Decoding Cognitive States from fNIRS Neuroimaging Data Using Machine Learning

by

Ruixue Liu

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Computer Science

Dec 11, 2020

APPROVED:

Professor Erin T. Solovey Worcester Polytechnic Institute Advisor	Professor Rodica Neamtu Worcester Polytechnic Institute Committee Member
Professor Ali Yousefi Worcester Polytechnic Institute Committee Member	Professor Catherine M. Arrington Lehigh University External Committee Member
Professor Craig E. Wills Worcester Polytechnic Institute Head of Department	

Abstract

Building brain-computer interfaces that can automatically adapt to an individual's changing cognitive states has important implications in many domains, such as gaming, driving, and learning. Recently, the use of functional near-infrared spectroscopy (fNIRS) has received focus because of its promise for detecting an individual's cognitive state in more ecologically valid studies.

In this dissertation, we focus on improving and expanding the usability of fNIRS for brain-computer interaction research. Particularly, we investigated the feasibility of using fNIRS to identify several user states that occur frequently in human-computer interaction, and that could inform adaptive user interfaces, but that are difficult to detect. We accomplished this goal by designing and conducting three human subjects experiments, collecting and curating fNIRS datasets, as well as developing and applying novel machine learning methods appropriate for the particular classification problem and that are tuned to the characteristics of fNIRS data. Particularly, we:

1. Explore mind wandering detection using fNIRS and develop a machine learning framework to incorporate individuals' differences in hemodynamic responses. Specifically, we conducted a study using fNIRS during the Sustained Attention to Response Task (SART) task to elicit mind-wandering states. We then built machine learning classifiers both on an individual level and at a group level to classify mind-wandering state versus on-task state. We also propose an individual-based novel window selection algorithm to incorporate individuals' differences in time window selection. Our results show that the proposed algorithm achieves significant improvements over the previous state-of-the-art in terms of brain-based detection of mind-wandering.

2. Explore driver cognitive load classification using fNIRS and investigate machine learning techniques for extracting spatial and temporal patterns from fNIRS data. Specifically, we conducted a study using fNIRS in a driving simulator with the n-back task used as a secondary task to impart structured cognitive load on drivers. We apply Convolutional Neural Networks (CNNs), multivariate Long Short Term Memory Fully Convolutional Networks (LSTM-FCNs), and Echo State Networks (ESNs) for fNIRS feature extraction and classification. Our results show that ESNs achieve state-of-the-art classification results for classifying different levels of driver cognitive load.
3. Explore cognitive processes associated with positive and negative learning outcomes using fNIRS and validate the generalizability of the proposed ESN models across tasks. Specifically, we conducted another study using fNIRS during a rule-learning task. We compare the classification results of CNNs, LSTM-FCNs, and ESNs for differentiating successful and unsuccessful rule learning processes. Our results show that ESNs achieve superior classification results and can extract distinct temporal patterns for different cognitive processes based on fNIRS data.

By improving and expanding the usability of fNIRS for identifying important user states for human-computer interaction, the results from this research serve as a foundation for future work that integrates fNIRS data for measuring an individual's changing cognitive states. Furthermore, findings from this work have important implications for building fNIRS-based brain-computer interfaces that can automatically adapt their behavior to better support the user and provide a better user experience.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank a group of amazing people for their support throughout my work on this dissertation.

Firstly, I would like to express my deepest gratitude to my advisor Prof. Erin Solovey for the continuous support of my PhD study and research, for her guidance, motivation, and patience. It is my great honor to be her first PhD student to complete the dissertation. I learned a lot from her on how to conduct human-computer interaction research, how to write, and present research work. She encouraged and trusted me during challenges, and she was always there for me when I needed her. This dissertation would not have been possible without her support.

Besides my advisor, I would like to thank the rest of my committee: Prof. Rodica Neamtu, Prof. Ali Yousefi, and Prof. Kate Arrington, for their valuable comments and suggestions. Their insightful questions also incented me to strengthen my research from various perspectives.

I would like to extend my sincere thanks to Prof. Erin Walker, Prof. Andrea Forte, and Prof. Gabriela Marcu for their advice and collaborations throughout my PhD study. My sincere thanks also go to Dr. Richard Harang and Dr. Advait Sarkar for the opportunities to join their teams as an intern, for mentoring me, and for inspiring me to explore challenging research. I also thank the rest of Microsoft Research Cambridge: Dr. Sebastian Tschiatschek, Dr. Cecily Morrison, Dr. Anja Thieme and Dr. Ed Cutell, for their advice, suggestions, and feedback.

I would like to thank my friends and labmates: Ke Yang, Haoyue Ping, Denisa Qori, Reza Moradinezhad, Janith Weerasinghe, and many others, for sharing this PhD journey with me, for discussions, suggestions, and collaboration.

I would like to thank my family. I'm deeply indebted to my parents for all their support and encouragement throughout the years. I am also grateful to have my brother, my sister-in-law, and my nieces in my life.

Finally, I am extremely thankful for my partner Sharon Tartarone, for her unwavering support, encouragement, care, and love.

Chapter 1

Introduction

Automatic detection of an individual's cognitive state in real-time has been an emerging field of research over the past decade. Researchers have been particularly interested in utilizing this information to enable adaptive human-computer interaction, with the goal of performance improvement. For example, if an online learning system can gain insight into the learner's cognitive states, it can adapt its behavior dynamically and provide a better learning experience.

Attention and cognitive load are among the most investigated cognitive states in the field of human-computer interaction (HCI) [1, 2]. Research has shown that users' attention and cognitive workload are highly correlated with their task performance [3, 4]. Attention refers to whether the orientation of users' senses is towards the current task. Attention regulates user's behavior by focusing on task-relevant stimuli [3, 5]. Cognitive load can be interpreted as an interaction between task demands and the users' capabilities or resources. Excessive levels of cognitive load can cause errors or delayed information processing [6], while low levels of cognitive could lead to annoyance and frustration in users when they are processing information [7].

However, these cognitive states have been challenging to identify using traditional measures. To infer users' dynamic attention states and cognitive load, prior work has in-

vestigated the use of behavioral data [8, 9, 10], eye activity [11, 12, 13], facial expression [14, 15, 16], and physiological sensing [17, 18, 19]. However, there exist some difficulties in building robust models using these approaches that have a general applicability[20]. These approaches depend on a set of measurable factors that have been shown to be related to the targeted cognitive states in specific tasks. As such, the models built are ad-hoc and can not apply to other tasks and domains. For example, ad-hoc subjective measurements were developed to quantify the mental workload experienced in specific domains, such as medical care [21] and web design [22]. In addition, it is difficult to define which factors can best describe these cognitive states. For instance, researchers have reported that changes in subjective workload do not always align with changes in task performance [4, 23]. Researchers also observed that user's states inferred from physiological data and self-report measurement sometimes conflict with each other [24].

Brain imaging and brain-sensing techniques have the potential to reduce the ambiguity surrounding the understanding of these states and provide an alternative for measuring these states objectively across different domains [4, 25]. Recently, the use of functional near-infrared spectroscopy (fNIRS) has received focus because of its promise for detecting a user's cognitive state in more ecologically valid studies than othere neuroimaging methods. fNIRS is a neuroimaging tool that is safe, portable, easy to use, and quick to set up — characteristics that have led to increasing adoption. It detects hemodynamic changes associated with neural activity in the brain while performing tasks (see Figure 1.1). Recent advances in fNIRS have shown the possibility of decoding cognitive states during various activities [26, 27, 28].

Nevertheless, there are some challenges regarding accurately detecting an individual's states based on the fNIRS signal. With the advancements in machine learning, researchers have attempted to move from offline statistical analysis of the fNIRS data to real-time automated classification of users' state. However, prior work has shown the difficulty of

Figure 1.1: A volunteer wears the fNIRS on the forehead.

achieving satisfactory results for fNIRS data classification based on traditional machine learning approaches [29, 30, 31].

1.1 Problem Statement

While prior work has shown the possibility of decoding cognitive states during various activities, there are some challenges that exist for building robust models based on fNIRS data that can accurately decode users' cognitive states and be used in real-world applications. These challenges are due to the inherent attributes of fNIRS data:

- Collecting and labeling brain data is costly and time-consuming. As such, the sizes of brain datasets are usually small. At the same time, advanced machine learning methods require a large number of training examples, and insufficient training data could lead to poor performance. For real-world applications, the accuracy of the

models needs to be improved [32].

- There are individual differences in fNIRS data. fNIRS signals vary across participants, which can be attributed to individual differences in hemodynamic responses and measurement variations [33]. However, since the process of collecting brain data is costly and time-consuming, to get an adequate amount of data for model training, there is a need to build models across participants [34, 35]. Individual differences in fNIRS data make it challenging to build robust machine learning models that can generalize across participants [36, 37].
- fNIRS data are high-dimensional and high volume time series data. For example, for an fNIRS device with eight channels and with a sample size of 10 Hz, for a one hour experiment, it will generate approximately 36,000 rows and 24 columns of data. Therefore, it requires powerful machine learning models that can accurately recognize the spatial and temporal patterns in fNIRS data [38].

1.2 Research Scope, Questions and Tasks

In this dissertation, we focus our work on improving and expanding the usability of fNIRS for brain-computer interaction research. Particularly, we investigated the feasibility of using fNIRS to identify several user states that occur frequently in human-computer interaction, and that could inform adaptive user interfaces, but that are difficult to detect. We distinguish the mind-wandering state from on-task states, classify different levels of driver cognitive load, and differentiate positive and negative cognitive processes during learning. We develop and apply novel machine learning methods appropriate for the particular classification problem and that are tuned to the characteristics of fNIRS data.

Given the complex characteristics of fNIRS data, it remains unclear whether it is feasible

to detect these user states using fNIRS and what machine learning approaches can be applied to improve the results.

RQ1: Can we use fNIRS to detect mind-wandering state and improve the accuracy by incorporating individual differences in fNIRS data?

Automatic detection of an individual's mind-wandering state has implications in many domains, such as driving and learning [39, 40]. However, it remains a challenge to detect mind-wandering state accurately [41].

RQ2:

Can we use fNIRS to classify different driver cognitive load and improve the accuracy by extracting spatial and temporal patterns in fNIRS data?

Understanding the cognitive load of drivers is crucial for road safety. Brain sensing has the potential to provide an objective measure of driver cognitive load. Previous studies in this direction utilized traditional signal processing methods to analyze fNIRS signals without using state-of-the-art machine learning algorithms [42, 43, 44].

RQ3:

Can we use fNIRS to differentiate cognitive processes associated with positive and negative learning outcomes by applying the proposed ESN model?

The main goal of intelligent tutoring systems is to facilitate robust learning. However, little is known about the underlying cognitive states that are associate with learning outcomes. On the other hand, it is important to develop machine learning models that are generalizable across different tasks [45].

