Summer 2015

Improving Engagement Assessment by Model Individualization and Deep Learning

Feng Li
Old Dominion University

IMPROVING ENGAGEMENT ASSESSMENT BY MODEL INDIVIDUALIZATION AND DEEP LEARNING

by

Feng Li
B.S. June 1994, Huazhong University of Science and Technology, China
M.S. June 1997, Huazhong University of Science and Technology, China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2015

Approved by:

Jiang Li (Director)

Dean J. Krusienski (Member)

Frederic D. McKenzie (Member)

Alan T. Pope (Member)

ABSTRACT

IMPROVING ENGAGEMENT ASSESSMENT BY MODEL INDIVIDUALIZATION AND DEEP LEARNING

Feng Li
Old Dominion University, 2015
Director: Dr. Jiang Li

This dissertation studies methods that improve engagement assessment for pilots. The major work addresses two challenging problems involved in the assessment: individual variation among pilots and the lack of labeled data for training assessment models.

Task engagement is usually assessed by analyzing physiological measurements collected from subjects who are performing a task. However, physiological measurements such as Electroencephalography (EEG) vary from subject to subject. An assessment model trained for one subject may not be applicable to other subjects. We proposed a dynamic classifier selection algorithm for model individualization and compared it to other two methods: base line normalization and similarity-based model replacement. Experimental results showed that baseline normalization and dynamic classifier selection can significantly improve cross-subject engagement assessment.

For complex tasks such as piloting an air plane, labeling engagement levels for pilots is challenging. Without enough labeled data, it is very difficult for traditional methods to train valid models for effective engagement assessment. This dissertation proposed to utilize deep learning models to address this challenge. Deep learning models are capable of learning valuable feature hierarchies by taking advantage of both labeled and unlabeled data. Our results showed that deep models are better tools for engagement assessment when label information is scarce.

To further verify the power of deep learning techniques for scarce labeled data, we applied the deep learning algorithm to another small size data set, the ADNI data set. The ADNI data set is a public data set containing MRI and PET scans of Alzheimer's Disease (AD) patients for AD diagnosis. We developed a robust deep learning system incorporating dropout and stability selection techniques to identify the different progression stages of AD patients. The experimental results showed that deep learning is very effective in AD diagnosis.

In addition, we studied several imbalance learning techniques that are useful when data is highly imbalanced, i.e., when majority classes have many more training samples than minority classes. Conventional machine learning techniques usually tend to classify all data samples into majority classes and to perform poorly for minority classes. Imbalanced learning techniques can balance datasets before training and can improve learning performance.

ACKNOWLEDGEMENTS

I would like to offer my sincerest appreciation to my advisor, Dr. Jiang Li. His direction, assistance, and guidance have been invaluable. I would like to thank the committee members: Dr. Dean J. Krusienski, Dr. Frederic D. McKenzie, and Dr. Alan T. Pope. I am extremely grateful for their assistance and suggestions throughout my projects and the writing of my dissertation. I would also like to express my deepest gratitude to my family for their wholehearted support, especially my wife Wen Hu, who always accompanies me to face any challenges.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

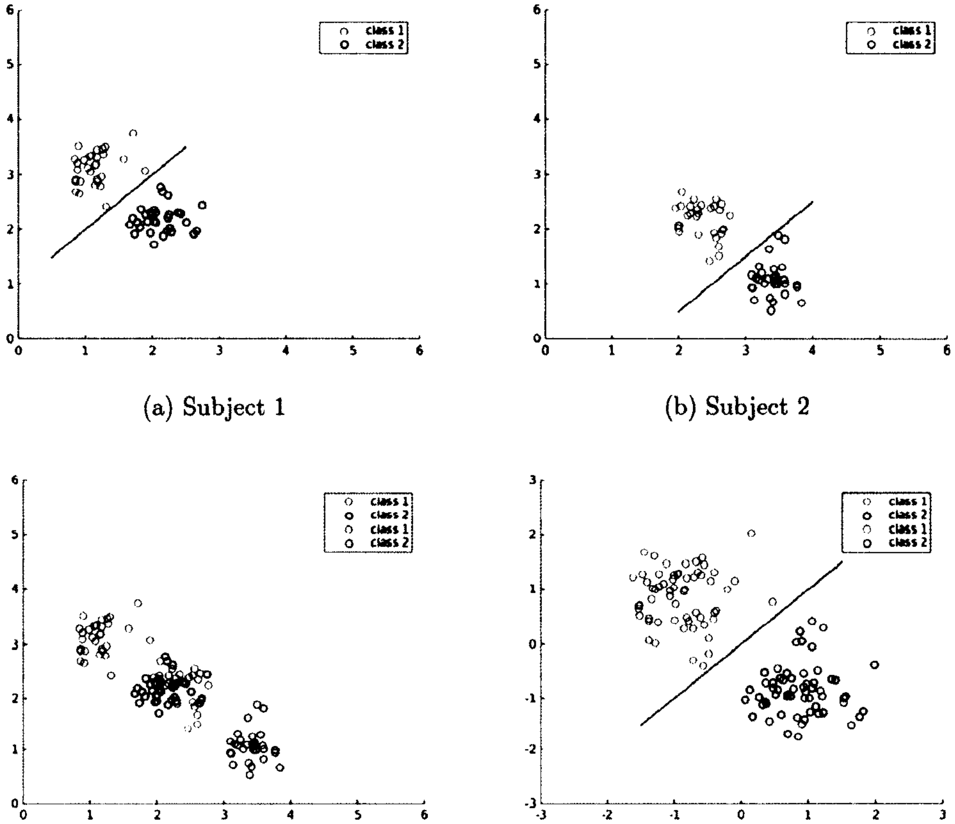
INTRODUCTION

1.1 PROBLEM STATEMENT

Task engagement is an "effortful striving toward task goals" [5]. It is an important factor that maintains operator performance during a cognitive task. This dissertation studies methods that improve engagement assessment for pilots. The major work addresses two challenges involved in the assessment: Individual variation among pilots and the lack of labeled data for training assessment models.

Most existing engagement assessment methods utilize physiological signals. However, physiological reaction, or individual response, varied among different subjects [6, 7]. This individual variation imposes a challenge on engagement assessment because a model trained for a specific subject cannot be applied across subjects. In practice, a general model is usually trained on combined signals collected from different subjects. Figure 1 shows an example that presents 2-D features for two subjects (Figure 1.(a), Figure 1.(b)). By inspecting the feature distribution, either subject 1 or 2 can draw a perfect decision boundary on its own feature space. However, when those two datasets combine together (Figure 1.(c)), data from different classes become overlapped and there is no way to draw a decision boundary for a good general model to discriminate the two classes. A normal method to solve this problem is normalization. For this example, the features of those two subjects follow similar distribution after normalization (Figure 1.(d)), and it is therefore possible to discriminate between the two classes. However, using only a normalization technique is not enough for complex tasks with large numbers of features. We need more advanced methods to solve this problem.

Another problem addressed by this dissertation is the lack of labeled data. For practical complex cognitive tasks, it is usually challenging and expensive to correctly label those cognitive states. Conventional labeling methods can be categorized as indirect and direct methods [8]. Indirect labeling methods usually evaluate cognitive levels in terms of task performances. These tasks have to be carefully designed so that



(a) Subject 1

(b) Subject 2

(c) Combined features

(d) Combined and normalized features

Figure 1: Individual variance

they can provide easy-to-measure task performances such as reaction time, error rate, etc. For complex tasks such as air flight and car driving, the performance metrics are not easy to obtain, and therefore indirect labeling methods cannot be applied to these tasks. Direct labeling methods are manipulated by the involved subjects themselves or by experts, where cognitive states are either self-assessed by the involved subjects or the experts after a task is finished. For complex and lengthy tasks, self-assessment is either not feasible or can only provide a rough estimate about the cognitive state. In contrast, experts can provide more precise assessments by observing the subjects' performance and considering the tasks' phases and progress [9]. Even though expert assessment is a feasible labeling method for complex tasks, it often provides scarce labeling data for classification. The reason is that a large amount of data in middle or unsure cognitive states has to be discarded. In addition, experts' time is expensive and usually only a small amount of data will be labeled.

This dissertation also discusses the imbalanced learning problem in engagement assessment. The problem refers to the situation where the training data is imbalanced, i.e., when majority classes have much more training samples than minority classes. Conventional machine learning techniques usually tend to classify all data samples into majority classes and perform poorly for minority classes.

This dissertation proposed several methods to address the above challenges.

1.2 CONTRIBUTIONS

The contributions of my dissertation work include:

1. A novel design of an enhanced committee machine for engagement assessment.
2. The first application of a dynamic classifier ensemble selection technique for model individualization.
3. The first attempt to utilize the deep learning framework for engagement assessment when the labeled data are scarce.

A committee machine is an "average" method which combines results achieved from a number of committee members (or models). It has the property to perform better than any individual committee member if its committee members are uncorrelated. In the dissertation, the committee machine is enhanced by making committee members more diverse using techniques include bootstrapping, feature selection, and model selection. It is also designed as a framework so that imbalanced learning and model individualization techniques can be implemented as modules and can be embedded into the framework.

Dynamic classifier ensemble selection is a technique that selects the best models from a number of models based on their performance on the test samples' neighbors in a validation dataset. Instead of training one general model from the combined datasets from all subjects, dynamic classifier ensemble selection aims to find a few best models from a model pool, which contains models from each individual subject.

Deep learning has been developed to automatically learn feature representations from unlabeled or labeled data. In this dissertation, the deep learning technique is first utilized to pre-train a model using large amount of unlabeled data. The model is then fine-tuned by limited labeled data. Experimental results show that deep learning performs well when labeled data is scarce. The deep learning technique is also verified with another small size data set (ADNI), and we find that dropout technique is very effective in preventing over-fitting and is therefore suitable for small size data sets.

1.3 ORGANIZATION OF THE DISSERTATION

Chapter 2 provides a survey on research of task engagement. Chapter 3 presents experimental design, data collection, and preprocessing for task assessment conducted in the dissertation. Chapter 4 focuses on model individualization methods. Chapter 5 demonstrates how deep learning was utilized to resolve the scarce labeled data challenge. Chapter 6 further studies the power of deep learning technique using another small size data set (ADNI). Chapter 7 describes imbalanced learning techniques, and Chapter 8 concludes the dissertation.

Chapter 2

RELATED WORK

2.1 TASK ENGAGEMENT

Task engagement is often involved in the research fields of Operator Functional State (OFS), Human-Computer Interaction (HCI), Brain-Computer Interfaces (BCI), and Ergonomics (or human factors).

OFS is defined as *the multidimensional pattern of processes that mediate task performance under stress and high workload, in relation to task goals and their attendant physiological and psychological costs* [10]. Hockey applied OFS as a framework to the assessment of performance degradation [10, 11]. He believed that OFS is a function of:

- current operator condition (sleep loss, fatigue, illness).
- pattern/mode of interaction with task goals (priorities, strategies, effort management, and control).
- stable operator characteristics (skill, motivational biases, coping style)

HCI is defined as a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of the major phenomena surrounding them [12, 13]. It studies a human and a machine in communication and utilizes knowledge from both the machine side and the human side. "On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, social psychology, and human factors such as computer user satisfaction are relevant. And, of course, engineering and design methods are relevant" [12]. The goal of HCI is to improve the interactions between users and computers by making computers more usable and receptive to users' needs.

BCI is a special type of HCI. BCI systems aim to provide assistive devices for people with severe disabilities that prevent them from performing physical movements [14, 15, 16]. So compared with other human-computer interfaces that require muscle activity, BCI provides "non-muscular" communication.

Ergonomics (or human factors) is defined as the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data, and methods to design in order to optimize human well-being and overall system performance [17, 18, 19]. Ergonomics contains three main fields of research: physical, cognitive, and organizational ergonomics. Physical ergonomics is concerned with human anatomical, anthropometric, physiological, and biomechanical characteristics as they relate to physical activity. Cognitive ergonomics is concerned with mental processes such as perception, memory, reasoning, and motor response as they affect interactions among humans and other elements of a system. And relevant topics include mental workload, decision-making, skilled performance, human-computer interaction, human reliability, work stress and training as these may relate to human-system design. Organizational ergonomics is concerned with the optimization of sociotechnical systems, including their organizational structures, policies, and processes.

OFS and HCI can be considered as two of the research fields of Ergonomics. The primary difference between Ergonomics and HCI is that HCI focuses on people working specifically with computers, while Ergonomics concerns many other types of machinery. Compared to the much wider application of Ergonomics, OFS focuses on some specific areas and tasks (aviation, driving) and aims to improve task performance with a feedback from OFS assessment.

Although differences exist among OFS, HCI, and ergonomics, how to increase or keep the operator's performance level is their ultimate goal, and task engagement is an important factor that influences performance. There are several definitions of "engagement" from literature. Matthews et al. defined task engagement as an "effortful striving toward task goals" [5]. He also pointed out that task engagement increases during a demanding cognitive task and decreases when participants perform a sustained and monotonous vigilance task. Berka and Levendowski treated engagement as a process related to information gathering, visual scanning, and sustained attention [20]. In [21], Stevens et al. believed that engagement is related to the level of mental vigilance and alertness during the task, and that the loss of engagement was considered as distraction. In [22], Channel et al. considered engagement as a particular emotion, which was "positive excited" in the valence/arousal model.

Engagement is often studied with other components that may affect task success, such as workload, attention, vigilance, fatigue, error recognition, and emotions [5, 20, 21, 22, 23, 24]. In [3], the connection between task engagement and those components is illustrated in Figure 2. The figure provides an view of a simplified characterization of the "constructs." (In HCI research, "constructs" is defined as patterns of users' states which could be used to characterize interactions.) The inner circles represent the HCI components: user, content, and interface. In the middle circles are the constructs for each related HCI component. From the figure, we can see that construct engagement is related to "content." The outer circles give a hint about what an evaluation would be useful for.

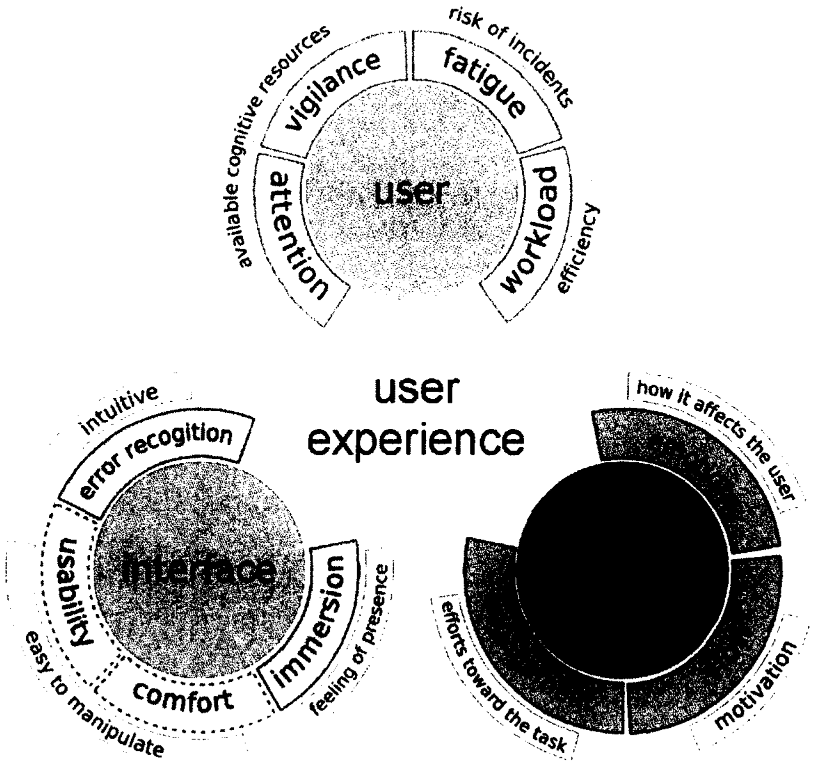


Figure 2: Engagement and other components from an HCI evaluation Perspective ([3]).

In [10], Hockey considered engagement (engaged/disengaged) as one of regulatory control modes when he studied human performance within the framework of

OFS. He proposed a compensatory control model with a central feature as the performance/cost trade-off. For example, when the current effort budget is inadequate to accomplish task goals, two optional routes (increase effort budget or reduce task goals) can be taken as regulatory controls. Hockey categorized the control modes as engaged, strain, and disengaged, and summarized them as shown in Table 1.

Table 1: Modes of regulatory control.

Control **mode**	**Environmental** **context**	**Performance** **(task goal)**	**Affective** **state**	**Stress** **hormones**
engaged	high demands high control	optimal (high)	effort without distress (anx- eff↑fat↓)	adrenaline ↑ cortisol ↓
strain	high demands low control	adequate (high)	effort with distress (anx↑eff↑fat ↑)	adrenaline ↑ cortisol ↑
disengaged	high demands low control	impaired (reduced)	distress without effort (anx↑eff↓fat-)	cortisol ↑ cortisol ↑

* anx: anxiety, eff: effort, fat: fatigue; ↑: increase, -: no change, ↓: decrease ([10])

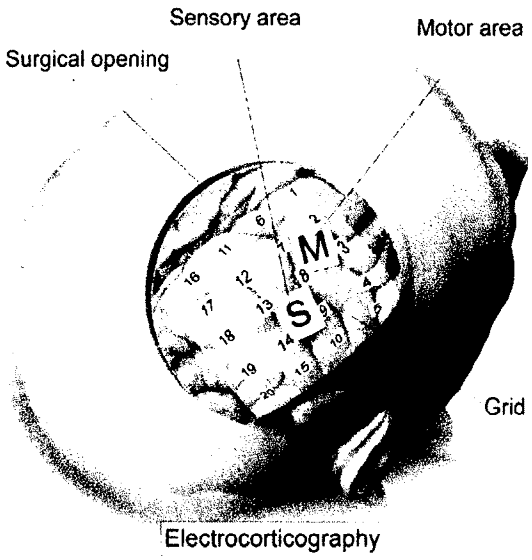
From the above literature review, we can conclude that engagement is the effort made by operator towards task goals. It can be measured as cognitive states and can be utilized as a control mode. In this dissertation, we will study engagement assessment for pilots while they are conducting a flight simulation from Seattle to Chicago.

2.2 NEUROIMAGING

Neuroimaging is currently becoming a major approach used to assess brain activities. Neuroimaging modalities can be categorized as invasive and non-invasive technologies.

Invasive modalities need to implant electrodes inside the skull (Figure 3). Two invasive modalities can be found in literature: electrocorticography (ECoG) [25, 26], and intracortical neuron recording [1]. The difference between ECoG and intracortical neuron recording is that ECoG places electrodes on the surface of the cortex and intracortical neuron recording implants electrodes inside the cortex. The method of intracortical neuron recording can record three types of signals: single-unit activity (SUA), multi-unit activity(MUA), and local field potentials (LFP).

Non-invasive neuroimaging methods do not need to implant electrodes, and include electroencephalography (EEG), magnetoencephalography (MEG), Functional magnetic resonance imaging (fMRI), and near-infrared spectroscopy (NIRS). EEG



from ***http://en.wikipedia.org/wiki/Electrocorticography***

Figure 3: Invasive neuroimaging: electrocorticography.

measures electric brain activity caused by the flow of electric currents during synaptic excitations of the dendrites in the neurons and is extremely sensitive to the effects of secondary currents [27]. MEG measures the intracellular currents flowing through dendrites which produce magnetic fields that are measurable outside of the head [28]. The advantage of MEG is that magnetic fields are less distorted by the skull and the scalp than electric fields [29]. fMRI detects changes in local cerebral blood volume, cerebral blood flow, and oxygenation levels during neural activation by means of electromagnetic fields [30]. NIRS is an optical spectroscopy method that employs infrared light to characterize noninvasively acquired fluctuations in cerebral metabolism during neural activity [31].

Neuroimaging methods can also be categorized as direct and indirect depending on whether the method is directly or indirectly related to neuronal activity. EEG, MEG, ECoG, and intracortical neuro recording measure electrophysiological activity and are considered as direct methods. In contrast, fMRI and NIRS are categorized as indirect methods, because they record the hemodynamic response, which is not directly related to neuronal activity.

Each neuroimaging method has its own characteristics. In [1], they were summarized as shown in Table 2 from the perspective of activity measured, direct/indirect, temporal resolution, spatial resolution, risk, and portability.

Table 2: Summary of neuroimaging methods ([1]).

Neuroimaging method	Activity measured	Direct/ Indirect measurement	Temporal resolution	Spatial resolution	Risk	Portability
EEG	Electrical	Direct	0.05 s	10 mm	Non-invasive	Portable
MEG	Magnetic	Direct	0.05 s	5 mm	Non-invasive	Non-portable
ECoG	Electrical	Direct	0.003 s	1 mm	Invasive	Portable
Intracortical neuron recording	Electrical	Direct	0.003s	0.5 mm(LFP) 0.1 mm(MUA) 0.05 mm(SUA)	Invasive	Portable
fMRI	Metabolic	Indirect	1 s	1 mm	Non-invasive	Non-portable
NIRs	Metabolic	Indirect	1 s	5 mm	Non-invasive	Portable

For OFS systems that usually involve health operators, non-invasive neuroimaging methods are obviously more welcome. Compared to other neuroimaging approaches, EEG is the most promising candidate due to its noninvasiveness, high temporal resolution, portability, and reasonable cost [32, 33]. In [34], the authors reviewed BCI articles published from 2007 to 2011 and concluded that EEG played a dominant role among all neuroimaging modalities. This dissertation will utilize EEG to access pilots' engagement whilesimulating a flight.

2.2.1 ELECTROENCEPHALOGRAPHY (EEG)

EEG Signals

EEG measures electric brain activity caused by the flow of electric currents during synaptic excitations of the dendrites in the neurons [27]. Researchers have found that energies in EEG frequency bands are correlated to cognitive states (Table 3).

A typical EEG recording system consists of electrodes, amplifiers, an A/D converter, and recording devices (Figure 4). The electrodes acquire signal from scalp, the amplifiers amplify the amplitude of the EEG signals, the A/D converters digitalize the amplified analog EEG signals, and the recording devices store the digital EEG signals.

It was recognized that the placement of EEG electrodes should be standardized and the first standard used was the 10-20 electrode system proposed by H.H. Jasper in 1958 [35], which defined locations of 21 electrodes (Figure 5). With the demand of increasing spatial resolution of EEG, a larger number of EEG electrodes were adopted and the original 10-20 system was upgraded to a 10-10 system [36], and a

Table 3: EEG frequency bands.

Band (Hz)	**Location**	**Normally**	**Pathologicallyc**
Delta **<4**	frontally in adults, posteriorly in children; high-amplitude waves	• adult slow-wave sleep • in babies • has been found during some continous-attention tasks	• subcortical lesions • diffuse lesions • metabolic encephalopathy hydrocephalus • deep midline lesions
Theta **4 - 7**	found in locations not related to task at hand	• higher in young children • drowsiness in adults and teens • idling • associated with inhibition of elicited responses (has been found to spike in situations where a person is actively trying to repress a response or action)	• focal subcortical lesions • metabolic encephalopathy • deep midline disorders • some instances of hydrocephalus
Alpha **8 - 15**	posterior regions of head, both sides, higher in amplitude on non-dominant side. Central sites (c3-c4) at rest	• relaxed/reflecting • closing the eyes • Also associated with inhibition control, seemingly with the purpose of timing inhibitory activity in different locations across the brain.	• coma
Beta **16 - 31**	both sides, symmetrical distribution, most evident frontally; low-amplitude waves	• range span: active calm ->intense -> stressed ->mild obsessive • active thinking, focus, hi alert, anxious	• benzodiazepines
Gamma **32+**	somatosensory cortex	• displays during cross-modal sensory processing (perception that combines two different senses, such as sound and sight) • also is shown during short-term memory matching of recognized objects, sounds, or tactile sensations	• a decrease in gamma-band activity may be associated with cognitive decline, especially when related to the theta band; however, this has not been proven for use as a clinical diagnostic measurement
Mu **8 - 12**	sensorimotor cortex	• shows rest-state motor neurons	• mu suppression could indicate that motor mirror neurons are working. Deficits in Mu suppression, and thus in mirror neurons, might play a role in autism.

* `http://en.wikipedia.org/wiki/Electroencephalography`

5% system [37].

EEG Artifacts Removal

EEG signals are very weak ($2 \sim 100\mu V$), hard to acquire, and of poor quality. In addition, EEG signals are easily contaminated by background noise generated either inside the brain or externally over the scalp. These contaminations are termed artifacts. Artifacts can be classified into three main categories: 1) physiological (arising from subject or patient, such as eye blinking/movement, heart beating, and movement of other muscle groups), 2) technological (arising from the electrode-subject interface, electrodes, electrode connection, amplifier and recording equipment), and 3) extrinsic (such as main line interference; other equipment connected to the patient; airborne sources, including electromagnetic, radio frequency, and electrostatic

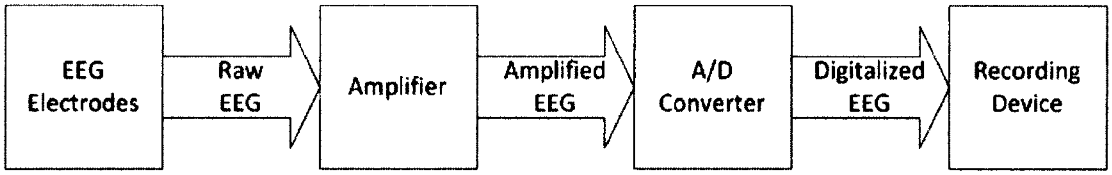


Figure 4: EEG recording system.

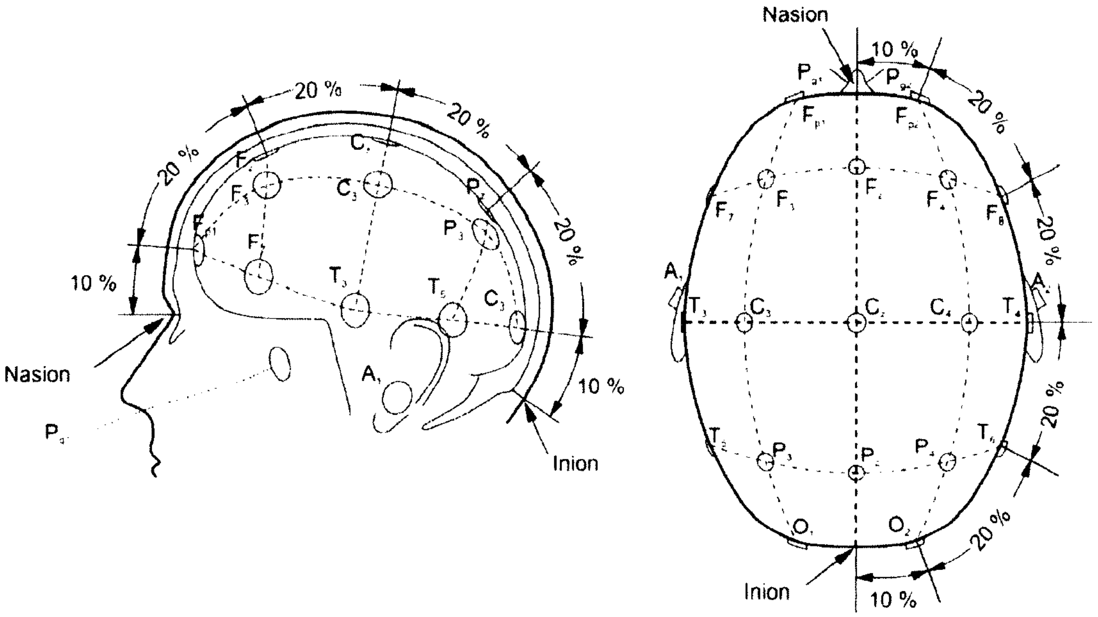


Figure 5: EEG 10-20 electrode system ([1]).

signals; and other environmental phenomena) [38].

An EEG recording with technologic and extrinsic artifacts cannot be recovered and should be rejected. For example, when the amplifier is saturated or the electrodes are malfunction or lose connection with scalp, recorded EEG signals are obviously invalid and are therefore useless. In contrast, biologic artifacts are mixed with valid EEG and can be removed using the appropriate algorithms.

Major physiological artifacts consist of Electromyogenic (EMG), Electrocardiogram (ECG), and Electrooculargram (EOG). EMG artifacts occur due to activities of muscles at rest and during contraction of frontal and temporal muscles (clenching of jaw muscles). The techniques proposed in literature for the removal of EMG artifacts include filters, adaptive filters, blind source separation, and Independent

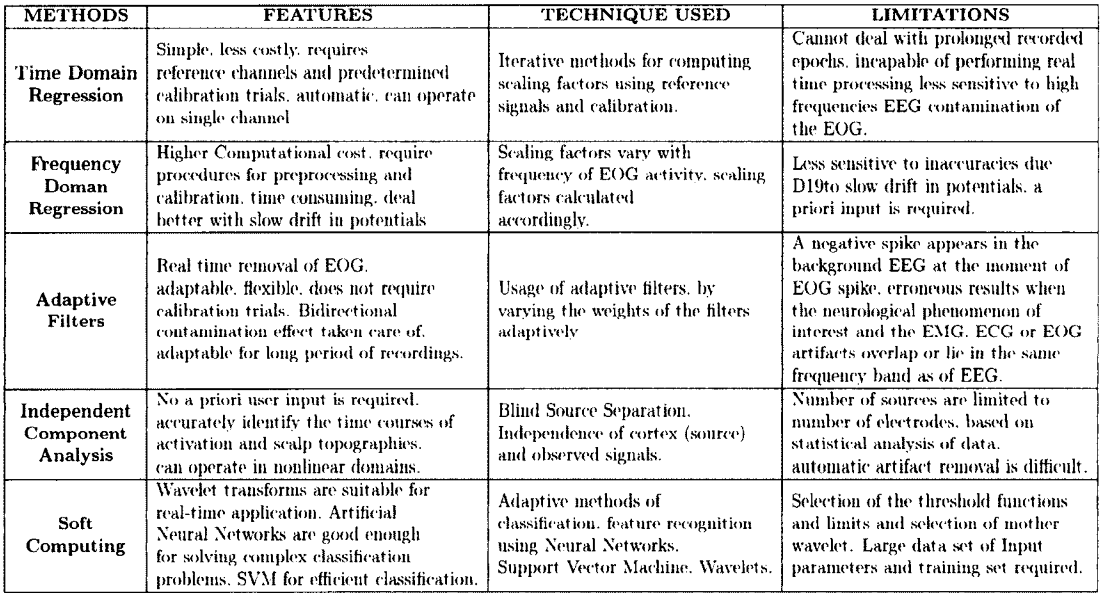
Component Analysis (ICA) [39]. In [40], a higher order statistical property, kurtosis, the forth cumulant of data, was used to distinguish non-artifact from artifact signal, and reject the later one. Gao et al. [41] have used the Canonical Correlation Analysis (CCA) technique which utilized a correlation threshold to remove the EMG artifacts automatically, without eliminating the signal of interest.

ECG artifacts come from the relatively high cardiac electrical field which affects the surface potentials on the scalp. Many efforts have been made to remove ECG artifacts. Fortgens and Bruin [42] have tried to subtract the weighted artifact of source signals of ECG. Nakamura and Shibasaki [43] and Harke et al. [44], studied the use of the Ensemble Average Subtraction method to correct ECG artifacts. Sahul et al. [45], proposed adaptive filtering (AF) using an ECG channel reference to remove them. The above methods utilized reference ECG signal and required consecutive R-waves of separate ECG channels to eliminate artifacts from EEG signal [46]. In 2000, Everson and Roberts [47] proposed an ICA-based artifact reduction method for EEG artifacts removal. In 2008, Devuyst et al. [48] implemented a variation of the ICA algorithm using a single-channel EEG and ECG. Their approach gave promising results as compared to earlier proposed techniques. Dewan et al. [49] utilized an adaptive thresholding method along with clustering to detect contaminated candidate R-spikes of ECG artifact, based on which a noise model of ECG artifact was built for decontamination.

EOG artifacts play significant detrimental effect on EEG signals due to eye activities. When human eyes blink or move, an electric field is created which can be 10 times larger in amplitude than an EEG and lasts for up to 400 ms [50]. Since eye movements are difficult to suppress over the period of EEG recording, almost all the EEG recordings become contaminated with EOG artifacts. EOG has been attributed to the fact that the eyeball acts as a dipole, where the external surface of the cornea (at the front of the eye) is positively charged with respect to the posterior surface of the retina (at the back of the eye). Therefore, each eyeball acts like a battery and generates an electric field, which interferes with the surface recording of the electrical activity of the brain, at particular electrode locations. In 1991, Berg et al. [51] presented a simplified model of the electric dipole within the eyeball. The direction of the dipole is aligned with the line of sight and the size of the dipole is determined by the amount of light hitting the retina in the back of the eye. EOG contamination is only dominant in the frontal EEG channels [52] where are close to eyes. The propagation of the EOG artifact from the eyes to the rest of the scalp locations is practically instantaneous [42]. Vertical eye movements will influence midline electrodes much more than lateral movements.

Over the decades, researchers have developed various algorithms to remove EOG, including regression techniques, filtering techniques, blind source separation (BSS), ICA, and soft computing techniques. In [53], Sood et al. reviewed those techniques for EOG removal and summarized them as Table 4

Table 4: Comparison between the various techniques used for artifacts removal from EEG signal



METHODS	FEATURES	TECHNIQUE USED	LIMITATIONS
Time Domain Regression	Simple. less costly. requires reference channels and predetermined calibration trials. automatic. can operate on single channel	Iterative methods for computing scaling factors using reference signals and calibration.	Cannot deal with prolonged recorded epochs. incapable of performing real time processing less sensitive to high frequencies EEG contamination of the EOG.
Frequency Doman Regression	Higher Computational cost. require procedures for preprocessing and calibration. time consuming. deal better with slow drift in potentials	Scaling factors vary with frequency of EOG activity. scaling factors calculated accordingly.	Less sensitive to inaccuracies due D19to slow drift in potentials. a priori input is required.
Adaptive Filters	Real time removal of EOG. adaptable. flexible. does not require calibration trials. Bidirectional contamination effect taken care of. adaptable for long period of recordings.	Usage of adaptive filters. by varying the weights of the filters adaptively	A negative spike appears in the background EEG at the moment of EOG spike. erroneous results when the neurological phenomenon of interest and the EMG. ECG or EOG artifacts overlap or lie in the same frequency band as of EEG.
Independent Component Analysis	No a priori user input is required. accurately identify the time courses of activation and scalp topographies. can operate in nonlinear domains.	Blind Source Separation. Independence of cortex (source) and observed signals.	Number of sources are limited to number of electrodes. based on statistical analysis of data. automatic artifact removal is difficult.
Soft Computing	Wavelet transforms are suitable for real-time application. Artificial Neural Networks are good enough for solving complex classification problems. SVM for efficient classification.	Adaptive methods of classification. feature recognition using Neural Networks. Support Vector Machine. Wavelets.	Selection of the threshold functions and limits and selection of mother wavelet. Large data set of Input parameters and training set required.

EEG Features

There are several types of EEG feature that have been often utilized, including power spectral density (PSD), event-related potential (ERP), use of more than two feature types, phase information, and others [34]. PSD has been used most often and in general, fast Fourier transform, wavelet transform, and autoregressive coefficients have been used to calculate PSD feature in literature. ERP is another widely used EEG feature and P300 has been used the most frequently. More than two feature types means that multiple features are utilized by either combining two more features or by using different features independently. Phase information is also a useful feature; due to the degree of phase, it is correlated to different brain regions. Other types of features discussed in literatures include correlation coefficients, time domain parameters, fractal dimensions, polynomial coefficients, local discriminative spatial patterns, approximate entropy, and time-embedded representations. We follow Berka et al.'s work and use 1-Hz PSD bins as features. This is described in section 3.3.

2.3 CLASSIFICATION ALGORITHMS

Numerous machine learning algorithms have been applied to cognitive states classification. Common algorithms include Bayesian analysis, linear discriminant analysis (LDA), support vector machine (SVM), K-nearest neighbour classifier (k-NNC), and artificial neural networks (ANN). In [1], those classification methods were summarized as shown in Table 5. ANN and SVM are two mature algorithms that have been widely used. From our experience, ANN is suitable to serve as committee members for the committee machine algorithm. It is also the base of the deep learning algorithm, which is studied in Chapter 5. Since deep neural networks work as a feature extractor instead of as a classifier, we need a classifier to compare the performance between the extracted features and the other types of features. We chose linear SVM because it is able to train stable models and works well for a fair comparison. A brief introduction to ANN and SVM is presented below.