We answered each of the research questions through three step-by-step research tasks:

RQ1. Can we use fNIRS to detect mind-wandering state and improve the accuracy by incorporating individuals' differences in fNIRS data?

T1:

We conducted a study using fNIRS during the Sustained Attention to Response Task (SART) to elicit mind-wandering states. We then built machine learning classifiers both on an individual level and at a group level to detect mind-wandering. We proposed an individual-based novel window selection (ITWS) algorithm to improve classification accuracy. We evaluated the performance of the ITWS algorithm with eXtreme Gradient Boosting (XGBoost), Convolutional Neural Networks, and Deep Neural Networks. Our results show that the proposed algorithm achieves significant improvement compared to the previous state of the art in terms of brain-based classification of mind-wandering, with an average F1-score of 73.2%.

RQ2. Can we use fNIRS to classify different driver cognitive load and improve the accuracy by extracting spatial and temporal patterns in fNIRS data?

We conducted a study using fNIRS in a driving simulator with the n-back task used as a secondary task to impart structured cognitive load on drivers. We investigate the application of Convolutional Neural Networks (CNNs), multivariate Long Short Term Memory Fully Convolutional Networks (LSTM-FCNs), and Echo State Networks (ESNs) for extracting spatial and temporal patterns from fNIRS data. We then compared the classification results. Our results show that the proposed ESN autoencoder achieves state-of-art classification results for group-level models without window selection, with accuracies of 80.61% and 52.45% for classifying two and four levels of driver cognitive load.

RQ3. Can we use fNIRS to differentiate cognitive processes associated with positive and negative learning outcomes by applying the proposed ESN model?

T3: We move beyond cognitive states that have been explored in previous work using fNIRS. We conducted a study with a rule learning task using fNIRS to elicit abstract rule induction processes. We compare the classification results of CNNs, multivariate

LSTM-FCNs, and ESNs for differentiating cognitive processes that lead to positive and negative learning outcomes. Our results show that ESN achieves superior classification results, with an accuracy of 87.95% and an F1-score of 85.64%. Visualization analysis of the ESN model shows that the temporal patterns extracted by ESN are discriminative for induction processes that lead to positive and negative learning outcomes.

1.3 Organization

This dissertation is organized as follows:

Chapter 2 provides a brief overview of the related work surrounding the two research areas of this dissertation work, including using fNIRS in HCI and challenges for fNIRS data modeling. Chapter 3 describes the general procedures for developing adaptive interfaces with fNIRS. Chapter 4 answers RQ1 and describes a study that uses fNIRS to detect mind-wandering states with personalized window selection. Chapter 5 answers RQ2 and describes a study which uses fNIRS for driver cognitive load classification, by applying CNNs, multivariate LSTM-FCNs, and ESNs. Chapter 6 answers RQ3 and describes a study which uses fNIRS for classifying successful and unsuccessful rule learning processes, by applying CNNs, multivariate LSTM-FCNs, and ESNs. Chapter 7 summarizes the main contributions of this work and discusses future opportunities.

Research Question	Task	Chapter
RQ1: *Can we use fNIRS to detect mind-wandering state and improve the accuracy by incorporating individuals' differences in fNIRS data?*	T1	§4 Detecting Mind-wandering State Using fNIRS With Personalized Window Selection
RQ2: *Can we use fNIRS to classify different driver cognitive load and improve the accuracy by extracting spatial and temporal patterns in fNIRS data?*	T2	§5 Classifying Driver Cognitive Load Using fNIRS with CNNs, Multivariate LSTM-FCNs and ESNs
RQ3: *Can we use fNIRS to differentiate cognitive processes associated with positive and negative learning outcomes by applying the proposed ESN model?*	T3	§6 Classifying Successful and Unsuccessful Rule Learning Processes Using fNIRS with CNNs, Multivariate LSTM-FCNs, and ESNs

Table 1.1: The organization of this dissertation.

Chapter 2

Background

Here we provide a brief overview of the related work surrounding the two research areas of this dissertation work. We will provide in-line discussions of the most relevant work in each research task.

2.1 Using fNIRS in HCI

Functional near-infrared spectroscopy (fNIRS) is a brain-imaging tool that is safe, portable, easy to use, and quick to set up—characteristics that have led to increasing adoption. It detects hemodynamic changes associated with neural activity in the brain while performing tasks [46]. Because fNIRS enables brain activity to be measured continuously during interactive tasks, it has promise for understanding user experience in realistic settings. fNIRS sensors use light to detect hemodynamic changes. The light sources send two wavelengths of near-infrared light into the forehead, where it continues through the skin and bone 1-3 cm deep into the cortex. Biological tissues are relatively transparent to these wavelengths, and oxygenated and deoxygenated hemoglobin are the main absorbers of this light. After the light scatters in the brain, some reaches the light detector. By determining the amount of light picked up by the detector, the amount of oxygenated and

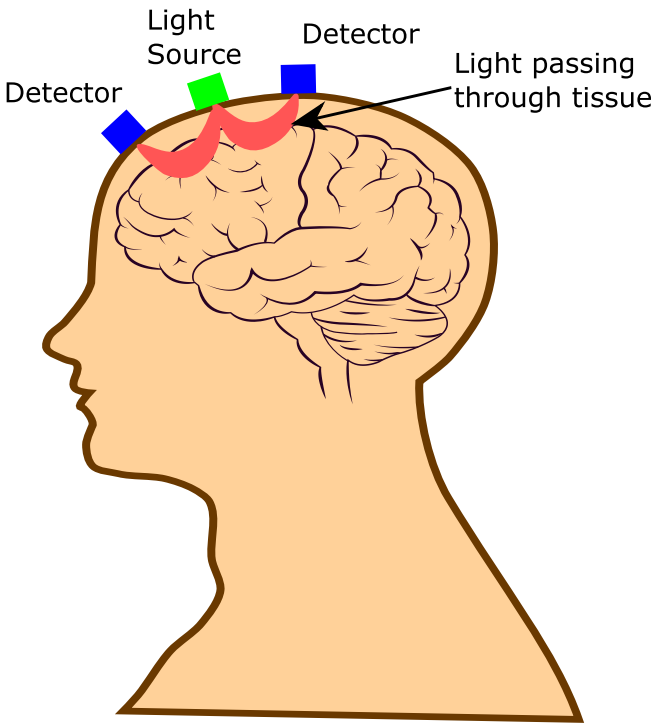

Figure 2.1: Illustration of path of near-infrared light between the source and detector

deoxygenated hemoglobin can be calculated in the area, which indicates hemodynamic activity associated with brain activation (see Figure 2.1). Thus, fNIRS measurements can be used to understand changes in a person's cognitive state while performing tasks [47, 48]. While fNIRS has been applied to various locations on the head, researchers have shown the most successful placement is on the forehead (Figure 1.1) [26]. As such, the anterior prefrontal cortex (PFC), which lies behind the forehead, has been the main target for fNIRS brain sensing in HCI. The PFC is responsible for many high-level processes and has been found to play a part in memory and executive control [49].

There are other techniques that can measure the changing state of the brain (e.g., fMRI, EEG, positron emission tomography (PET), and magnetoencephalography (MEG)). These tools are often prohibitively expensive and require restrictions on the study partic-

ipant that are not reasonable for use in realistic settings. Also, PET requires ingestion of hazardous material, and fMRI exposes individuals to loud noises that may interfere with the study [50]. The strong magnetic field prevents typical computer usage in both fMRI and MEG. EEG is less intrusive, more portable, and less expensive than these other tools, and has been widely used in brain-computer interface research. However, it can have a significant setup time and has a limited spatial resolution. Electronic devices in the room can also interfere with the signal, and it is susceptible to artifacts in the data due to user movement.

fNIRS avoids many of the restrictions of other techniques and therefore has promise for use in real-world settings such as classrooms or driving. It has been shown to be robust in typical human-computer interaction scenarios, including during typing and mouse clicking [47, 51], and verbalization [52]. Real-time fNIRS brain data has been used to make appropriate adaptations to user interface elements [53] as well as to modulate interruptions [54, 55] and enable attention-aware systems [56]. fNIRS hyperscanning has also shown promising to monitor multiple participants' brain activation simultaneously during their natural interactions [57, 58]. Significant improvements have been made recently in terms of fNIRS hardware to make it wearable and wireless, and we foresee it being increasingly integrated with wearable computing platforms currently being developed [59, 60, 61, 62].

2.2 Challenges for fNIRS data modeling

An increase in oxygenated hemoglobin (HbO) and a decrease in deoxygenated hemoglobin (HbR) have been shown to be related to activation in the associated brain area [63]. Therefore, researchers have focused on using fNIRS data to obtain the brain's activation patterns while performing tasks. Furthermore, to enable brain signals as input for application in-

terfaces, researchers have used classification methods to identify different brain patterns from fNIRS data. However, there exist a few challenges for machine learning modeling of fNIRS data.

2.2.1 Extracting Spatial-Temporal Patterns

Traditional machine learning methods, such as Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Hidden Markov Models (HMM), and Artificial Neural Networks (ANN), have been widely applied to detect patterns from fNIRS data [64]. These classifiers gained popularity in the fNIRS research community because of their simplicity and low computational requirements. However, these machine learning classifiers cannot adequately consider the spatial-temporal dynamics of fNIRS data. The results of these classifiers depend heavily on the features extracted from fNIRS data. However, due to the time-series nature of fNIRS data, it can be difficult for researchers to identify and extract informative features from fNIRS data mnaually. Also, these classifiers often treat features from different channels separately, without considering the correlation between different channels, which contains important spatial information for brain activation [65].

More recent work has investigated using deep learning methods for fNIRS data classification, including Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), and Long Short-Term Memory (LSTM) network [66, 67, 68]. These approaches have shown their ability to detect spatial patterns and long-term dependencies from time-series data by automatically extracting higher-level features. However, most research did not adequately consider the spatial-temporal dynamics of fNIRS data when applying these models. In most work, fNIRS data of multiple channels were transformed to one-dimension data through dimensionality reduction [66, 68]. This makes it difficult for the models to capture the spatial-temporal patterns in data.

2.2.2 Sample Window Selection

In most previous work using hand-crafted features as well as CNN-based methods for fNIRS-based cognitive load classification, window selection methods were utilized to carefully pick a small segment of a fixed size from the original data as the input. While this method might yield better classification results, it ignores the global temporal information and could result in overly optimistic classification results for real-world applications. Moreover, research has shown that due to the latency of the underlying physiological processes, fNIRS cognitive load classification may require a minimum window length of 10 seconds [69, 70]. Some previous work has not met this requirement which could lead to unreliable results. For example, Saadati *et al.* used fNIRS data from a 3-seconds window to build CNN models [71]. Even though they achieved an accuracy of 89% for classifying cognitive load tasks, continuous time-windows from a single trial were used to form multiple samples in their work. This violates a key assumption behind machine learning techniques that samples are independent and make their results unreliable.