2.3.1 ARTIFICIAL NEURAL NETWORKS

The idea of ANNs was inspired by how the brain processes information. An ANN comprises a set of nodes and connections that are updated during the training process. The ANN is fed with a set of training examples and the output is observed. The difference between desired and actual outputs is calculated, and the internal weights are modified by the training algorithm to minimize the difference. The procedures keep iterating until the network gets converged.

The most widely used ANN is multilayer perceptron (MLP). An MLP is composed of several layers of neurons: an input layer, one or several hidden layers, and an output layer. At the beginning, the practical MLPs usually only contain one hidden layer due to the difficulty to train NN with multiple hidden layer (deep NN), i.e., the error becomes too small when layers to be trained are far away from the highest layer, which makes the learning impossible. In 2006, Hinton proposed the deep learning algorithm that can successfully train deep NN [54] and deep NN became prevailing.

2.3.2 SUPPORT VECTOR MACHINE

Table 5: Summary of classification methods ([1]).

Classification algorithms	Properties
Bayesian analysis	• Assigns the observed feature vector to the labeled class to which it has the highest probability of belonging • Produces nonlinear decision boundaries • Not very popular in the BCI systems
LDA	• Simple classifier with acceptable accuracy • Low computation requirements • Fails in the presence of outliers or strong noise. Regularization required • Usually two class. Extended multiclass version exits. • Improved LDA versions: BLDA, FLDA
SVM	• Linear and non-linear (Gaussian) modalities • Binary or multiclass method • Maximizes the distance between the nearest training samples and the hyperplanes • Fails in the presence of outliers or strong noise. Regularization required • Speedy classifier
k-NNC	• Uses metric distances between the test feature and their neighbors • Multiclass • Efficient with low dimensional feature vectors. Very sensitive to the dimensionality of the feature vectors
ANN	• Very flexible classifier • Multiclass • Multiple architectures (PNN, Fuzzy ARTMAP ANN, FIRNN, PeGNC)

An SVM uses a discriminant hyperplane to identify classes. SVM selects the hyperplanes that maximize the margins, i.e, the distance between the nearest training samples and the hyperplanes (see Figure 6). So the basis of SVM is to map data into a high dimensional space and find a separating hyperplane with the maximal margin. SVM with linear decision boundaries and regularization has been successfully applied to many problems [55, 56]. It is also possible to create an SVM with non-linear decision boundary by means of a kernel function K(x, y). Non-linear SVM leads to a more flexible decision boundary in the data space, which may increase classification accuracy. The kernel generally used is the Gaussian or Radial Basis Function (RBF) kernel:

$$K(x,y) = exp\left(\frac{-\left\|x-y\right\|^2}{2\sigma^2}\right) \tag{1}$$

SVM has been widely used in cognitive states assessment, because it is simple,

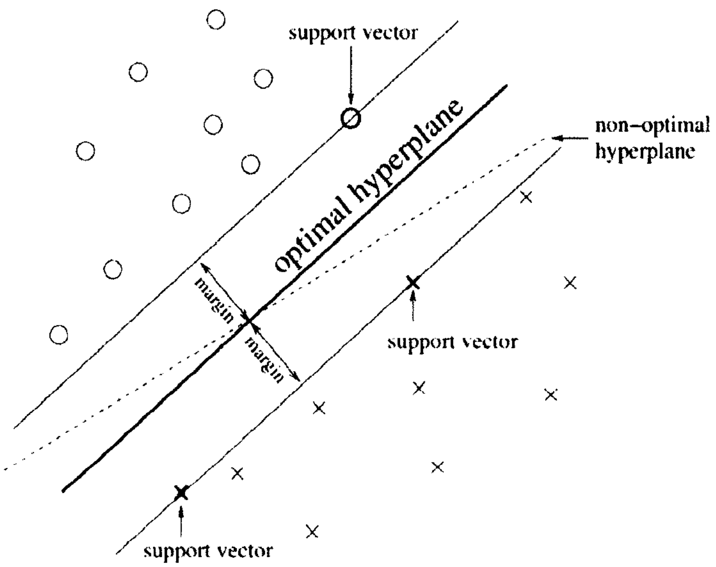


Figure 6: SVM finds the optimal hyperplane([4]).

performs well, and is robust with regard to the curse of dimensionality. So SVM can work well without a large training, even with very high dimensional feature vectors.

2.4 MODEL INDIVIDUALIZATION

Most existing OFS assessment methods utilizes psychophysiological signals to index the level of cognitive demand associated with a task [57, 58], with fatigue [59, 60], with engagement [21, 61, 6], and with other functional state dimensions [10]. Usually, a general model is first trained based on signals collected from different subjects. The trained model is then applied to different subjects directly or with a minimum of adaptation. However, individual cognitive state differences exist among different subjects as referred to idiosyncratic regularities of the physiological reaction, or individual response specificity (IRS) [6, 7]. The inconsistency of individual response over time is also documented by Forster [62]. The general model usually does not perform well due to the large individual variance. Attempts at model individualization have been made to address the variance and can be divided into two categories [63]: basic research and statistical approaches. The basic research tries to discover the individual physiological response differences, which reflect OFS differences, among subjects. The identified differences are then used for model adaptation to compensate for individual variance. Statistical approaches are purely data driven and do not rely on an understanding of the physiological differences. Basic research may lead to more interpretable models. Unfortunately, most of the current methods belong to the second category because the nature of OFS is still not well understood.

In [63], Erik Olofsen presented a neurobiological model that integrated the two-process model of sleep regulation [64] with the flip-flop sleep switch model [65], which can be used to identify the physiological parameters that underlie individual differences in resilience during sleep deprivation. As a statistic approach, the mixed-effects modeling method is considered suitable for modeling of longitudinal data, explicitly accounting for inter-individual differences [66, 67]. Erik Olofsen developed nonlinear mixed-effects modeling for individualized prediction of fatigue and performance [68]. Van Dongen also compared nonlinear mixed-effects modeling with two other implementations of a mixed-effects modeling approach: the standard two stages (STS) and restricted maximum likelihood (REML) [69]. Rajaraman et al. developed a method for predicting the cognitive performance of individuals with total sleep loss [70]. They individualized their model by combining the performance information with a priori performance information using a Bayesian framework. In our previous work, we presented a similarity-based approach for model individualization [71], in which we identified similar subjects from the training data pool and used their data together with the limited data from the test subject to build an individualized OFS assessment model. Our approach was built upon the assumption that if a subject has "similar" data as another, then the two subjects will have a "similar" behavior in cognitive state. The similar metrics was defined as the Euclidean distance in feature space.

2.5 LEARNING WITH SCARCE LABELED DATA

Assessment of human cognitive states such as vigilance, fatigue, and engagement has attracted a lot of attention in recent decades [72, 20, 60, 73]. For complex tasks like car or plane driving, the driver's cognitive states can be evaluated by many approaches, such as the subjective report [23], [74], biological measures (EEG [75], [76], electrocardiogram (ECG) [77], [78], electro-oculography (EOG) [79], [80], surface electromyogram (sEMG) [81], [82]), physical measures (eye tracking [83], [84], fixed gaze [85], [86], mouth activities [87], [88], head pose or nodding [89], [90]), driving performance measures [91], [92], and hybrid measures [93], [94]. However, correctly labeling these cognitive states is challenging and expensive. Conventional labeling methods can be categorized as indirect and direct methods [8, 10]. Indirect labeling methods usually evaluate cognitive levels in terms of task performances; these tasks have to be carefully designed so that they can provide easy-to-measure task performances such as reaction time, error rate, etc. For complex tasks such as air flight and car driving, the performance metrics are not easy to obtain, and therefore indirect labeling methods cannot be applied to these tasks. Direct labeling methods are manipulated by the involved subject themselves or by experts, while cognitive states are either self-assessed by the involved subjects or by the experts after a task is finished. For complex and longtime tasks, self-assessment is either not feasible or can only provide a rough estimate of the subjects' cognitive state. In contrast, experts can provide more precise assessments by observing the subject's performance and considering the task's phases and progress [9]. Even though expert assessment is a feasible labeling method for complex tasks, it often provides scarce labeling data for classification. This is because a large amount of data in middle or unsure cognitive states has to be discarded during the labelling process and because examining a large data set is time consuming.

Most of the supervised algorithms suffer from the lack of labeled data, which could lead to the overfitting problem [95]. The problem becomes more severe when the model to be trained contains a large number of parameters. Unsupervised and semi-supervised algorithms utilize unlabeled data (large quantity compared with labeled data) for training, and they could improve model's generalization capability [96, 97, 98]. Deep learning techniques can be considered as a semi-supervised algorithm that utilizes both labeled and unlabeled data for training [99]. The Deep learning scheme utilizes a multi-layer neural network which is capable of learning complex hierarchical nonlinear features. However, local gradient-based optimization algorithms, such as Back Propagation (BP), usually perform poorly when the multi-layer neural network is initialized with random weights because the training can get trapped in bad local minima [100]. A significant breakthrough was made by Hinton in 2006 [54, 101]. He proposed an efficient algorithm to pre-train Deep Belief Networks (DBNs), a deep structure network, in a layer-by-layer fashion using Restricted Boltzmann Machines (RBMs). The pre-trained deep structure can then be efficiently trained by the BP algorithm. The training can be further improved by using "dropout", which significantly reduces overfitting [102].

In this dissertation, we studied engagement assessment in a complex task with scarce labeled data. We conducted a 4-hour flight simulation for 15 pilots and EEG data were recorded during the simulation. For data from each pilot, experts labeled 10-minute recordings as engaged and other 10-minute recordings as disengaged. We utilized deep learning techniques to solve the scarce labeled data problem and the details are presented in Chapter 5.

Chapter 3

DATA COLLECTION AND PROCESSING

3.1 FLIGHT SIMULATION AND DATA COLLECTING

In order to study engagement, we conducted simulations in a fully equipped Boeing 737 simulator (Figure 7). Involved pilots had varying levels of experience with different types of aircraft. All had instrument ratings and held commercial/private/ATP (Airline Transport Pilot) licenses with experience in Single-Engine Land (SEL), Multi-Engine Land (MEL), Jet, or Turboprop.

Figure 7: Flight simulator.

The simulations involved a flight from Seattle Tacoma International Airport to Chicago O'Hare International Airport. The details of the flight have been extracted from an actual American Airlines flight which took place on May 10th, 2010.

In order to study the effects of sleep-loss related fatigue on engagement, all pilots were scheduled to arrive at 5:30pm and were asked to avoid drinking caffeinated drinks such as coffee during the day of the experiment. An orientation video was shown to the subjects before the scheduled experiment time. The video contained a description of the experiment as well as a Control & Display Unit (CDU) programming training section. The video included a description of the sensors and of the video recording devices that were used during the experiment, as well as the responsibilities that the experimenters would have during the simulation. The details shared with the subjects did not include information on the probes that were used to measure engagement levels, so that the pilots would not anticipate these probes throughout the simulation. During the simulation, one member of the staff controlled the simulation computer to play pre-recorded audio files which can mimic ATC transmissions. An experimenter was in charge of tagging the data to make sure that the proper labels were put in the data sheets to identify the phases of the experiment as well as the times when the pilot responded to ATC. At the end of the experiment, the subjects filled out a subjective survey to assess their workload, fatigue and situational awareness during different phases of flight.

The simulation included three events inserted into the flight scenario. The events were scheduled to occur at predetermined times to observe and measure how the pilots responded to them. The first event was an ATC call asking the pilot to report when the aircraft was at 29000 feet. This call came while the aircraft was crossing 19000 feet. The aim of this event was to assess whether the pilot would remain engaged at the early stage of initial ascent. The second event was another ATC call that asked the pilot to report his/her position at 20 miles east of HLN (one of the waypoints). This call came at the early stages of level flight. The goal of this event was to determine whether the pilot would remember to call back ATC at the designated point. The third probe was a failure event. Half of the subjects received a failure signal at the time of one hour into the simulation. The other half received it at the time of three hours into the simulation. This approach was preferred because if all runs had both failures, the pilots might have remained in an engaged state throughout the flight after the first failure, with the expectation that such failures might be inserted into the scenario to test his/her performance. The data collected during these events could be compared, in order to establish the difference in the two engagement states in terms of physiological measures and subjective ratings. The event was selected such that it wouldn't prompt a drastic decision such as an emergency landing but it would allow the pilot to solve the problem with onboard capabilities.

In the experiments to be performed, we included several subjective rating scales that were collected after each simulation, including: the Situational Awareness Rating Technique ([103]), the Bedford workload scale [104], the NASA Task Load Index (NASA TLX [104]), the Samn-Perilli fatigue scale [105]., and the boredom proneness scale [106]. In order to minimize the effects of intrusive questioning, a post-experiment survey was conducted. Each subject was asked about his/her perceived level of workload, boredom proneness, situation awareness and fatigue during different phases of flight.

In additional to flight technical data (altitude, speed, etc.), objective data collection was achieved with the use of three sensors, including eye tracking cameras, a EEG net, and a EKG sensor. The sensors of the EEG net contained 32 channels which were located on scalp as shown in Figure 8. In addition, performance data, such as response time to ATC calls or pump failure, were also collected.

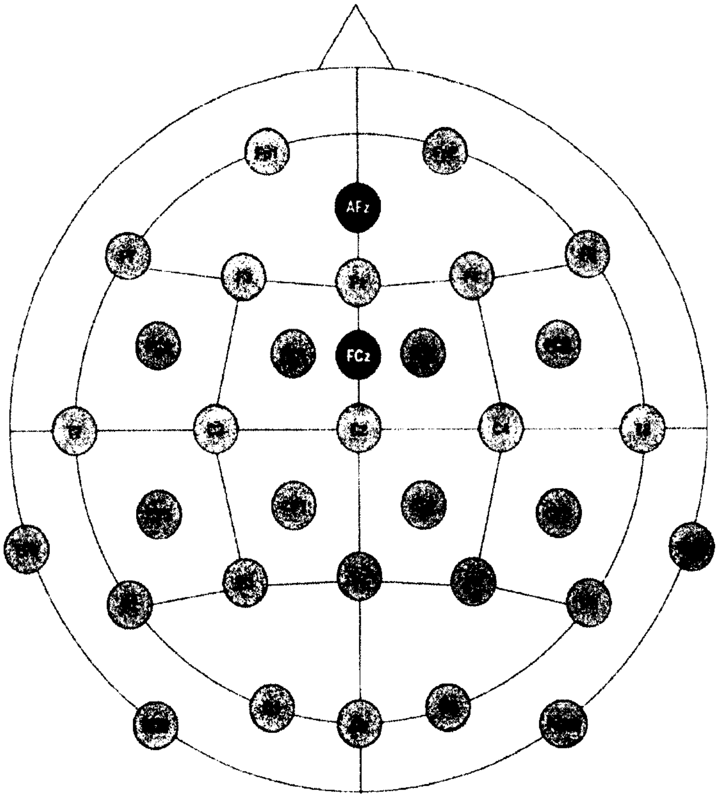


Figure 8: EEG sensor location.

3.2 ENGAGEMENT GROUND TRUTH FINDING

Before an engagement assessment model can be deployed, it needs to be trained based on the engagement ground truth and corresponding input information (physiological signals, performance, and others). However, there does not exist a sensor to provide engagement ground truth. In this paper, we created an engagement ground truth assessment model that incorporated subjective evaluation, behavioral measures (such as communications with ATC and real time performance data), and sensor measures (such as EEG, eye tracking and EKG data).

The subjective evaluation data were collected after each pilot completed his/her flight simulation. We divided the whole flight simulation into 11 phases: 1) takeoff to 19k, 2) 19k to 29k, 3) 29k to 37k, 4) Seattle center, 5) failure / Seattle center, 6) Salt Lake center, 7) Minneapolis center, 8) Chicago center, 9) Chicago center to call to descend, 10) call to descent to leveling at 9000, and 11) final descent to land. For each phase, each pilot gave a score for each dimension in the SART, the Bedford workload scale [104], the NASA TLX, and the Samn-Perilli fatigue scale. Each pilot also rated his boredom proneness based on the survey.

To derive an engagement profile as ground truth for OFS model training, we need to consider different sources of information. Three major steps are followed: baseline construction, degradation/recovery, and refinement based on strong indicators.

1. Baseline construction. An engagement baseline is constructed based on possible incentives/motivations. A pilot with strong motivation or in a mission with a high incentive usually has a relatively higher engagement level. In this paper, for simplicity, we set the engagement to a constant highest level.

2. Degradation/recovery. Engagement status usually changes due to workload/task change and/or occurrence of unexpected events. Expected events include regular ATC calls or corresponding replies. Although those expected events do not have a precise time schedule, their happening would not surprise the pilots. When expected events happen, the operator can be awakened so that his/her engagement level increases by a certain amount. On the other hand, unexpected events are those that the pilots are not prepared for. In our experiment, the pilots were not aware of the pump failure event in advance. We hypothesize that a pilot has a more rapid engagement recovery when an unexpected event happens, and it can keep the pilot alert for a longer period of time, which indicates a slower degradation speed in engagement level.

3. Refinement based on strong indicators. Strong disengagement /engagement indicators based on measurements (such as eye closure/head drooping due to fatigue indicating a disengaged state; shorter R-R interval or fast heart beat indicating an engaged state) shall be utilized for engagement refinement. In this paper, we used pilots' R-R (heart beat) interval as an indicator for their engagement level. High R-R interval values imply a relaxed stage in which the pilot's engagement level will degrade, and low R-R interval values indicate an engagement recovery stage. The degradation/recovery speed of engagement is individual dependent (based on boredom proneness for example) and is also dependent on an individual's physical fitness. The speed is controlled by the subjective evaluations, such as the boredom proneness scale and the real-time workload level. An easily bored operator usually gets distracted faster, and the lower the workload is, the faster the engagement level drops. In summary, the schema is shown in Figure 9.

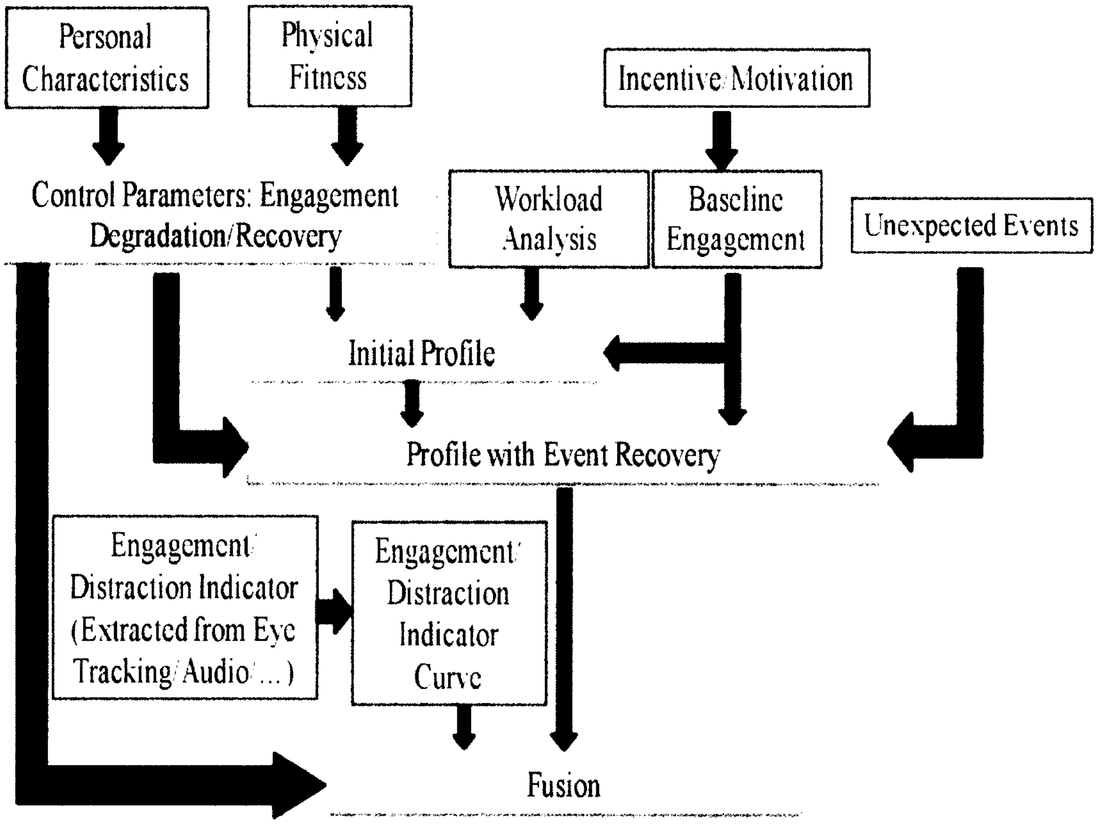

Personal Characteristics
Physical Fitness
Incentive/Motivation
Control Parameters: Engagement Degradation/Recovery
Workload Analysis
Baseline Engagement
Unexpected Events
Initial Profile
Profile with Event Recovery
Engagement/ Distraction Indicator (Extracted from Eye Tracking/Audio/...)
Engagement/ Distraction Indicator Curve
Fusion

Figure 9: Engagement ground truth finding

3.3 EEG DATA PROCESSING AND FEATURE EXTRACTION

Our engagement assessment model uses power features calculated from the collected EEG recordings. The literatures suggest that some EEG channels are highly correlated with engagement and other cognitive states. Berka et al. [20] suggested bi-polar sites, Fz-POz, Cz-POz for engagement assessment, C3-C4, Cz-POz, and F3-Cz, F3-C4, Fz-C3, Fz-POz for workload assessment. A bipolar EEG signal is the potential difference between two EEG electrodes, which can be directly recorded if there is an EEG amplifier for each pair of electrodes. It can also be derived from unipolar measurements (e.g. $Potential_{C3-C4} = Potential_{C3} - Potential_{C4}$). Trejo [60] emphasized that mental fatigue was associated with Fz, P7 and P8. Pope [61] studied the "combined" powers of Cz, Pz, P3, and P4 for evaluating indices of operator engagement. We started with all of channels mentioned in these studies, but the data collected from sensors Fz and P4 were not adopted due to hardware problems that made the system fail to record signal (value is 0) or saturated (a very large number). Since the EEG sensors we used (actiCAP) did not provide signals from channel POz, we selected Oz as a substitute, which is the nearest sensor to POz. To make it comparable, the sensors P7, P8, Pz, P3 were paired with Oz, respectively. The final selected EEG sensors were Cz-Oz, C3-C4, F3-Cz, F3-C4, P7-Oz, P8-Oz, Pz-Oz, and P3-Oz.

EEG recordings are known to be contaminated by both physiological and non-physiological artifacts [107]. In this work, we developed a procedure of preprocessing as shown in Figure 10 to remove the artifacts in EEG recordings. First, spikes, amplifier saturations, and excursions were identified and removed from EEG recordings. Base on Berka et al.'s model [20], spikes and excursions can be identified when the EEG amplitude changes significantly (e.g., $> 40uV$) over short durations (e.g., $12 - 27ms$) and saturation can be considered when the difference between two adjacent data points exceeds the predefined thresholds. In our experiment, the spike, excursion, or saturation was recognized if the difference of the maximum and minimum value of the adjacent 6 points (30 ms) was over 3 times that of the STD of the channel. The detected spikes, excursions, and saturations were rejected as in Figure 11 . Second, a high-pass filter with 0.5-Hz cutoff frequency is designed to remove baseline drift and a 60-Hz notch filter was implemented to delete the electrical interference. Finally, we implemented a Wavelet-based method to remove physiological noises such as ocular and muscular artifacts [108]. Our EEG signals were sampled at 200 Hz. The EEG data was decomposed using a six level stationary wavelet transformation, yielding a set of wavelet bands: 0-1.56, 1.56-3.13, 3.13-6.25, 6.25-12.5, 12.5-25, 25-50 and 50-100 Hz. For each wavelet band, the mean and standard deviation of the coefficients were calculated. Coefficients in the band were set to its mean if the absolute difference between the coefficient and the mean was larger than 1.5 times of the standard deviation in that band. Finally, the EEG signals were reconstructed from the modified coefficients. Figure 12 shows an example of EEG signals before and after artifact removal.

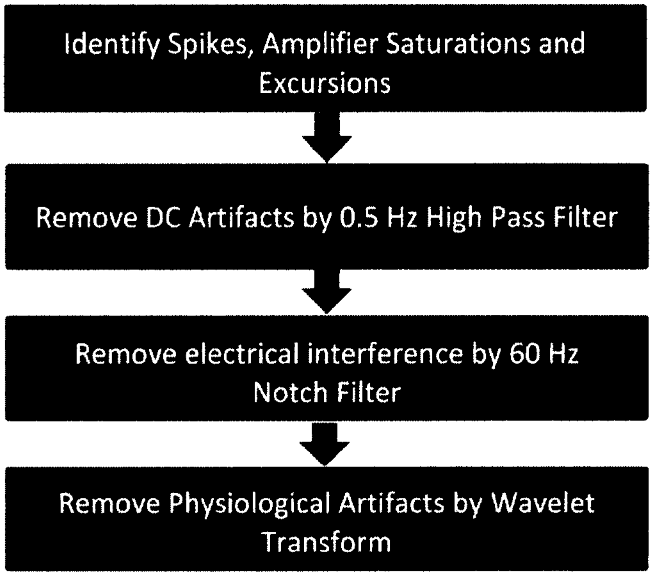
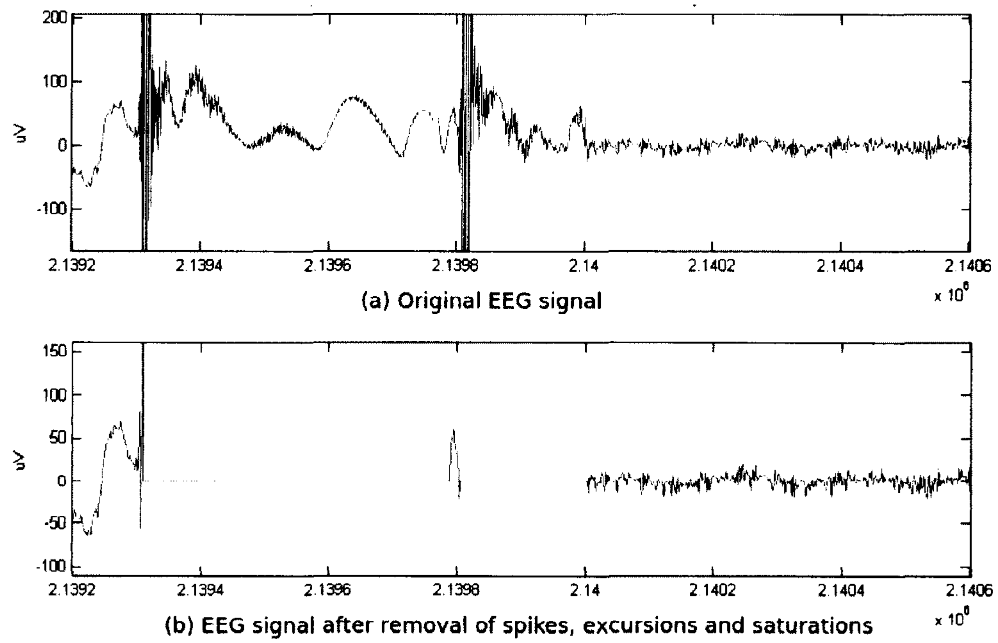

Identify Spikes, Amplifier Saturations and Excursions
Remove DC Artifacts by 0.5 Hz High Pass Filter
Remove electrical interference by 60 Hz Notch Filter
Remove Physiological Artifacts by Wavelet Transform

Figure 10: EEG artifact removal.

The decontaminated EEG signals were divided to three-second EEG segments with two-second overlapping between adjacent EEG segments. For each three-second EEG segment, the power spectral density (PSD) values from 1 Hz to 40 Hz with 1Hz resolution were computed as features. Therefore, each channel resulted in 39 features, generating 312 (39 × 8) features from all 8 channels.

3.4 FEATURE ANALYSIS

To better understand those computed features, we performed a feature analysis in our study. There are two main goals we wanted to achieve from feature analysis. The first was to find the most valuable features for each individual subject, to verify whether the feature extraction method was effective by observing the distribution of the selected features. Another goal was to analyze variance among subjects in the feature space, which could inspire new methods for model individualization.



(a) Original EEG signal

(b) EEG signal after removal of spikes, excursions and saturations

Figure 11: Spike, excursion and saturation identification and rejection.

To find valuable features, we used the Fisher score to rank features for each subject. Because of the variance among subjects, the top ranked features for different subjects may not be the same. To find features that were important to all or most of the subjects, we calculated the histogram of the selected features, which provided the probability of one feature being selected among subjects. Another tool that we employed for feature analysis is one-way ANOVA. It provides an intuitive way to present the distribution of the data points belonging to different groups/states of each individual or across subjects. To evaluate the effectiveness of the model individualization methods that directly adjust the value of features to mitigate differences among subjects, the one-way ANOVA is also a good tool to use.

3.5 FEATURE COMPARISON

In [61], Pope et al. evaluated indices of operator engagement in automated tasks. They concluded that the index constructed according to the formula, $betapower/(alphapower + thetapower)$, reflected task engagement. We designed an experience to compare the performance of Pope's index, the PSD of common EEG bands (theta, alpha, beta, and gamma), and our 312 1-Hz PSD bins. The number

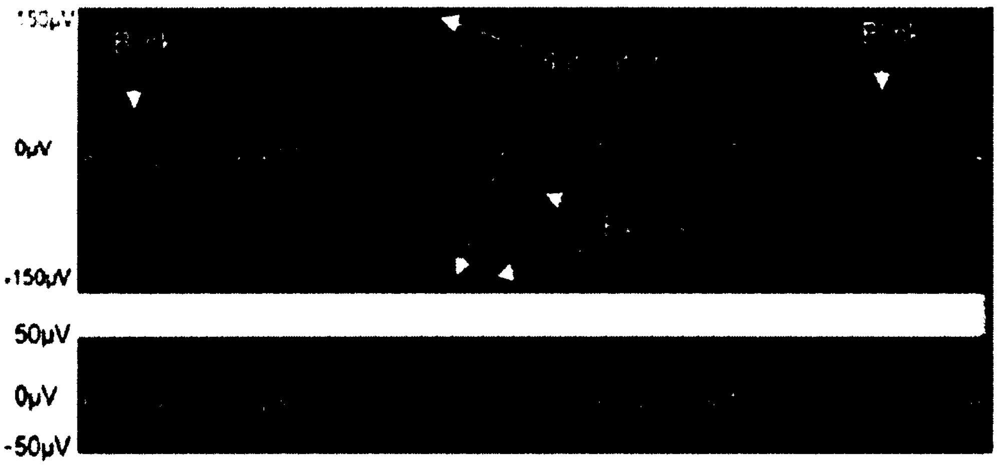


Figure 12: EEG signal before and after artifacts removal

of features for Pope's index and other EEG bands were all eight, since there were eight channels being selected. For each type of feature, a linear SVM classifier was trained using top 20% data samples and then it was applied to the remaining 80% data samples. The classification accuracy for each type of feature was calculated for comparison. The relationship between the features and their channel and PSD bin information is shown in Table 6.

Table 6: Feature index and corresponding EEG channel / PSD bin

Feature index	**EEG channel**	**PSD bins (Hz)**
1 - 39	Cz - Oz	1 - 39
40 - 78	C3 - C4	1 - 39
79 - 117	F3 - Cz	1 - 39
118 - 156	F3 - C4	1 - 39
157 - 195	P7 - Oz	1 - 39
196 - 234	P8 - Oz	1 - 39
235 - 273	Pz - Oz	1 - 39
274 - 312	P3 - Oz	1 - 39

3.6 RESULTS AND DISCUSSIONS

3.6.1 FEATURE ANALYSIS

To find the most valuable features, we calculated the Fisher score for each feature and ranked the features based on the Fisher score. Table 7 lists the top 15 features for each subject. The result is presented directly using the feature index from $1-312$. Find the corresponding channel and frequency information from Table 6. From the table, we can hardly find common features between subjects, which implies significant difference or individual variation between subjects.

Table 7: Top 15 ranked features.

Subject	**Ranked Top 15 features**														
1	39	273	38	37	272	36	271	35	30	33	312	270	32	267	117
2	187	304	307	298	306	297	301	311	290	294	296	302	176	291	292
3	70	57	81	72	174	150	78	109	71	94	179	69	180	59	187
4	120	121	81	119	82	122	80	118	83	125	234	230	123	229	124
5	227	305	230	273	211	266	228	231	233	234	222	306	220	223	225
6	312	311	310	57	133	56	95	134	97	289	94	96	281	135	288
7	152	153	107	146	147	108	104	154	114	103	113	151	156	143	145
8	4	199	3	237	238	221	277	276	197	159	5	198	196	220	160
9	198	237	236	235	197	2	196	159	276	199	275	3	1	274	193
10	82	117	305	309	86	81	306	121	311	310	35	270	308	303	36
11	192	193	194	187	195	310	309	304	232	221	311	189	183	191	231
12	39	156	207	135	133	11	139	208	212	132	206	41	227	149	138
13	273	272	271	270	269	251	268	264	263	262	267	266	254	265	260
14	188	189	210	226	178	230	225	224	209	222	190	223	221	231	211
15	9	165	243	204	8	282	244	126	283	242	38	234	164	203	281

To further study the effectiveness and the distribution of the features, we designed three methods, as below.

Check effectiveness of features. For each subject, we plotted the two highest ranked features to see if we could distinguish the samples belonging to engaged and disengaged states. Figure 13 is an example of such a figure for four subjects. This figure visualizes the effectiveness of the extracted features and shows that, for these subjects, the highest two ranked features could effectively discriminate the engaged and disengaged states, which proved that we had extracted effective features.

Find distribution of high ranked common features across subjects. To identify the most highly ranked common features, we calculated the histogram of the top 30 features from all subjects (Figure 14). There were 40 features being selected more than four times and all eight bi-polar sites are involved. The distribution of those features about which EEG channel they were recorded on and what frequency range they fell in is presented in (Table 8). It shows that most of the features were in the frequency range of 14 – 40 Hz, which crosses the alpha, beta and gamma

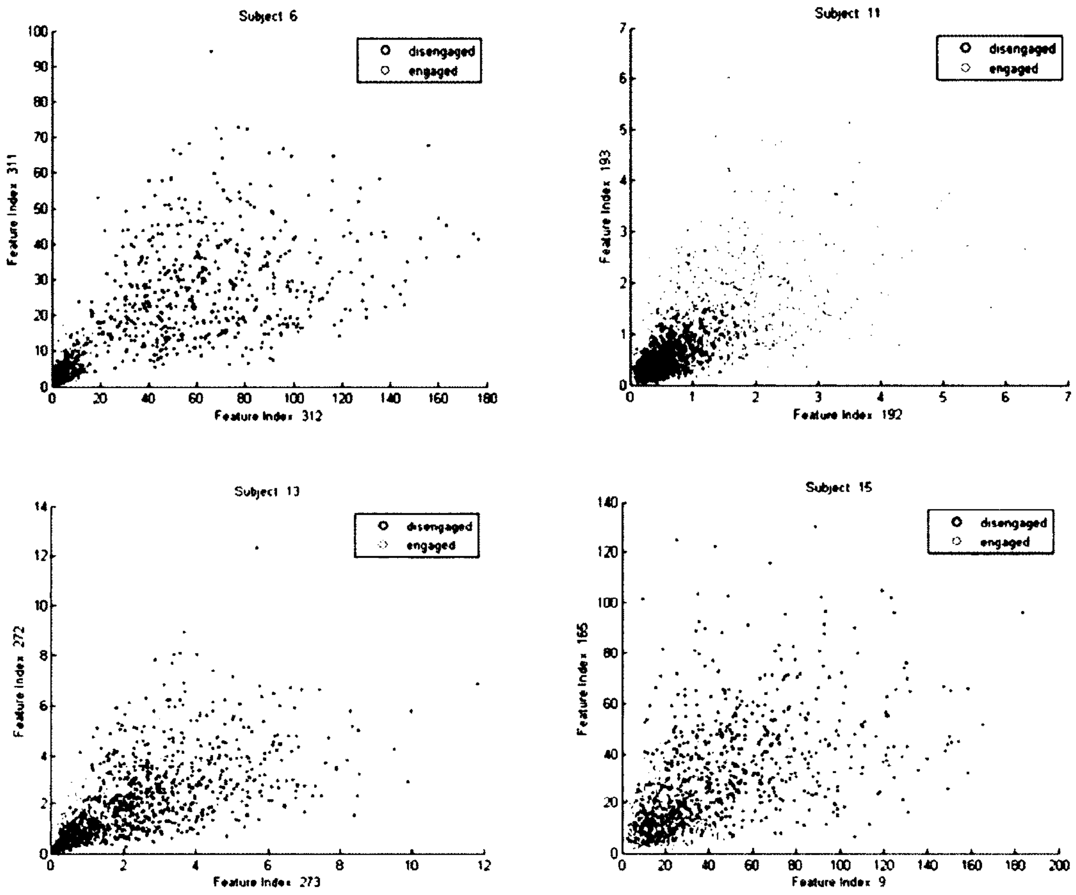


Figure 13: Top two ranked features.