2.2.3 Individual vs. Group Models

Most previous work builds individual models for fNIRS-based classifications. However, research has shown that due to the small dataset of an individual participant and the high feature space of brain data, building individual models could lead to overfitting and overly-optimistic results [72]. Therefore, researchers have shown the need for building group models (across participants) for fNIRS data classification [34, 73], which can enable researchers to get a larger dataset for model training and achieve more reliable results, as well as reduce the time for collecting brain data from a particular individual. However, due to inter-subject variability in hemodynamic responses, it is difficult to build robust models across participants based on fNIRS data [74, 75, 76, 77].

2.2.4 Satisfying Diverse Classification Requirements

Each classification task based on fNIRS data is unique and has diverse requirements. Therefore, it is difficult to establish a standard machine learning method for fNIRS data classification. First, the sizes of datasets are different, depending on the number of trials in the experiments and the number of participants, which can affect the choice of machine learning models. For example, deep learning methods require a large number of training examples. Therefore, for data sets with a smaller sample size, deep learning methods might not be able to achieve satisfactory results. Second, the associated cognitive state of the tasks can also affect the classification results on fNIRS data. Different cognitive states are mapped with different brain activation patterns and could require different classification techniques [78]. Therefore, one machine learning method might be able to achieve good classification results for distinguishing between left and right finger tapping, but not be able to distinguish between different levels of cognitive load [69, 79]. Third, different classification problems might have different requirements for training time and computer resources. For example, for real-world applications that use fNIRS data as training data, the training time of the machine learning models needs to be fast. In this case, deep learning models might not be suitable due to the significant time and computational resources required.

In this work, we explore novel solutions to the machine learning challenges of fNIRS data, with the goal of improving the classification performance for decoding cognitive states. Specifically, we investigate novel machine learning methods that are tuned to the characteristics of fNIRS data and are appropriate for the classification task. We explore possible development to traditional machine learning classifiers as well as deep learning techniques while classifying fNIRS data. We demonstrate these approaches in three research tasks with different targeted cognitive states.

Chapter 3

Toward Adaptive Brain-Computer Interfaces Using fNIRS

Most work on developing adaptive interfaces using fNIRS has taken the cognitive states as implicit inputs [80]. In contrast to explicit inputs, implicit inputs are user actions or situational contexts that a computer can understand that are not explicitly given commands by the user. These implicit inputs can then lead to systems that adapt appropriately to changes in the user's states [81]. In this dissertation, we propose to use fNIRS data for attention-aware interfaces that can automatically detect mind-wandering states without interrupting the task with experience sampling probes; for building driver support systems that can automatically measure drivers' cognitive load; as well as for developing adaptive learning systems that can detect learners' cognitive state. For example, when a system automatically detects an individual is mind-wandering during online learning, it could change the presentation to help the user focus on important tasks and materials.

To build multimodal interfaces with fNIRS, Solovey *et al.* pointed out that there are some common high-level phases [82], with calibration phase, modeling phase, and real-time classification phase being the main phases for real-time applications [82]. During the calibration phase, users are asked to perform a set of cognitive benchmark tasks. The

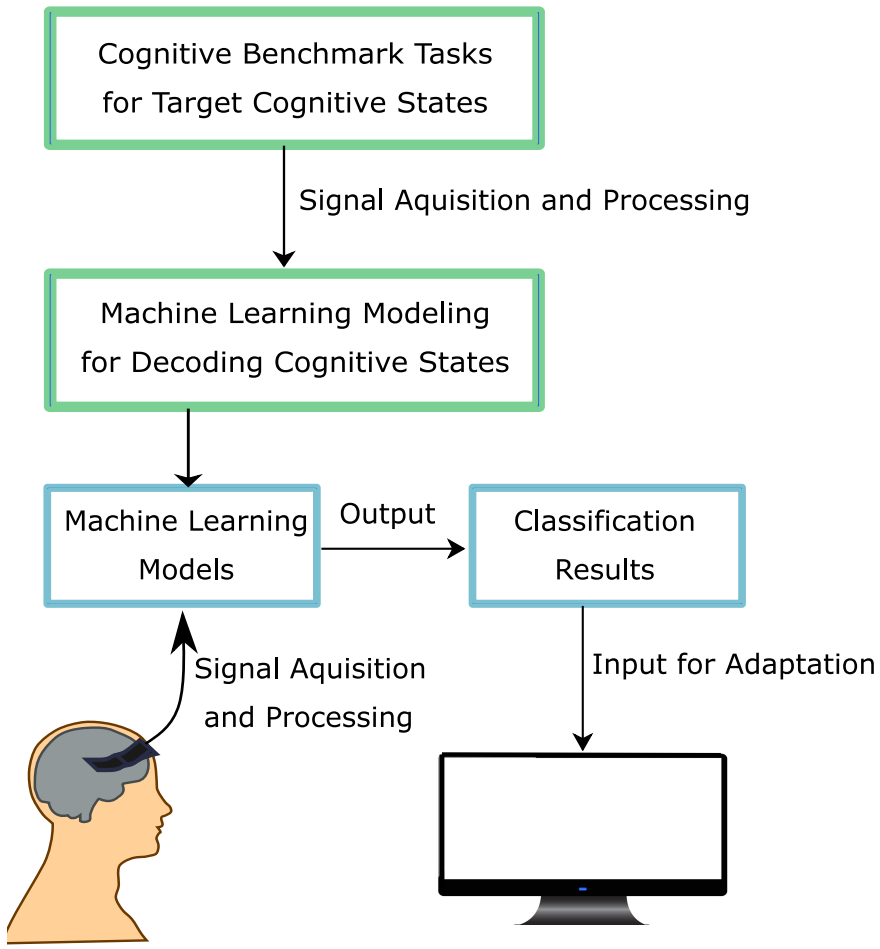

Figure 3.1: The workflow of developing adaptive Brain-Computer interfaces using fNIRS.

cognitive benchmark tasks are experiment tasks from cognitive psychology that can elicit different targeted cognitive states [49]. fNIRS data recorded during the cognitive benchmark tasks is then used to train machine learning classifiers. In the real-time classification phase, the machine learning model continuously classifies the new data coming in. The classification results can then be sent to the system for necessary adaptations. We describe these phases for developing adaptive interfaces with fNIRS in Figure 3.1.

To move toward this goal, classification accuracy for decoding cognitive states from fNIRS data needs to be higher than shown in previous work [31, 70, 83]. To do this, appropriate datasets need to be created for validating algorithms for detecting particular

cognitive states. Further, we need to explore advanced machine learning frameworks that are suitable for fNIRS data.

Chapter 4

Detecting Mind-Wandering State Using fNIRS with Personalized Window Selection

In this chapter, we investigate the feasibility of using fNIRS to detect mind-wandering states[1]. Particularly, we explore automated window selection for fNIRS data when classifying mind-wandering episodes versus on-task episodes. We also propose an individual-based time window selection (ITWS) algorithm to incorporate individual differences in window selection.

4.1 Introduction

Mind-wandering occurs when an individual is engaging in internal non-task thoughts, instead of processing external task-related information [84]. Even though people may be generally unaware of when it occurs, mind-wandering could occupy 46.9% of daily life [85]. While some studies suggest that mind-wandering may contribute to future planning and creative problem solving, mind-wandering has shown to be disruptive and detrimen-

[1]The work in this chapter was originally described in Liu, et al. "fNIRS-based Classification of Mind-wandering with Personalized Window Selection for Multimodal Learning Interfaces". *Journal on Multimodal User Interfaces* [75].

tal to individuals' performance when it happens during cognitively demanding tasks [86]. Therefore, the detection of mind-wandering states is important for many domains, and particularly for learning and training. For example, when a student is engaging in cognitively demanding tasks such as learning, mind-wandering would negatively affect task performance and lead to errors [87].

While technology-enhanced learning such as intelligent tutoring interfaces and Virtual Reality (VR) environments show promise for enhancing learning and training experiences, research shows not all interface features or virtual environments elements increase the effectiveness [88, 89]. As such, identification of a user's mind-wandering episodes and on-task episodes in a learning interface could inform evaluations. Further, detecting mind-wandering states is an important step towards attention-aware systems, which can dynamically update interfaces and content to facilitate users' focus on task-related information. For example, when the system detects an individual is mind-wandering during training sessions, it could change the presentation to help the user focus on important materials [90].

To measure mind-wandering, many researchers use an experience sampling methodology. With this method, researchers ask individuals to self-report when mind-wandering occurs during a task or place thought probes during the task, which periodically ask individuals whether they are mind-wandering. However, these methods have a limitation due to their dependence on participants to be aware of their mind-wandering episodes and respond accurately. Also, the thought probes interrupt both the task and the mind-wandering episodes [87].

One possible solution to address these limitations is to examine an individual's brain activity directly and use the brain data to disentangle focused states from mind-wandering states. Functional magnetic resonance imaging (fMRI) studies show that mind-wandering is associated with activation in the default network [84]. Several default mode network

areas have shown consistent activation during mind-wandering, including the medial prefrontal cortex, medial temporal lobe, posterior cingulate cortex, and bilateral inferior parietal lobule [91]. Moreover, as non-invasive neuroimaging techniques become less expensive and more portable, we can monitor brain activity during various activities.

Recently, the use of functional near-infrared spectroscopy (fNIRS) has received focus because of its promise for detecting an individual's user's cognitive state in more ecologically valid studies. While fMRI has become the gold standard for brain imaging, in real-world environments, fNIRS is a more convenient and more affordable technology than fMRI [92]. fNIRS emits near-infrared light into the brain, and the light returned to the surface is measured and used to calculate oxygenation in the blood. This calculation reflects brain activity in that particular area. Prior work has shown the potential of using fNIRS data to identify brain activation related to mind wandering episodes [31].

In this paper, we aim to build on previous findings and present a data-driven classification framework to improve mind-wandering classification accuracy. Since prior fNIRS studies have shown that the classifier performance can be improved by focusing on a specific window [93, 94, 95], we utilize a moving window method for the classification of mind-wandering, which can select the best window for classification during a time period. In addition to building models for each individual, we also demonstrate the feasibility of building machine learning models across individuals to differentiate mind-wandering episodes versus on-task episodes.

For individual-level classification, we use the moving window method combined with a shrinkage LDA classifier to find the best window for detecting mind-wandering. For group-level classification, to incorporate individual differences in window selection and hence improve the classification results, we propose a novel individual-based time window selection (ITWS) algorithm. The ITWS algorithm iteratively chooses the best window for each individual through embedded individual-level classifiers, and then uses data

from these windows as training data and test data for the group-level classifier. We validate the framework using an fNIRS dataset we collected with mind-wandering episodes and on-task episodes during the Sustained Attention to Response Task (SART). The errors during the SART have been shown to be correlated with mind-wandering [87], and thus form a ground truth for our classification results.