EEG bands. From this result, it seems that the engagement state is not related with theta EEG band, which is different from Pope's index $(betapower/(alphapower + thetapower))$ that is determined by the theta, alpha, and beta EEG bands.

analyze the distribution of a common feature for different subjects. From the previous histogram analysis, we found that the most frequently selected feature was the 39 $\sim$ 40 Hz PSD bin from the EEG node pair P3-Oz. We applied the ANOVA analysis to this feature for all subjects and the result is shown in Figure 15. We observed significant differences between the engaged and disengaged states for some subjects, such as subjects 1, 2, 3, 5, 9, 10, and 15. We also found that for subjects 1, 2, 3, 5, and 9, the means of the feature for the engaged state were larger than those for the disengaged state. However, the tendency is reversed for subjects 10 and 15.

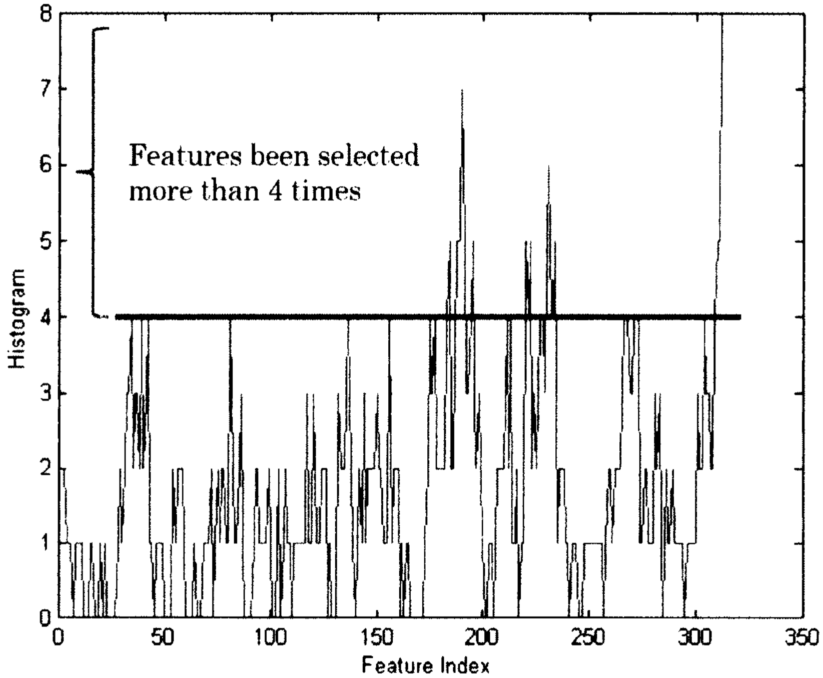


Figure 14: Histogram of the highest ranked 30 features for 15 subjects.

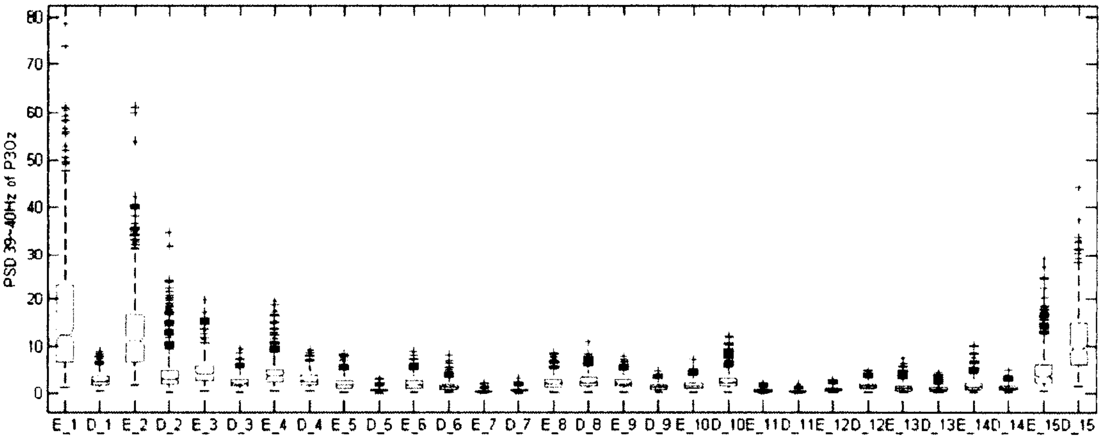
Overall, it can be concluded that effective EEG features have been extracted, but there is a large individual variance among the different subjects in the feature space.

3.6.2 FEATURE COMPARISON

We calculated classification accuracy to compare the performance of different types of features, including the PSD of common EEG bands (theta, alpha, beta, and gamma), Pope's index ($betapower/(alphapower + thetapower)$), combination of five types of features (theta band, alpha band, beta band, gamma band, and Pope's index), and our 312 1-Hz PSD bins. To do a quick comparison of the performance for the different types of features, we chose the top 20% of labeled data to train a linear SVM classifier and then applied the trained model on the remaining data. The reason why we chose top 20% data for training is explained in 4.3.1. The result shows that the combination feature and feature types of 312 1-Hz PSD bins outperformed the other types of features (Table 9). We expect that the performance using 312 1-Hz PSD bins as features could be further improved by utilizing more advanced techniques (enhanced committee machine and deep learning) that will be studied in this dissertation. So we will continue to use the 312 1-Hz PSD bins as features for

Table 8: Distribution of high ranked features.

	1-4Hz	5-7Hz	8-13Hz	14-24Hz	25-40Hz
Cz-Oz	0	0	0	0	2
C3-C4	1	0	0	0	0
F3-Cz	0	1	0	0	0
F3-C4	0	0	0	1	1
P7-Oz	0	0	0	2	9
P8-Oz	0	0	0	2	10
Pz-Oz	0	0	0	0	6
P3-Oz	0	0	0	0	5



For label of x axis, E denotes engaged, D denotes disengaged, numbers denote which subject.

Figure 15: ANOVA analysis.

the following research work.

Table 9: Feature comparison

Feature for comparison	Classification accuracy (%)
theta band	72.99
alpha band	71.55
delta band	73.40
gamma band	73.47
Pope's index	72.65
theta + alpha + delta + gamma band + Pope's Index	81.76
312 1-Hz bins	83.40

Chapter 4

MODEL INDIVIDUALIZATION

We proposed an individualized engagement assessment system, which consists of three modules, EEG signal processing, the enhanced committee machine and model individualization as shown in Figure 16. The EEG signal processing module cleans the EEG recordings and extracts features from the cleaned EEG data for classification. The enhanced committee machine trains multiple models based on the extracted features and the model individualization module implements the dynamic classifier selection strategy for model individualization. The module of EEG signal processing has been illustrated in Chapter 3. We will focus on the enhanced committee machine and model individualization in this chapter.

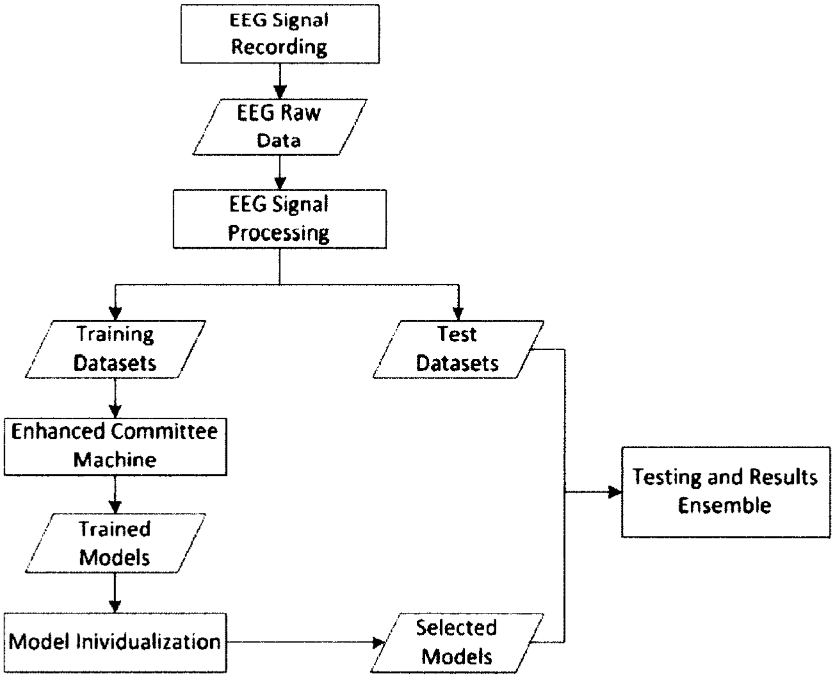


Figure 16: Individualized engagement assessment system.

4.1 ENHANCED COMMITTEE MACHINE

A committee machine is a strategy to improve classification/regression performance by combining results from multiple committee members (Figure 17). A theoretic interpretation to the improvement is that errors from individual committee members can be cancelled to some extent if they are uncorrelated [109]. Furthermore, since the committee machine "averages" its individual member's estimation, the variance of the final estimation can be significantly reduced. As a consequence, the performance of the combination of the estimation from committee members is often more superior and stable than that of any committee member.

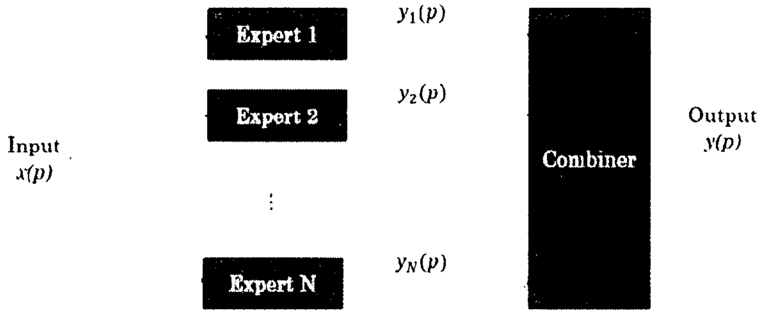


Figure 17: Original committee machine.

To enhance the committee machine, we would like to train more diverse committee members. The training procedure shown in Figure 18 presents how diversity property is achieved. First, each model was trained using datasets from one subject only, instead of combining all of the datasets together and training one general model. Second, multiple models were trained using a bootstrapped dataset from each of those subjects. Collecting EEG signals from pilots is expensive; using the above techniques we can train multiple models for each subject and simultaneously attain diversity among those trained models because each of those models was trained based on one subject's data only. Third, an advanced feature selection algorithm, PLOFS [110], can be utilized to select different features for each committee member. Finally, to make the model more diverse, we can train models using different classifier algorithms (SVM or multilayer perceptron classifier (MLP) for example). As a result, a set of models based on different datasets, features and classifier algorithms can be obtained.

During the test procedure, only part of the models will be selected, based on specific criteria, and the selected models become valid committee members. Our model individualization methods are implemented and integrated to the committee machine at this step (Figure 19). Finally, all of the selected models are applied on

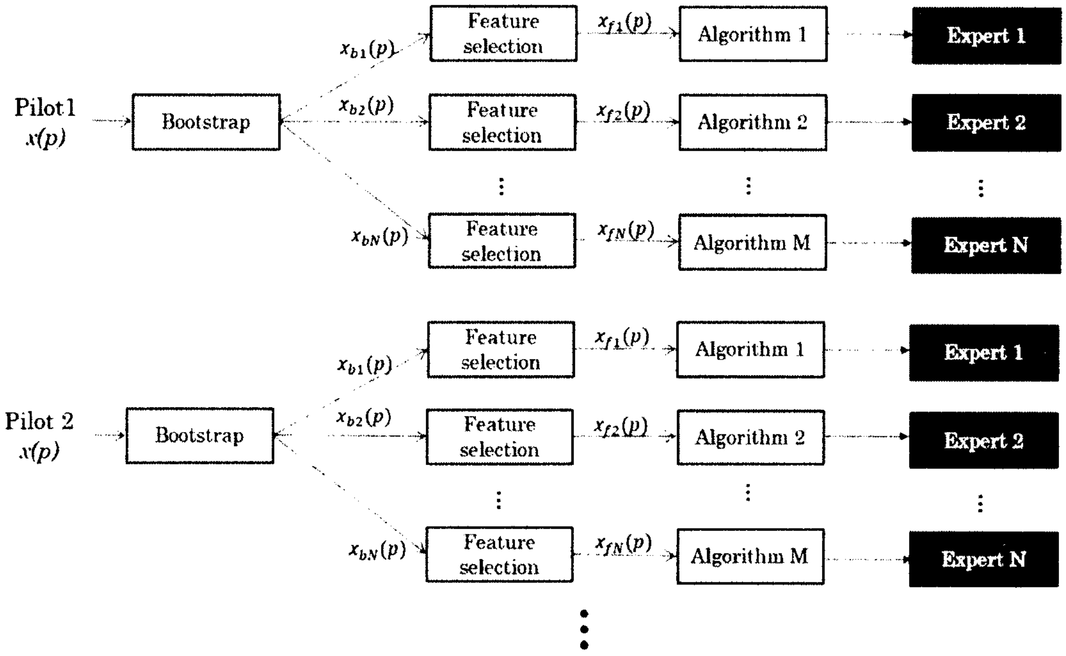


Figure 18: Training procedure of enhanced committee machine.

testing data and the outputs are combined using majority voting scheme.

4.2 MODEL INDIVIDUALIZATION

4.2.1 BASELINE NORMALIZATION

It has been observed that there were significant differences in the PSD features among different subjects. To mitigate the individual variance, we normalized each data set before training. For each subject to be tested, we assumed that a small set of data samples were available (i.e., from baseline experiments) before the experiment, and normalized the feature data of the subject using the means and standard deviations computed from the subjects' available dataset. This method ensured that the training data sets from different subjects and the testing data would be in the same scale in the feature space.

4.2.2 SIMILARITY-BASED MODEL INDIVIDUALIZATION

In many cases, it may be infeasible or too expensive to label enough data for

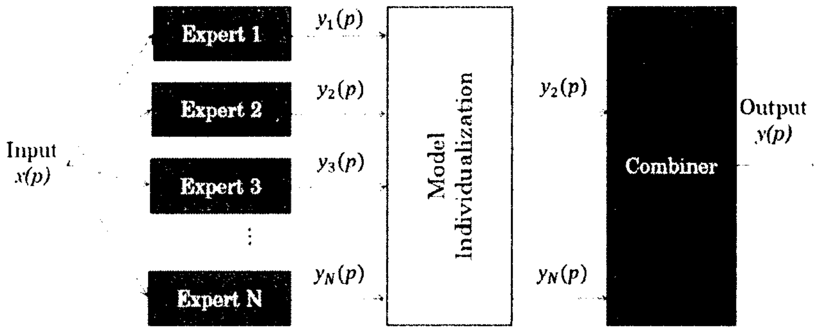


Figure 19: Test procedure of enhanced committee machine.

training an individualized OFS model. Though we observed that variance usually existed among different subjects, we also hypothesized that with enough subjects available, we might be able to find some subjects who would show some similarities in the feature space. Our motivation was to improve cross-subject performance by identifying similar subjects and directly use models from the identified "similar" subjects. In this dissertation, we investigated three different similarity measures, relative entropy, Bhattacharyya distance, and ROC, for similar subject identification. Relative entropy [111], also known as Kullback-Leibler distance or divergence between two probability density functions $f(x)$ and $g(x)$, is defined as,

$$D(f\|g) = \int f(x) log \frac{f(x)}{g(x)} dx \tag{2}$$

and the divergence satisfies three properties:

- Self-similarity: $D(f\|f) = 0$.
- Self identication: $D(f\|g) = 0$ only if f = g.
- Positivity: $D(f\|g) \geq 0$ for all f, g.

For two Gaussians $\hat{f}$ and $\hat{g}$,the KL divergence has a closed formed expression,

$$D(\hat{f}\|\hat{g}) = \frac{1}{2}\left[log \frac{\sum_{\hat{g}}}{\sum_{\hat{f}}} + T_r[\sum_{\hat{g}}^{-1} \sum_{\hat{f}}] - d + (\mu_{\hat{g}} - \mu_{\hat{f}})^T (\sum_{\hat{g}})^{-1} (\mu_{\hat{g}} - \mu_{\hat{f}}) \right] \tag{3}$$

Bhattacharyya distance [112] provides the upper and lower bounds of the Bayes error. For two normally distributed classes, the Bhattacharyya distance is defined as follows:

$$b = \frac{1}{8}(\mu_2 - \mu_1)^T \left[\frac{\sum_1 + \sum_2}{2}\right]^1 (\mu_2 - \mu_1) + \frac{1}{2} ln \frac{(\sum_1 + \sum_2)/2}{|\sum \sum_1 |^{1/2} | \sum_2 |^{1/2}} \tag{4}$$

where u_i and $_i$ are the mean vector and covariance matrix of class i, respectively.

The Receiver Operating Characteristic (ROC) curve [113] provides a useful method to evaluate diagnostic accuracy of a test and to compare the performance of different tests for the same outcome. The ROC curve is a graph of sensitivity against 1-specificity. If two data sets cannot be separated, the ROC curve of the discriminative test between the two data sets is a straight line with a slope of 1, and the area under the ROC curve (AUC) has a value of '0.5'. The larger the AUC value is, the easier to separate the two data sets. Two similar data sets can be identified if the AUC value is close to '0.5'.

4.2.3 DYNAMIC-BASED MODEL INDIVIDUALIZATION

Dynamic classifier selection has been proposed for many applications [114, 115, 116, 117]. To the best of our knowledge, it has not been utilized for model individualization. In dynamic classifier selection, the data is divided into three parts: the training data set, the validation data set, and the testing data set. The training set is used to train a set of classifiers. The validation data set is utilized to evaluate the performance of the trained models. In the testing phase, for each testing data point, its k-nearest neighbours in the validation set are first identified. Then, all trained models are applied on those neighbours and only those classifiers with performance better than a threshold will be used as a committee members for the testing point.

For each testing data point, the selected nearest neighbours are different; thus, the procedure is dynamic. Figure 20 shows the system diagram of the dynamic ensemble selection method. For test sample X, we first found the k nearest neighbours of the test sample in a validation dataset. We then used the k neighbours to evaluate the available classifiers and selected a subset of the classifiers, which could best classify those neighbours. Finally, we utilized the selected ensemble to classify the given test sample. Figure 21 shows details in the feature space. For the given test sample represented by the Red Cross, a set of nearest neighbours (blue circles) are found in the validation dataset. Note that there are several different decision boundaries (blue curves) formed by the available trained classifiers. Those blue circles were then used to evaluate all classifiers and a set of best classifiers was then be selected to classify the test sample.

It is worthy to note that this technique is different from the similarity-based individualization technique, which is based on a static selection procedure. In dynamic ensemble selection, each classifier's accuracy is estimated in the local feature space surrounding the data similar to an unknown test sample from the individual. The first few top classifiers are then selected to classify the test sample by majority voting. The rational of this design is based on the assumption that if the test sample is similar to its local validation samples in the feature space, we may achieve a good assessment result for the test data point by utilizing those classifiers which perform well on the adjacent validation data points.

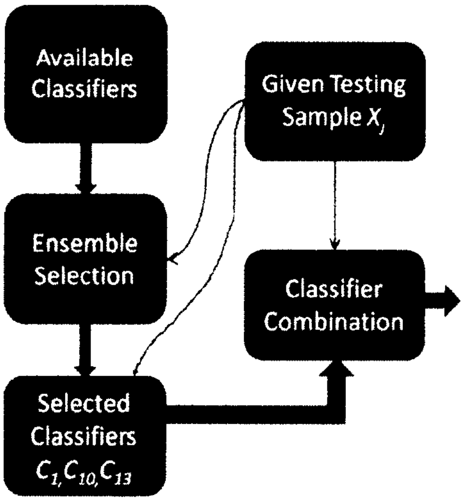

Available Classifiers
Given Testing Sample X_j
Ensemble Selection
Classifier Combination
Selected Classifiers C_1, C_{10}, C_{13}

Figure 20: System diagram of dynamic ensemble selection.

4.3 RESULTS

The proposed system was evaluated with experimental data collected from 15 subjects (pilots). For each subject, 20-minute data were labeled using the method described in [9] as engaged or disengaged. The labeled EEG signals were preprocessed using the steps described in section 3.3. The PSD features were then computed and fed into the enhanced committee machine training framework, which resulted

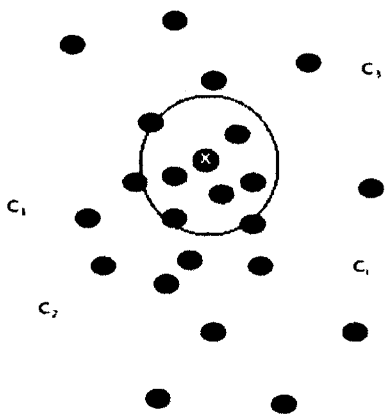


Figure 21: Detailed illustration for ensemble selection.

in engagement assessment models. We also performed feature analysis and model individualization and those results are presented in this section.

4.3.1 BASELINE NORMALIZATION FOR MODEL INDIVIDUALIZATION

The feature analysis sections showed the individual variance among subjects in the feature space making a general model usually not performing well. We utilized the baseline normalization method to improve the modeling performance. In our experiment, we used the top 20%, or two-minute data of engaged and two-minute data of disengaged to calculate the parameters for normalization (mean and standard deviation). The top 20% data was also used as training data to train classification models. The reasons not to use whole data or different numbers of data(top 1%, 3%, 5%, 10%, 30%, ...) to calculate the parameters for normalization are based on the following considerations,

- We considered engagement assessment as a real-time problem and in a practical situation, the raw EEG data was fed as a stream data. So we wanted to collect a small amount of data at the beginning period of the experience for normalization parameters, in stead of using whole data.
- The proportion of top 20% was determined by empirical experiments. We wanted the amount of data for normalization parameters to be as small as possible, but we also wanted to be able to normalize the new coming data to

be roughly in the range [-1, 1] with mean 0. We tried several portions of data (top 1%, 3%, 5%, 10%, 30%, ...) and found that 20% worked well enought to balances the requirement using less data but having good performance.

In order to evaluate the effect of normalization approach, the performances of committee machine using models trained from non-normalized features and normalized features were calculated and compared. Table 10 shows the cross-subject classification accuracy using models trained from non-normalized features. The first column of the table is the index for the subject to be tested, and the first row is the model index trained from the subjects' data. The last column is the classification performance using majority voting based on testing results in the corresponding row. The table shows that the performance of the cross-subject classifiers was fairly poor. Many of them performed not better than a random classification, i.e. the classification accuracy was around or less than 50%. It reminded us the challenge of the problem of individual variations.

Table 11 shows results using normalized features. It shows that the performance of many classifiers dramatically increased. For example, for subject 1, the accuracy using classifier from subject 11 increased from 50.18% to 77.79%. If we denote this example as $AccImprove(1, 11, 50.18, 77.79)$, we can find a number of similar examples, such as $AccImprove(2, 1, 53.96, 86.27)$, $AccImprove(3, 11, 50.32, 81.99)$, $AccImprove(6, 8, 57.4, 79.98)$, and $AccImprove(14, 2, 58.11, 79.25)$ etc. Its voting results also surpassed those using non-normalized data. Figure 22 illustrates that the performance using normalized data was better than those using non-normalized data for most subjects. The statistical analysis also proved this. The mean accuracy increased from 60.17% to 68.51% and the paired t-test result shows the normalization method was significantly better (p-value = 0.0049)

4.3.2 SIMILARITY-BASED MODEL INDIVIDUALIZATION

To study the effectiveness of the similarity-based model individualization technique, we first identified a similar subject to the subject to be tested, using the similarity measure (entropy, Bhattacharyya distance or ROC) in the training data pool. We then use the identified similar subject's data in order to build a model for the testing subject. The similarity metric was computed using the top 20% data of the testing subject with all of the data from each of the other subjects. The evaluation results for the three similarity metrics are shown in Table 12, Table 13,

Table 10: Classification accuracy based on non-normalized data.

Subjects	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Voting
1		46.34	52.7	60.86	51.38	65.31	49.94	51.62	51.14	50.06	50.18	46.7	50.06	49.94	55.34	65.67
2	53.96		40.99	55.48	84.37	45.09	54.62	47.86	45.38	24.02	45.38	65.3	44.9	60.82	28.88	45.76
3	59.81	50.16		61.09	57.23	50.16	49.84	58.36	55.31	54.66	50.32	42.44	50.16	53.86	49.36	62.06
4	63.26	53.46	62.77		49.42	46.54	52.31	67.38	66.56	46.54	46.62	50.58	46.54	53.46	47.12	70.26
5	87.81	52.83	69.7	70.85		47.17	12.81	75.35	76.41	48.32	81.54	46.47	47.17	85.78	46.91	82.07
6	56.22	40.13	33.18	49.27	55.03		49.27	57.4	49.36	50.64	50.73	51.46	81.81	70.75	72.85	70.02
7	55.48	55.58	33.26	42.15	64.15	45.25		82.33	38.02	44.63	22.21	63.64	44.42	72.42	55.17	50.83
8	53.56	49.95	50.05	59.02	46.35	50.05	50.69		53.19	51.62	49.4	51.53	50.05	53.19	50.05	56.34
9	75.61	50.04	47.2	62.49	65	49.96	47.95	77.61	-	44.11	55.39	52.13	49.87	59.65	49.04	75.94
10	32.99	50.6	50.52	43.9	23.63	49.4	60.82	60.82	50.6		42.78	52.41	49.4	46.31	49.66	44.07
11	52.07	50.64	82.3	59.19	68.84	49.36	43.1	53.6	49.53	49.36		49.36	49.36	66.38	49.36	51.14
12	50.62	49.02	38.7	32.03	50.18	51.33	50.53	55.25	52.22	50.98	50.09		50.98	49.02	51.33	43.42
13	33.76	50.85	57.18	85.81	34.36	49.23	67.95	92.74	53.42	67.52	47.78	42.74		88.03	49.74	72.65
14	70.8	58.11	59.11	78.05	87.61	41.89	56.7	66.26	63.24	40.08	52.37	25.38	41.89		40.79	66.67
15	45.6	48	57.2	42.3	45.1	45.6	54.4	55.2	45.6	58.5	45.6	56.2	45.2	60.5		45.6

Table 11: Classification accuracy based on normalized data.

Subject	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Voting results
1		68.07	57.26	64.23	76.59	32.17	32.05	66.87	77.55	29.53	77.79	57.14	23.17	71.19	54.86	67.35
2	86.27		80.84	44.52	86.56	10.39	42.52	74.83	82.08	20.69	90.37	41.28	12.11	86.94	25.74	75.69
3	54.34	77.81		73.63	63.67	19.77	26.53	51.29	74.76	64.15	81.99	46.14	41.64	60.45	41.16	67.68
4	62.44	43.82	65.65		49.09	60.79	36.24	74.79	71.75	76.44	53.13	51.65	67.71	57.33	68.53	72.49
5	85.69	81.71	81.45	73.85		16.43	21.38	75.18	78.45	37.81	83.83	50.18	17.49	86.13	41.25	82.33
6	71.39	34.19	33.73	70.57	57.04		58.59	79.98	56.67	63.25	38.94	43.51	69.65	79.34	73.4	69.93
7	46.18	42.36	44.73	40.91	48.66	81.51		82.02	61.16	44.32	34.61	57.54	79.55	69.42	43.9	63.53
8	51.43	47.27	51.9	61.24	53.1	60.04	58.19		69.47	57.72	47.73	56.06	63	62.16	39.04	67.53
9	79.11	76.69	69.42	63.41	68.42	27.57	38.93	76.27		36.68	78.95	58.65	26.9	81.2	45.86	78.11
10	26.72	40.21	63.49	59.79	35.48	69.67	44.59	52.15	46.48		36.68	53.09	79.38	33.42	57.9	45.45
11	76.63	89.33	90.09	64.01	83.66	11.26	22.44	72.99	79.85	39.8		48.52	32.77	82.64	26.42	76.8
12	42.79	51.78	41.01	36.92	46	50.09	62.99	51.87	52.85	29.18	47.15		36.12	51.78	56.94	41.81
13	44.53	29.57	61.62	70.77	50.43	82.99	73.5	90	57.44	74.53	43.25	33.5		83.16	64.62	81.2
14	70.49	79.25	75.23	51.26	86.4	27.59	63.04	81.17	82.38	30.21	78.15	44.81	35.65		41.99	85.5
15	26.8	35.9	35.1	59.3	30	78.9	71.7	36.5	43	63.3	16.6	78.7	82.6	48.2		52.3

Table 14.

We also calculated the cross-subject performance and tried to find its relationship with similarity matrixes. This is shown in Table 15, where each row shows the testing performances for the subject indexed by the number in the first column using the model trained by the dataset from the subject indexed by the number in the first row. For example, the number shown in red represents the testing accuracy on subject 2 using the model trained by data from subject 3. This table gives us an overall cross subject performances.

The value in the similarity matrixes indicates the ability to distinguish the data sets from each other in the feature space. So a smaller value means more similar. To evaluate which method is more effective to find similar subjects, we calculated the

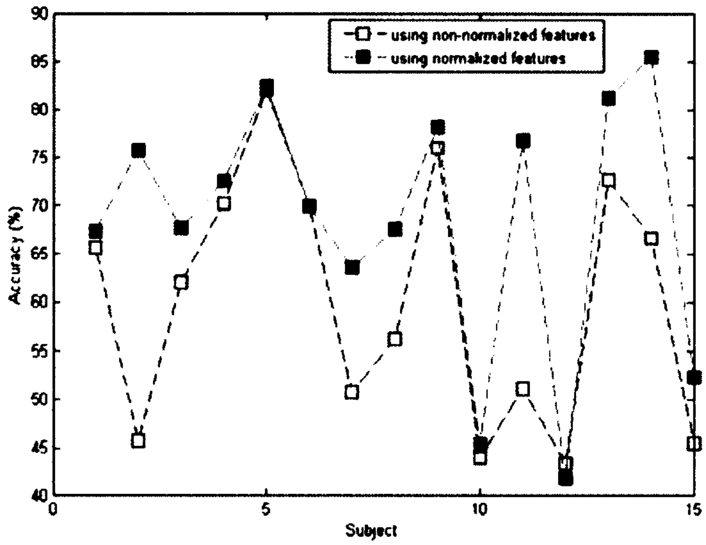


Figure 22: Performance comparison using normalized and non-normalized features.

correlation between the similarity matrixes and the performance matrix and achieved the values of -0.096, -0.154, and -0.277. It is obvious that the ROC metric is highly negatively correlated with the performance. The lower ROC value is between two subjects' dataset, the better the performance can be achieved. Based on the test, we concluded that the ROC metric was the best metric for the similarity-based model individualization method. The average accuracy using the models ranked the highest by ROC was 46.12%, much worse than the general cross-subject model.

Table 12: Similarity matrix of entropy.

Subject	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.05	11.19	10.37	20.78	194.69	63.03	39.32	73.28	129.98	124.75	47.70	15.08	13.48	153.72	24.92
2	10.67	0.16	5.09	28.16	20.84	5.55	135.90	29.38	40.98	25.19	117.36	36.73	18.39	21.93	4.62
3	18.58	15.26	0.22	1.61	3.26	55.92	4.84	2.24	6.14	2.61	5.06	1.74	1.22	2.64	20.98
4	16.08	89.12	8.04	0.13	6.89	538.83	2.54	4.33	2.95	7.65	2.52	1.47	9.68	5.59	33.33
5	50.07	18.78	5.73	4.45	0.11	52.33	9.96	2.71	2.81	1.59	5.88	9.78	0.49	0.74	10.36
6	37.93	15.45	2.95	22.14	5.16	0.13	32.05	22.20	34.38	10.10	101.10	54.73	27.03	1.55	43.24
7	43.29	81.42	5.22	2.51	28.02	580.27	0.27	4.32	11.60	11.59	8.98	1.34	20.18	18.59	69.97
8	59.15	56.68	5.80	3.52	3.74	68.67	4.59	0.20	5.82	2.38	5.70	5.10	5.11	3.38	66.37
9	63.52	66.92	13.34	7.90	0.94	113.40	10.60	5.04	0.10	1.09	2.87	12.62	2.95	1.62	18.35
10	92.96	52.26	3.70	5.85	0.90	61.40	8.13	1.68	2.65	0.14	4.24	10.93	2.00	1.26	33.20
11	68.22	81.41	7.73	3.75	3.24	817.60	2.49	6.62	4.15	4.87	0.11	4.48	37.87	3.79	51.13
12	13.68	20.29	1.03	1.85	10.61	78.18	1.46	3.04	6.41	6.25	3.43	0.21	5.20	7.66	9.34
13	54.80	24.99	4.89	8.03	0.49	43.66	17.05	2.69	2.78	1.20	4.24	9.92	0.08	0.48	17.27
14	35.31	17.04	3.59	30.35	1.15	62.09	12.28	5.65	2.91	1.73	5.08	5.24	1.55	0.10	8.03
15	12.91	3.29	6.39	47.58	10.19	6.09	159.91	26.03	34.00	24.49	98.02	36.86	17.84	10.09	0.12

Table 13: Similarity matrix of Bhattacharyya distance.

Subject	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	0.04	4.52	1.96	2.31	4.91	2.10	1.81	3.54	2.69	3.19	2.34	2.53	0.64	2.41	1.18
2	1.93	0.02	1.11	3.50	3.78	0.66	2.89	2.99	2.22	4.92	2.31	1.64	1.05	1.72	3.98
3	2.56	3.58	0.02	1.52	0.52	0.84	0.30	0.32	0.56	0.17	1.08	0.35	0.09	0.15	3.80
4	1.99	4.61	0.32	0.02	2.31	1.40	0.55	0.49	0.45	1.32	0.93	0.30	0.27	0.19	1.99
5	2.72	1.30	0.11	1.63	0.15	0.93	0.44	0.47	0.39	0.51	0.48	0.48	0.05	0.06	2.62
6	3.04	1.24	0.35	4.51	4.94	0.02	1.85	1.20	1.14	5.10	2.01	1.73	0.28	0.15	6.75
7	2.26	3.38	0.30	1.01	1.25	1.44	0.04	0.47	0.60	0.61	0.50	0.13	0.38	0.33	2.79
8	3.35	4.64	0.44	1.01	1.81	1.29	0.39	0.04	0.69	0.55	1.18	0.76	0.23	0.24	3.21
9	2.42	2.08	0.39	2.44	1.23	1.46	0.86	0.58	0.02	0.18	0.40	1.06	0.16	0.09	4.50
10	3.46	3.97	0.33	2.11	0.30	1.21	0.48	0.11	0.53	0.03	1.07	0.64	0.14	0.11	4.44
11	2.34	2.50	0.35	0.64	0.83	1.37	0.33	1.53	0.60	0.72	0.03	0.83	0.49	0.20	2.44
12	2.03	3.00	0.12	0.51	1.34	1.07	0.14	0.66	0.58	0.65	0.51	0.10	0.18	0.30	1.94
13	2.60	2.26	0.10	1.69	0.25	0.96	0.40	0.41	0.38	0.39	0.33	0.57	0.00	0.04	3.47
14	2.36	2.01	0.12	1.63	0.21	0.95	0.32	0.51	0.38	0.37	0.36	0.36	0.12	0.02	2.69
15	1.33	1.15	1.49	2.73	3.74	0.90	2.80	3.23	2.21	3.64	2.36	2.20	1.65	1.58	0.10

4.3.3 DYNAMIC ENSEMBLE SELECTION FOR MODEL INDIVIDUALIZATION

To evaluate the dynamic ensemble selection approach for engagement assessment, five scenarios were designed as below:

Scenario 1: We utilized the first 20% of data from one subject for training (9 committee members in total) and the remaining 80% data for testing the same subject, thus obtaining the individual model performance.

Scenario 2: Generalized model performance. For each subject, we trained 9 committee members/models using data from each of other subjects. Since there were 15 subjects in total, 9 × 14 = 126 committee members were trained for the testing subject. This gave us the baseline generalized model performance.

Scenario 3: Everything was the same as that in Scenario 2 except that the dynamic ensemble selection technique was applied and the validation data was a combination of top 20% data from each of other subjects.

Scenario 4: Everything was the same as that in Scenario 3 except that the validation dataset for dynamic ensemble selection was from the testing subject (first 20%).

Scenario 5: Everything was the same as that in Scenario 4 except that the models trained from the top 20% of the testing subject in Scenario 1 were also added as candidates.

Table 14: Similarity matrix of ROC.

Subject	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	0.04	0.25	0.27	0.35	0.38	0.30	0.39	0.36	0.35	0.34	0.38	0.35	0.18	0.33	0.19
2	0.26	0.03	0.28	0.33	0.38	0.17	0.39	0.40	0.39	0.39	0.41	0.29	0.32	0.41	0.23
3	0.33	0.33	0.05	0.16	0.17	0.24	0.24	0.16	0.18	0.15	0.26	0.17	0.14	0.15	0.32
4	0.35	0.37	0.21	0.05	0.20	0.34	0.14	0.24	0.19	0.24	0.13	0.14	0.22	0.19	0.30
5	0.33	0.34	0.11	0.15	0.03	0.27	0.18	0.17	0.12	0.12	0.22	0.16	0.10	0.10	0.33
6	0.33	0.26	0.15	0.25	0.22	0.04	0.32	0.23	0.25	0.26	0.34	0.27	0.12	0.12	0.26
7	0.39	0.38	0.21	0.17	0.16	0.37	0.05	0.25	0.19	0.21	0.09	0.14	0.31	0.18	0.33
8	0.36	0.42	0.21	0.26	0.21	0.32	0.25	0.05	0.21	0.13	0.26	0.26	0.20	0.18	0.37
9	0.36	0.40	0.24	0.26	0.10	0.36	0.26	0.20	0.04	0.12	0.20	0.22	0.18	0.15	0.35
10	0.37	0.41	0.24	0.23	0.11	0.34	0.25	0.16	0.15	0.04	0.24	0.18	0.17	0.12	0.41
11	0.39	0.40	0.23	0.15	0.21	0.37	0.16	0.28	0.22	0.25	0.03	0.24	0.34	0.22	0.31
12	0.33	0.29	0.14	0.14	0.24	0.31	0.17	0.23	0.21	0.21	0.24	0.05	0.16	0.19	0.34
13	0.31	0.35	0.10	0.20	0.11	0.29	0.21	0.20	0.14	0.15	0.23	0.17	0.03	0.10	0.35
14	0.30	0.34	0.10	0.19	0.13	0.28	0.21	0.18	0.13	0.16	0.24	0.16	0.14	0.06	0.29
15	0.26	0.17	0.31	0.32	0.36	0.18	0.38	0.39	0.38	0.43	0.39	0.38	0.40	0.34	0.05

Table 15: Cross-subject performances.

Subject	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	99.1	70.16	60.57	65.07	78.71	28.19	26.99	64.02	77.81	31.93	78.41	58.32	21.29	67.92	56.07
2	83.43	97.85	84.03	43.03	86.89	10.13	43.15	74.26	82.48	24.08	89.03	43.86	14.06	85.46	26.58
3	47.99	74.5	97.99	72.69	57.03	22.69	30.92	44.98	69.68	63.05	77.51	45.38	44.38	53.82	45.38
4	62.92	45.83	62.2	98.97	49.54	61.28	41.92	74.36	70.44	73.12	50.05	54.99	67.46	59.73	69.21
5	83.65	78.45	78.9	74.7	96.91	19.23	22.76	72.04	75.69	39.01	80.22	52.71	19.78	84.09	47.07
6	77.14	39.09	40.46	76.69	61.49	100	51.77	85.03	62.86	63.31	.45.49	42.86	67.54	84.11	71.43
7	42.89	36.43	45.22	44.7	45.09	84.75	99.74	80.62	58.79	50.39	26.74	58.01	83.33	63.7	48.45
8	48.21	45.9	49.13	58.5	50.52	58.15	56.76	98.03	67.05	56.53	45.66	55.72	62.43	54.68	36.76
9	81.3	74.82	68.55	67.61	69.49	28.11	34.9	76.28	96.76	39.08	79.41	57.89	26.44	79.21	47.44
10	29.75	44.47	67.13	57.57	39.53	66.49	39.63	52.95	47.8	98.6	41.78	52.74	77.23	35.77	56.71
11	78.71	88.56	89.72	63.24	85.59	11.12	22.78	74.26	81.25	36.23	99.79	47.14	31.36	84.96	26.48
12	42.6	50.61	35.93	32.04	42.83	53.95	70.19	55.39	51.95	25.7	45.38	99.22	34.48	52.84	55.84
13	49.36	32.16	61.86	68.59	52.35	80.34	71.15	89.1	59.51	69.34	46.9	36.43	100	80.56	61.75
14	69.56	75.6	71.7	50.69	85.28	29.94	67.92	80.5	78.99	28.55	74.09	48.18	36.35	98.87	44.91
15	25.87	38.38	33.5	55.13	27.5	78.38	74.25	32.13	43	60.12	15.63	83.88	83.88	48.75	99.25

The five scenarios were summarized in Table 16 and the performance of engagement assessment for all the subjects is shown in Table 17.

It appears that the top 20% data from each subject is sufficient to train a reasonably good individual model for engagement assessment (Scenario 1). This performed better than the generalized model (Scenario 2). The dynamic ensemble selection strategy (Scenario 3) seemed not helpful if only other subjects' data is used for validation. However, the performance was boosted from 67.59% in Scenario 2 to 80.27% in Scenario 3 if the top 20% data of the testing subject was used as validation data, though the performance was still worse than the individual model, as in Scenario

1. If we also added the models trained from those top 20% to the model pool as candidates (Scenario 5), the performance was further improved to 86.47%, which is significantly higher than the individual model (p-value = 0.0013).

Table 16: Dynamic ensemble experiment setup.

Scenario	No. of Models	Ensemble method	Validation dataset
1	9	Majority voting	N/A
2	9*14	Majority voting	N/A
3	9*14	Dynamic	From all other subjects
4	9*14	Dynamic	Top 20% of data from the testing subject
5	9 + 9*14	Dynamic	Top 20% of data from the testing subject

Table 17: Dynamic ensemble results.

Subject	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
1	79.16	67.17	67.77	87.71	87.88
2	84.15	77.23	86.29	84.27	88.08
3	83.94	62.25	71.89	82.53	86.82
4	77.55	72.09	74.05	83.93	87.15
5	75.58	80.55	74.92	78.45	82.07
6	95.31	76.69	75.66	86.51	95.89
7	84.75	61.76	56.72	79.72	87.81
8	70.87	60.58	62.43	63.7	74.56
9	82.97	78.68	83.07	84.22	88.64
10	88.72	47.48	42.75	80.34	90.03
11	94.17	77.75	86.76	94.81	96.53
12	67.52	39.49	42.27	46.94	65.93
13	93.06	80.24	70.94	85.15	91.03
14	77.99	84.4	75.97	76.73	81.87
15	90.25	50.75	42.38	89	92.7
Mean	83.07	67.81	67.59	80.27	86.47
STD	8.36	13.68	15.24	11.51	7.91

* The first row is the index of scenarios and the first column is the index of subjects.

4.4 DISCUSSIONS

The results of feature analysis and individual model trained using top 20% data show that our EEG processing algorithm could successfully extract useful features for individual engagement assessment. The average classification accuracy was over 80% when only used the top 20% data for training. However, the cross-subject models performed poorly due to the individual variance. One direct piece of evidence was that there was no common feature found from the ranked features for the 15 subjects; for the subjects having some common features, those features were not in the same scale. Therefore, there was a need to individualize an average model that could perform well across subjects.

Normalization using the baseline experiment data set is a straight forward method and it improved the performance of the average model in our study. It is worthy to note that we selected top 20% data from engaged and disengaged data segments, respectively. In practice, it is difficult to collect disengaged data in a baseline experiment partially because the disengaged functional state is difficult to mimic. We tried to select the top 2-minute engaged data only for normalization, and the trained models tended to classify all testing data to the engaged state.

The similarity-based method for model individualization did not perform well in our study with low correlation coefficients among the cross-subject similarities and classification performances (the highest was -0.277). The classification accuracy using models from the most similar subject was also not good, much lower than the general model. A possible reason is that we didn't have enough data sets to guarantee a really similar subject for each testing subject.

The term "similar subjects" means the subjects have similar range of value in the feature space, but it does not mean they have similar distributions for different states. For example, from the ANOVA analysis (Figure 15), subject 2 and subject 15 are similar in the feature space of P3Oz of 39 – 40Hz PSD. However, the feature distributions for engaged and disengaged states were reversed.

The proposed dynamic classifier selection algorithm proved to be an effective model individualization strategy for across subject engagement assessment. We assumed that a small data set from a baseline experiment would be available for each subject. Those baseline data sets could be utilized to train an individual model for each subject for engagement assessment (scenario 1) or to individualize the average models trained from other subjects (scenario 4) or to do both (scenario 5). In scenario 5, we obtained the best performance that was even better that the individual model (scenario 1). This was possibly because the training data was limited in scenario 1; if we had started with the average model and use the baseline data to individualize the average model for the testing subject, the information in the average model might have helped the assessment performance (scenario 5). It was also observed that the baseline data played a critical role in the model individualization because, in our study, just using data sets from other subjects in the dynamic classifier selection did not help the performance (scenario 3).

4.5 CONCLUSIONS

This chapter explored an EEG-based engagement assessment mechanism considering individual variance in an aviation environment. We tried three individualization methods to improve performance of cross-subject classification. The normalization method significantly improved the performance, but similarity based methods seemed not to help. From the experience designed for the dynamic classification method (scenarios 3, 4, and 5), we found that if a small amount of baseline data were available for model training and were used as validation dataset, we could achieve better performance than we could with individual models.

Our enhanced committee machine provided a mechanism to integrate similarity-based and dynamic ensemble methods in an elegant way. From the view of committee machine, either the similarity-based method or dynamic ensemble method just provides a criterion to select appropriate models from the model pool. Such a mechanism also made it easy to implement in a real-time system.

Chapter 5

DEEP LEARNING FROM SCARCE LABELED DATA

We proposed an engagement assessment system in a complex task with scarce labeled data. The system consisted of three modules, EEG signal processing, deep learning, and classification,as in Figure 23. The EEG signal processing module cleaned the EEG recordings and extracted features from the cleaned EEG data (see Chapter 3). The deep learning module learned high level features from both labeled and unlabled EEG features. The classification module utilized SVM to evaluate engaged levels based on learned features using deep learning techniques.

We will focus on deep learning techniques in this chapter.

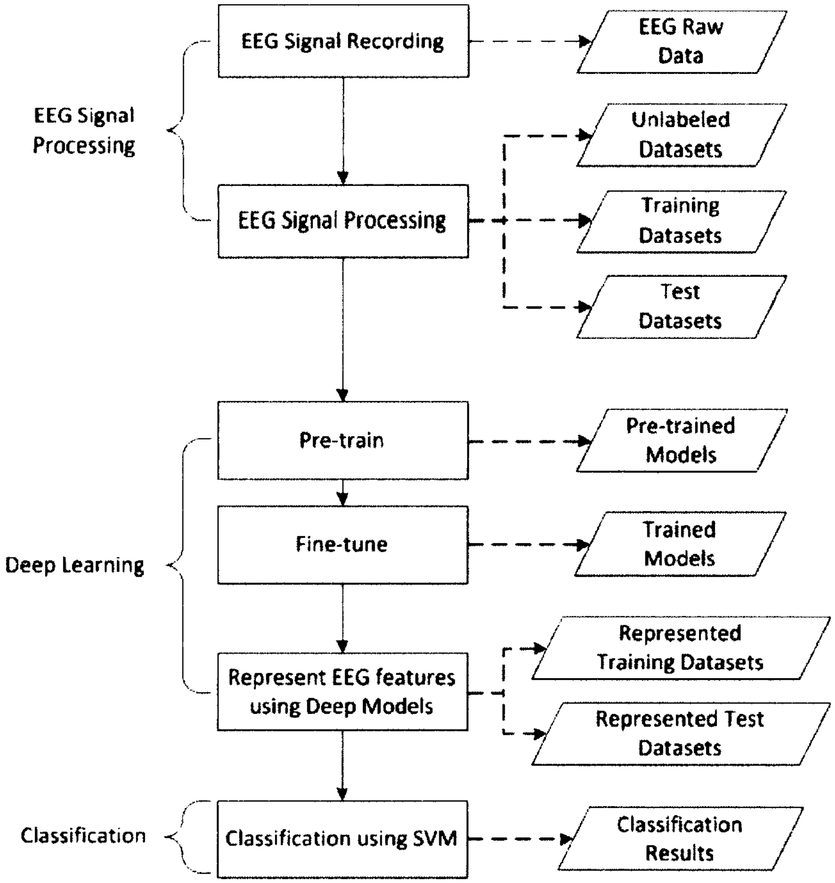


Figure 23: Deep learning system.

5.1 DEEP LEARNING MODELS

In contrast to the traditional three-layer neural network (shallow structure), deep learning is based on a deep architecture consisting of many layers of hidden neurons for modeling. A shallow architecture would involve many duplications of effort to express things; such a fat architecture has been shown to suffer from the problem of over-fitting, which leads to a poor generalization capability. Instead, deep architecture could more gracefully reuse previous computations and discover complicated relations of input [118].

To train a deep architecture, the standard Backpropagation (BP) algorithm did not work well with randomly initialized weights because the error feedback became progressively noisier as it went back to lower levels (closer to inputs), making the low-level weight updates less effective. Even though experiments have shown that if the top layers have enough units, the deep structure can still bring down training errors to be small enough, it cannot generalize well to new data [99]. This is because the top layers can be effectively trained by gradient based algorithms but low-layers cannot. The randomly initialized low-layer layers behave like random feature detectors so good representations for original data were not achieved, leading to degraded generalization capability [99]. In 2006, a breakthrough in deep learning made deep architecture training possible by utilizing the restricted Boltzmann machine (RBM) to initialize multiple hidden layers one layer at a time in an unsupervised manner [54]. With unsupervised learning, deep learning tries to understand data first, i.e., to obtain a task specific representation from data so that a better classification can be achieved. It has been experimentally proven that the unsupervised learning step plays a critical role in the success of deep learning [119]. The proposed deep model consists of several components that will be described bellow.

5.1.1 PRE-TRAINING WITH RBM

Each layer in the proposed deep model is an RBM and the deep model used in this paper consists of a stack of RBMs. RBM is an energy-based model in which a scalar energy is associated with each configuration of the variables in the model, and a probability distribution function (PDF) through the energy function is defined. The purpose of learning is to modify the energy function so that a desirable PDF can be achieved, i.e., to have low energy. A basic RBM model having a visible (input) layer and a hidden (output) layer is shown in Fig. 24. The visible layer of the bottom RBM contains real-valued units (receiving data) and all other RBM layers have binary units. Let $v \in R^M$ represent input data (visible units) and $h \in 0, 1^N$ denote binary hidden units for the bottom RBM. We used Gaussian-Bernoulli RBMs to train it [99, 120]. All other RBMs were trained by utilizing Bernoulli-Bernoulli distribution. Variables v and h have a joint probability distribution defined as

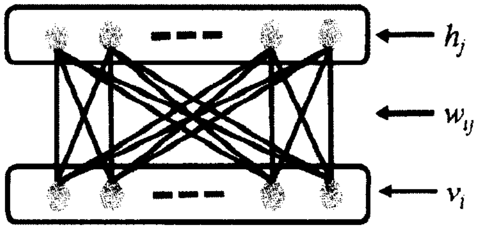


Figure 24: A basic RBM model.

$$p(v, h) = \frac{1}{Z} exp^{-E(v,h)}, \tag{5}$$

where $E(v, h)$ is an energy function and Z is a normalization constant. For real-valued visible layer RBMs, $E(v, h)$ is defined as

$$E(v, h) = \frac{1}{2\sigma^2} \sum_i v_i^2 - \frac{1}{\sigma^2}(\sum_i c_i v_i + \sum_j b_j h_j + \sum_{i,j} v_i w_{ij} h_j), \tag{6}$$

where c_i and b_j are biases of the ith and jth units in the visible and hidden layers, respectively. w_{ij} is the weight connecting v_i and h_j, and σ^2 is the variance of v. The conditional probability distributions are

$$P(h_j = 1|v) = sigmoid(\frac{1}{\sigma^2}(\sum_i w_{ij} v_i + b_j)), \tag{7}$$

$$P(v_i|h) = \mathcal{N}(\sum_j w_{ij} h_j + c_i, \sigma^2). \tag{8}$$

If both visible and hidden layers are binary, the energy function and conditional probability distributions are defined as

$$E(v, h) = -(\sum_i c_i v_i + \sum_j b_j h_j + \sum_{ij} v_i w_{ij} h_j), \tag{9}$$

$$P(h_j = 1|v) = sigmoid(\sum_i w_{ij}v_i + b_j), \tag{10}$$

$$P(v_i = 1|h) = sigmoid(\sum_j w_{ij}h_j + c_i). \tag{11}$$

Model parameters w, b and c are updated using contrastive divergence [54]. For RBM having a real-valued visible layer, the formulas for updating those parameters during each iteration are

$$\Delta W_{ij}^{t+1} = \eta \Delta W_{ij}^{t} - \epsilon(< \frac{1}{\sigma_i^2} v_i h_j >_d - < \frac{1}{\sigma_i^2} v_i h_j >_m), \tag{12}$$

$$\Delta b_i^{t+1} = \eta \Delta b_i^{t} - \epsilon(< \frac{1}{\sigma_i^2} v_i >_d - < \frac{1}{\sigma_i^2} v_i >_m), \tag{13}$$

$$\Delta c_j^{t+1} = \eta \Delta c_j^{t} - \epsilon(< h_j >_d - < h_j >_m). \tag{14}$$

where $< \cdot >_d$ and $< \cdot >_m$ denote the expectation computed over data and model distributions accordingly, t is iteration index, η is momentum and ϵ is learning rate. For binary RBM, equations (12) and (13) become

$$\Delta W_{ij}^{t+1} = \eta \Delta W_{ij}^{t} - \epsilon(< v_i h_j >_d - < v_i h_j >_m), \tag{15}$$

$$\Delta b_i^{t+1} = \eta \Delta b_i^{t} - \epsilon(< v_i >_d - < v_i >_m). \tag{16}$$

Note that the pre-training of RBM was unsupervised, i.e., class label (classification task) or desired output (regression) was not needed in the training. After the pre-training, we attached the class label on top of the stacked RBMs and utilized an adaptive backpropagation algorithm to fine-tune the weights in the model. All of the binary layers were also converted to real-valued units by using their continuous activities. Thus the deep learning model turned out to be a traditional multilayer perceptron (MLP), but its weights were initialized by RBM.

5.1.2 FINETUNING

After pre-training, two types of multi-layer network were constructed by adding a label layer (Type I network, deep classifier) or by "unfolding" the pretrained model (Type II network, deep autoencoder) as shown in Figure25. Both models will be initialized with the learned parameters (W and b) and fine-tuned using the BP algorithm [101]. The layer under the label layer in Type I network and the middle layer in Type II network were new representations for the original features shown in yellow in Figure25. A SVM classifier was then be trained using the learned features for engagement assessment.

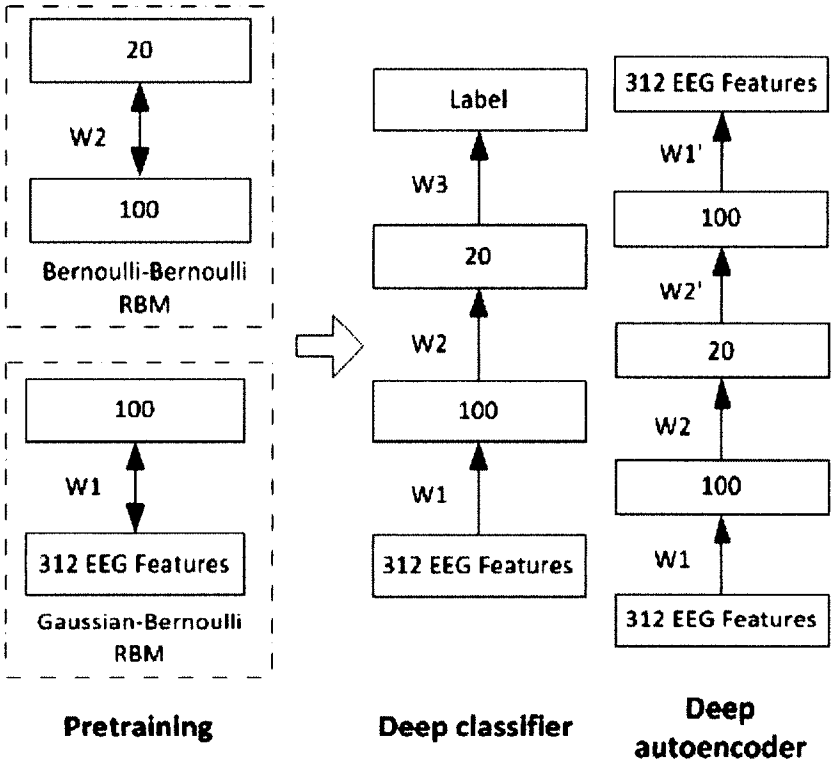


Figure 25: Deep learning models.

5.1.3 DROPOUT

Deep learning achieves excellent results in applications where the training data size is large. For small-sized data sets, such as the one in this paper, it is still possible for a deep structure to over-fit the data, given the fact that it usually has tens of thousands or even millions of parameters. To improve the generalization capability of the model, the dropout technique tries to prevent weight co-adaptation by randomly dropping out some units in the model during training [102, 121]. In the training process, each hidden unit is randomly omitted or dropped out from the network with a probability of p (usually $p = 0.5$), which can decrease the correlations among different hidden units. Previous experiments [102] showed that it was also beneficial if we applied the "dropout" process to the input layer but with a lower probability (i.e., 0.2 in this paper). In the testing procedure, all hidden units and inputs were used to compute model outputs for a testing case with appropriate compensations, i.e., weights between inputs and the first hidden layer were scaled by 0.8 and all other weights were halved.

5.2 EXPERIMENTAL SETUP

5.2.1 DATA PREPROCESSING AND FEATURE EXTRACTION

The procedures of data preprocessing and feature extraction were similar to those described in section 3.3. The simulated flight lasted for about 4.5 hours. Experts labeled 10-minute data as engaged and another 10-minute data as disengaged [9]. The EEG signals were first preprocessed including smoothing and artifacts removal. Then a 3-second window was designed as "Epoch" and was shifted along EEG signals with a step size of one second, making a two-second overlapping between any two adjacent Epochs. For each Epoch, 1Hz frequency bin power spectral density (PSD) from $1-39$ Hz for the selected 8 EEG channels was calculated and yielded 312 (39×8) 1Hz bin PSDs as features. This procedure produced a feature vector for every signal Epoch resulting in 1200 feature vectors for the total 20 minute labeled data. Those feature vectors were then fed into the deep learning framework to learn a new feature representation for the original features. Finally, the new feature representations were used for engagement assessment by classifying a feature vector to either an "engaged" or "disengaged" category using a linear SVM.

5.2.2 ENGAGEMENT ASSESSMENT THROUGH 5-FOLD CV

We first conducted 5-fold CV on the labeled data from 15 pilots to study engagement assessment. In 5-fold CV, we randomly divided the labeled data into five parts. We first used four parts to train a model; the trained model was then applied to the remaining part for evaluation. This procedure was repeated five times, such that each part was tested once. There were several hyperparameters associated with the deep learning scheme including number of hidden layer in the deep structure and number of hidden units in each hidden layer. Due to the computational complexity of deep learning and the limited labeled data, it was difficult to determine the optimal values for the hyperparameters in the deep learning structure. Instead, we studied the effects of these hyperparameters by performing 5-fold CV on the available data with different combinations of parameter values, and we compared the performances resulting from these combinations. In addition, we also studied whether dropout could improve the performance of deep learning for engagement assessment.

5.2.3 ENGAGEMENT ASSESSMENT WITH SCARCE LABEL INFORMATION

In the 5-fold CV evaluation, we randomly divided the labeled data into 5 parts without considering the time information associated with each of the data points. In practice, a trained model may be applied for a certain amount of time without retraining, making the testing data continuous in time. To mimic the practical application scenario, we conducted a more restricted engagement assessment for pilots using the limited labeled data in this study. We designed experiments in which we utilized the continuous top 1%, 3%, 5%, 10%, 15%, and 20% of all labeled data, respectively, for training, and utilized the corresponding remaining data for testing. In each experiment, we first used all data points (without labels) to pre-train the deep structure. After pre-training, we fine-tuned the deep structure utilizing labels from the training data or using an autoencoder. Finally, the learned 20 features were used to train a linear SVM classification model. The testing data points were fed into the fine-tuned deep structure to obtain their new feature representations and were subsequently classified by a trained linear SVM classifier.

Two experiments were designed for comparison. The first one used all 312 EEG features as inputs and the second one utilized PCA to reduce the dimensionality of the original EEG features to 30. Both models used linear SVM for classification. In addition, we studied the effect of the hyperparameters in this evaluation.

5.3 RESULTS AND DISCUSSIONS

In this section, we present the engagement assessment performances conducted on the labeled data. The effects of hyperparameters in deep model on the performances will also be discussed.

5.3.1 RESULTS OF 5-FOLD CV WITH DIFFERENT HYPERPARAMETERS

Effects of momentum and learning rate in pre-training. A good combination of learning rate and momentum is crucial for the convergence of RBM learning. Usually, a small value for the learning rate and a large value for the momentum are helpful for the convergence. Figure26 shows reconstruction errors for a Gaussian-Bernoulli RBM with a structure of 312-600. We set the momentum value as 0.9 and the learning rate value as 0.005, 0.01, and 0.018 respectively. It can be observed that the error decreased faster if a larger learning rate was used. However, when we increased the learning rate to 0.02, the training failed to converge. We also tried different values for the momentum and found that the training failed to converge when momentum value was less than 0.6 with a learning rate of 0.018. To guarantee a safe convergence of the RBM training, in subsequent experiments, we set the momentum as 0.9 and the learning rate as 0.01.

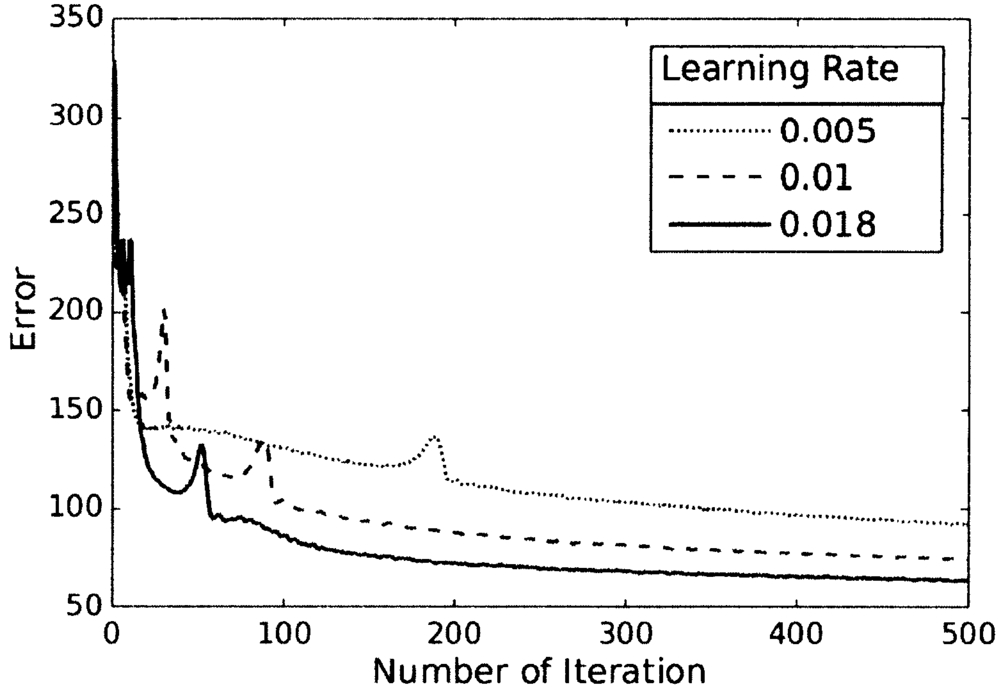
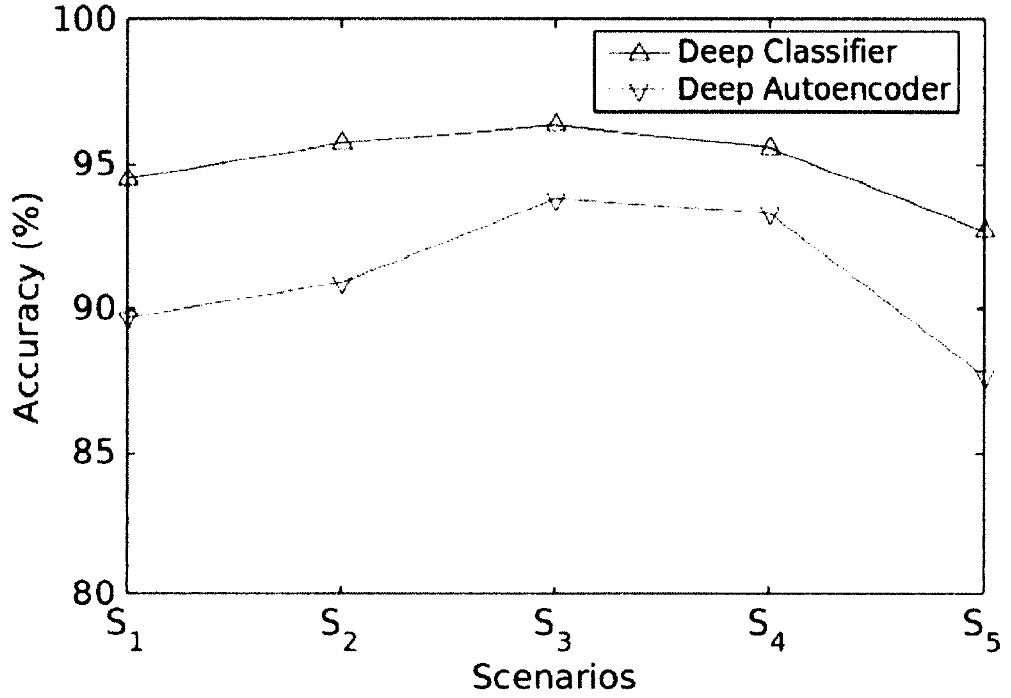


Figure 26: Learning error for different learning rate

Effect of the deep network structure. We conducted 5-fold CV on the available data set using five different structures including 600-200-100-20, 200-100-20, 100-20, 20, and 800-200-100-10. For each structure, we first pre-trained the network and then the network was fine-tuned as a deep classifier and a deep autoencoder, respectively. The number of pre-training iterations for all networks was set as 3000 for a fair comparison. Each experiment was repeated five times and the average accuracy was computed as shown in Figure27. It is observed that the highest accuracy of 96.36% was achieved by the network that had a structure of 100-20 and was fine-tuned by labels (deep classifier). Based on this result, the structure of 100-20 was chosen for subsequent engagement assessment.



S_1 : 600 − 200 − 100 − 20, S_2 : 200 − 100 − 20, S_3 : 100 − 20, S_4 : 20, S_5 : 800 − 200 − 100 − 10

Figure 27: Accuracies of networks with different structures.

Effect of dropout. We tested whether the dropout technique could improve the classification performance. We designed three scenarios for the network with structure of 100-20: S_1) fine-tuning without dropout, S_2) fine-tuning with dropout (drop out probability for visible/hidden layer are 0/0.5), and S_3) fine-tuning with dropout (drop out probability for visible/hidden layer are 0.2/0.5). Table18 illustrates the classification performance, the numbers shown in the table are averaged accuracies (in %) from five runs and their corresponding standard deviations are shown in blanket. For the deep classifier, the performance differences among these three scenarios were very small. However, for the deep autoencoder, the model without dropout was outperformed by the other two models with dropout; the model with dropout probabilities of 0.2/0.5 performed the best. Therefore, dropout with probability of 0.2/0.5 will be utilized for subsequent experiments.

Table 18: Results of dropout

	Scenario 1	Scenario 2	Scenario 3
Deep Classifier	97.53(0.13)	97.43(0.17)	97.16(0.04)
Deep Autoencoder	91.42(0.21)	93.60(0.26)	93.86(0.22)

Effect of number of iterations in pre-training and fine-tuning. A model trained with a large number of iterations does not always perform better than those with less training because a model can be over-fitted [122], [123]. We tried different number of iterations in pre-training (0, 100, 500, 1000, and 2000) and fine-tuning (0 $\sim$ 3000) and monitored the performances of the model. Our results are listed in Figure28 (deep classifier) and Figure29 (deep autoencoder). Other hyperparameters in the learning were set as momentum=0.9, learning rate=0.01, and dropout probability = 0.2/0.5.

If the number of the fine-tuning iteration was 0, the networks were initialized with pre-trained weights without fine-tuning. Note that the pre-training was unsupervised and that label information was not yet built in the model. However, the features learned from pre-training could already achieve over 94% accuracy by both the deep classifier and the deep autoencoder. As a comparison, both models obtained around 50% accuracy, equivalent to random guess, if models were not pre-trained and were not fine-tuned. Although pre-training played a positive role, increasing iteration of pre-training did not always offer a better result. In this study, 100 iterations of pre-training can already achieve good enough performance.

With the increase of fine-tuning iteration, all deep classifier models continued to improve, and the performance of the pre-trained autoencoders maintained the same or slightly dropped. However, there was no trend of over-training. We believe this is because of the regularization effect of the dropout technique that could prevent weight from co-adaption [102].

It was also observed that pre-training had a large effect on the deep autoencoder. Figure29 shows that the deep autoencoder without pre-training did not perform well, even with an increased number of fine-tunings. However, the deep classifier seemed to depend less on pre-training; fine-tuning could improve the model even if it was not pre-trained. The reason could be that the deep classifier has a relative simple structure ([312-100-20-2]), and also in 5-fold CV, the training data set was relatively larger than those in the experiments that we would perform in section 5.3.2. However, for the deep autoencoder, the network structure was more complicated ([312-100-20-100-312]), and pre-training became a key procedure to improve performance.

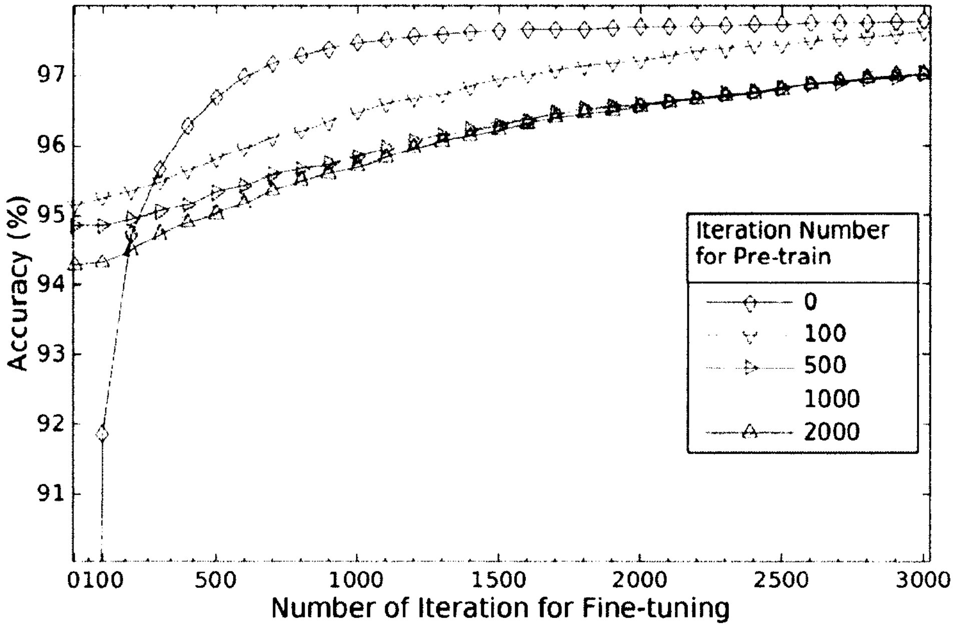


Figure 28: Performance with different number of iterations (deep classifier).