The main contributions of this work are as follows:

- We propose to use fNIRS brain data for evaluating learning interfaces and for attention-aware systems that can automatically detect mind-wandering state without interrupting the task with experience sampling probes.
- We describe a study in which we collected fNIRS brain data during the SART task. This dataset provides examples of mind-wandering and on-task episodes, defined based on behavioral data, that can be used to investigate robust classification algorithms. We confirm that there are differences in frontal lobe blood oxygenation patterns between mind-wandering episodes and on-task episodes.
- To improve classification accuracy, we investigate window selection when classifying mind-wandering states versus on-task state using fNIRS. We show individual-level classifiers can achieve better classification results when focusing on specific windows rather than those using the entire episodes.
- To further improve model robustness and performance, we extend the window selection method for group-level classification. We propose a novel individual-based time window selection (ITWS) algorithm to incorporate individual differences in window selection when building group-level classifiers. We show that the ITWS algorithm can improve the group-level classification result by comparing with other methods that do not use the ITWS algorithm.

4.2 Background

4.2.1 Detection of Mind-wandering in Multimodal Learning Interfaces

Technology-enhanced learning is increasingly adopted for providing novel solutions to educational and training activities, such as intelligent tutoring interfaces, serious games, and VR environments. Previous studies have shown the positive effects of such applications in improving students' cognitive states during learning, including motivation and attention [96, 97]. However, in addition to motivation and attention, mind-wandering has also shown to play an important role in students' learning performance. Mind-wandering can be detrimental to student learning, where instead of processing external task-related information, students engage in internal non-task thoughts [98]. Therefore, detection of mind-wandering would be valuable for understanding users' attention control mechanisms during these interfaces. Nevertheless, since mind-wandering involves internal thoughts instead of expressive behaviors and the dynamics of mind-wandering remain elusive, detecting mind-wandering is a challenging task [99]. Prior research has investigated using physiological and behavioral metrics, as well as brain data for mind-wandering detection.

Physiological and behavioral metrics of mind-wandering

Probe-caught mind-wandering has been predicted using eye gaze [100, 101], physiological sensing [102, 103], behavioral indices [104, 105], and facial expression [106]. Hutt *et al.* used eye gaze and contextual cues as features to predict mind-wandering state when participants were interacting with an intelligent tutoring system. Participants were randomly probed to report mind-wandering instances. They achieved a prediction accuracy of about 25% above chance [101]. Physiological features, including heart rate

[103] and skin conductance [102], have also been used for mind-wandering detection. Blanchard *et al.* measured participants' skin conductance and skin temperature to detect mind-wandering during a reading task. They achieved 22% above chance accuracy [102].

Some researchers also used behavioral indices, including reading behaviors and textual features, to detect mind-wandering during reading tasks [104, 105]. The resulting accuracy is 20% above the chance accuracy for a somewhat naturalistic reading paradigm [105]. However, this method is limited to reading tasks.

Another approach is using facial expressions and movements to detect mind-wandering states. Bosch and D'Mello applied this approach in a laboratory study where participants read a text and in a classroom study where high school students learned biology from an intelligent tutoring system. After extracting facial and movement features from the recorded video and applying machine learning classifiers, they achieved 25.4% and 20.9% above-chance accuracy for detecting mind-wandering in the lab and classroom, respectively [106].

For all of these investigations, the models built are ad-hoc and depend on a set of measurable factors that have been shown to be related to mind-wandering in the specific task.

Brain-based metrics of mind-wandering

Brain sensing techniques provide an alternative to detect mind-wandering objectively across different domains. Some researchers explored using EEG brain signals to differentiate mind-wandering versus on-task. Kawashima *et al.* used EEG variables to estimate mind-wandering intensity through support vector machine regression during a sustained attention task [107]. However, the mind-wandering intensity was determined by thought probes, which were placed at a fixed interval. This could lead to individuals anticipating the probe occurrence and becoming more conscious of mind-wandering. Jin *et al.* trained

machine-learning models on EEG markers to determine participants' state as either mind wandering or on-task, and they achieved a mean accuracy of 64% for a sustained attention task [99].

Considerations

In all these studies, probes were used to catch mind wandering by asking participants whether they are mind-wandering. Researchers then focused on an interval of time that precedes the probes (10s or 30s). These probes allow researchers to mark the time point when mind wandering is actually happening. However, it interrupts the mind wandering episodes, and can only collect the mind wandering episodes that participants are aware of. Therefore, exploring the detection of mind-wandering episodes without interruption would be an important step toward fully automated attention-aware systems and environments. In this work, we explore fNIRS brain measures of mind wandering and use the SART task to elicit mind-wandering episodes, since the errors during the SART task have been shown to be correlated with mind-wandering [87].

4.2.2 Mind-wandering Classification with fNIRS

Accuracy of mind-wandering detection with fNIRS

As mentioned earlier, activation of the medial prefrontal cortex during mind-wandering has been detected using fNIRS during a sustained attention task [31]. This study showed promise for detecting default network activations related to mind-wandering from fNIRS data. However, this work also highlighted the difficulty of real-time detection of mind-wandering using only fNIRS data. Their machine learning model achieved a mean accuracy of 56% for classifying mind-wandering episodes versus on-task episodes using Linear Discriminant Analysis for each individual separately. For real-world use, this ac-

curacy would need improvement. Therefore, there is a need to explore methods that can achieve higher accuracy. Two approaches that may hold promise are 1) exploring automatic detection of optimal time windows, and 2) exploring both individual and group models.

Optimal time windows for classification

While many studies build and evaluate machine learning classifiers using fNIRS data associated with the entire episodes (e.g., entire mind-wandering episode or on-task episode [31]), other fNIRS studies have shown that we can improve the classifier performance by focusing on a specific window, instead of using the fNIRS data from the overall task period [93, 94, 95]. Naseer *et al.* used fNIRS data to classify right- and left-wrist motor imagery task and they analyzed six different temporal windows within an overall 10s task. They showed that the 2-7s period after the stimulus was the most critical period and they could enhance the average classification accuracy by around 4% by focusing on this period [93]. Khan *et al.* used linear discriminant analysis to find the best window size for detecting drowsiness using fNIRS [94]. They analyzed three different time windows ((0-3 sec, 0-4 sec, and 0-5 sec), and proposed drowsiness detection in 0-4 second window when using fNIRS. These approaches compare a few pre-defined windows and select the one with the best outcomes.

Researchers have also used the moving window method to explore all windows with a specific size and find the best window for classification using fNIRS data [95, 108]. For example, Shin *et al.* conducted two fNIRS experiments (left versus right-hand motor imagery; mental arithmetic versus resting state), and used a 3s moving window with 1s step size to find the maximum classification accuracy over time. The classification accuracy achieved by the best window is significantly higher than those for the other windows [95]. Hennrich *et al.* adopted an n-back task with fNIRS to induce different

levels of workload and extracted 10s windows for workload classification. Their results show that classification accuracy differs between different windows, and peak around 10s after the trial start [69].

From all these studies, the results show that the optimal windows vary between different participants and different tasks. Moreover, both the window sizes and the offset from start time can affect the accuracy of the classification results. Therefore, in this work, we use the moving window method along with different window sizes to find the best window for mind-wandering classification.

Individual vs. group models

In prior work, machine learning models were built for each participant separately (individual-level models) [31]. Considering the small dataset of each participant and the high feature space of the brain data, building models per-participant could lead to overfitting and achieving overly-optimistic results [72]. As such, researchers have shown the need for building models across participants [34].

Building models across participants (i.e., group-level models) can enable researchers to get an adequate amount of data for model training while reducing the time for collecting brain data. Compared to individual-level models, the group-level models are more robust and can achieve more reliable results. However, due to the individual differences in hemodynamic responses, it is a challenge to build robust models across participants based on fNIRS data.

To solve this issue, prior work has investigated optimal feature combinations for each participant [109, 110]. For example, Noori *et al.* used the hybrid genetic algorithm to choose the optimal feature for each participant [110]. Hossein *et al.* applied a personalized feature normalization approach to standardize the extracted feature values of each participant to improve the performance of group-level models. However, even though

prior work shows that the optimal windows vary between different participants, little attention has been paid to the effect of individual differences in window selection on the performance of group-level models. In this paper, we investigate possible methods for incorporating individual differences in window selection for group-level modeling.

4.3 Data Collection

We set out to build a dataset of fNIRS data associated with mind-wandering episodes without using experience sampling probes and to investigate methods of distinguishing mind-wandering states from on-task states with high accuracy. To do this, we conducted a human-subject study that was approved by our institutional review board and informed consent was obtained for all participants.

4.3.1 Sustained Attention to Response Task

To elicit mind wandering, we used a well-studied paradigm called the Sustained Attention to Response Task (SART) [111]. The SART shows a number (0-9) at the center of a blank white screen for 0.5 seconds, followed by a blank screen for 1.0 seconds. Participants were instructed to respond by pressing a key for each stimulus that appears except for the target stimulus, the number 3. When a '3' is shown, the participant is instructed not to press any key and to wait for the next number. For typical SART tasks, the target stimuli occur around 5% to 11% of all stimuli [111, 112]. Since prior work has shown that a low proportion of target stimuli allows increased mind-wandering during the task [111], we adopted a frequency of 5% for the target stimuli to elicit mind-wandering states from the participants. Also, following previous work [111], targets are presented pseudorandomly throughout all trials and are arranged to ensure that they did not appear immediately next to each other.

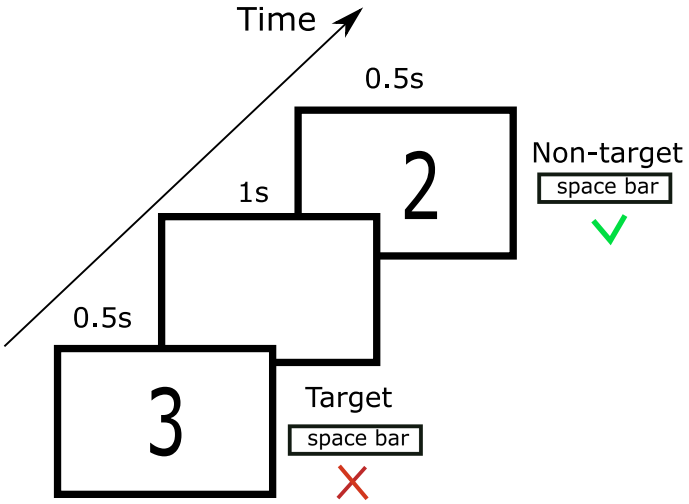

Figure 4.1: Time course of the SART protocol. The number was shown on a white screen for 0.5 seconds, followed by a blank screen for 1.0 seconds. Participants were asked not to press the space bar for the target number '3' and press the space bar for any other numbers.

Figure 4.1 shows the time course of the SART protocol. An incorrect keypress for the target stimulus has been associated with mind-wandering, while a correct response indicates on-task behavior [113].