5.3.2 RESULTS OF ENGAGEMENT ASSESSMENT WITH SCARCE LABEL INFORMATION

Model comparison results. We are more interested in the model performance for engagement assessment when there are only limited labeled data available and the model performance is tested on sequential data. We compared four models for this purpose including a linear SVM using all 312 raw features (method 1), a linear SVM using top 30 principal components of the raw features (method 2), a deep classifier and a deep autoencoder. We studied the effects of dropout and pre-training on model performances for the two deep models and results are illustrated in Table19 and Figure30. Other hyperparameters of the deep learning were set as: momentum = 0.90, learning rate = 0.01, and network structure = [100-20], and the dropout probabilities were chosen as 0.2 for the visible layer and 0.5 for the hidden layer. Each experiment except those using the two compared models was repeated five times and the average accuracies of the five runs are shown in Table19. The

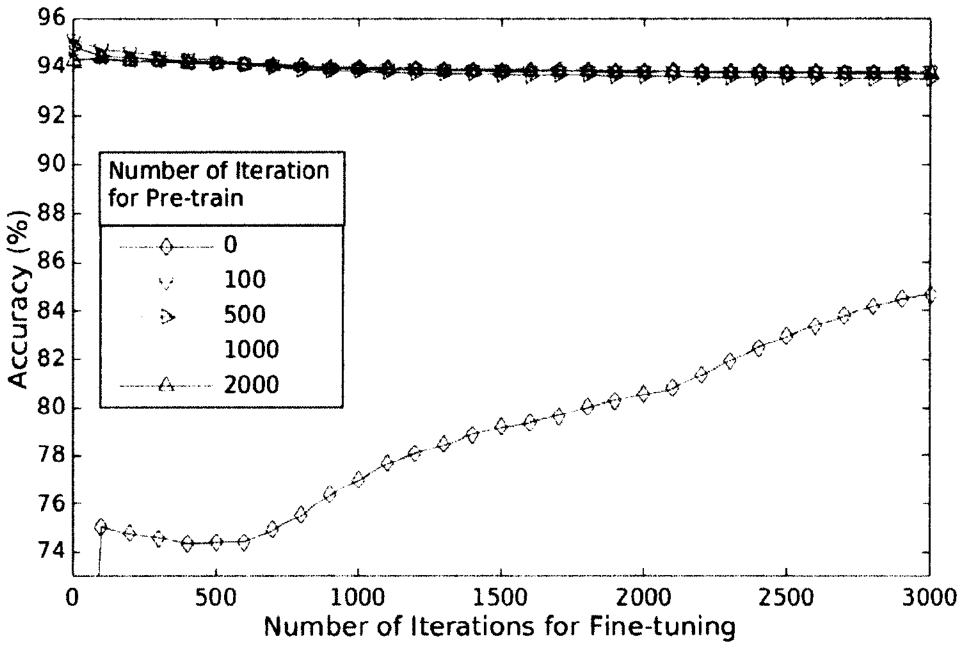


Figure 29: Performance with different number of iterations (deep autoencoder).

standard deviations of the average accuracies are shown inside the brackets.

It is observed that both of the deep learning models outperformed the two compared models, especially when there were not enough labeled data for training (less than 15% of the labeled data for training). The two deep models performed similarly in all of the experiments. These experimental results show that deep model is especially suitable for data modeling in the situation if labeling data is difficult to perform.

It also can be observed in Table19 and Figure30 that the deep models' performances dropped significantly if the models were not pre-trained or were not trained by the dropout technique when labeled data was limited. With more labeled data for training (the last column in Table19), the deep classifier performed relatively well even without pre-training or dropout. On the contrary, the deep autoencoder still could not perform well for this case if the deep structure were not pre-trained or were trained without the dropout technique.

The above results are not surprising because the pre-training step in the two deep models may help classification (modeling $p(y|x)$) by modeling $p(x)$ first [118]. In other words, if data labels are limited, understanding the data itself may be important in data classification. The dropout technique is a method for mitigating the over-fitting problem and it is especially helpful if labeled data is limited.

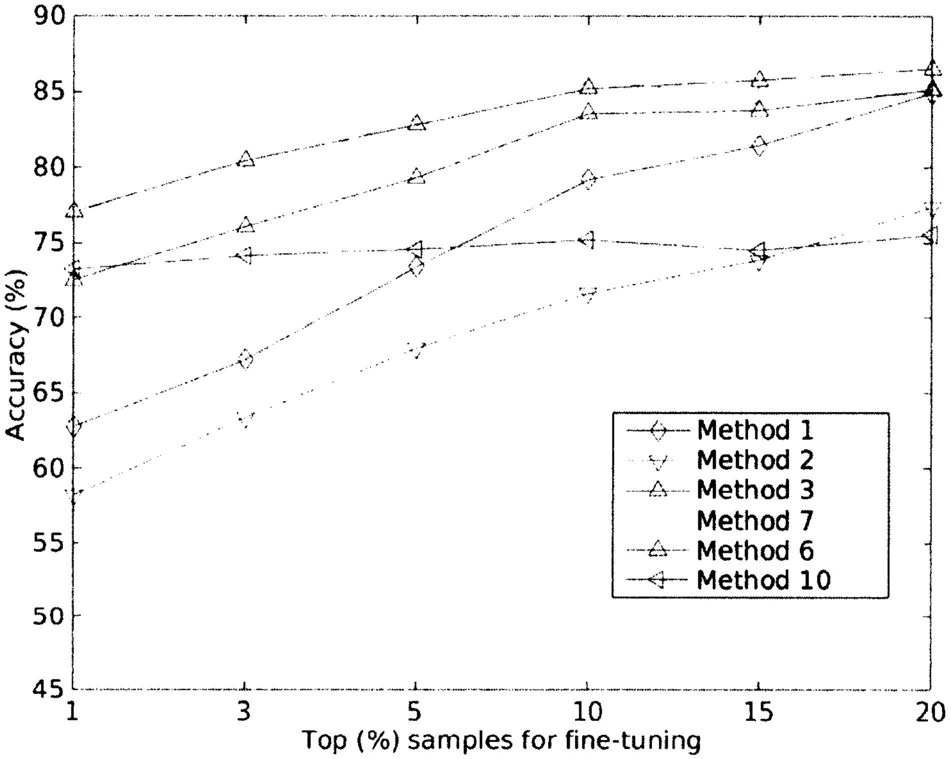


Figure 30: Engagement assessment results

Our results from 5-fold CV ($\sim$ 97%) are comparable to other functional state assessment studies in literature. For example, using a multi-class SVM classifier based on EEG signals, Shen et al. achieved a 10-fold CV accuracy of 91.2% for fatigue modeling [73]. In [60], EEG-based models for mental fatigue obtained 5-fold CV accuracies ranging from 90% to 100% with a mean of 97% to 98%. In [72], an EEG-based cognitive state estimation system achieved an accuracy of 98%. However, the more strict evaluation using continuous data blocks for training and testing in our study showed that the CV accuracies of deep models dropped from $\sim$ 97% to $\sim$ 85%. The more strict evaluation scheme is similar to a real application scenario. We must be aware of this performance drop if we want to deploy such systems in real applications.

5.3.3 COMPUTATIONAL EFFICIENCY

The running time for deep learning algorithms depends on the number of training samples, the structure of the network, and the number of iterations. Once a model has been trained, it requires almost no time for testing. For our cases, the deep learning algorithms were used for feature extraction, and linear SVM models were trained for classification, so the running time consists of time for training both deep networks and SVM models. All of the computations were done on an HP-Z800 workstation and hardware configuration included two Intel Xeon x5660, 48Gb memory and a GeForce GTX 780 GPU with 6Gb RAM. The program was developed in Matlab and accelerated with GPU. For a typical scenario, 1200 data samples, with the network structure as 100-20, the number of iterations set to 2000 for both pre-training and fine-tune, and the linear SVM as classification model, the running time was around ten minutes for either deep classifier or deep autoencoder for one subject. The running time should be timed 5 for 5-fold CV.

5.4 CONCLUSIONS

This chapter studied pilot's engagement assessment under complex task when only very limited labeled EEG data were available. We proposed deep learning models that are able to learn valuable high-level features by taking advantage of both unlabeled and labeled data. The two deep models, deep classifier and deep autoencoder, were studied, and both models outperformed baseline methods.

Table 19: Engagement Assessment Results.

	Top 1%	Top 3%	Top 5%	Top 10%	Top 15%	Top 20%
Method 1 low level features	62.73	67.19	73.38	79.18	81.47	84.92
Method 2 PCA features	58.21	63.35	67.94	71.65	73.86	77.39
Method 3 Deep classifier + pre-train + dropout	77.07 (1.02)	**80.45** **(1.31)**	82.82 (1.21)	85.22 (1.67)	**85.78** **(0.64)**	**86.52** **(0.72)**
Method 4 Deep classifier - pre-train + dropout	74.66 (1.43)	78.6 (2.38)	80.54 (1.22)	83.68 (0.59)	85.35 (0.26)	85.34 (0.79)
Method 5 Deep classifier + pre-train - dropout	70.81 (2.12)	76.54 (1.28)	79.45 (1.62)	84.27 (0.76)	84.37 (0.84)	86.00 (0.70)
Method 6 Deep classifier - pre-train - dropout	72.53 (1.74)	76.08 (2.43)	79.32 (1.98)	83.56 (0.64)	83.76 (0.72)	85.17 (1.02)
Method 7 Deep autoencoder + pre-train + dropout	**77.09** **(0.59)**	79.54 (0.30)	**83.32** **(0.58)**	**85.74** **(0.39)**	84.8 (0.63)	84.83 (0.44)
Method 8 Deep autoencoder - pre-train + dropout	72.49 (0.93)	75.72 (0.15)	76.94 (1.00)	79.53 (0.91)	78.62 (0.43)	79.21 (0.25)
Method 9 Deep autoencoder + pre-train - dropout	71.28 (1.76)	75.77 (0.40)	75.43 (0.50)	75.61 (0.40)	76.32 (0.28)	77.03 (0.22)
Method 10 Deep autoencoder - pre-train - dropout	73.24 (1.36)	74.13 (0.71)	74.59 (0.45)	75.22 (0.38)	74.53 (0.23)	75.53 0.86)

Note: sign + and - mean using or not using the followed technique

Chapter 6

DEEP MODEL FOR IMPROVED CLASSIFICATION OF AD/MCI PATIENTS

6.1 INTRODUCTION

Alzheimer's Disease (AD) is the sixth-leading cause of death in the United States [124]. Accurate classification of AD and its prodromal stage, Mild Cognitive Impairment (MCI), plays a critical role in possibly preventing progression of memory impairment and improving quality of life for AD patients. For each of these stages, significant amounts of research has been conducted aiming to understand the underlying pathological mechanisms. In addition, imaging biomarkers have been identified using different imaging modalities such as magnetic resonance imaging (MRI) [125], positron emission tomography (PET) [126], and functional MRI (fMRI) [127]. Imaging biomarkers are a set of indicators computed from image modalities which can be used for early detection of AD disease. It has been shown that fusing these different modalities may lead to more effective imaging biomarkers [2].

The first successful deep learning framework, auto-encoder, was developed in 2006 [54]. It was subsequently used in other application fields and achieved state-of-the-art performance in speech recognition, image classification and computer vision [119]. Deep learning itself also evolved after 2006. For instance, the multimodal deep learning framework boosted speech classification by learning a shared representation between video and audio modalities [128]. A dropout technique further improved zip code recognition, document classification, and image recognition [102, 121].

In this paper, we developed a robust deep learning framework for AD diagnosis by fusing complementary information from MRI and PET scans. These 3D scans were preprocessed and their features were further extracted. Specifically, we first applied principal component analysis (PCA) to obtain PCs as new features. We then utilized the stability selection technique [129] together with the least absolute shrinkage and selection operator (Lasso) method [130] to select the most effective features. The selected features were subsequently processed by the deep learning structure. Model weights in the deep structure were first initialized by unsupervised training and then were fine-tuned by AD patient labels. During the fine-tuning phase, the dropout technique was employed to improve the models' generalization capabilities. Finally, the learned feature representation was used for AD/MCI classification by a support vector machine (SVM).

In addition to discrete patient labels (AD, MCI, or Healthy), there are two additional clinical scores, namely Minimum Mental State Examination (MMSE) and Alzheimer's Disease Assessment Scale-Cognitive subscale (ADAS-Cog) associated with each patient. MMSE is a 30-point questionnaire widely used to measure cognitive impairment [131]. It is used to estimate the severity and progression of cognitive impairment, instead of providing any AD information. ADAS-Cog is the most popular cognitive testing instrument to measure the severity of the most important symptoms of AD, including disturbances of memory, language, praxis, attention and other cognitive abilities, which have been considered to be the core symptoms of AD [132]. The information from these scores is related, and identifying the commonality among them may help in AD diagnosis. We configured the deep learning structure as a multi-task learning (MTL) framework, and treated the learning of class label, MMSE, and ADAS-Cog as related tasks to improve the prediction of main task (class label).

We evaluated the proposed method on the ADNI[1] data set and compared it with a baseline method and a similar deep learning system, where the auto-encoder was used as a feature extractor for AD diagnosis [2]. The baseline method contains feature selection and SVM steps but does not use deep learning. We also evaluated the impact on performance of each of the components in the proposed system. A brief version of this paper was published in [133].

6.2 MATERIALS AND METHODS

The proposed system consists of multiple components, including PCA, stability selection, unsupervised feature learning, multi-task deep learning and SVM training, as shown in Fig. 31. We detail each of these components in the following subsections.

6.2.1 DATA PREPROCESSING

[1] Available at http://www.loni.ucla.edu/ADNI.

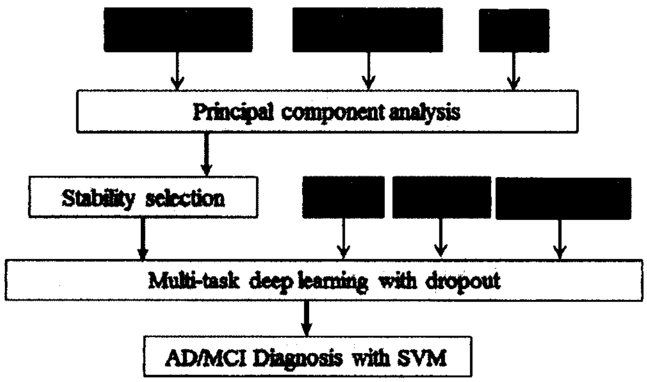

Principal component analysis
Stability selection
Multi-task deep learning with dropout
AD/MCI Diagnosis with SVM

Figure 31: Diagram of the proposed multi-task deep learning framework.

We utilized the public ADNI data set to validate our proposed deep learning framework. The data set consisted of MRI, PET, and CSF data from 51 AD patients, 99 MCI patients (43 MCI patients who converted to AD (MCI.C), and 56 MCI patients who did not progress to AD in 18 months (MCI.NC)) as well as 52 healthy normal controls. In addition to the crisp diagnostic result (AD or MCI), this data set contained two additional clinical scores, MMSE and ADAS-Cog, for each patient. A typical procedure of image processing was applied to the 3D MRI and PET images [125, 134, 135] including anterior commissure-posterior commissure correction, skull-stripping, cerebellum removal, and spatially normalization. Finally, we extracted 93 region-of-interest (ROI) based volumetric features from MRI and PET images, respectively, which together with three CSF biomarkers, i.e., $A\beta_{42}, t-$tau, and p-tau, summed up to 189 features for each subject.

6.2.2 PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is a linear orthogonal transformation that converts a set of features into linearly uncorrelated variables in which each of the new variables is a linear combination of all of the original features [136]. The first principal component (PC) is defined as the one that can explain the largest variance in the original data set. The second PC has the second largest variance under the constraint that it is orthogonal to the first component. If correlations exist among features, the number of PCs that can be found is usually less than the number of features in the original data. PCA is optimal for preserving energy and it is often used for dimensionality reduction by just keeping the first few PCs.

Let $\mathbf{F}$ denote a feature data set with a size of $n \times p$, where n is the number of data samples and p is the number of features in the data, and each column in $\mathbf{F}$ is centered. PCA can be achieved by performing the singular value decomposition (SVD) on $\mathbf{F}$ as

$$\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{17}$$

where $\mathbf{U}$ is an $n \times n$ matrix with orthogonal unit columns (left singular vectors of $\mathbf{F}$), Σ is an $n \times p$ diagonal matrix consisting of singular values of $\mathbf{F}$ from the largest to least, and $\mathbf{V}$ is an $p \times p$ matrix whose columns are orthogonal unit vectors (right singular vectors of $\mathbf{F}$).

To achieve dimensionality reduction, the first l columns in $\mathbf{V}$ corresponding to the first l largest singular values of $\mathbf{F}$ can be used as a transformation matrix to be applied on $\mathbf{F}$,

$$\mathbf{x} = \mathbf{F}\mathbf{V}_l, \tag{18}$$

where $\mathbf{V}_l$ consists of the first l columns of $\mathbf{V}$.

Geometrically, PCA analysis rotates data to align its maximum variance direction of the data with the coordinate system as illustrated in Fig. 32. PCA is an effective tool for dimensionality reduction but the preserved PCs may not be useful for classification. The two dimensional artificial data set in Fig. 32 consists of 'blue' and 'red' classes. After PCA, the whole data set was rotated and its main axis was aligned with the coordinate system. However, even though PC 1 had the largest variance, it did not contain any discriminating information for the two classes. For the purpose of classification, PC 2 was preferred and a feature selection step was necessary. This example shows that feature selection may be applied after PCA to retain discriminating information for classification.

6.2.3 STABILITY SELECTION

In this paper, we first applied PCA to the 189 features and used the resulting PCs as new features. We then applied Lasso [130] to identify the most effective features for AD diagnosis. Lasso tries to minimize the following cost function for feature selection:

$$\min_{\mathbf{s}} ||\mathbf{t} - \mathbf{x}\mathbf{s}||_2^2 + \lambda||\mathbf{s}||_1, \tag{19}$$

where $\mathbf{t} \in \{+1, -1\}^n$ is a class label vector of size $n \times 1$ associated with the feature matrix $\mathbf{x}$ of size $n \times l$, where l is the number of features (PCs) found in PCA,

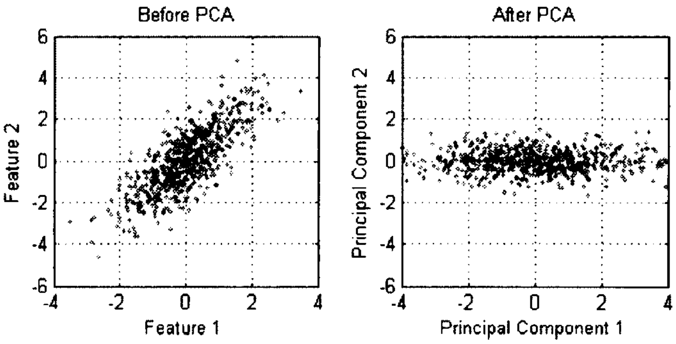


Figure 32: Principal component analysis example. PC 1 contains the most energy of the data but does not have any discrimination information for the 'red' and 'blue' classes.

$\mathbf{s} = [s_1, s_2 ... s_l]^T$ is the weight vector associated with the l features (columns in $\mathbf{x}$), λ is a regularization parameter, and $||\cdot||_2$ and $||\cdot||_1$ denote L_2 and L_1 norms, respectively. Because of the L_1 norm constraint on the weight magnitude, the solution minimizing the above cost function is usually sparse, meaning that if a feature is not correlated with the target class label, the feature will have a zero value for its weight. Features having nonzero weights will be selected and otherwise will be excluded.

It is well known that the solution of L_1 norm based optimizations are sensitive to the choice of λ, and it is difficult to determine how many features should be kept in the model. A recent breakthrough sheds a light on selecting the right amount of regularization for stability selection [129]. The idea is to repeat the feature selection procedure multiple times based on bootstrapped data sets and compute the probability of the features to be selected. The final selected features are those having probabilities above a predefined threshold t_h. It has been shown experimentally and theoretically that the feature selection results vary little for sensible choices in a range of the cut-off value for t_h [129]. We incorporated the stability selection concept into the AD patient diagnosis in this paper. In particular, we repeated the Lasso procedure 50 times and each time with a different value for the parameter λ (We used the SLEP toolbox for Lasso[2]). A probability, p_i, for the ith feature was computed by counting the frequency of the feature being selected in the 50 experiments. The ith feature was selected if p_i is larger than a pre-defined threshold t_h.

6.2.4 MULTI-TASK DEEP LEARNING WITH DROPOUT

[2] Available at http://www.public.asu.edu/ jye02/Software/SLEP/index.htm

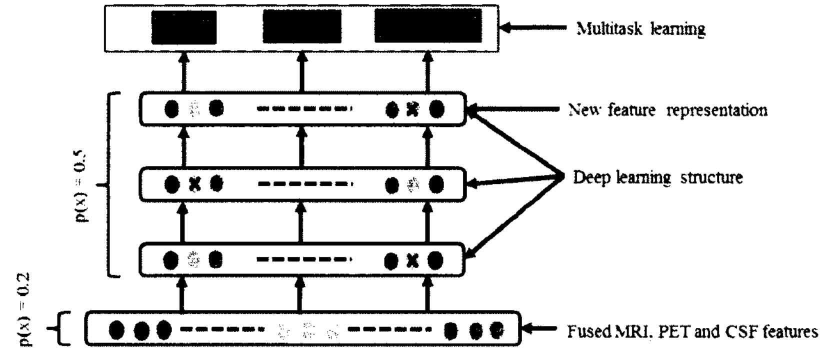


Figure 33: Multi-task deep learning with dropout. "x" denotes a dropped unit.

As presented at Chapter 5, deep learning techniques are available to work as strong representations for original data. In this paper, we incorporated deep learning with multi-task learning to further improve performance. Related tasks are learned simultaneously by extracting and utilizing appropriate shared information across tasks to improve performance. The proposed deep model shown in Fig. 33 consists of several components that will be described below.

Pre-training with RBM

Please see details in Section 5.1.1.

Multi-task learning

In multi-task learning, related tasks are learned simultaneously by extracting and utilizing appropriate shared information across tasks to improve performance. It has received attention in broad areas, such as machine learning, data mining, computer vision, and bioinformatics [137, 138, 139] recently. This approach is particularly effective when only limited training data for each task is available. It is worth noting that neural networks can simultaneously model multiple outputs, making deep learning a natural multi-task learning framework if multiple tasks share inputs [140]. The proposed multi-task deep learning framework is shown in Fig. 33, where we treated the predictions of class label, MMSE and ADAS-Cog as three different tasks and modeled them simultaneously. MMSE, and ADAS-Cog were normalized to the range of [0,1] and we used the deep structure as a regression model. The class label was coded by the 1-of-k scheme. To classify an input vector, we checked the corresponding k outputs and assigned it to the class having the largest output. One drawback of deep model is over-fitting due to large capacity. This is more prominent if training data is limited. To overcome this limitation, we utilized the dropout technique to improve training.

Dropout with adaptive adaptation

Deep learning achieved excellent results in applications where training data size was large. For small sized data sets such as the one in this paper, it is still possible for a deep structure to over-fit the data given the fact that it usually has tens of thousands or even millions of parameters. To improve the generalization capability of the model, the dropout technique tries to prevent weight co-adaptation by randomly dropping out some units in the model during training [102, 121]. We incorporated the dropout technique in the multi-task learning context to improve AD diagnosis as shown in Fig. 33. In the training process, each hidden unit in the model was dropped with a probability of 0.5 when a batch of training cases were present. Previous experiments [102] showed that it is also beneficial if we apply the "dropout" process to the input layer but with a lower probability (i.e., 0.2 in this paper). In the testing procedure, all of the hidden units and inputs were used to compute model outputs for a testing case with appropriate compensations, i.e., weights between inputs and the first hidden layer were scaled by 0.8 and all other weights were halved.

During the multi-task fine-tuning step, the stochastic gradient descent method with a fixed learning factor is usually utilized as [54],

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \alpha \frac{\partial L}{\partial w_{ij}}, \tag{20}$$

where $\frac{\partial L}{\partial w_{ij}}$ is the gradient of the cost function L and α is a learning factor. Sometimes, the weights update may contain a momentum term [102]. We proposed an adaptive learning factor to speed up the adaptation. The motivation of the adaptive learning is that the learning factor should be large at locations where the gradient is small and vice versa. Assume the decrease of L due to the change in w_{ij} is approximated by

$$\Delta L^{ij} = L^{ij}_{new} - L^{ij}_{old} \approx \frac{\partial L}{\partial w_{ij}} \times \Delta w_{ij} = -\alpha [\frac{\partial L}{\partial w_{ij}}]^2, \tag{21}$$

then ΔL due to all w_{ij} can be computed as

$$\Delta L = -\alpha \sum_i \sum_j [\frac{\partial L}{\partial w_{ij}}]^2. \tag{22}$$

Suppose we want to decrease L by $\beta\%$, then $L_{new} = (1 - \beta)L_{old}$, and an adaptive learning factor α can be determined as

$$\alpha = \frac{\beta L_{old}}{\sum_i \sum_j [\frac{\partial L}{\partial w_{ij}}]^2}. \tag{23}$$

We set β as 10% in our experiments in this paper. Once the new feature representation was learned, an SVM classifier [141] was trained using the learned feature representation.

6.2.5 SVM CLASSIFIER

Given a set of data pairs $\{\mathbf{r}_i, t_i\}_{i=1}^n$, where $\mathbf{r}_i \in R^M$ is the learned feature representation from subjects, $t_i \in \{+1, -1\}$ is a class label (e.g., AD vs. non-AD) associated with $\mathbf{r}_i$. An SVM defines a hyperplane

$$f(\mathbf{r}) = \mathbf{k}^T \phi(\mathbf{r}) + e = 0 \tag{24}$$

separating the data points into 2 classes. In equation (24), $\mathbf{k}$ and e are the hyperplane parameters, and $\phi(\mathbf{r})$ is a function mapping the vector $\mathbf{r}$ to a higher dimensional space. The hyperplane (24) was determined using the concept of *Structural Risk Minimization* [141] by solving the following optimization problem,

$$\min_{\mathbf{k},e,\xi} \left(\frac{1}{2}\mathbf{k}^T\mathbf{k} + C \sum_{i=1}^{n} \xi_i \right), \tag{25}$$

subject to

$$t_i(\mathbf{k}^T \phi(\mathbf{r}_i) + e) \geq 1 - \xi_i, \xi_i \geq 0, \tag{26}$$

where C is a regularization parameter and ξ_i is a slack variable. After the hyperplane is determined, an AD case is declared if $f(\mathbf{r}_i) > 0$, or otherwise a non-AD case is declared.

6.3 RESULTS AND DISCUSSIONS

6.3.1 EXPERIMENTAL SETUP

Ten-fold cross-validation

We considered four classification tasks including AD patients vs Healthy Control subjects (AD vs HC), MCI patients vs HC (MCI vs HC), AD patients vs MCI patients (AD vs MCI) and MCI-converted vs MCI-non converted (MCI.C vs MCI.NC). For each task, we utilized a ten-fold cross-validation (CV) scheme to evaluate the proposed method. In the ten-fold CV, we randomly divided the data set into 10 parts and for one run, we separated one part for testing and applied the proposed framework to the remaining data to train a classification model. This procedure was repeated 10 times so that each part was tested once. Finally, testing accuracies were computed. To obtain a more reliable estimate of the performance, we repeated the ten-fold CV ten times for each task with different random data partitions and computed average accuracy. To compare different classification models, we kept the same data partitions in the ten-fold CV and utilized the paired-t test to evaluate if there was a significant performance difference.

Hyperparameter determination

We did preliminary experiments to determine the structure of the deep learning model. It was found that using three hidden layers with hidden units of 100-50-20 worked the best among the candidate structures considered, and thus, this method was utilized in our experiments. For the SVM classifier, we tried different kernels and a linear kernel was chosen. We also did a grid search for the "soft margin" parameter in the linear kernel SVM model but it did not improve the classification accuracies. Therefore, in all of the experiments, we utilized a three hidden-layer model with a structure of 100-50-20 for feature learning and a linear SVM with default soft margin as the classifier.

Impact assessment for individual component

There are four components in the proposed framework including PCA, stability selection, dropout and multi-task learning. Inspired by "sensitivity analysis" and "impact assessment" that analyze inputs of or components in a model and identify their impacts on the model objectives by varying the inputs [142], we incorporated a

Table 20: Performance comparison (in%) of the competing methods. The proposed method consists of four components. "-PCA" stands for "the proposed method without the PCA component" and "SS" stands for stability selection, "Baseline" denotes the framework without the deep learning component.

Tasks	Proposed	-PCA	-Dropout	-SS	-MultTask	Baseline
AD vs HC	**91.4**(1.8)	89.6(1.3)	84.2(3.0)	89.4(1.6)	90.3(1.7)	86.4(2.0)
MCI vs HC	**77.4**(1.7)	76.4(1.5)	73.1(3.1)	74.3(1.6)	75.6(1.7)	72.1(3.0)
AD vs MCI	**70.1**(2.3)	69.5(2.7)	65.1(3.7)	68.7(2.1)	67.1(2.9)	61.5(2.9)
MCI.C vs MCI.NC	57.4(3.6)	**58.1**(1.8)	50.2(3.3)	57.7(1.8)	56.7(3.0)	50.6(4.7)
Average	**74.1**	73.4	68.2	72.5	72.4	67.7

Table 21: Paired-*t* test between results of the proposed method vs deep learning without dropout. The methods of "SAEF" and "LLF+SAEF" were proposed by Suk [2]. "SAEF" stands for Stacked Auto-Encoder Features and "LLF" denotes Low Level Features.

Tasks	Proposed	-Dropout	Improvement	*p*-value	SAEF	LLF+SAEF
AD vs HC	**91.4**(1.8)	84.2(3.0)	7.2	$< 10^{-3}$	83.2(2.7)	85.3(3.2)
MCI vs HC	**77.4**(1.7)	73.1(3.1)	4.3	0.0034	70.1(2.8)	76.9(2.3)
AD vs MCI	**70.1**(2.3)	65.1(3.7)	5.0	0.0017	N/A	N/A
MCI.C vs MCI.NC	57.4(3.6)	50.2(3.3)	7.2	$< 10^{-3}$	58.4(4.1)	**60.3**(2.3)
Overal Average	**74.1**	68.2	5.9	N/A	N/A	N/A
Average w/o AD vs MCI	**75.4**	69.2	6.2	N/A	70.6	74.2

similar concept to evaluate the impact of each component on model performance by varying the component (presence vs absence). 'Absence' means that the component was not included in the model.

Methods for comparison

We compared the proposed method with a baseline method and a deep learning system similar to that proposed in [2]. The baseline method consisted of all components in the proposed system except the deep learning step. The work by Suk in [2] is an auto-encoder-based deep learning method in which features representations for MRI, PET, and CSF from the same data set were learned separately and were combined by a linear SVM classifier. They also combined the learned representations with original features for AD diagnosis.

6.3.2 RESULTS

Table 20 shows the overall performances of the proposed method and the impact of each component in the framework. The proposed method performed the best in diagnosing AD and MCI patients, and in discriminating MCI patients from AD patients with accuracies of 91.4%, 77.4% and 70.1%, respectively. It was significantly better than the baseline method that obtained accuracies of 86.4%, 72.1%, and 61.5% for the diagnoses. In the MCI conversion diagnosis (MCI.C vs MCI.NC), the PCA component slightly degraded the proposed method (from 58.1% to 57.4%), but it was still significantly better than the baseline method (57.4% vs 50.6%).

Among those components, it is obvious that "dropout" has the most significant impact on the performances. Without "dropout", deep learning did not significantly improve the baseline method (68.2% vs 67.7% in terms of average acc.). The least important component is "PCA", i.e., the average acc. slightly dropped from 74.1% to 73.4% without the PCA component. Without "stability selection" and "multi-task learning", the average accuracy dropped slightly from 74.1% to 72.5% and 72.4%, respectively.

We conducted a paired-t test between results by the proposed method and those from classical deep learning ("-Dropout"). Table 21 lists the improvements and p-values. The average improvement was 5.9%, and the improvements for all the four classification tasks were significant.

The work by Suk [2] on the same data set is also shown in Table 21, where "SAEF" corresponds to the method using features learned by a deep auto-encoder and "LLF+SAEF" represents the method that combines original features with the SAEF features for AD diagnosis. The AD vs MCI classification experiment was not conducted in [2]. The proposed method (75.4%) outperformed the SAEF method (with an average accuracy of 70.6%). By combining SAEF with LLF (LLF+SAEF), the average accuracy was increased to 74.2% (Last column in Table 21).

6.3.3 DISCUSSIONS

There are usually two ways to increase the generalization capability of a model: adding regularization (L_1 or L_2 norm) on weights or using a committee machine. However, solving the regularization problem is usually challenging, especially in the deep learning context. In addition, the committee machine technique requires averaging many separately trained models to compute a prediction for a testing case, which is time consuming for deep learning. The dropout procedure does them both (constraint and committee machine) simultaneously in a very efficient way. 1) Each sub-model in training is a sampled model from all of the possible ones, and all sub-models share weights. The weight sharing property is equivalent to the L_1 or L_2 norm constraint on weights, and 2) the testing procedure is an approximation of averaging all trained sub-models for a testing case, but it does not separately store them because they share weights. This is an extremely efficient and smart implementation of a committee machine [102, 121].

The impact evaluation method was inspired by the "sensitivity analysis" and "impact assessment" [142]. We were aiming to identify the impact on performance of each component in the model by excluding the component from the pipeline. Note that we did not try to decouple the component from the system. This evaluation method may not be a strict sensitivity analysis or impact assessment by means of their definitions, but we could verify each component if it can improve the AD diagnosis when it is included in the proposed system. Our experiments showed that the dropout component had the largest impact on the performance, multi-task learning ranked second, stability selection third, and PCA had the least impact on the performance.

In terms of stability selection and computational efficiency, there were usually around 40 features left after the stability selection, and it took about one hour for a personal computer to conduct a ten-fold CV evaluation for one task. The number of features that were chosen was determined by stability selection, in which the Lasso algorithm ran 50 times with different values of regularization parameter (λ). In each run, Lasso chose different features and a probability of being chosen for each feature was computed in the 50 runs. Finally, a feature was chosen if its probability was larger than 0.5.

It is worth it to note that the results obtained by the proposed method in Table 20 and Table 21 only used the new representations learned by the deep model. We tried to combine the new representations with the original features, but the combination did not improve performance. In [2], new representations learned from auto-encoder did not perform well unless they were combined with the original features. Our experiment also showed that the deep model without dropout performed comparably to the baseline method. It seems that traditional deep learning cannot extract information effectively from small data sets unless it is regularized by techniques such as dropout.

In [143], a multi-kernel SVM (MK-SVM) method was applied to the same data set to combine the original LLF features for AD diagnosis, and achieved 93.2% and 76.4% for AD vs HC and MCI vs HC classifications, respectively. The MCI conversion diagnosis and the AD vs MCI classification were not conducted. In [2], utilizing the MK-SVM method to combine SAEF features from MRI, PET, and CSF boosted the performances to 95.9%, 85.0% and 75.8% for the three tasks (AD vs MCI classification was not performed), respectively. Since the dropout technique improved upon the basic deep learning, we are currently investigating weather the MK-SVM method can further boost the performance of the proposed system.

We did not attempt to perform a comprehensive comparison study of the proposed method with others that have been applied to this data set in the literature. Instead, we have evaluated some recently proposed advanced machine learning techniques for AD diagnosis including Lasso, stability selection, multi-task learning, deep learning and dropout. The dropout technique seems to be an effective method of regularization for learning with small data sets. Without dropout, deep learning has no advantage over the baseline method on ANDI data set (68.2% vs 67.7%). Note that dropout is computationally very efficient as compared to either L_1 norm based regularization or committee machine and it can be extended to many models other than the deep model as discussed in this paper.

6.4 CONCLUSION

Our proposed method achieved 91.4%, 77.4%, 70.1% and 57.4% accuracies for AD vs HC, MCI vs HC, AD vs MCI, and MCI.c vs MCI.NC classifications, respectively. The framework consisted of multiple components including PCA, stability selection, dropout and multi-task deep learning. We showed that dropout is the most effective one. This is not surprising because the size of ADNI data is relatively small compared to that of the deep structure utilized in this paper. Classical deep learning does not perform well on this small data set, but with the dropout technique, the average accuracy was improved by 5.9%, on average. We plan to incorporate MK-SVM [2] into our method for further improving AD diagnosis.

Chapter 7

IMBALANCED LEARNING

7.1 INTRODUCTION

To successfully perform OFS assessment, researchers often face the challenge of modeling imbalanced data sets because OFS assessment data sets usually have much more data samples for some OFSs than others [71]. In the machine learning community, those OFSs having lots of data samples are named as 'majority' classes while those having less samples are called 'minority' classes. Traditional classifiers tend to classify all data samples into majority classes resulting in poor performances for minority classes [144], which is not acceptable for OFS assessment.

Many imbalanced learning techniques have been proposed to balance performances among majority and minority classes. Those techniques could be divided into four categories [144]: sampling methods, cost-sensitive methods, kernel-based methods, and active learning methods. Sampling methods aim to reduce the data imbalance by removing (under-sampling) samples from majority classes or by generating (over-sampling) more training samples for minority classes [145]. Cost-sensitive methods improve classification performance by using different cost matrices to compensate for imbalanced classes [146]. Kernel based methods, such as the support vector machine (SVM), are based on the principles of statistical learning and Vapnik-Chervonenkis (VC) dimensions [141]. Active learning is often integrated into kernel-based learning methods by selecting the closest instance to the current hyperplane from the unseen training data and adding it to the training set in order to retrain the model [147].

In this chapter, we implemented five sampling methods including random under-sampling, random over-sampling, synthetic minority over-sampling technique (SMOTE), borderline-SMOTE, and adaptive synthetic sampling (ADASYN) [144], and we integrated those methods into a committee classifier for OFS assessment. We validated our technique on a driving test benchmark dataset by treating the OFS assessment as a classification problem.

7.2 IMBALANCED LEARNING TECHNIQUES

There exist many imbalanced learning techniques proposed in the literature as described in the review paper [144]. In our study, we implemented five of them as described below.

- Random under-sampling
- Random over-sampling
- Synthetic minority over-sampling technique (SMOTE)
- Borderline-SMOTE
- Adaptive synthetic sampling (ADASYN)

All the methods have been illustrated in detail in [144], including their implementation, performance, and limitations. The overall goal of the methods was to make data samples balanced among classes. We briefly describe their basic ideas here.

7.2.1 RANDOM UNDER-SAMPLING

This method randomly samples majority classes and keeps minority classes unchanged to balance data distribution among classes (Figure 34). The method is simple and usually will reduce the number of data points available for training.

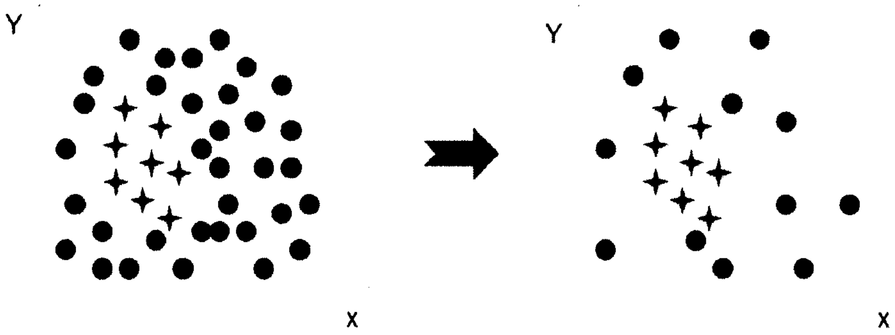


Figure 34: Random under-sampling

7.2.2 RANDOM OVER-SAMPLING

Similar to the random under-sampling technique, this method randomly over-samples minority classes and keeps majority classes unchanged to balance data distribution (Figure 35). The method will usually increase the number of data points available for training.

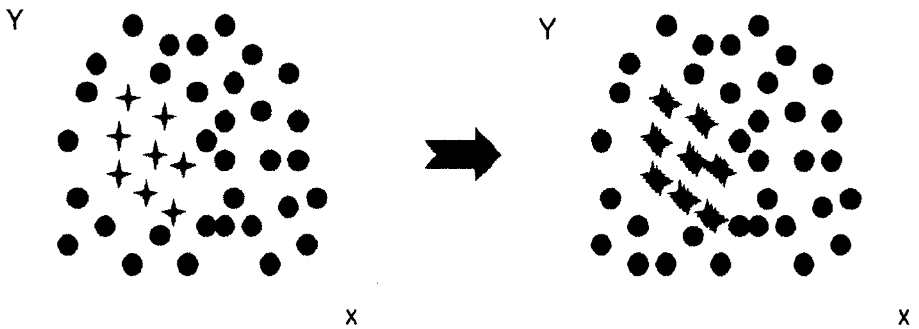


Figure 35: Random over-sampling

7.2.3 SMOTE

Different from the random over-sampling method that copies samples for minority classes, SMOTE generates or synthesizes new samples for minority classes. To create a new synthetic sample for a sample (seed) of minority class, it first randomly selects one of its K-nearest neighbors belonging to the minority class. Then a random point that is on the line between the seed and the selected neighbor will be synthesized as a new data sample (Figure 36).

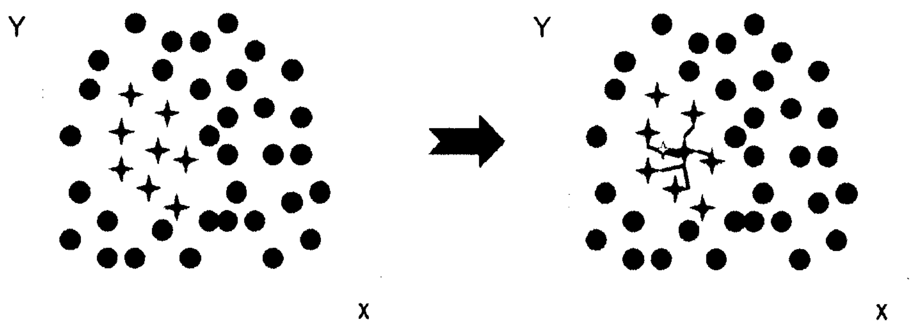


Figure 36: SMOTE

7.2.4 BORDERLINE-SMOTE

The difference between Borderline-SMOTE and SMOTE is how they select seeds. SMOTE may select any minority sample as a seed. However, Borderline-SMOTE only selects seed from minority samples that are on the borderline between the minority samples and the majority samples (Figure 37). A minority sample is considered as on the borderline when more than half of its M nearest samples belong to majority classes.

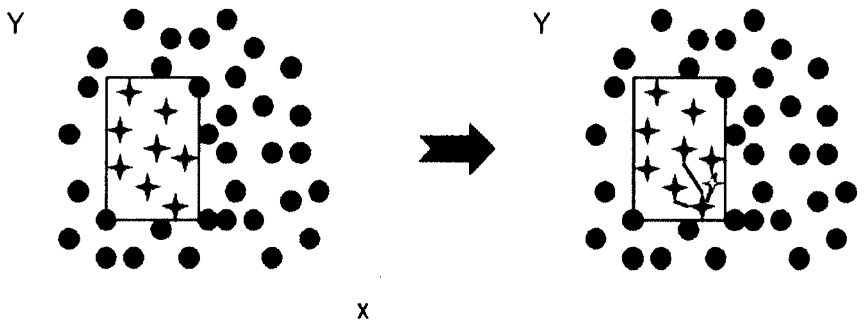


Figure 37: Borderline SMOTE

7.2.5 ADASYN

The difference between ADASYN and SMOTE is the amount of new samples that need to be synthesized for each seed of minority classes. SMOTE generates the same number of synthetic data samples for each seed while ADASYN constructs new data samples based on the distribution of seeds (Figure 38). Considering the K nearest neighbors of a seed, the more neighbors that belong to majority classes, the more samples need to be synthesized for the seed.

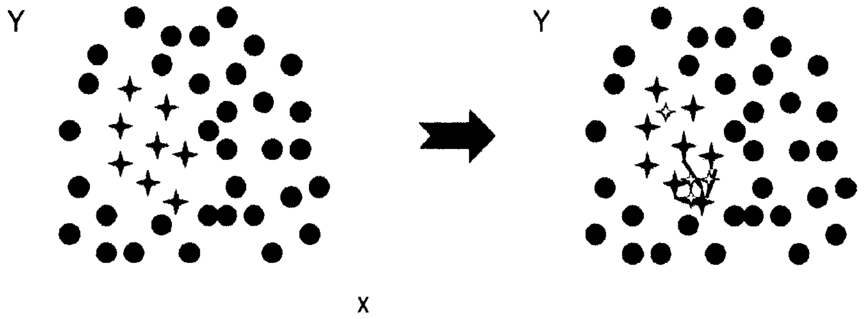


Figure 38: ADASYN

7.3 EXPERIMENT DESIGN

7.3.1 THE DRIVING TEST DATASET

The imbalanced learning techniques were originally validated using a driving test dataset. We found that those techniques were helpful for improving classification performance and embedded it to a committee machine, which was used for the pilot dataset as default. Since the pilot dataset were already balanced, there was no significant difference between using or not using balanced learning techniques. So in order to show the advantage of the imbalanced learning techniques, we utilized the driving test dataset (which was highly imbalanced) to validate our proposed method for OFS assessment. The dataset was collected when participants were performing a driving test during a course of two hours. The collected information includes description of the task, system dynamics related information, performance measures, physiological signals (128-channel EEG, ECG, respiration, etc.), and eye tracking. The workload was also analyzed according to the driving conditions (city-driving, stopped, highway passing, etc.). Based on the above information, seven OFSs, which were considered as seven classes in the committee machine, were defined.

Six subjects participated in the driving test and data was recorded in a separate file for each participant, resulting in six individual datasets. Each dataset had seven operator functional states that were considered as seven classes by our committee classifier. In the dataset, the number of data samples in each class was not balanced. Four classes (minority class) have many fewer data samples than other three majority classes. Table 22 and Figure 39 show the proportion of data samples of each class.

Class distributions were similar for all subjects. Class 2 had the largest number of samples (about 35% of data samples). Class 3 and 4 had the second largest number of samples (about 20%). Therefore, about 75% of samples belonged to these three classes. Class 7 had the smallest samples with a portion of less than 1%, and subject 2, 4 and 6 have no sample belonging to class 7. Class 6 was the second smallest class, having about 3% samples. Both class 1 and 5 had about 5% samples.

7.3.2 IMBALANCED LEARNING

Table 22: Proportion of samples for each class.

Class	Data set 1 (%)	Data set 2 (%)	Data set 3 (%)	Data set 4 (%)	Data set 5 (%)	Data set 6 (%)
1	6.17	8.70	6.29	5.59	3.52	3.86
2	38.34	39.24	33.83	39.66	32.65	39.87
3	19.96	21.42	24.56	32.94	26.39	20.16
4	23.55	19.40	21.07	16.43	31.24	27.05
5	8.03	8.25	11.30	3.03	2.99	6.10
6	3.89	2.98	2.67	2.35	2.99	2.96
7	0.06	0.00	0.28	0.00	0.22	0.00

To implement the five imbalanced techniques, we first computed a desired percentage of data samples per class as,

$$N_d = 100/No.ofclasses$$

We then calculated a threshold for the number of data samples for each class as,

$$T_H = N_d * (1 + 10\%)$$

$$T_L = N_d * (1 - 10\%)$$

Classes having more samples than T_H were considered as majority classes while classes with fewer samples than T_L were considered as minority classes and others were treated as medium classes.

For our case, there are seven classes and N_d, T_L and T_H were 14.29%, 12.86% and 15.71%, respectively. Referring to Table 22, it is clear that classes 2, 3 and 4 were majority classes. Class 1, 5, 6 and 7 are minority classes and there was no medium class in our datasets. In order to achieve a balanced dataset, the data portions in both majority and minority classes were made roughly the same as Nd. Medium classes were kept unchanged.

We applied the random under-sampling technique to the majority classes and four over-sampling methods to the minority classes, resulting in four balanced datasets, as shown in Figure 40. All the balanced datasets shared the same majority and medium classes' data samples but had different data samples from the minority classes, depending upon which oversampling method was used.

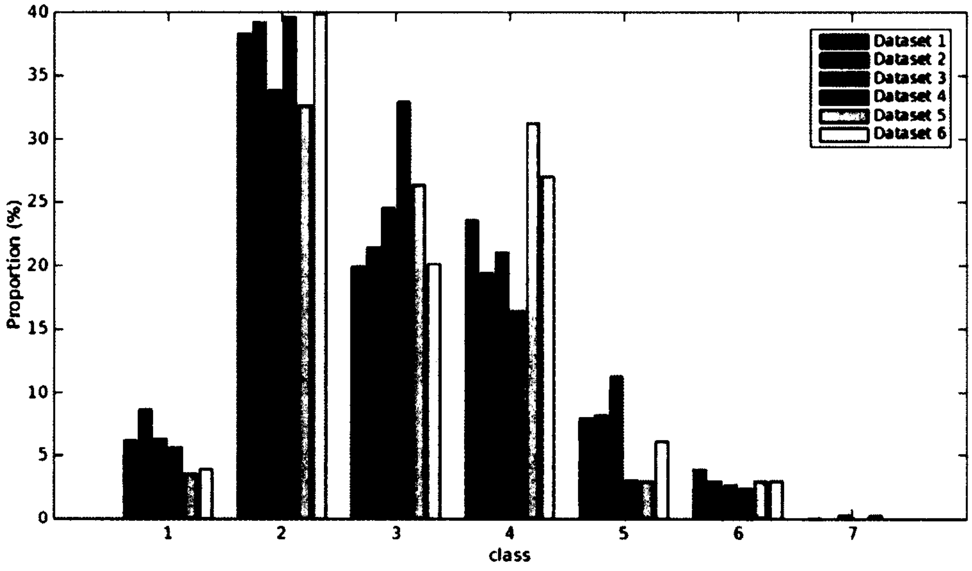


Figure 39: Proportion of samples for each class

7.3.3 COMMITTEE CLASSIFIER

The committee classifier consisted of a bootstrap procedure, a feature selection process, and a majority voting scheme (see Figure 41). A MLP trained by the BP algorithm was implemented as the base classification model. Basic procedures performed by the committee classifier were as follows:

1. Randomly divide a subject dataset to two parts for training and testing.

2. Generate M bootstrapped datasets.

3. Apply one of the imbalanced learning techniques to the bootstrapped datasets. A balanced dataset was then obtained for each of the M datasets.

4. Select a set of most useful features for each of the balanced datasets using the PLOFS algorithm. Selected features for different datasets were usually different.

5. Train a MLP classifier for each of the datasets using the features selected for that dataset.

6. Apply the trained MLP to the training and testing datasets.

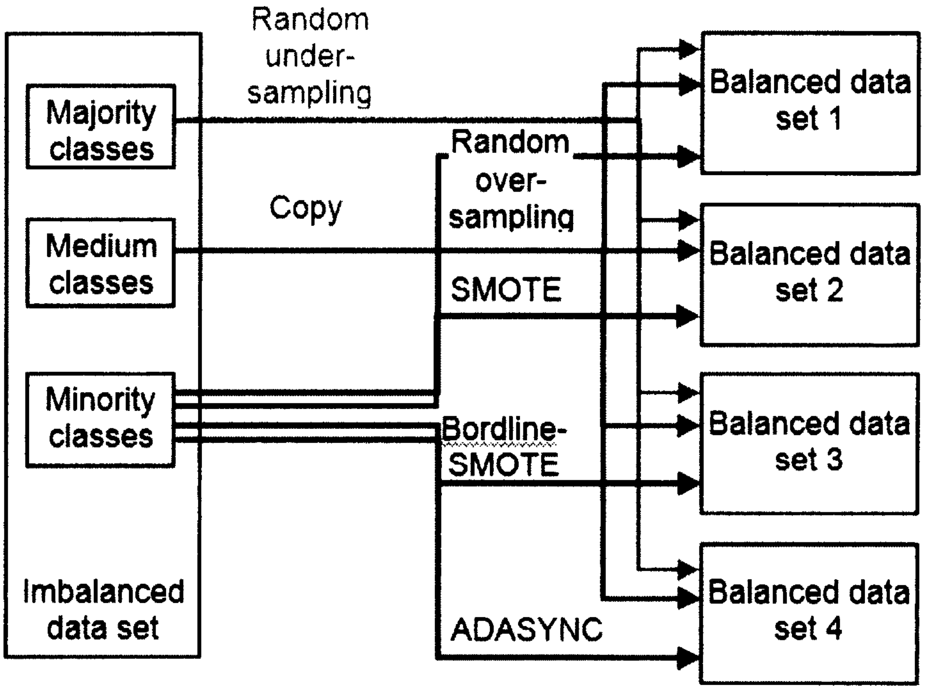


Figure 40: Generation of balanced datasets

7. Generate the final classification result by majority voting. MLPs having training accuracies greater than 50% were used only. Repeat the above procedures by exchanging the role of training and testing datasets.

8. Repeat the above steps for each of the imbalanced learning techniques.

7.4 RESULTS AND DISCUSSIONS

We trained a committee classifier for each of the six participants (datasets) and then calculated the average accuracy for each of seven classes. The results are shown in Table 23 and Figure 42.

In Table 23, the column "Imbalance" means classification accuracies (in percentage) for each class while no imbalanced technique is applied. The other four columns present classification performance (in percentage) for each class utilizing the four imbalanced learning techniques. The last row shows the overall accuracies achieved by each of the techniques. Together with the accuracy of each class using each imbalanced technique, the difference between its and "Imbalance" accuracy is also

Table 23: Result using imbalance techniques

Class	Imbalance (%)	Oversample (%)	Smote (%)	Border (%)	AdaSyn (%)
1	81.51	94.78 (+13.27)	96.57 (+15.06)	95.13 (+13.62)	95.47 (+13.96)
2	98.42	96.52 (-1.90)	96.45 (-1.97)	96.93 (-1.49)	96.98 (-1.44)
3	86.64	75.03 (-11.61)	70.16 (-16.48)	78.44 (-8.2)	75.75 (-10.89)
4	63.34	52.82 (-10.52)	45.85 (-17.49)	57.75 (-5.59)	55.90 (-7.44)
5	25.17	52.51 (+27.34)	68.00 (+42.83)	35.90 (+10.73)	42.90 (+17.73)
6	57.19	91.51 (+34.32)	87.71 (+30.52)	78.56 (+21.37)	85.77 (+28.58)
7	0.00	100.00 (+100)	86.31 (+86.31)	91.07 (+91.07)	86.31 (+86.31)
Average	58.90	80.45 (+21.55)	78.72 (+19.82)	76.25 (+17.35)	77.01 (+18.11)

presented in a pair of parentheses.

It is observed that the classification accuracies were highly imbalanced if no imbalanced learning technique was used. For instances, the minority class 7 always had 0% accuracy for all subjects but good performances were achieved for majority classes 1, 2, and 3. After imbalanced techniques had been applied, while the classification accuracies of majority classes (class 2, 3, 4) slightly decreased, accuracies of minor classes (class 1, 5, 6, 7) significantly increased. As a result, the performances of majority and minor classes became more balanced and the overall performance increased significantly. Different sampling algorithms appeared to perform similarly.

7.5 CONCLUSIONS

We have implemented five different imbalanced techniques for OFS assessment and validated our methods on a driving test benchmark dataset. Experimental results show that classification accuracies for minority classes are improved dramatically with a cost of slight performance degradations for majority classes, indicating that imbalanced learning techniques could be very useful for OFS assessment.

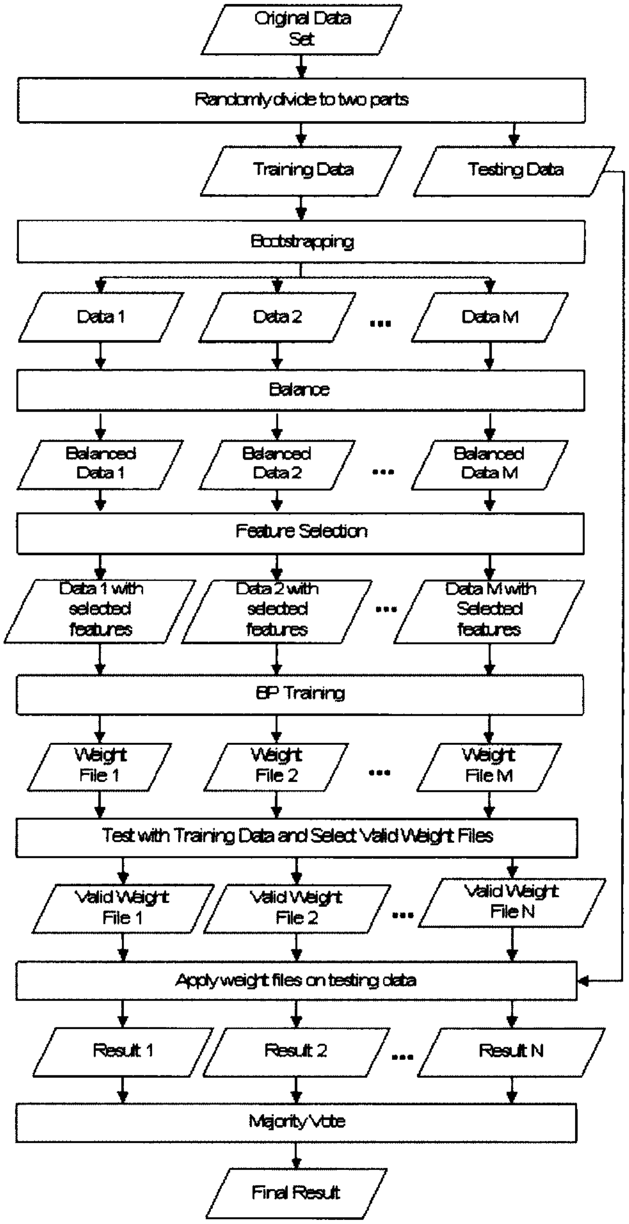


Figure 41: Design of the committee classifier

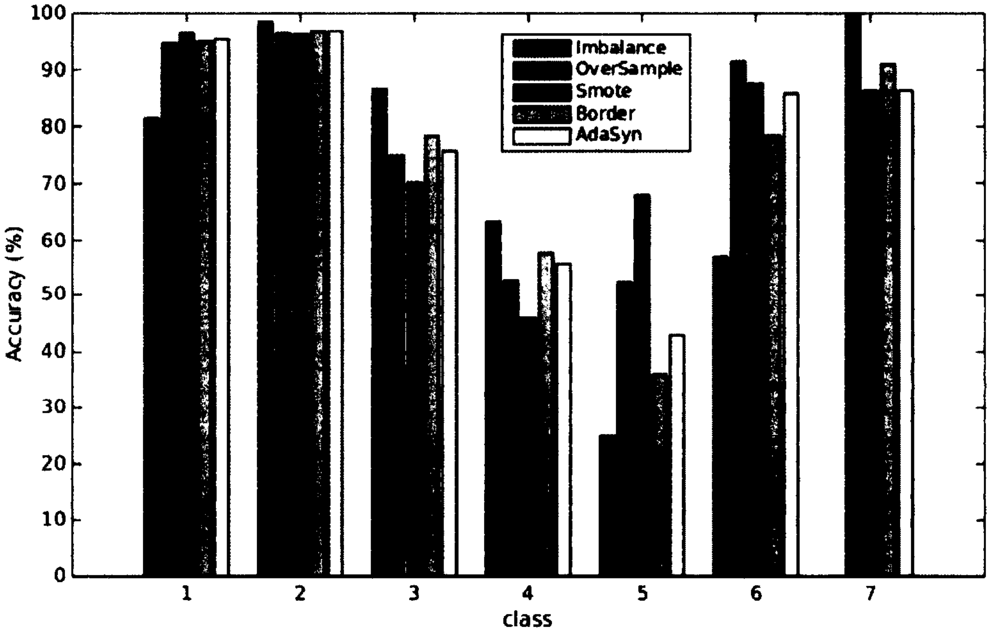


Figure 42: Result using imbalance techniques

Chapter 8

CONCLUSION

This dissertation studied methods for improving the performance of pilots' engagement assessment. Specifically, the dissertation addressed three challenge problems, individual variation, lack of labeled data, and imbalanced labeled data. Our work began from a systematic framework for data collecting and processing. Under the framework, experiments were conducted in a flight simulation, and then EEG and several other types of data were recorded and a small portion of the data were labeled by incorporating multiple resources. EEG features (1-Hz PSD bins) were extracted after artifacts had been removed from raw EEG signals. By analyzing the extracted features, we found that the features had been effectively extracted from the EEG data for individual subjects, but distribution of the highest ranked features for different subjects were significant different, which implied the problem of individual variation.

We proposed model individualization methods to solve the problem of individual variation. The dynamic classifier selection algorithm was proposed for model individualization and was compared to other two methods, base line normalization and similarity-based model replacement. Experimental results showed that baseline normalization and dynamic classifier selection could significantly improve cross-subject engagement assessment. It is worth it to note that our enhanced committee machine provided a mechanism to integrate similarity based and dynamic ensemble methods in an elegant way. Either the similarity-based method or dynamic ensemble method can be considered as a filter that deletes poorly performing models before voting.

We proposed a deep learning algorithm to address the challenge of learning with scarce label information. The deep learning method is able to learn valuable high-level features by taking advantage of both labeled and unlabeled data. The performance of deep models is sensitive to the selection of parameters and we discussed the strategy to find appropriate learning parameters. Our results showed that deep models incorporating dropout technique were better tools for engagement assessment when label information is scarce. The same conclusion was verified using another small size data set (ADNI).

For the problem of imbalanced labeled data, we implemented several imbalanced learning techniques that can balance datasets before training and therefore can improve learning performance. Experiments on extreme unbalanced driving data showed that imbalanced learning techniques significantly improved overall classification performance. This technique was embedded to both the enhanced committee machine and deep learning framework.

The contributions of this dissertation include several aspects. First, we proposed the novel design of an enhanced committee machine for engagement assessment. The enhanced committee machine utilized bootstrapping, feature selection, and different classifier algorithms to make the trained models more diverse, which is beneficial for the final voting performance. It also made it possible to train multiple models based on one data set. This property was fairly important, since our data size is relatively small but we needed lots of trained models for the voting procedure of the committee machine. Second, it was the first application of the dynamic classifier selection algorithm for model individualization. The dynamic classifier selection method provided a way to evaluate and choose well performing cross-subject models for the committee so to improve cross-subject classification performance. Finally, it was the first attempt to utilize the deep learning algorithm for the problem of scarce labeled data learning in engagement assessment for pilots. Deep learning techniques were originally successfully applied to large data sets. We extended deep learning algorithms to be utilized for small-sized data.

Our study has limitations. First, we need to collect more data from more subjects. It is difficult to draw statistically significant conclusion with current data size. Second, we need new methods to efficiently find ground truth. The current labeling method is inefficient and it is difficult to label large number of data. Finally, it is necessary to study the significance of EEG features in terms of neuroscience.

References

[1] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.

[2] H.-I. Suk and D. Shen, "Deep learning-based feature representation for ad/mci classification," in *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2013*, pp. 583–590, Springer, 2013.

[3] J. Frey, C. Mühl, F. Lotte, and M. Hachet, "Review of the use of electroencephalography as an evaluation method for human-computer interaction," *arXiv preprint arXiv:1311.2222*, 2013.

[4] F. Lotte, M. Congedo, A. Lécuyer, and F. Lamarche, "A review of classification algorithms for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 4, 2007.

[5] G. Matthews, S. E. Campbell, S. Falconer, L. A. Joyner, J. Huggins, K. Gilliland, R. Grier, and J. S. Warm, "Fundamental dimensions of subjective state in performance settings: task engagement, distress, and worry.," *Emotion*, vol. 2, no. 4, p. 315, 2002.

[6] B. T. Engel, "Stimulus-response and individual-response specificity," *AMA Archives of General Psychiatry*, vol. 2, no. 3, pp. 305–313, 1960.

[7] M. Marwitz and G. Stemmler, "On the status of individual response specificity," *Psychophysiology*, vol. 35, no. 01, pp. 1–15, 1998.

[8] G. Wilson, W. Fraser, M. Beaumont, M. Grandt, G. Varoneckas, H. Veltman, E. Svensson, A. Burov, B. Hockey, G. Edgar, *et al.*, "Operator functional state assessment," *NATO RTO Publication RTO-TR-HFM-104, NATO Research and Technology Organization, Neuilly sur Seine*, 2004.

[9] W. W. Guangfan Zhang *et al.*, "A systematic approach for real-time operator functional state assessment," in *MODSIM World Conference*, (Virginia Beach), 2011.

[10] G. R. J. Hockey, "Operator functional state as a framework for the assessment of performance degradation," *NATO SCIENCE SERIES SUB SERIES I LIFE AND BEHAVIOURAL SCIENCES*, vol. 355, pp. 8–23, 2003.

[11] G. R. J. Hockey, *Operator functional state: The prediction of breakdown in human performance.* Oxford, UK: OUP, 2005.

[12] T. T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, and W. Verplank, *ACM SIGCHI curricula for human-computer interaction.* ACM, 1992.

[13] J. Kjeldskov and C. Graham, "A review of mobile hci research methods," in *Human-computer interaction with mobile devices and services*, pp. 317–335, Springer, 2003.

[14] E. A. Curran and M. J. Stokes, "Learning to control brain activity: a review of the production and control of eeg components for driving brain–computer interface (bci) systems," *Brain and cognition*, vol. 51, no. 3, pp. 326–336, 2003.

[15] T. M. Vaughan, W. J. Heetderks, L. J. Trejo, W. Z. Rymer, M. Weinrich, M. M. Moore, A. Kübler, B. H. Dobkin, N. Birbaumer, E. Donchin, *et al.*, "Brain-computer interface technology: a review of the second international meeting.," *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 11, no. 2, pp. 94–109, 2003.

[16] B. Graimann, B. Allison, and G. Pfurtscheller, "Brain–computer interfaces: A gentle introduction," in *Brain-Computer Interfaces*, pp. 1–27, Springer, 2010.

[17] B. ISO, "6385: 2004 ergonomic principles in the design of work systems," *European Committee for Standardization*, 2004.

[18] IEA, *What is Ergonomics*, 2014 (accessed 10/26/2014).

[19] N. A. Stanton, M. S. Young, and C. Harvey, *Guide to methodology in ergonomics: Designing for human use.* CRC Press, 2014.

[20] C. Berka, D. J. Levendowski, M. N. Lumicao, A. Yau, G. Davis, V. T. Zivkovic, R. E. Olmstead, P. D. Tremoulet, and P. L. Craven, "Eeg correlates of task engagement and mental workload in vigilance, learning, and memory tasks," *Aviation, space, and environmental medicine*, vol. 78, no. Supplement 1, pp. B231–B244, 2007.

[21] R. H. Stevens, T. Galloway, and C. Berka, "Eeg-related changes in cognitive workload, engagement and distraction as students acquire problem solving skills," in *User Modeling 2007*, pp. 187–196, Springer, 2007.

[22] G. Chanel, C. Rebetez, M. Bétrancourt, and T. Pun, "Emotion assessment from physiological signals for adaptation of game difficulty," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 41, no. 6, pp. 1052–1063, 2011.

[23] J. Smallwood, J. B. Davies, D. Heim, F. Finnigan, M. Sudberry, R. O'Connor, and M. Obonsawin, "Subjective experience and the attentional lapse: Task engagement and disengagement during sustained attention," *Consciousness and cognition*, vol. 13, no. 4, pp. 657–690, 2004.

[24] M. Chaouachi, P. Chalfoun, I. Jraidi, and C. Frasson, "Affect and mental engagement: towards adaptability for intelligent systems," in *Proceedings of the 23rd International FLAIRS Conference, Daytona Beach, FL. http://citeseerx. ist. psu. edu/viewdoc/download*, Citeseer, 2010.

[25] T. N. Lal, T. Hinterberger, G. Widman, M. Schröder, N. Hill, W. Rosenstiel, C. E. Elger, N. Birbaumer, and B. Schölkopf, "Methods towards invasive human brain computer interfaces," in *Advances in Neural Information Processing Systems*, pp. 737–744, 2005.

[26] E. C. Leuthardt, K. J. Miller, G. Schalk, R. P. Rao, and J. G. Ojemann, "Electrocorticography-based brain computer interface-the seattle experience," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 14, no. 2, pp. 194–198, 2006.

[27] S. Baillet, J. C. Mosher, and R. M. Leahy, "Electromagnetic brain mapping," *Signal Processing Magazine, IEEE*, vol. 18, no. 6, pp. 14–30, 2001.

[28] S. Waldert, T. Pistohl, C. Braun, T. Ball, A. Aertsen, and C. Mehring, "A review on directional information in neural signals for brain-machine interfaces," *Journal of Physiology-Paris*, vol. 103, no. 3, pp. 244–254, 2009.

[29] R. Salmelin, M. Hámäaläinen, M. Kajola, and R. Hari, "Functional segregation of movement-related rhythmic activity in the human brain," *Neuroimage*, vol. 2, no. 4, pp. 237–243, 1995.

[30] N. Weiskopf, K. Mathiak, S. W. Bock, F. Scharnowski, R. Veit, W. Grodd, R. Goebel, and N. Birbaumer, "Principles of a brain-computer interface (bci) based on real-time functional magnetic resonance imaging (fmri)," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 6, pp. 966–970, 2004.

[31] S. M. Coyle, T. E. Ward, and C. M. Markham, "Brain–computer interface using a simplified functional near-infrared spectroscopy system," *Journal of neural engineering*, vol. 4, no. 3, p. 219, 2007.

[32] M. van Gerven, J. Farquhar, R. Schaefer, R. Vlek, J. Geuze, A. Nijholt, N. Ramsey, P. Haselager, L. Vuurpijl, S. Gielen, *et al.*, "The brain–computer interface cycle," *Journal of Neural Engineering*, vol. 6, no. 4, p. 041001, 2009.

[33] C.-T. Lin, L.-W. Ko, M.-H. Chang, J.-R. Duann, J.-Y. Chen, T.-P. Su, and T.-P. Jung, "Review of wireless and wearable electroencephalogram systems and brain-computer interfaces–a mini-review," *Gerontology*, vol. 56, no. 1, pp. 112–119, 2009.

[34] H.-J. Hwang, S. Kim, S. Choi, and C.-H. Im, "Eeg-based brain-computer interfaces: A thorough literature survey," *International Journal of Human-Computer Interaction*, vol. 29, no. 12, pp. 814–826, 2013.

[35] H. H. JASPER, "The ten twenty electrode system of the international federation," *Electroencephalography and clinical neurophysiology*, vol. 10, pp. 371–375, 1958.

[36] G. Chatrian, "Ten percent electrode system for topographic studies of spontaneous and evoked eeg activity," *Am J Electroencephalogr Technol*, vol. 25, pp. 83–92, 1985.

[37] R. Oostenveld and P. Praamstra, "The five percent electrode system for high-resolution eeg and erp measurements," *Clinical neurophysiology*, vol. 112, no. 4, pp. 713–719, 2001.

[38] D. W. Klass, "The continuing challenge of artifacts in the eeg," *American Journal of EEG Technology*, vol. 35, pp. 239–269, 1995.

[39] R. Vigário, J. Sarela, V. Jousmiki, M. Hamalainen, and E. Oja, "Independent component approach to the analysis of eeg and meg recordings," *Biomedical Engineering, IEEE Transactions on*, vol. 47, no. 5, pp. 589–593, 2000.

[40] A. Delorme, T. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis," *Neuroimage*, vol. 34, no. 4, pp. 1443–1449, 2007.

[41] J. Gao, C. Zheng, and P. Wang, "Online removal of muscle artifact from electroencephalogram signals based on canonical correlation analysis," *Clinical EEG and neuroscience*, vol. 41, no. 1, pp. 53–59, 2010.

[42] C. Fortgens and M. De Bruin, "Removal of eye movement and ecg artifacts from the non-cephalic reference eeg," *Electroencephalography and clinical neurophysiology*, vol. 56, no. 1, pp. 90–96, 1983.