4.3.2 Procedure

Participants were given an overview and instructions for the task and informed about the brain sensing equipment that would be worn during the study. After providing informed consent, each participant was given instructions about the SART task and the opportunity to ask questions. Participants were equipped with the fNIRS sensors on their foreheads. Then participants performed the SART task on a computer. The experiment consists of 6 sections, with 10 targets and 190 non-targets. In between sections, there was a ten-second break.

At the end of the experiment, individuals were given a questionnaire where they were asked how focused they were throughout the task (scale of 1-7), and if they experienced

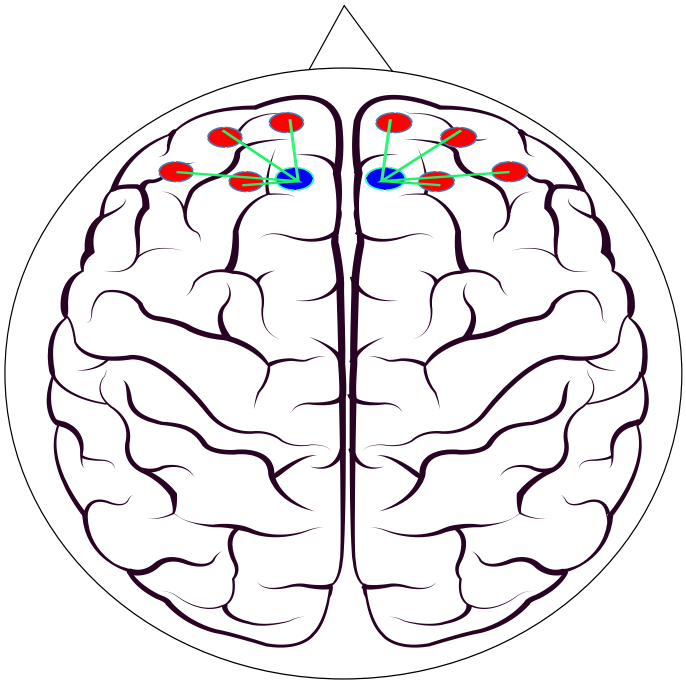

Figure 4.2: Placement of fNIRS sources (red circles) and detectors (blue circles). The green solid line indicates fNIRS channels.

unrelated thought or drowsiness (from ‘never’, ‘rarely’, ‘occasionally’, ‘sometimes’, ‘frequently’, to ‘very frequently’, later converted to a 6-point scale).

4.3.3 fNIRS Recording

The fNIRS data was acquired using a multichannel frequency domain Imagent from ISS Inc. (Champaign, IL). Two probes were placed on the forehead to measure the two hemispheres of the anterior prefrontal cortex (Fig. 4.2). The source-detector distances were 0.8 cm or 3 cm. Each light source emits two light wavelengths (690 nm and 830 nm) to detect and differentiate between oxygenated and deoxygenated hemoglobin. The sampling rate was 6.649 Hz. The sensors were kept in place using headbands, which can also reduce light interference.

4.3.4 Participants

The study included 11 healthy volunteers (5 males) between the ages of 18 and 41 (average 26.27).

4.4 Dataset Curation

Based on the fNIRS data collected during the experiment, we built the dataset for investigating the classification of *on-task* and *mind-wandering* states. We also analyze participants' performance on the task and compare the results with prior work.

4.4.1 General Dataset Description

The dataset consists of fNIRS data of 6 channels, from 11 participants. Since the two short-separation channels (0.8cm) contain mostly noise, we only analyze fNIRS signals from the six long-separation channels (Fig. 4.2).

4.4.2 Dataset Preprocessing

The fNIRS signals from the device may contain noise from various sources, including instrumental noise, motion artifact, and physiological noise [114]. Following typical preprocessing techniques [115], we used a band-pass filter with a high pass value of 0.02 and a low pass value of 0.5 to remove the physiological noise (e.g., heart rate, respiration) and the instrumental noise. The motion artifacts were removed using a wavelet-based denoising and correction procedure [114]. Raw light intensity data was then converted to oxygenated and deoxygenated hemoglobin values using the Modified Beer-Lambert Law. All preprocessing was completed in MATLAB using HomER [116].

4.4.3 Dataset Labeling

To prepare the datasets for analysis and classification, following the work of Durantin *et al.* [31], for each target episode, we extracted fNIRS data from 30 seconds before the target and 10 seconds after the target. Target episodes with a correct response were labeled as *on-task* episodes, while target episodes with an incorrect response were labeled as *mind-wandering* episodes. All non-target episodes were ignored for this analysis since they were not indicative of our target classes. Figure 4.3 shows the number of mind-wandering episodes and the number of on-task episodes from each participant's dataset. Due to the nature of the task, the number of on-task and mind-wandering episodes varied across participants. For each participant, the number of mind-wandering episodes ranged from 8 to 33 out of 60 total targets and the number of on-task episodes varied from 27 to 52 out of 60 total targets (Fig. 4.3). Across all participants, the dataset contains 239 mind-wandering episodes and 421 on-task episodes in total.

4.4.4 Behavioral Data

For the 60 target episodes of the experiment, the mean accuracy across all participants was 0.63 (SE: 0.044) (Fig. 4.3). For the non-target episodes, the mean accuracy across all participants was 0.98 (SE: 0.002). These are not used for our classifier. Participants made significantly more errors on the target episodes than on the non-target episodes (The Wilcoxon signed-rank test, $p < 0.05$), which is consistent with prior work [87]. For the post-survey, the mean level of focus participant reported was 4.45 (SE: 0.35, a scale of 1-7), and the mean frequency of unrelated thoughts and drowsiness was 4.18 (SE: 49) and 4.18 (SE: 0.45), respectively (converted to a scale of 1-6 from 'never' being 1 to 'very frequently' being 6). This shows that participants experienced mind-wandering states during the study.

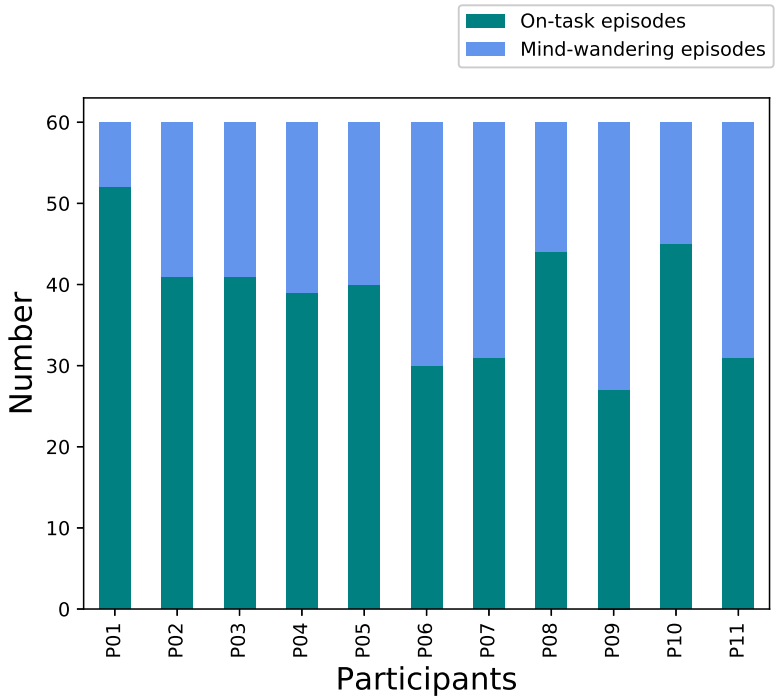

Figure 4.3: The number of mind-wandering episodes and the number of on-task episodes from each participant. Each episode consists of 30 seconds before the target and 10 seconds after the target.

4.4.5 Dataset Overview

For the overview of the dataset, we calculated the folded average of oxygenated hemoglobin (HbO) and the deoxygenated hemoglobin (HbR) change across all participants for the on-task (correct) and mind-wandering (incorrect) target responses. Specifically, we calculated the folded average of all long-separation channels on the left side of the head and all long separation channels on the right side of the head separately. From Fig. 4.4, we can see the average change in HbO from the right side of the cortex showed a significant increase during 30s to 15s prior to an incorrect response to the target, followed by a decrease before the target. From the left side of the cortex, there was a slight increase in HbO change around 25s to 15s before a target error and then return to normal. The average change in HbR on the left side of the cortex showed a slight decrease around 10

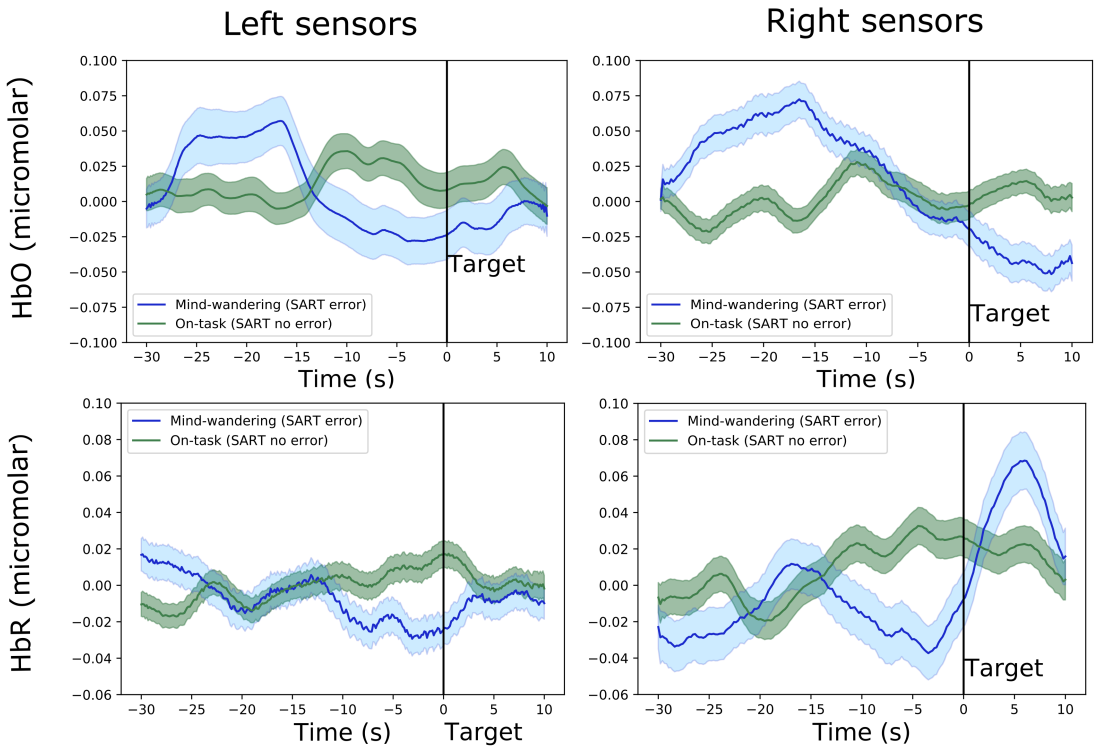

Figure 4.4: Variation of the oxygenated hemoglobin (HbO) and deoxygenated hemoglobin (HbR) concentration for the mind-wandering episodes (SART error, in blue) and on-task episodes (SART no error, in green). The figures show the mean (averaged across individuals) and standard error over the 40 seconds. The figures on the left are data from the sensor on the left side of the head, and the figures on the right are data from the right side of the head. Shaded areas represent the standard error of the mean for each condition.

seconds before an incorrect response to the target. The average change in HbR on the right side of the cortex showed a decrease around 15 seconds before an incorrect response to the target and followed by an increase immediately before the target.

Consistent with prior findings [31, 91], our results suggest there are differences in frontal lobe blood oxygenation patterns between mind-wandering episodes and on-task episodes. Also, our results indicate activation in the prefrontal area preceding mind-wandering occurrence, as the level of HbO increases on both sides of the prefrontal cortex before SART errors. This is consistent with the findings of previous investigations, which

suggest that the prefrontal area contributes to the switching from an on-task state to mind-wandering [31, 91]. In contrast with the previous findings of Durantin *et al.* [31], where they found no significant variations on the HbR relative to incorrect responses to the target, our results showed a decrease on both sides of the cortex before incorrect responses to the target. Since both a decrease in HbR and an increase of HbO indicate cerebral activation, our results are consistent and suggest activation at the prefrontal area at the beginning of mind-wandering episodes.

Moreover, from Fig. 4.4, we can see that the time series behaviors of the hemodynamic patterns are different in different windows during the mind-wandering episodes. In the next section, we investigate window selection for detecting mind-wandering and develop a data-driven classification framework.

4.5 Data-driven Classification Framework

Using the fNIRS dataset that we built and validated above with mind-wandering episodes and on-task episodes, we develop a data-driven classification framework for detecting mind-wandering.

In this section, we investigate window selection when classifying mind-wandering episodes versus on-task episodes using fNIRS data, with the goal of improving the classification accuracy. In addition to individual-level classification, we also explore the feasibility of building machine learning models across participants for detecting mind-wandering. We evaluate the window selection method by comparing the results with the same classifiers, but without window selection.

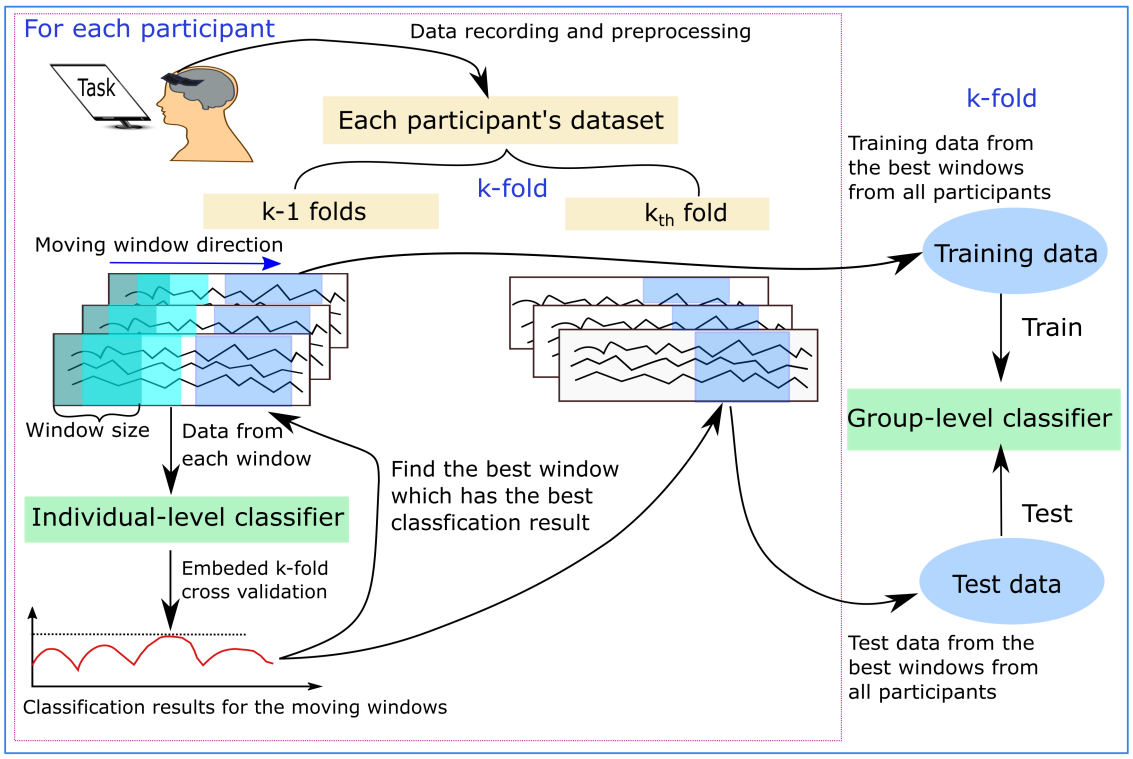

Figure 4.5: Structure of the ITWS algorithm. The dataset (episodes of 40s) of each participant is divided into k folds. In each fold, k-1 folds of the dataset are used to find the best window for classification, and the data from this window of these k-1 folds are later used as training data for the group-level classifier. The data from the same window of the remaining fold are used as the test data to evaluate the classifier. For each participant, a moving window method combined with an individual-level classifier was used to obtain the best window, which has the best cross-validation results.

4.5.1 Individual-level Classification

We start with building models for each participant to classify mind-wandering episodes versus on-task episodes. The goal is to determine if the window selection method can improve the individual-level classification accuracy.

Moving window method

We use the moving window method to find the best sub-window. The moving window method iterates through all the windows with a specific size during a period, and then

all data processing is performed separately on each time window, i.e., feature extraction and classification. Therefore, the moving window method requires a predefined window size and a step size. To investigate the effect of the window size on classification results, we use three different window sizes that are commonly used in previous fNIRS studies [69, 93, 117], which are 5s, 10s, and 15s. For each of the window sizes, we use a 1s step size [117]. The best sub-window is defined as the window with the best classification result.

Individual-level classifier

Because of its simplicity and low computational requirements, linear discriminant analysis with shrinkage (shrinkage LDA) is commonly used as the classifier in fNIRS studies [64]. Particularly, shrinkage LDA has shown advantages when dealing with datasets with small sample size and a large number of features [118]. LDA uses discriminant hyperplane(s) to separate data from different classes [119]. It assumes the class covariance are identical and then models the class conditional distribution of the data for each class. However, with a small sample size, the number of features of each sample could exceed the number of samples in each class. In this case, the empirical sample covariance is a poor estimator [120]. Using a shrinkage estimator of the covariance matrix can help solve this issue [121]. In this study, considering the small sample size for each participant, we use the shrinkage LDA as the individual-level classifier.

From Fig. 4.3, we can see that for most participants (ten out of eleven participants), the dataset is not balanced between the two classes. Most participants had fewer mind-wandering episodes compared to on-task episodes. To avoid the bias of training the classifier towards one class, we use the synthetic minority oversampling technique (SMOTE) to balance the training data for each participant. SMOTE is an oversampling method that has shown effectiveness in many imbalanced datasets. It can generate new synthetic ex-

amples by finding the nearest neighbors of the examples from the minority class [122]. This oversampling method is used only during the training process.

All long-separation channels are used to build the classifier for each participant. The average values of HbR and HbO and the slope over the moving time window are used as features [110]. Each feature is normalized.

Then, in combination with the moving window method, we apply shrinkage LDA to build individual-level classifiers and find the best window for classification.

4.5.2 Group-level Classification

Individual-level classifiers often rely on a relatively small dataset with high feature space, which could lead to model overfitting. Group-level models can solve this issue by training on data collected from all participants. However, it is difficult to achieve high accuracy on group-level models due to individual differences. We propose an individual-based time window selection (ITWS) algorithm to improve the group-level classification results.

ITWS algorithm

When using the window selection method, the best windows could vary between different participants. If we use the standard moving window method to build group-level classifiers across individuals, then data from the same windows from all individuals will be used to build and evaluate the classifier. However, since the best window could vary for different participants, using the standard moving window method could lead to suboptimal classification results.

We propose a novel individual-based time window selection (ITWS) algorithm to select the best window for each individual when building the group-level classifier. Figure 4.5 shows the structure of the algorithm. The main principle of the ITWS algorithm is to use an embedded individual-level classifier to determine the best window for each partic-

Algorithm 1 Individual-based time window selection (ITWS) algorithm

Initialize Divide the dataset of each participant into k folds. k-1 folds of the dataset are used as the training data and to obtain the best window for this specific participant, and the remaining one fold is used as the test data. Set the group-level training data and group-level test data to empty
for current k **in** k-fold cross-validation **do**
    for participant **in** all participants **do**
        Generate all moving windows by sliding the window on the data from the k-1 folds
        for window **in** all moving windows **do**
            Use embedded subject-level classifier to obtain the classification score on this window (embedded k-fold cross-validation)
        end for
        Select the best window by finding the maximum cross-validation score from all moving windows
        Add the data from the best window from the k-1 folds to the group-level training data
        Add the test data from the best window from the remaining fold to the group-level test data
    end for
    Train the group-level classifier on the group-level training data
    Obtain the test results by applying the group-level classifier on the group-level test data
end for
Calculate the average test result after 5-fold cross-validation.

ipant. The embedded individual-level classifiers are used in combination with the moving window method and are applied on the entire episode. Data from each participant are first separated into two blocks (training data and test data for the group-level classifier). Then, the embedded individual level classifiers are trained and evaluated only on one block (embedded k-fold cross-validation).

To effectively assess the performance of the machine learning models, the algorithm can be used together with k-fold cross-validation for the group-level classifier. The flow of the ITWS algorithm is described in Algorithm 2. Specifically, for k-fold cross-validation, we first divide the dataset (episodes of 40s) from each individual into k folds. Then, during each fold, k-1 folds of the dataset are used to find the best window for classification. The data from this window of these folds are later used as the training data for the group-level classifier. The data from the same window of the remaining fold are used as the test data to evaluate the group-level classifier. We repeat this procedure for all individuals. At each fold, training data from all individuals together are used to train the group-level classifier, and test data from all individuals are used to evaluate the classifier. The test result from all folds are then averaged to give the final mean test results.

Embedded individual-level classifier and group-level classifier

We use the shrinkage LDA as the embedded individual-level classifier as described in Sect. 4.5.1 (line 6 in Algorithm 2). Also, similar to individual-level classification, we examine the effect of window sizes by using 5s, 10s, and 15s as the window size for the moving window method.