[43] M. Nakamura and H. Shibasaki, "Elimination of ekg artifacts from eeg records: a new method of non-cephalic referential eeg recording," *Electroencephalography and clinical neurophysiology*, vol. 66, no. 1, pp. 89–92, 1987.

[44] K. Harke, A. Schlögl, P. Anderer, and G. Pfurtscheller, "Cardiac field artifact in sleep eeg," *Proceedings EMBEC99*, pp. 482–483, 1999.

[45] Z. Sahul, J. Black, B. Widrow, and C. Guilleminault, "Ekg artifact cancellation from sleep eeg using adaptive filtering," *Sleep Research A*, vol. 24, p. 486, 1995.

[46] J. Mejia-García, J. Martínez-de Juan, J. Saiz, J. García-Casado, and J. Ponce, "Adaptive cancellation of the ecg interference in external electroenterogram," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 3, pp. 2639–2642, IEEE, 2003.

[47] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, vol. 46. John Wiley & Sons, 2004.

[48] S. Devuyst, T. Dutoit, P. Stenuit, M. Kerkhofs, and E. Stanus, "Removal of ecg artifacts from eeg using a modified independent component analysis approach," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 5204–5207, IEEE, 2008.

[49] M. A. A. Dewan, M. Hossain, M. Hoque, and O. Chae, "Contaminated ecg artifact detection and elimination from eeg using energy function based transformation," in *Information and Communication Technology, 2007. ICICT'07. International Conference on*, pp. 52–56, IEEE, 2007.

[50] T.-P. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. J. Sejnowski, "Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects," *Clinical Neurophysiology*, vol. 111, no. 10, pp. 1745–1758, 2000.

[51] P. Berg and M. Scherg, "Dipole models of eye movements and blinks," *Electroencephalography and clinical Neurophysiology*, vol. 79, no. 1, pp. 36–44, 1991.

[52] G. Gratton, M. G. Coles, and E. Donchin, "A new method for off-line removal of ocular artifact," *Electroencephalography and clinical neurophysiology*, vol. 55, no. 4, pp. 468–484, 1983.

[53] M. Sood, V. Kumar, and S. V. Bhooshan, "Review of state of art in electrooculogram artifact removal from electroencephalogram signals," *International Journal of Enhanced Research in Science Technology & Engineering*, vol. 2, no. 4, pp. 32–41, 2013.

[54] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[55] B. Blankertz, G. Curio, and K.-R. Muller, "Classifying single trial eeg: Towards brain computer interfacing," *Advances in neural information processing systems*, vol. 1, pp. 157–164, 2002.

[56] G. Garcia, T. Ebrahimi, and J. Vesin, "Support vector eeg classification in the fourier and time-frequency correlation," in *Proceedings of the IEEE-EMBS First International Conference on Neural Engineering 2003*, pp. 591–594, IEEE, 2003. LTS-CONF-2003-030.

[57] P. C. Schutte, K. H. Goodrich, D. E. Cox, E. B. Jackson, M. T. Palmer, A. T. Pope, R. W. Schlecht, K. K. Tedjojuwono, A. C. Trujillo, R. A. Williams, *et al.*, "The naturalistic flight deck system: An integrated system concept for improved single-pilot operations," *NASA Technical Memorandum. NASA/TM-2007-215090*, 2007.

[58] W. Boucsein and R. W. Backs, "Engineering psychophysiology as a discipline: Historical and theoretical aspects," *Engineering psychophysiology. Issues and applications*, pp. 3–30, 2000.

[59] M. E. Smith, L. K. McEvoy, and A. Gevins, "Neurophysiological indices of strategy development and skill acquisition," *Cognitive Brain Research*, vol. 7, no. 3, pp. 389–404, 1999.

[60] L. J. Trejo, K. Knuth, R. Prado, R. Rosipal, K. Kubitz, R. Kochavi, B. Matthews, and Y. Zhang, "Eeg-based estimation of mental fatigue: convergent evidence for a three-state model," in *Foundations of Augmented Cognition*, pp. 201–211, Springer, 2007.

[61] A. T. Pope, E. H. Bogart, and D. S. Bartolome, "Biocybernetic system evaluates indices of operator engagement in automated task," *Biological psychology*, vol. 40, no. 1, pp. 187–195, 1995.

[62] F. Foerster, "Psychophysiological response specificities: A replication over a 12-month period," *Biological psychology*, vol. 21, no. 3, pp. 169–182, 1985.

[63] E. Olofsen, H. P. Van Dongen, C. G. Mott, T. J. Balkin, and D. Terman, "Current approaches and challenges to development of an individualized sleep and performance prediction model," *Open Sleep J*, vol. 3, pp. 24–43, 2010.

[64] A. A. Borbély, "A two process model of sleep regulation.," *Human neurobiology*, 1982.

[65] C. B. Saper, T. C. Chou, and T. E. Scammell, "The sleep switch: hypothalamic control of sleep and wakefulness," *Trends in neurosciences*, vol. 24, no. 12, pp. 726–731, 2001.

[66] P. Burton, L. Gurrin, and P. Sly, "Tutorial in biostatistics. extending the simple linear regression model to account for correlated responses: an introduction to generalized estimating equations and multi-level mixed modeling," *Statistics in medicine*, vol. 17, pp. 1261–1291, 1998.

[67] S. L. Zeger, K.-Y. Liang, and P. S. Albert, "Models for longitudinal data: a generalized estimating equation approach," *Biometrics*, pp. 1049–1060, 1988.

[68] E. Olofsen, D. F. Dinges, and H. Van Dongen, "Nonlinear mixed-effects modeling: individualization and prediction," *Aviation, space, and environmental medicine*, vol. 75, no. Supplement 1, pp. A134–A140, 2004.

[69] H. Van Dongen, G. Maislin, and D. F. Dinges, "Dealing with inter-individual differences in the temporal dynamics of fatigue and performance: importance and techniques," *Aviation, space, and environmental medicine*, vol. 75, no. Supplement 1, pp. A147–A154, 2004.

[70] S. Rajaraman, N. J. Wesensten, T. J. Balkin, and J. Reifman, "Individualized management of fatigue and cognitive performance impairment through biomathematical modeling," tech. rep., DTIC Document, 2009.

[71] G. Zhang, R. Xu, W. Wang, A. A. Pepe, F. Li, J. Li, F. McKenzie, T. Schnell, N. Anderson, and D. Heitkamp, "Model individualization for real-time operator functional state assessment," *Advances in Human Aspects of Aviation*, p. 417, 2012.

[72] T. Lan, A. Adami, D. Erdogmus, and M. Pavel, "Estimating cognitive state using eeg signals," *Journal of Machine Learning*, vol. 4, pp. 1261–1269, 2003.

[73] K.-Q. Shen, X.-P. Li, C.-J. Ong, S.-Y. Shao, and E. P. Wilder-Smith, "Eeg-based mental fatigue measurement using multi-class support vector machines with confidence estimate," *Clinical Neurophysiology*, vol. 119, no. 7, pp. 1524–1533, 2008.

[74] M. Ingre, T. ÅKerstedt, B. Peters, A. Anund, and G. Kecklund, "Subjective sleepiness, simulated driving performance and blink duration: examining individual differences," *Journal of sleep research*, vol. 15, no. 1, pp. 47–53, 2006.

[75] Y.-T. Wang, K.-C. Huang, C.-S. Wei, T.-Y. Huang, L.-W. Ko, C.-T. Lin, C.-K. Cheng, and T.-P. Jung, "Developing an eeg-based on-line closed-loop lapse detection and mitigation system," *Frontiers in neuroscience*, vol. 8, 2014.

[76] H. Almahasneh, W.-T. Chooi, N. Kamel, and A. S. Malik, "Deep in thought while driving: An eeg study on drivers cognitive distraction," *Transportation research part F: traffic psychology and behaviour*, vol. 26, pp. 218–226, 2014.

[77] C.-P. Chua, G. McDarby, and C. Heneghan, "Combined electrocardiogram and photoplethysmogram measurements as an indicator of objective sleepiness," *Physiological measurement*, vol. 29, no. 8, p. 857, 2008.

[78] G. Yang, Y. Lin, and P. Bhattacharya, "A driver fatigue recognition model based on information fusion and dynamic bayesian network," *Information Sciences*, vol. 180, no. 10, pp. 1942–1954, 2010.

[79] I. G. Damousis and D. Tzovaras, "Fuzzy fusion of eyelid activity indicators for hypovigilance-related accident prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 491–500, 2008.

[80] S. Hu and G. Zheng, "Driver drowsiness detection with eyelid related parameters by support vector machine," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7651–7658, 2009.

[81] C. Katsis, N. Ntouvas, C. Bafas, and D. Fotiadis, "Assessment of muscle fatigue during driving using surface emg," in *Proceedings of the IASTED International Conference on Biomedical Engineering*, vol. 262, 2004.

[82] V. Balasubramanian and K. Adalarasu, "Emg-based analysis of change in muscle activity during simulated driving," *Journal of Bodywork and Movement Therapies*, vol. 11, no. 2, pp. 151–158, 2007.

[83] T. DOrazio, M. Leo, C. Guaragnella, and A. Distante, "A visual approach for driver inattention detection," *Pattern Recognition*, vol. 40, no. 8, pp. 2341–2355, 2007.

[84] J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1139–1152, 2014.

[85] S. J. Lee, J. Jo, H. G. Jung, K. R. Park, and J. Kim, "Real-time gaze estimator based on driver's head orientation for forward collision warning system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 254–267, 2011.

[86] A. Tawari and M. M. Trivedi, "Robust and continuous estimation of driver gaze zone by dynamic analysis of multiple face videos," in *IEEE Intelligent Vehicles Symposium*, 2014.

[87] X. Fan, B.-C. Yin, and Y.-F. Sun, "Yawning detection for monitoring driver fatigue," in *Proceedings of the 2007 IEEE International Conference on the Machine Learning and Cybernetics, Hong Kong, China*, vol. 2, pp. 664–668, 2007.

[88] E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan, "Drowsy driver detection through facial movement analysis," in *Human-Computer Interaction*, pp. 6–18, Springer, 2007.

[89] J. Pohl, W. Birk, and L. Westervall, "A driver-distraction-based lane-keeping assistance system," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 4, pp. 541–552, 2007.

[90] L. M. Bergasa, J. M. Buenaposada, J. Nuevo, P. Jimenez, and L. Baumela, "Analysing driver's attention level using computer vision," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 1149–1154, IEEE, 2008.

[91] T. Wakita, K. Ozawa, C. Miyajima, K. Igarashi, I. Katunobu, K. Takeda, and F. Itakura, "Driver identification using driving behavior signals," *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 3, pp. 1188–1194, 2006.

[92] T. Ersal, H. J. Fuller, O. Tsimhoni, J. L. Stein, and H. K. Fathy, "Model-based analysis and classification of driver distraction under secondary tasks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 692–701, 2010.

[93] A. Doshi and M. Trivedi, "Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 887–892, IEEE, 2009.

[94] B. Cyganek and S. Gruszczyński, "Hybrid computer vision system for drivers' eye recognition and fatigue monitoring," *Neurocomputing*, vol. 126, pp. 78–94, 2014.

[95] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[96] O. Chapelle, B. Schölkopf, A. Zien, *et al.*, *Semi-supervised learning*, vol. 2. MIT press Cambridge, 2006.

[97] M. A. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 3, pp. 381–396, 2002.

[98] M. Weber, M. Welling, and P. Perona, *Unsupervised learning of models for recognition*. Springer, 2000.

[99] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[100] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[101] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[102] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[103] D. G. Jones, "Subjective measures of situation awareness," *Situation awareness analysis and measurement*, pp. 113–128, 2000.

[104] V. J. Gawron, *Human performance, workload, and situational awareness measures handbook.* CRC Press, 2008.

[105] S. W. Samn and L. P. Perelli, "Estimating aircrew fatigue: a technique with application to airlift operations," tech. rep., DTIC Document, 1982.

[106] R. Farmer and N. D. Sundberg, "Boredom proneness–the development and correlates of a new scale," *Journal of personality assessment*, vol. 50, no. 1, pp. 4–17, 1986.

[107] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. Mckeown, V. Iragui, and T. J. Sejnowski, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, no. 02, pp. 163–178, 2000.

[108] V. Krishnaveni, S. Jayaraman, L. Anitha, and K. Ramadoss, "Removal of ocular artifacts from eeg using adaptive thresholding of wavelet coefficients," *Journal of neural engineering*, vol. 3, no. 4, p. 338, 2006.

[109] V. Tresp, "Committee machines," 2001.

[110] J. Li, M. T. Manry, P. L. Narasimha, and C. Yu, "Feature selection using a piecewise linear network," *Neural Networks, IEEE Transactions on*, vol. 17, no. 5, pp. 1101–1115, 2006.

[111] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. IV–317, IEEE, 2007.

[112] E. Choi and C. Lee, "Feature extraction based on the bhattacharyya distance," *Pattern Recognition*, vol. 36, no. 8, pp. 1703–1709, 2003.

[113] S. H. Park, J. M. Goo, and C.-H. Jo, "Receiver operating characteristic (roc) curve: practical review for radiologists," *Korean Journal of Radiology*, vol. 5, no. 1, pp. 11–18, 2004.

[114] K. Woods, K. Bowyer, and W. P. Kegelmeyer Jr, "Combination of multiple classifiers using local accuracy estimates," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pp. 391–396, IEEE, 1996.

[115] G. Giacinto and F. Roli, "A theoretical framework for dynamic classifier selection," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 2, pp. 8–11, IEEE, 2000.

[116] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognition*, vol. 34, no. 9, pp. 1879–1881, 2001.

[117] A. H. Ko, R. Sabourin, and A. S. Britto Jr, "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, no. 5, pp. 1718–1731, 2008.

[118] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[119] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.

[120] K. Cho, A. Ilin, and T. Raiko, "Improved learning of gaussian-bernoulli restricted boltzmann machines," in *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 10–17, Springer, 2011.

[121] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[122] I. V. Tetko, D. J. Livingstone, and A. I. Luik, "Neural network studies. 1. comparison of overfitting and overtraining," *Journal of chemical information and computer sciences*, vol. 35, no. 5, pp. 826–833, 1995.

[123] L. Prechelt, "Early stopping-but when?," in *Neural Networks: Tricks of the trade*, pp. 55–69, Springer, 1998.

[124] A. Association *et al.*, "2012 alzheimers disease facts and figures," *Alzheimer's & Dementia*, vol. 8, no. 2, pp. 131–168, 2012.

[125] C. Davatzikos, P. Bhatt, L. M. Shaw, K. N. Batmanghelich, and J. Q. Trojanowski, "Prediction of mci to ad conversion, via mri, csf biomarkers, and pattern classification," *Neurobiology of aging*, vol. 32, no. 12, pp. 2322–e19, 2011.

[126] A. Nordberg, J. O. Rinne, A. Kadir, and B. Långström, "The use of pet in alzheimer disease," *Nature Reviews Neurology*, vol. 6, no. 2, pp. 78–87, 2010.

[127] M. D. Greicius, G. Srivastava, A. L. Reiss, and V. Menon, "Default-mode network activity distinguishes alzheimer's disease from healthy aging: evidence from functional mri," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 13, pp. 4637–4642, 2004.

[128] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.

[129] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[130] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[131] V. C. Pangman, J. Sloan, and L. Guse, "An examination of psychometric properties of the mini-mental state examination and the standardized mini-mental state examination: implications for clinical practice," *Applied Nursing Research*, vol. 13, no. 4, pp. 209–213, 2000.

[132] E. Kolibas, V. Korinkova, V. Novotny, K. Vajdickova, and D. Hunakova, "Adas-cog (alzheimer's disease assessment scale-cognitive subscale)-validation of the slovak version," *Bratislavské lek ́arske listy*vol. 101, no. 11, pp. 598–602, 2000.

[133] F. Li, L. Tran, K.-H. Thung, S. Ji, D. Shen, and J. Li, "Robust deep learning for improved classification of ad/mci patients," in *Machine Learning in Medical Imaging*, pp. 240–247, Springer, 2014.

[134] N. Kabani, D. MacDonald, C. Holmes, and A. Evans, "Automated 3-d extraction of inner and outer surfaces of cerebral cortex from mri," *NeuroImage*, vol. 7, no. 4, p. S717, 1998.

[135] C. Hinrichs, V. Singh, G. Xu, S. C. Johnson, A. D. N. Initiative, *et al.*, "Predictive markers for ad in a multi-modality framework: an analysis of mci progression in the adni population," *Neuroimage*, vol. 55, no. 2, pp. 574–589, 2011.

[136] I. Jolliffe, *Principal component analysis.* Wiley Online Library, 2002.

[137] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio, "Categorization by learning and combining object parts," in *Advances in neural information processing systems*, pp. 1239–1245, 2001.

[138] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proceedings of the 26th annual international conference on machine learning*, pp. 457–464, ACM, 2009.

[139] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with dirichlet process priors," *The Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[140] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias1," in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 41–48, Citeseer.

[141] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[142] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer.* John Wiley & Sons, 2008.

[143] D. Zhang, Y. Wang, L. Zhou, H. Yuan, D. Shen, A. D. N. Initiative, *et al.*, "Multimodal classification of alzheimer's disease and mild cognitive impairment," *Neuroimage*, vol. 55, no. 3, pp. 856–867, 2011.

[144] H. He and E. A. Garcia, "Learning from imbalanced data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1263–1284, 2009.

[145] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[146] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, pp. 973–978, Citeseer, 2001.

[147] S. Ertekin, J. Huang, and C. L. Giles, "Active learning for class imbalance problem," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 823–824, ACM, 2007.

Appendix A

SOFTWARE OF ENHANCED COMMITTEE MACHINE

A.1 INTRODUCTION

The software is developed for EEG signal preprocessing and engagement assessment. It includes 13 packages.

1. Common Toolbox. The package provides common functions, such as normalization, saving data as self-defined format.

2. EEG Toolbox. It includes the common individual functions for EEG signal preprocessing, such as filtering, spikes finding, artifacts removing, and power spectrum calculation.

3. EEG Code. It is for the generation of the data sets for the classification model.

4. Committee Machine. It implemented two fusion methods for multi-classification problem, voting and dynamic classification. This package depends on package Common Toolbox, Committee, PLNFeaSel, BpOrTraining, BpOrTesting, LibSVM and MatlabArsenal. By using LibSVM and MatlabArsenal, the package can utilize many models as classifiers, including SVM, KNN, Gaussian Mixture, Linear Discriminant Analysis, Maximum entopy, etc.

5. TestDll. A package to test "EngagementEvaluation", showing an example how to extract features from raw data. It depends on package ReadEegFile.

6. Committee. Fuse the classification results by voting and dynamic classification methods.

7. PLNFeaSel. Feature selection.

8. BpOrTraining. Train models with neural network method. BpOrTesting. Test with the model trained by BpOrTraining. ReadEegFiles. Read EEG, ECG, Eye tracking and time information from raw files provided by IOWA.

9. EngagementEvaluation. Online package provides two important interface, online feature extraction and online testing.

10. LibSVM. LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It is developed by Chih-Chung Chang and Chih-Jen Lin and downloaded from `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

11. MatlabArsenal. MATLABArsenal is an open-source MATLAB package that encapsulates a number of popular classification algorithms. It is developed by Rong Yan and downloaded from `http://www.informedia.cs.cmu.edu/yanrong/MATLABArsenal/MATLABArsenal.htm`

The relationship between packages can be illustrated as Figure 43.

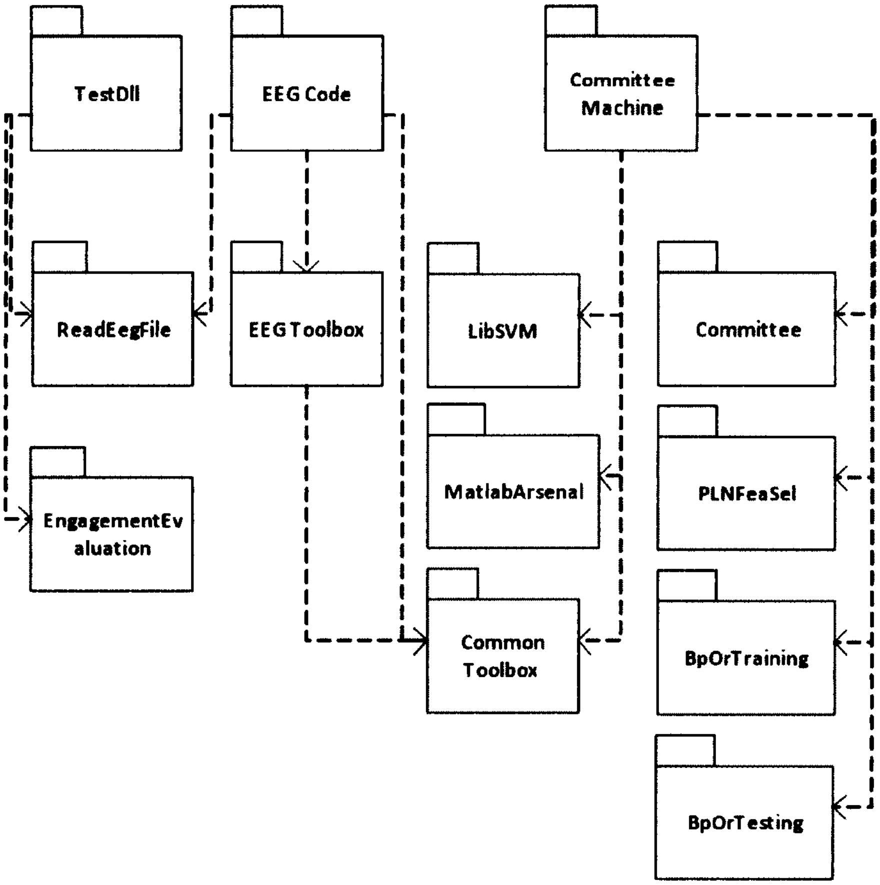


Figure 43: Relationship between packages.

A.2 DESIGN OF THE SOFTWARE

A.2.1 COMMITTEE MACHINE

The package Committee Machine was developed in Matlab using object-orient idea. The classes include:

1. CMaster. The manage class, providing interface to the outside and in charge of the inside procedures.
2. CConfig. The class CConfig is the entry to all configure parameters, which is composited by other four classes, CCfgBasic, CCfgClassifier, CCfgFeatureSelection, CCfgBagging.
3. CCfgBasic. Basic parameters.
4. CCfgClassifier. Parameters for each classifier.
5. CCfgBasicClassifier. Provide basic parameters for CCfgClassifier.
6. CCfgFeatureSelection. Parameters for feature selection program.
7. CCfgBagging. Parameters for bagging methods, including voting and dynamic classification.
8. CClassifier. Superclass for each classifier. Only provide interface like train, test, and validate, which should be implemented in the subclasses.
9. CClassifier_Neural. Subclass implementing the back propagation neural network method, depending on package BpOrTraining for training and BpOrTesting for testing and validating.
10. CClassifier_LibSVM. Subclass implementing support vector machine using package LibSVM.
11. CClassifier_Arsenal. Subclass implementing many different kinds of classifiers by using the package MatlabArsenal.

The structure of the package committee machine is illustrated as Figure 44.

A.2.2 ENGAGEMENT EVALUATION

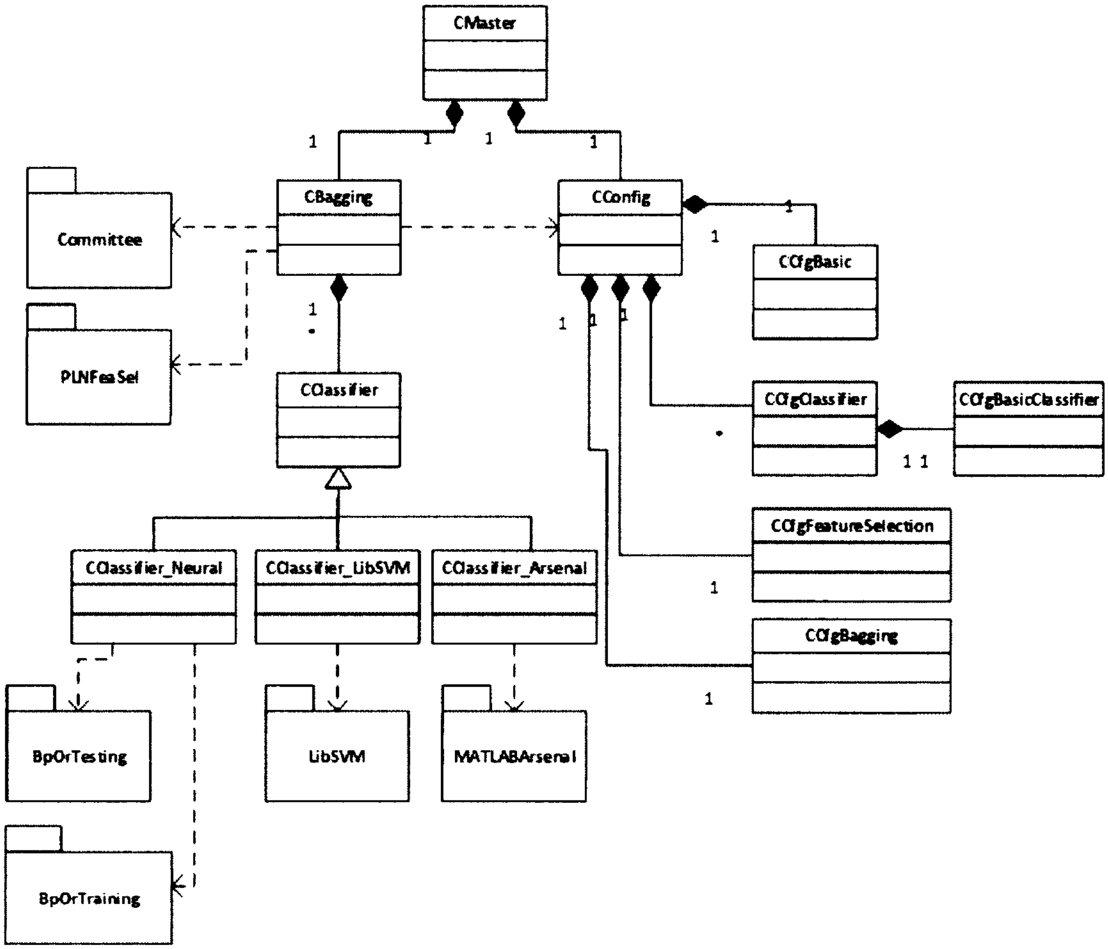


Figure 44: Static structure of committee machine.

Package Engagement Evaluation is developed as a dynamic link library, and provides interfaces for both online feature extraction and online classification testing. The static structure is shown as Figure 45. The following is a brief description of the classes.

1. CInterface exposes five interfaces to any host program that will invoke them.
2. CMaster is the manager class.
3. CPreprocess does the EEG signal preprocessing work, including artifact removal, resampling, and feature extraction (Power spectrum).
4. CNormalize normalizes the extracted features.
5. CModel are the models that have already been trained before testing procedure.

6. CCommittee calculate the testing IDs based on all models and achieve the final result by majority voting.

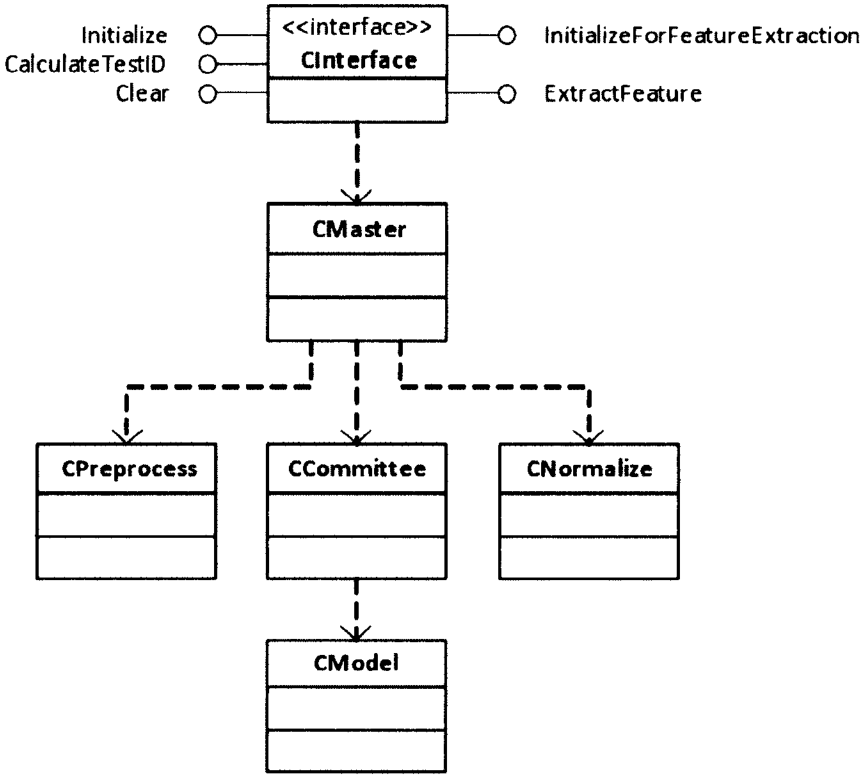


Figure 45: Static structure of package engagement evaluation.

A.2.3 FUNCTIONS IN PACKAGE COMMON TOOLBOX

1. formFiles_N_Fold. Forms files for N-fold classification.

2. getNormalizedData. Normalize a dataset either by its own mean and stand deviation or by given mean and stand deviation.

3. getOtherItems. Get items in a 1*n matrix excluding a specific item. For example, the return items for item 3 and matrix [1 2 3 4 5] is the matrix [1 2 4 5].

4. SaveAsTrainValidateTestFiles. Divide a giving file to three files, training file, validating file and testing file.

5. SaveCrossvalindFiles. Divide a giving file to two files using the Matlab's built-in function crossvalind.

6. SaveDataToFile. Save a matrix to file with specific format.
7. SaveDataToFile_all_int. Save a matrix with all integer elements to a file.
8. SaveFileAsSVM. Save file to another file with the format required by package LibSVM.

A.2.4 FUNCTIONS IN PACKAGE EEG TOOLBOX

1. calculatePSD. Calculate power spectrum.
2. designHighPass. Get a high pass filter.
3. designNotch. Get a notch filter.
4. doWavelet. Remove artifacts using wavelet methods.
5. doWavelet_multiChannels. Apply "doWavelet" on EEG data with multi-channels.
6. filterByNotch_Highpass. Fitler by using the given notch and high pass filters.
7. findSpike. Find spikes and excursions.
8. findSpike_MultiChannels. Apply "findSpike" on EEG data with multi-channels.
9. getFilteredData. Filtering the data by segments dividing by spikes or excursions.
10. getReferenceData. Do the Re-Reference operation on EEG data.
11. matchTime_Label. Match the time list with labels.
12. removeArtifact. Divide the dataset to segments and then implement function "findSpike_MultiChannels".

A.2.5 FUNCTIONS IN PACKAGE EEG CODE

1. calculatePsdFromOrgData. Calculate power spectrum from the original dataset.

2. doSmooth. Smooth a dataset by give number of data points.
3. formFiles_AverageModel. Generate file for average classification model.
4. getCleanData. Get clean EEG dataset by filtering, removing spikes and artifacts.
5. splitToClassFiles. Split a data file to several files according to the class ID. As a result, each produced file only contains data points for one class.
6. test_ExtractFeatures. Testing code for feature extraction.
7. test_RemoveArtifacts. Testing code for artifacts removing.
8. formFiles_AverageModel. Form files for training, testing for individual and average models based on extracted features.

A.3 INSTALLATION AND CONFIGURATION

1. Download and install QT from `http://qt.nokia.com/products/`, which is depended by package "ReadEegFiles".
2. Copy and unzip all packages.
3. Compile all c++ packages.
4. Make a subdirectory "support_packages" under package "CommitteePackages", and copy "BpOrTesting.exe", "BpOrTraining.exe", "Committee.exe", and "PLNFeaSel.exe" to this directory.
5. Download package "libSVM", unzip it to directory "support_packages".
6. Download pakage "MatlabArsenal", then unzip it to any directory and add the directory to Matlab path.
7. Add package "Common Toolbox" and "EEG Toolbox" to Matlab path.

Now the software is ready to use. It is recommended that use the online package "EngagementEvaluation" to extract features. Then use "formFiles_AverageModel" (package "EEG_Code") to form files for training and testing. Finally run package 'Committee Machine to do classification.

A.4 TYPICAL USAGE OF THE SOFTWARE

A.4.1 FEATURE EXTRACTION

Features can be extracted by using Matlab code in the package "EEG Code", but the recommended way is to use the online package "EngagementEvaluation", which is a "dll" and provides interfaces for both feature extraction and classification testing. The procedures can follow the example given by "TestDll".

1. Find ground true and the associated time slot. And then use them in function initSubjectTimeID() to initialize the program.

2. Utilize package "ReadEegFiles" to extract EEG segments from the raw signals.

3. Initialize package "EngagementEvaluation" by calling interface "InitializeForFeatureExtraction".

4. Call another interface "ExtractFeature" by feeding one-second data samples each time.

5. Save extracted features.

A.4.2 OFFLINE TRAIN AND TEST

Offline training and testing are implemented by package "Committee machine". To make it run as flexible as possible, many parameters need to be configured. The parameters are categorized to five groups and correspondingly, five structures are designed. Please follow file "main.m" to initialize these parameters and run the program.

1. CCfgBasic. Define basic parameters, including 5 members.

   (a) m_vSubjects. Serial number or label for each subject. Must to be number, not necessary to be consequent. For example, if we have subject 2, 4, 5, 6, then we define it as: cfgBasic. m_vSubjects = [2, 4, 5, 6].

   (b) m_strDataRootDir. The root directory for data sets. The data set for each subject will be at the directory "m_strDataRootDir/datax", where x is the serial number of the subject. For example, if

$m\_strDataRootDir =' /home/datasets'$, then data directory for subject 2 is "$/home/datasets/data2$".

(c) m_strResultDir. The directory for any output results, including selected features, trained models, testing results, etc. If $m\_strResultDir =' situation\_1'$ and $m\_strDataRootDir =' /home/datasets'$, then the actual output directory is, "$/home/datasets/data2/situation\_1$".

(d) m_nInput. Number of features

(e) m_nOutput. Number of class ID. Class ID must be integer numbers starting from 1. If $m_nOutput = 2$, then valid class IDs must be 1 and 2.

2. CCfgClassifier. Parameters for each classifier.

   (a) m_nIndex. Index of the classifier, starting from 1.

   (b) m_nType. Type of the classifier. 1: neural network; 2: libSVM; 3: arsenal classifier.

   (c) m_nConfig. Specific parameters for different classifier and parameters for neural network classifier:

      i. m_strTrainProgram, set the training program, default is "$./support\_packages/BpOrTraining.exe$".

      ii. m_strTestProgram, set the testing program, default is "$./support\_packages/BpOrTesting.exe$".

      iii. m_nHiddenUnits, number of hidden units, default is 5.

      iv. m_nIteration, number of iterations, default is 15.

   (d) m_bTrain, set to "true" if need to train models; otherwise 'false.

   (e) m_bTest, set to "true" if need to get testing result; otherwise "false".

   (f) m_cfgBasic, another structure setting suffixes for training, testing and validating data files.

3. CCfgBasicClassifier. Set suffixes for training, testing and validating data files.

   (a) m_strTrainDataFile_Suffix.

   Set suffix for training data file name. Explain it with an example. If $CCfgBasic :: m\_strDataRootDir =' /home/datasets'$, serial number of the subject is 2, and $m\_strTrainDataFile\_Suffix =' train\_'$, then the train file name is, $'/home/datasets/data2/train\_2.txt'$.

(b) m_strTestDataFile_Suffix. Similar to (1).

(c) m_strValidateDataFile_Suffix. Similar to (1).

4. CCfgFeatureSelection. Parameters for feature selection program.

   (a) m_bFeatureSelection. Set to true if need to do feature selection.

   (b) m_nMaxCluster_Sel. Set maximum number of clusters. The number may need to try several different values, starting from the 1/3 of the number of features.

   (c) m_nMinFeatureNum. Set minimum acceptable number of selected features. If the number of selected features is less than m_nMinFeatureNum, then the feature selection program will run again, but it only try 10 times at most.

   (d) m_nMaxFeatureNum. Set maximum acceptable number of selected features. If more features have been selected, only top m_nMaxFeatureNum features are kept.