Comparing to individual-level classification, the group-level classification can be trained on a larger dataset from all participants. Therefore, we aim to use modern machine learning models that take advantage of the larger sample size. Modern machine learning models, including XGBoost and deep learning techniques, have achieved state-of-the-art

results on many machine learning problems [123, 124, 125], and have shown promise for fNIRS data classification [66, 126]. Therefore, to evaluate the performance of the ITWS algorithm, we use Deep Neural Networks (DNNs), Convolution Neural Networks (CNNs), and XGBoost as the group-level classifier (line 12 in Algorithm 2).

XGBoost is a gradient tree boosting system that builds trees sequentially, such that each subsequent tree learns from its predecessors to reduce the errors of the previous tree [125]. Specifically, a greedy algorithm is used in the model, which starts from a single leaf, and iteratively adds branches to the tree by evaluating every possible split loss reduction. The ensemble model gives the aggregate output from all trees. To prevent overfitting, we set the learning rate to be 0.01, the maximum depth of a tree to be 4, the number of estimators to be 200, and the subsampling ratio of training instance subsample to be 0.8.

A DNN is a layered organization of connected neurons. Between the input and output layers, there are multiple hidden layers. During each hidden layer, each neuron is associated with a weight that is used to compute the weighted input. The weighted inputs are then summed and transformed by the activation function to determine the output of the neuron. By adjusting the weights of neurons, DNNs can model complex non-linear relationships between the input and output [123]. In this work, we use a network consists of three hidden layers with rectified linear unit (ReLU) activation function. Each hidden layer has 300 units, 100 units, and 40 units, respectively. We implemented an optimizer using RMSprop with a learning rate of 0.01.

CNNs are neural networks that use convolutions over the input layer. The hidden layers of a CNN typically consist of a series of convolutional layers, ReLU layers, and pooling layers. By performing specific functions, each layer learns a useful representation from the input [124]. In this work, our CNN architecture has three convolutional layers, which consist of 32 filters of size 3×1, 64 filters of size 3×1, and 64 filters of size

5×1, respectively. Each of them is followed by a batch normalization layer and a ReLU layer. Then, a max-pooling layer and a dropout layer are utilized to prevent overfitting. Finally, a fully connected layer with 64 input neurons and two output neurons is used for the binary classification. We implemented an optimizer using SGD with a learning rate of 0.01.

Similar to individual-level classification (see Sect. 4.5.1), the samples from the two classes are first balanced using SMOTE [122]. We used the same features for the embedded individual-level classification and group-level classification, which include the average values and slope of HbR and HbO from all long-separation channels. Each feature is normalized as well. All features are then used as the input for XGBoost and DNNs. For the input of CNNs, features of each channel are concatenated into a 2D matrix (*number of channels* $\times$ *number of features*).

Then, following the ITWS algorithm, we iteratively choose the best windows from each individual. Data from these windows are then used as training data and test data for the group-level classifier.

4.6 Evaluation

4.6.1 Methodology

We evaluate the effectiveness of our window selection method by comparing the results with the same classifiers, but without window selection.

For individual-level classification, our research questions are whether focusing on a specific window will improve the classification results, and whether the window size of the moving window method can affect the classifier's performance. Therefore, we compare the classification results achieved using the moving window method with 5s, 10s, and 15s as the window size, as well as with the classification results achieved using

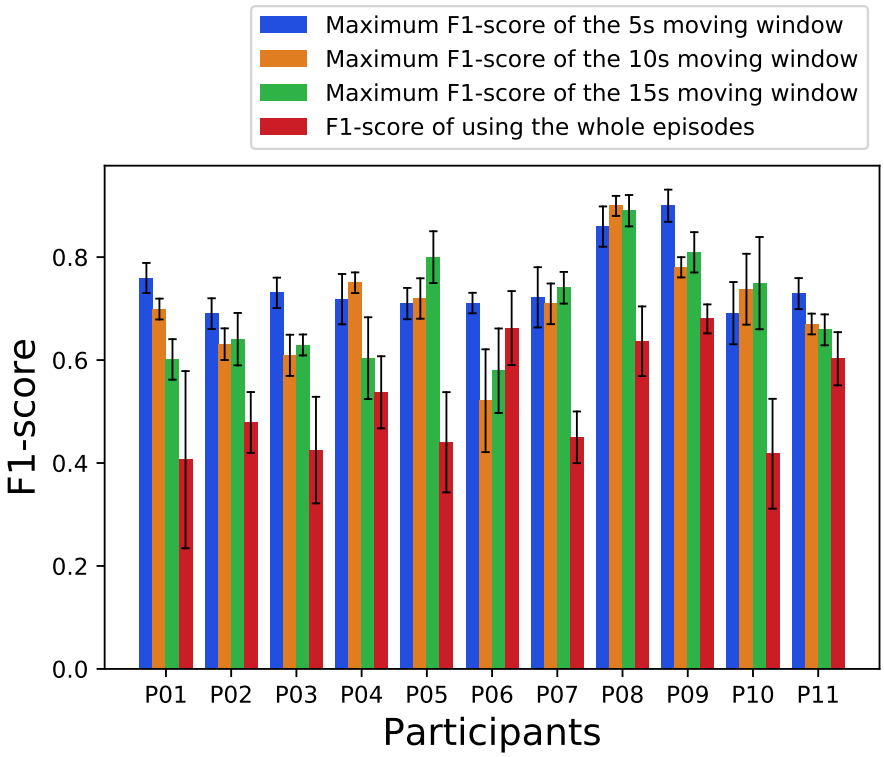

Figure 4.6: Comparison results of maximum F1-score achieved using the moving window method (with 5s, 10s, and 15s as the window size) and the F1-score achieved using the whole episodes (5-fold cross-validation)

the entire episodes.

For group-level classification, our research questions are whether the ITWS algorithm can improve the group-level classification results, and whether the choices of window sizes and classifiers can affect its performance. Therefore, when XGBoost, DNNs, and CNNs are used as the group-level classifier, we compare the classification results achieved using the ITWS algorithm with classification results achieved using a standard moving window method, as well as using the entire episodes. Specifically, we compare the results when 5s, 10s, and 15s are used as the window size for the moving window method.

Due to the imbalance in our dataset, the test accuracy of the classifiers could be misleading. Therefore, we report F1-scores of our classifiers. F1-scores are commonly used

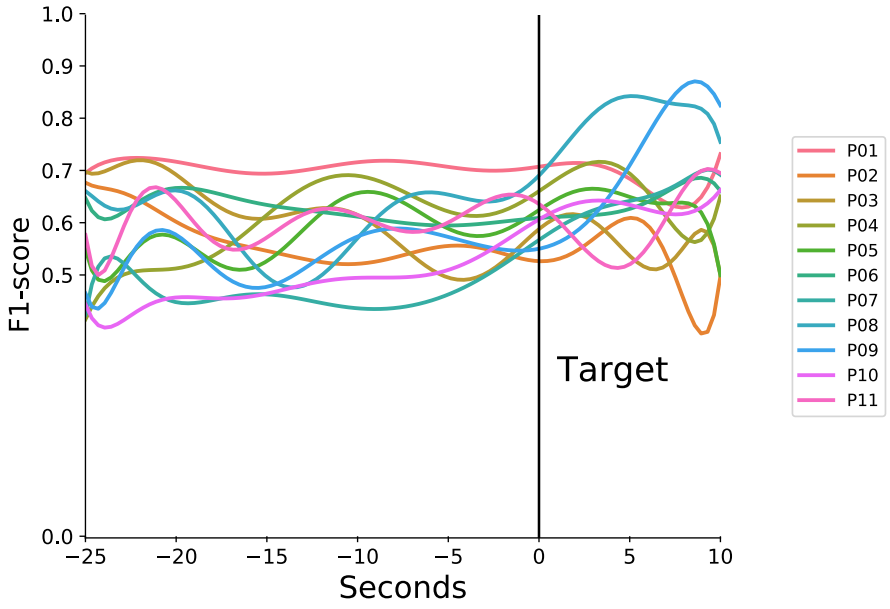

Figure 4.7: Classification results for 5s moving windows for each individual over the 40s time period, the x-axis indicates the right edge of the moving time window. The F1-score represents the mean F1-score of the 5-fold cross-validation on each window.

to account for the imbalance of the dataset. We also use 5-folds cross-validation to assess the performance of the classifiers.

4.6.2 Results

Individual-level classification using moving window

Figure 4.6 shows comparative results of maximum F1-score achieved using the moving window method, with the window size of 5s, 10s, and 15s, respectively, and the F1-score achieved using the whole episode. We can see that, for all participants, the maximum F1-scores achieved by using the moving window method with all three window sizes are significantly higher than the F1-score achieved when using the whole episodes (Wilcoxon signed-rank test, $p < 0.05$). When using the whole episode, only four participants achieved an F1-score over 60%. The average F1-score for all participants was $52.1 \pm 3.0\%$. When

Table 4.1: Comparative results of using the ITWS algorithm, the moving window method, and using the whole episodes for group-level classification (5-fold cross-validation). The F1-score of the moving window method represents the maximum F1-score.

	XGBoost	CNNs	DNNs
Whole episodes	45.4 ± 0.80	52.6 ± 0.14	48.8 ± 0.12
Moving window method (5s window)	57.0 ± 0.31	60.5 ± 0.13	55.2 ± 0.24
Moving window method (10s window)	52.3 ± 0.42	58.8 ± 0.22	52.5 ± 0.70
Moving window method (10s window)	51.6 ± 0.36	59.5 ± 0.53	54.6 ± 0.68
ITWS algorithm (5s window)	$\mathbf{73.2 \pm 0.18}$	$\mathbf{72.8 \pm 0.13}$	$\mathbf{69.4 \pm 0.08}$
ITWS algorithm (10s window)	70.1 ± 0.10	72.4 ± 0.06	68.7 ± 0.11
ITWS algorithm (15s window)	71.3 ± 0.21	70.7 ± 0.07	66.3 ± 0.07

using moving the window method with different window sizes, the window size of 5s achieved the highest average value ($74.8 \pm 2.0\%$) for all participants' maximum F1-score, while the average values for all individuals' maximum F1-scores are $70.0 \pm 2.8\%$ and $70.2 \pm 3.0\%$ with window sizes of 10s and 15s respectively. Particularly, for each individual, six out of eleven participants achieved a maximum mean F1-score when using the window size of 5s, while two and three participants achieved a maximum mean F1-score when using the window size of 10s and 15s, respectively. Furthermore, Fig 4.7 shows the F1-score for the moving windows for each participant with the window size of 5s. For each participant, we can see that the mean cross-validation F1-score varies for different windows.

These results suggest that for each participant, focusing on a specific window can achieve better classification results than using the whole episode. Also, the window size of the moving window method can slightly affect the classification results for different participants.