   (e) m_strFeatureSelProgram. Program for feature selection. Default is $'./support\_packages/PLNFeaSel.exe'$.

5. CCfgBagging. Parameters for bagging methods, including voting and dynamic classification.

   (a) m_nBalanceMethod. Set type of balance methods. If an valid method is set and the number of data samples belonging to different classes are not balanced, or in other word, the data samples of one class are much more then data sample of another class, then the data samples will be made balanced before training. The value could be, 0: No balance; 1: Downsampling; 2: Upsampling; 3: Smote; 4: BorderSmote; 5: AdaptiveSyn.

   (b) m_strBaggingProgram. Set bagging program. Default is $'./support\_packages/committee.exe'$.

   (c) m_nNeighbors. Set the number of neighbors for each testing sample. This parameter is only needed when dynamic classification is utilized as bagging method.

  (d) m_nBaggingMethod. Set bagging or fusing method. The value could be, 0: no bagging, 1: only voting, 2: both voting and dynamic classification.

A.5 ONLINE TEST

The package EngagementEvaluation implements online testing. Before use it, neural network models should be trained in advance. Each model contains two files, Model_x.txt and selectedFeatures_x.txt, where x is the index of the subject. The procedures to utilize this dynamic link library include:

1. Call function bool Initialize(int nSamplingFreq, std::string strModelDir, int nNumModels). The parameters:

   (a) nSamplingFreq, sampling frequence.

   (b) strModelDir, the directory containing models.

   (c) nNumModels, the number of models.

2. Call function int CalculateTestID(const double* pEEGData, int nNum) every second and the return value is the testing ID. The possible testing ID,

   (a) -1, invalid value, may be caused by spikes.

   (b) 1, disengaged.

   (c) 2, engaged.

   The parameters,

   (a) nNum, number of data items in the array pEEGData.

   (b) pEEGData, data samples. Here we assume 32-channel EEG data are coming in, so each data sample is a 1*32 matrix. If the sampling frequency is 500, and the function is called each second, then input EEG data is a 500*32 matrix. Here the data should be arranged row by row to a 1D double array with 500*32=16000 items.

3. Before the host program exits, call function void clear() to release memory.

Appendix B

SOFTWARE OF DEEP LEARNING

B.1 INTRODUCTION

The deep learning package was implemented in Matlab. It supports pre-training using RBM and fine-tuning as two types of network: autoencoder or classifier. It is also capable of multi-modality problem. The software can be considered as a data-driven framework. The structure of the networks, the parameters for training, and which modality got involved, are defined in a data structure, which will be transferred to the software and control the behaviour of the deep learning program.

This manual consists of three parts. First, a big picture of the design idea is introduced. Second, the framework of the software is illustrated. Finally, an example is given to present how to design, train and use a deep network.

B.2 BIG PICTURE OF THE DESIGN

The design considers the networks as multi-modalities problem and one modality problem is a special case. For different modality, the pre-train runs independently. Or in other words, there are no interactions between different modalities during pre-training procedures. Then the pre-trained weights of the networks are utilized for fine-tune. During fine-tune, the modalities get connected with a shared layer of the networks, and the shared layer is the output as high-level features. Typically, there are basically two type of fine-tune networks: deep auto-encoder using folded networks, and deep classifier using class labels. For multi-modality networks, there are more variance. If we define the networks above the shared layer as upper networks, and the networks below the shared layer as lower layer, then there exist many possibilities to combine different upper or lower networks.

To make the structure of the networks most flexible, the frame of the networks is initialized with maximum networks. Each network could be activated or deactivated. Only activated networks get involved in fine-tune. By selecting and setting interested networks to be activated, we can easily define deep auto-encoder, deep classifier networks with one or more modalities.

Here is an example that consists of two modalities (Figure 46). Modality one contains 312 EEG features and the structure of networks is defined as 312-600-100-20. Modality two contains 10 ECG and EOG features and the structure of networks is defined as 10-50-20. The layer with 20 units is shared between two modalities. During pre-train, two networks run RBM layer by layer independently (Figure 47). In fine-tune procedure, many variants of networks can be generated an by activating different modality of the networks, for example, networks for modality one as Figure 48, networks for modality two as Figure 49, and networks for two modalities as Figure 50.

There is a special case that is also support by the software. After pre-train, we can first build an deep auto-encoder network. Then the upper network is replaced with ID layer but with the lower network and the shared layer unchanged, and therefore a deep classifier is built. So in this case, the model is fine-tuned using deep auto-encoder network and then further fine-tuned using deep classifier.

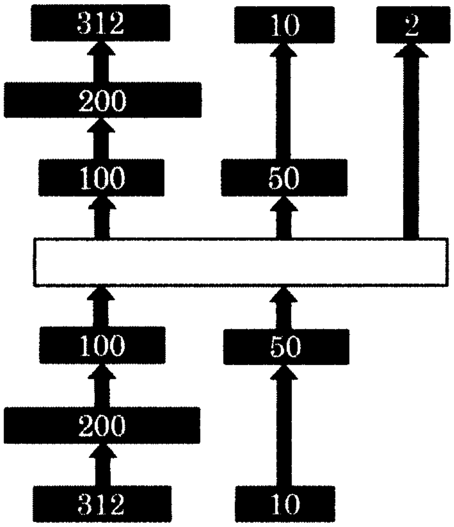


Figure 46: Two modalities with whole networks available.

B.3 DETAIL DESIGN

There are three classes designed for the deep learning software. Class *CTestMN* exposes an interface to caller, and is in charge of the organizing of the networks, and the process for pre-train and fine-tune. Class *CTestMN* depends on two other classes, Class *CRBM* and Class *CMN*. Class *CRBM* processes pre-train using RBM for an layer of networks. Class *CMN* manages the fine-tune procedures for a network.

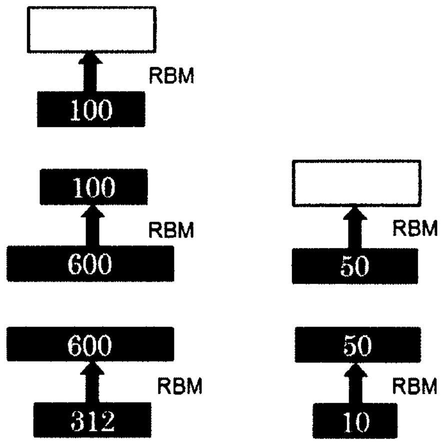


Figure 47: Two modalities of networks are independently pre-trained using RBM layer by layer.

B.3.1 CTESTMN

Data members for *CTestMN*,

1. m_DataSet, data for training and testing.
2. m_OptSet, definition of parameters.
3. m_vCfgFineTune, definition of the structure of networks for fine-tune.
4. m_MN, instance of CMN, in charge of the procedures of fine-tune.
5. m_vRBM, instance of CRBM, in charge of the procedures of pre-train.
6. m_theResults, fine-tune results for output.

Important method members for *CTestMN*

1. CTestMN, constructor function of CTestMN, initialize m_DataSet, m_OptSet, and m_vRBM
2. SetFineTuneCases, initialize m_vCfgFineTune.
3. PreTrain, manage the procedures of pre-train.
4. DoPreTrain, called by PreTrain and do specific pre-train.

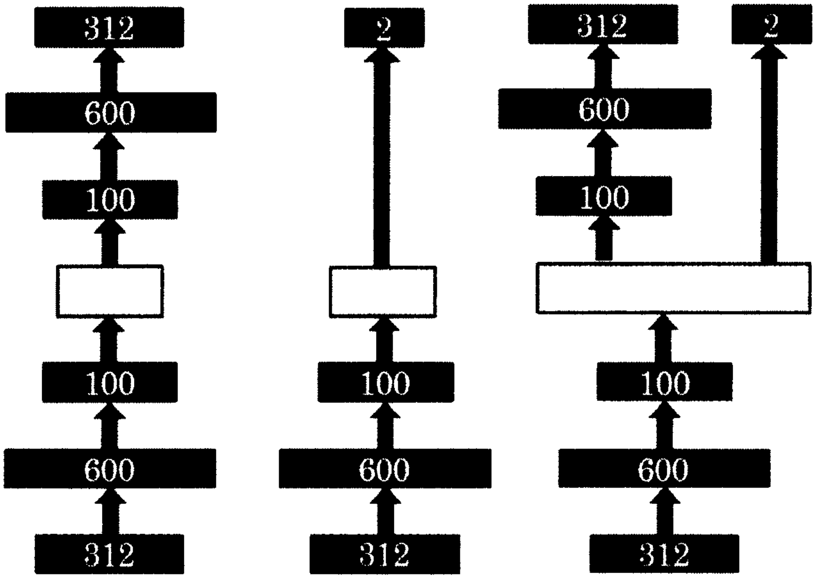


Figure 48: Modality One Variants, including standard deep auto-encoder, classifier, and auto-encoder + classifier

5. FineTune, manage the procedures of fine-tune.
6. DoFineTune, called by FinTune and do specific fine-tune.
7. ProcessDefault, initialized the default configurations and run default pre-train and fine-tune.

B.3.2 CRBM

Data members for *CRBM*,

1. m_sizes, define the layers and the number of units in each layer.
2. m_rbm, is a structured data set for each layer of rbm.
3. m_bUseGPU, set if use or not use GPU

Method members for *CRBM*,

1. CRBM, constructor function for initialization
2. Train, manage the procedure of pre-train.

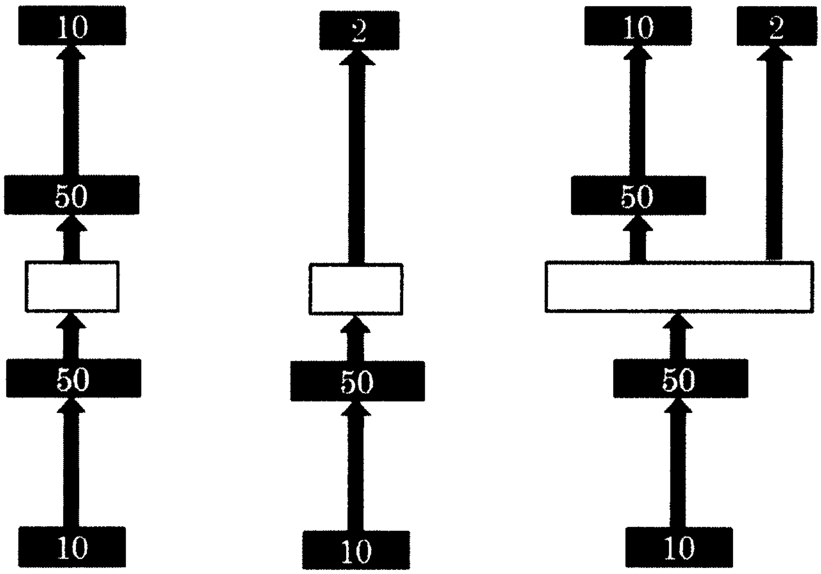


Figure 49: Modality two variants, including standard deep auto-encoder, classifier, and auto-encoder + classifier.

3. DoRBMTrain, called by Train and run specific RBM

B.3.3 CMN

Data members for *CMN*,

1. m_MN, a dataset that controls the fine-tune procedures. It contains a number of data members,
   (a) nShareLayerSize, the number of units in the share layer.
   (b) nNumModalities, number of modalities.
   (c) vLowerNet, the networks below the shared layer
   (d) vUpperNet, the networks above the shared layer
   (e) opt, define the parameters for fine-tune, including momentum, learning rate, scaling learning rate, weight penalty for L2 regularization, non-sparse penalty, sparse target, and fraction of zero mask.
2. m_bUseGPU, set if use or not use GPU for fine-tune.

Method members for *CMN*

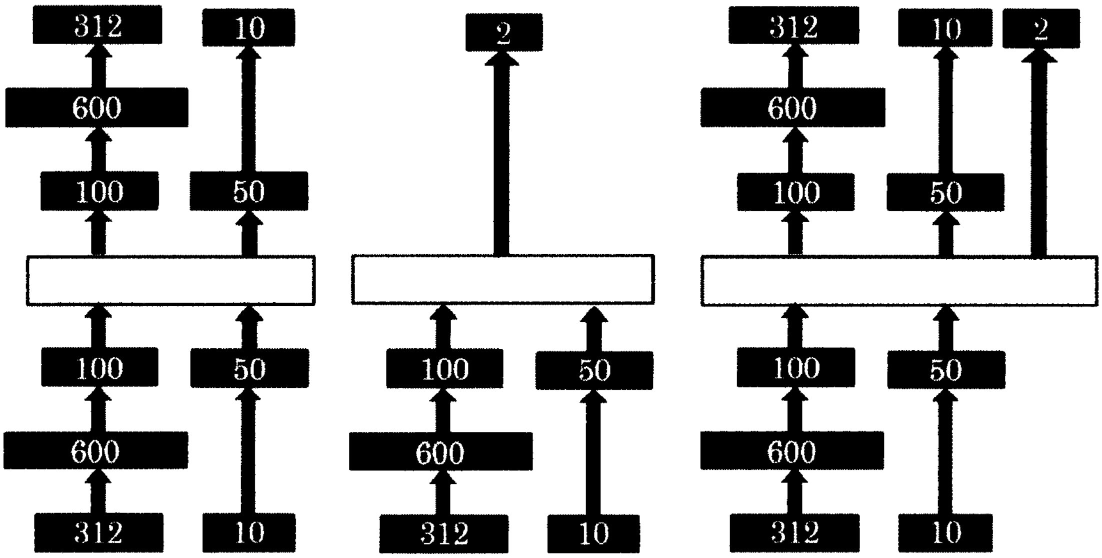


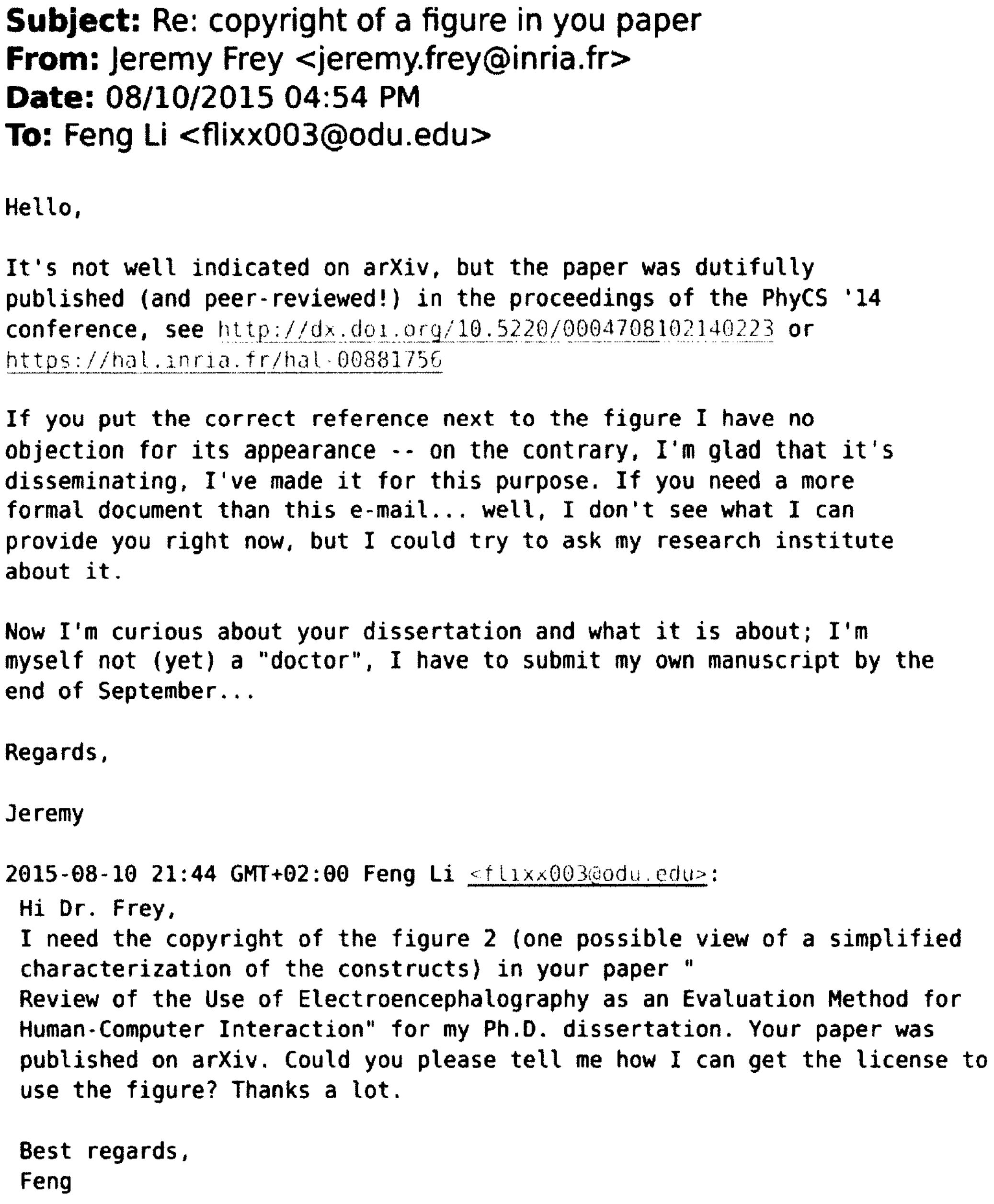
Figure 50: Two modalities variants, including standard deep auto-encoder, classifier, and auto-encoder + classifier.

1. CMN, constructor for initialization.
2. Initialize, do specific initialization.
3. SetActiveNets, set those networks involved in fine-tune to be active.
4. SetOpts, set parameters for fine-tune.
5. SetWeights, initialize the weights of the networks with pre-trained weights.
6. TrainNet, do fine-tune.

Appendix C

REUSE LICENSES

Figure 51: Reuse license for Figure 2



Subject: Re: copyright of a figure in you paper
From: Jeremy Frey <jeremy.frey@inria.fr>
Date: 08/10/2015 04:54 PM
To: Feng Li <flixx003@odu.edu>

Hello,

It's not well indicated on arXiv, but the paper was dutifully published (and peer-reviewed!) in the proceedings of the PhyCS '14 conference, see http://dx.doi.org/10.5220/0004708102140223 or https://hal.inria.fr/hal-00881756

If you put the correct reference next to the figure I have no objection for its appearance -- on the contrary, I'm glad that it's disseminating, I've made it for this purpose. If you need a more formal document than this e-mail... well, I don't see what I can provide you right now, but I could try to ask my research institute about it.

Now I'm curious about your dissertation and what it is about; I'm myself not (yet) a "doctor", I have to submit my own manuscript by the end of September...

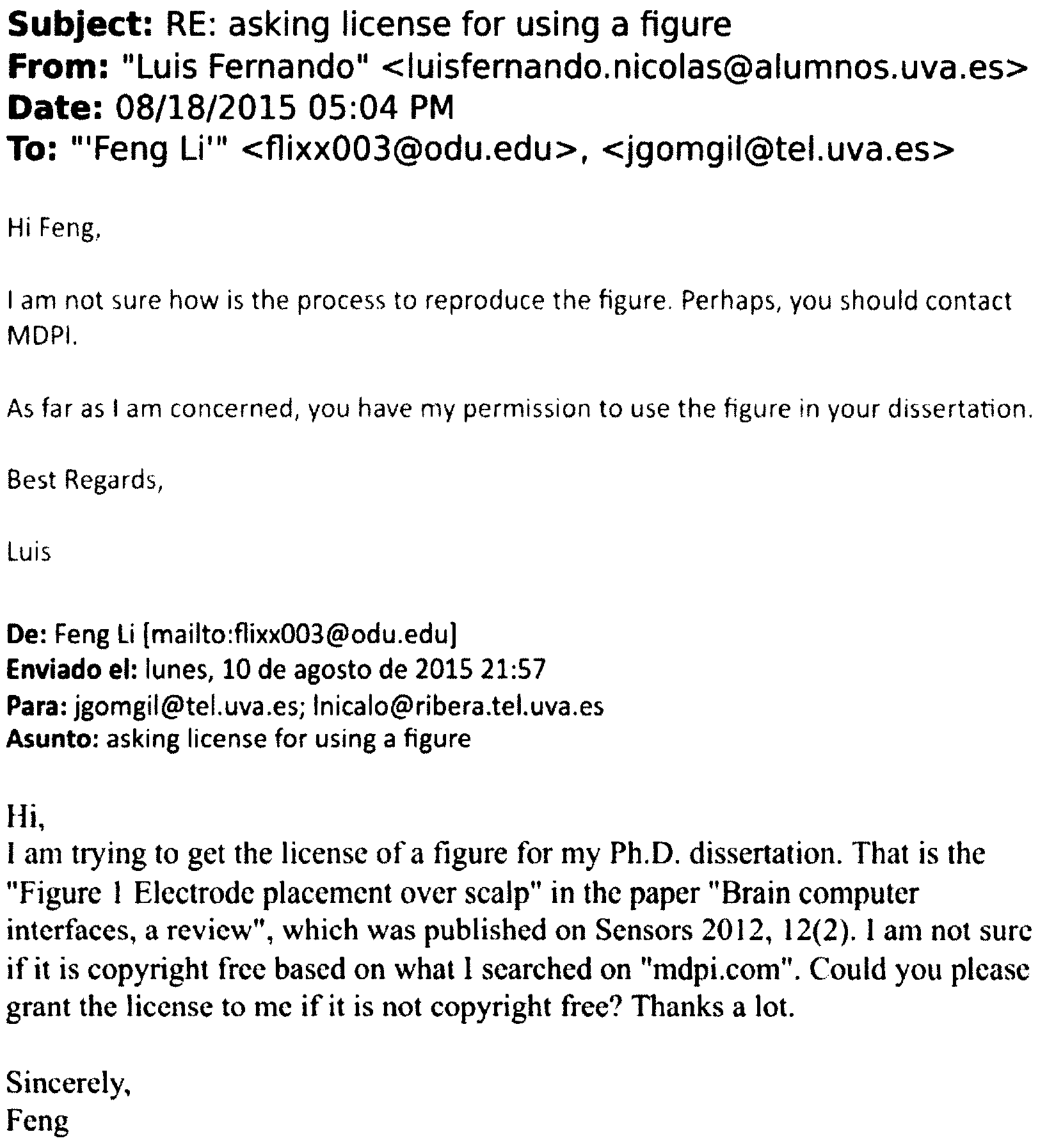
Regards,

Jeremy

2015-08-10 21:44 GMT+02:00 Feng Li <flixx003@odu.edu>:

> Hi Dr. Frey,
> I need the copyright of the figure 2 (one possible view of a simplified characterization of the constructs) in your paper "
> Review of the Use of Electroencephalography as an Evaluation Method for Human-Computer Interaction" for my Ph.D. dissertation. Your paper was published on arXiv. Could you please tell me how I can get the license to use the figure? Thanks a lot.
>
> Best regards,
> Feng

Figure 52: Reuse license for Figure 5



Subject: RE: asking license for using a figure
From: "Luis Fernando" <luisfernando.nicolas@alumnos.uva.es>
Date: 08/18/2015 05:04 PM
To: "'Feng Li'" <flixx003@odu.edu>, <jgomgil@tel.uva.es>

Hi Feng,

I am not sure how is the process to reproduce the figure. Perhaps, you should contact MDPI.

As far as I am concerned, you have my permission to use the figure in your dissertation.

Best Regards,

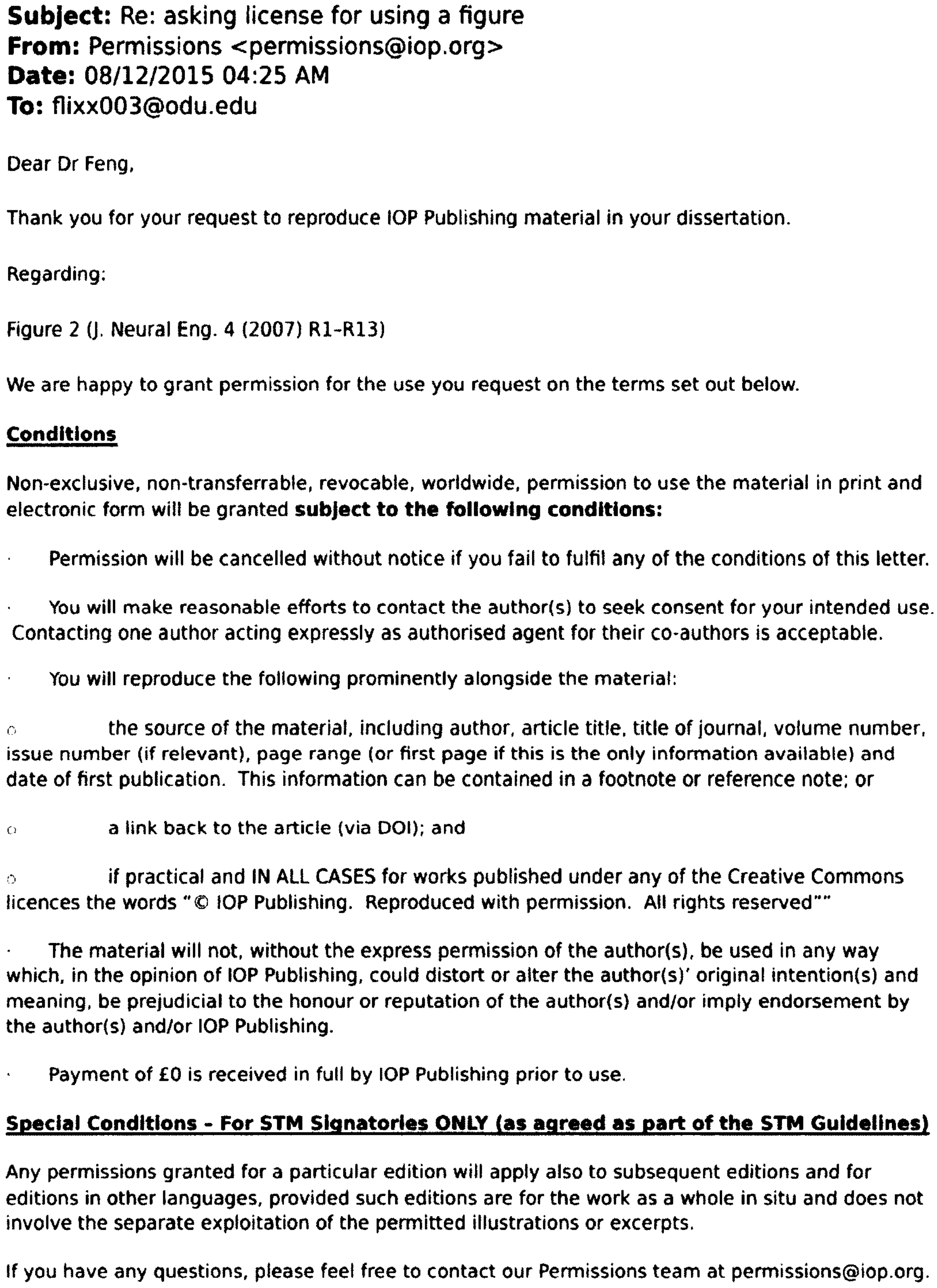
Luis

De: Feng Li [mailto:flixx003@odu.edu]
Enviado el: lunes, 10 de agosto de 2015 21:57
Para: jgomgil@tel.uva.es; lnicalo@ribera.tel.uva.es
Asunto: asking license for using a figure

Hi,
I am trying to get the license of a figure for my Ph.D. dissertation. That is the "Figure 1 Electrode placement over scalp" in the paper "Brain computer interfaces, a review", which was published on Sensors 2012, 12(2). I am not sure if it is copyright free based on what I searched on "mdpi.com". Could you please grant the license to me if it is not copyright free? Thanks a lot.

Sincerely,
Feng

Figure 53: Reuse license for Figure 6



Subject: Re: asking license for using a figure
From: Permissions <permissions@iop.org>
Date: 08/12/2015 04:25 AM
To: flixx003@odu.edu

Dear Dr Feng,

Thank you for your request to reproduce IOP Publishing material in your dissertation.

Regarding:

Figure 2 (J. Neural Eng. 4 (2007) R1-R13)

We are happy to grant permission for the use you request on the terms set out below.

Conditions

Non-exclusive, non-transferrable, revocable, worldwide, permission to use the material in print and electronic form will be granted **subject to the following conditions:**

- Permission will be cancelled without notice if you fail to fulfil any of the conditions of this letter.

- You will make reasonable efforts to contact the author(s) to seek consent for your intended use. Contacting one author acting expressly as authorised agent for their co-authors is acceptable.

- You will reproduce the following prominently alongside the material:

  - the source of the material, including author, article title, title of journal, volume number, issue number (if relevant), page range (or first page if this is the only information available) and date of first publication. This information can be contained in a footnote or reference note; or

  - a link back to the article (via DOI); and

  - if practical and IN ALL CASES for works published under any of the Creative Commons licences the words "© IOP Publishing. Reproduced with permission. All rights reserved""

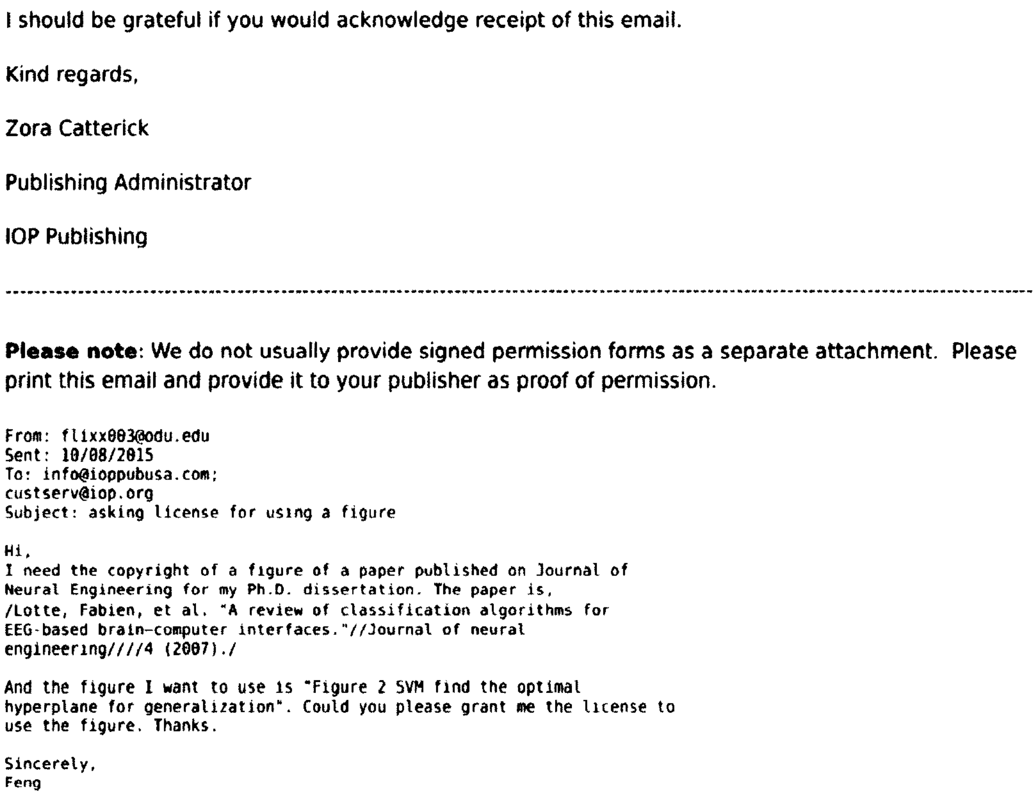
- The material will not, without the express permission of the author(s), be used in any way which, in the opinion of IOP Publishing, could distort or alter the author(s)' original intention(s) and meaning, be prejudicial to the honour or reputation of the author(s) and/or imply endorsement by the author(s) and/or IOP Publishing.

- Payment of £0 is received in full by IOP Publishing prior to use.

Special Conditions - For STM Signatories ONLY (as agreed as part of the STM Guidelines)

Any permissions granted for a particular edition will apply also to subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted illustrations or excerpts.

If you have any questions, please feel free to contact our Permissions team at permissions@iop.org.



I should be grateful if you would acknowledge receipt of this email.

Kind regards,

Zora Catterick

Publishing Administrator

IOP Publishing

Please note: We do not usually provide signed permission forms as a separate attachment. Please print this email and provide it to your publisher as proof of permission.

From: flixx003@odu.edu
Sent: 10/08/2015
To: info@ioppubusa.com;
custserv@iop.org
Subject: asking license for using a figure

Hi,
I need the copyright of a figure of a paper published on Journal of Neural Engineering for my Ph.D. dissertation. The paper is, /Lotte, Fabien, et al. "A review of classification algorithms for EEG-based brain-computer interfaces."//Journal of neural engineering////4 (2007)./

And the figure I want to use is "Figure 2 SVM find the optimal hyperplane for generalization". Could you please grant me the license to use the figure. Thanks.

Sincerely,
Feng

VITA

Feng Li
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

Feng Li achieved B.S. and M.S. in Thermal and Power Engineering from Huazhong University of Science and technology, China in 1994 and 1997 respectively. In 2010, he began to continue his education at Old Dominion University in pursuit of a Ph.D. Degree in Electrical and Computer Engineering. He was supported by university Graduate Assistantship during his Ph.D. work.

Typeset using LaTeX.

LIST OF PUBLICATIONS

SUBMITTED MANUSCRIPTS

1. **Feng Li**, Guangfan Zhang, Roger Xu, Tom Schnell, Jonathan Wen, and Jiang Li, Deep Models for Engagement Assessment with Scarce Label Information, revised to *IEEE Transactions on Human-Machine Systems*

JOURNAL ARTICLES

1. **Feng Li**, Loc Tran, Kim-Han Thung, Shuiwang Ji, Dinggang Shen, and Jiang Li. "A Robust Deep Learning for Improved Classification of AD/MCI Patients", *IEEE Journal of Biomedical and Health Informatics*, 2015, in press

2. Hoang-Anh T. Nguyena, John Mussonb, **Feng Li**, Wei Wang, Guangfan Zhang, Roger Xu, Carl Richey, Tom Schnell, Frederic D. McKenzie, Jiang Li, "EEG Artifact Removal Using A Wavelet Neural Network", in *Neurocomputing*, Volume 97, 15 November 2012, Pages 374-389

BOOK CHAPTER

1. **Feng Li**, Loc Tran, Kim-Han Thung, Shuiwang Ji, Dinggang Shen, and Jiang Li. "Robust Deep Learning for Improved Classification of AD/MCI Patients," In *Machine Learning in Medical Imaging*, pp. 240-247. Springer International Publishing, 2014.

2. Guangfan Zhang, Roger Xu, Wei Wang, Aaron A. Pepe, **Feng Li**, Jiang Li, Frederick McKenzie, Tom Schnell, Nick Anderson, and Dean Heitkamp, "Model individualized for real-time operator functional state assessment," in *Advances in Human Aspects of Aviation*, Jul 2012 , 417 -426

3. Jiang Li, Natalie Chuang, VamsiMantena, **Feng Li** and Frederick D. McKenzie, "Application of Machine Learning Techniques to Biomarker Identification for Prostate Cancer," in *Chapter 5 in Computational Techniques and Algorithms for Image Processing: Reviews, Principles and Applications on Pattern*

Recognition, Image Enhancement, Compression and Watermarking, ISBN: 978-3-8433-5802-6, edit by: S. Ramakrishnan and Ibrahiem M. M. EI Emary, Publisher: Lambert Academic Publishing (LAP), Germany, Oct. 2010.

CONFERENCE PUBLICATIONS

1. **Feng Li**, Jiang Li, Frederic McKenzie, Guangfan Zhang, Wei Wang, Aaron Pepe, Roger Xu, Tom Schnell, Nick Anderson and Dean Heitkamp, "Engagement assessment using EEG signals", *Modsim*, Virginia Beach, VA., Oct. 2011

2. Guangfan Zhang, Wei Wang, Aaron Pepe, Roger Xu, To Schnell, Nick Anderson, Dean Heitkamp, Jiang Li, **Feng Li** and Frederic McKenzie, "A systematic approach for real-time operator functional state assessment", *ModSim*, Virginia Beach, VA., Oct. 2011.

3. **Feng Li**, Frederick McKenzie, Jiang Li, Guangfan Zhang, Roger Xu, Carl Richey and Tom Schnell, "Imbalanced Learning for Functional State Assessment", *ModSim*, Hampton, VA., Oct. 2010