Group-level classification using the ITWS algorithm

Table 4.1 represents the F1-score achieved for group-level classification with different classifiers, when using the ITWS algorithm, the standard moving window method, and

using the whole episodes as input respectively. We can see that the ITWS algorithm greatly improved the group-level classification results with all three different window sizes (5s, 10s, and 15s), as well as with all three classifiers. Particularly, for different window sizes, applying the ITWS algorithm with the window size of 5s achieved the highest performance. This is easy to understand since most participants achieved the best individual-level classification results when using the window size of 5s (see Sect. 4.6.2). Specifically, when using the ITWS algorithm with a window size of 5s and with XGBoost as the group-level classifier, we achieved the highest average F1-score of $73.2 \pm 2.0\%$, while CNNs and DNNs achieved an average F1-score of $72.8 \pm 0.13\%$ and $69.4 \pm 0.08\%$ respectively. Also, it is worth noting that even with the window size of 10s and 15s, for all classifiers, the ITWS algorithm achieved superior performance than the standard moving window method, as well as using the whole episodes as input. Therefore, we can conclude that the proposed ITWS algorithms can improve the classification result for detecting mind-wandering episodes across-participants using fNIRS, and is generally not affected by choice of window sizes and classifiers.

To further investigate the effectiveness of the ITWS algorithm, we analyzed the selected windows for each participant by using the ITWS algorithm with a window size of 5s. Figure 4.8 shows the distribution of the right edge of selected time windows for each individual during the 5-fold cross-validation. For each individual, the box shows the quartiles with the inner line indicating the mean value. The whiskers extend to show the rest of the distribution, and the points are the "outliers" determined as a function of the inter-quartile range. Even though the selected best window for each individual varies during each fold, we can still see there are individual differences related to window selection. For example, the selected best windows for individual P03 concentrate around 20s before the target, while the selected best window for individual P10 centered around 5s after the target. Also, while some participants show a broader spread of window selection

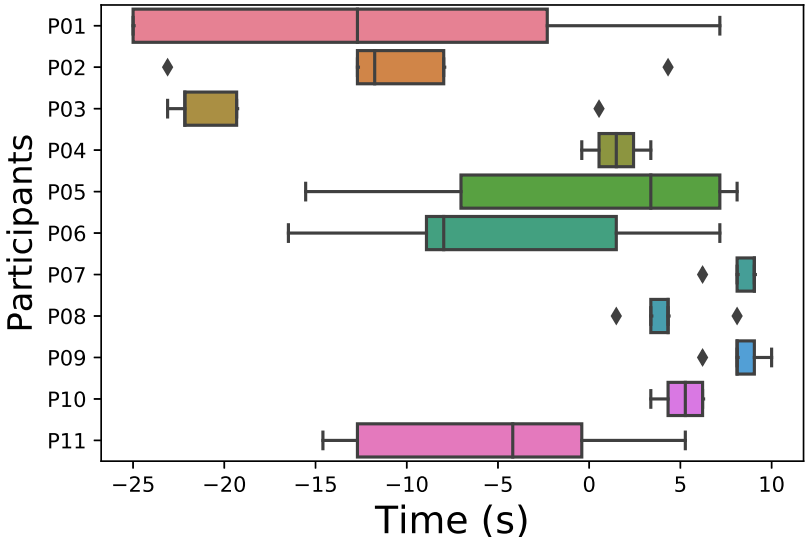

Figure 4.8: The distribution of the selected best windows (the right edge) for each individual during the 5-fold cross-validation, when using a window size of 5s. 0s represents the timing of the targets.

than the others, the classification results for test data from each participant did not show any differences. These findings further confirm that the proposed ITWS algorithms can incorporate individuals' differences in window selection and ensure the best window for classification for each individual is used to build the final classifier across individuals.

4.7 Discussion

Our study aimed to build classifiers based on fNIRS data to detect whether an individual is mind-wandering or focusing on-task. To build a dataset for exploration, we conducted a study using fNIRS during the SART task. The errors during the task are correlated with mind-wandering [87]. Consistent with previous findings, we showed individuals made a higher number of errors for the target than non-target trials. We also showed activation in the prefrontal cortex during mind-wandering episodes, as the changes of HbO increase

and the changes of HbR decrease before the targets with incorrect responses. All individuals retrospectively reported mind-wandering during the task in the post-survey.

For classification, we labeled the target episodes (30s before the target and 10s after the target) with a correct response as the on-task episodes, and we labeled the target episodes with incorrect response as the mind-wandering episodes. Particularly, we investigated window selection during the episodes when building classifiers both on an individual-level and group-level.

Compared to the previous state of the art in terms of brain-based classification of mind-wandering [31, 99], our proposed approach achieved significant improvement. Previous work using EEG to predict task-general mind-wandering achieved a mean accuracy of 64% [99], while prior work using fNIRS for mind-wandering classification achieved a mean accuracy of 56% [31]. Our results suggest that focusing on a specific window can improve the classification results for individual-level classifiers. For group-level classification, we proposed a novel algorithm to incorporate individuals' differences in window selection. We show that when using the XGBoost as the group-level classifier and 5s as the window size, the proposed ITWS algorithm achieved a mean F1-score of 73.2%. Moreover, we show that even though the window size can slightly affect the individual-level classification results for different participants, the performance of the ITWS algorithm is generally not affected by the choice of window sizes. Also, our results show that the ITWS algorithm can improve the classification results when used with different classifiers (XGBoost, CNNs, and ANNs).

Our findings have important implications for designing and evaluating engaging and effective learning interfaces, as well as building attention-aware systems that can automatically detect mind-wandering states using fNIRS. For real-time applications, labeled brain data is required to train the classifier, which can then detect the activation at the prefrontal area associated with mind-wandering. However, the classification of mind-wandering is

a challenging task. Different windows during mind-wandering episodes exhibit different time series behavior for each individual. As such, machine learning models trained on different windows of the labeled data can have different classification performance. Our classification methods serve the role of finding the best windows of training data for real-world applications. Classifiers trained on these windows can then be used to predict the label of real-time data. To do so, the first step is to collect labeled brain data from individuals. Then, the ITWS algorithm can be used to incorporate individuals' differences in window selection and determine the best windows for building the final classifier.

Our results show that the spread of selected windows varies a lot for some participants during cross-validation while applying the ITWS algorithm. This could be due to the overfitting of the individual-level classifiers since the dataset for each participant is small. Therefore, even though the classification results for test data from each participant did not show any differences in our work, further work that explores methods for more robust window selection can potentially improve the overall group-level classification results.

A limitation of this study is the mind-wandering episodes are inferred from behavioral responses and explicit reports of mind wandering. We aimed to avoid interrupting the mind-wandering episodes and therefore chose to determine mind-wandering episodes by SART errors, instead of using experience sampling probes. While previous research supports that SART errors are linked to mind-wandering [87, 111, 127], there is also research suggesting that SART errors could be related to impulsivity in individuals' responses [31]. Therefore, further investigation using experience sampling protocols and analyzing the window selection during the mind-wandering periods would be needed to confirm our findings.

4.8 Conclusion

In this chapter, we investigated window selection for classifying mind-wandering episodes and on-task episodes using fNIRS. The proposed classification framework is data-driven and enables a more accurate detection of mind-wandering. The findings from this study also reveal individual differences in window selection for mind-wandering detection. This work could inform further research about the time course aspects of mind-wandering, and it builds a foundation for both evaluation of multimodal learning interfaces and future attention-aware systems based on fNIRS data.

Chapter 5

Classifying Driver Cognitive Load Using fNIRS with CNNs, Multivariate LSTM-FCNs and ESNs

In the previous chapter, we explored mind-wandering detection using fNIRS and developed a data-driven classification framework[1]. In this chapter, we investigate classifying different levels of driver cognitive load using fNIRS. In particular, we aim to apply advanced machine learning approaches that are specially designed to extract spatial and temporal patterns. We apply Convolutional Neural Networks (CNNs), multivariate Long Short Term Memory Fully Convolutional Networks (LSTM-FCNs), and Echo State Networks (ESNs) for fNIRS data classification.

5.1 Introduction

Road traffic accidents have claimed more than 1.35 million deaths each year around the world, with around 50 million people injured [129]. Meanwhile, according to a report

[1]Part of the work in this chapter was originally described in Liu, et al. "Unsupervised fNIRS feature extraction with CAE and ESN autoencoder for driver cognitive load classification". *Journal of Neural Engineering* [128].

from the National Highway Traffic Safety Administration (NHTSA), 36,560 lives were lost on U.S. roads in 2018, with around 400,000 people injured and 2,841 people killed by distracted drivers [130]. Distractions are often caused by a mix of auditory, vocal, visual, manual, and cognitive demands (e.g. [131]). As a complex and intensive activity, driving requires a driver to focus on not only the car, but also on factors such as nearby vehicles, traffic signs, pedestrians, and lights. At the same time, the increased number of mobile devices and advanced in-car communication and infotainment systems are imposing different levels of cognitive load on the driver [132]. Research has shown both under-load and overload of driver's cognitive resources are related to road accidents [133]. When drivers are under-loaded, they can experience fatigue or drowsiness, and this may lead to reduced alertness and lowered attention. When drivers are overloaded, drivers are under stress and this may lead to insufficient attention and inadequate capacity and time for information processing [134]. As a result, understanding the cognitive load of drivers has the potential to contribute to avoiding future accidents and hazards on the road [135].

Previous research has used several approaches to assess drivers' cognitive load, which can be divided into three main categories: subjective measures, performance measures, and physiological measures [135, 136]. Each of these approaches has both advantages and disadvantages [134]. Subjective measures can provide strong periodic indicators of load but require interrupting the task flow with probes or recalling events post hoc. Continuous objective measures, such as those that are physiological-based, can provide greater sensitivity to the time course changes in cognitive load during driving [137]. As such, various types of physiological data have been collected for driver cognitive load studies, e.g. electroencephalogram (EEG) data [138, 139], heart rate [135, 137, 140], skin conductance [135, 137, 141] and eye movements [142].

Functional near-infrared spectroscopy (fNIRS) is a brain imaging technique, which has been shown to be useful for evaluating human cognitive load and working memory de-

mand under various circumstances [29, 30, 66, 69, 143]. fNIRS emits near-infrared light into the brain. By measuring the light returned to the surface, the amount of oxygenated hemoglobin (HbO) and deoxygenated hemoglobin (HbR) can be calculated, which can indicate hemodynamic activity associated with brain activation in that area. As a portable and non-invasive technique, it has the potential to be used for driver cognitive load estimation [42, 44].

Most previous studies in this direction utilized traditional signal processing methods to analyze fNIRS signals without using state-of-the-art machine learning algorithms [42, 43, 44]. fNIRS data are high-dimensional and high volume time series data. However, these studies either used a small segment or simple statistics to describe fNIRS data. The former approach requires the selection of small windows from the whole series and ignores global temporal dynamics, while statistics-based features lose both amplitude and temporal details. Motivated by this, we aim to explore advanced machine learning methods for fNIRS data, to improve the classification accuracy for differentiating different levels of driver cognitive load using fNIRS.

Recent advances in deep learning allow task-specific features to be deep learned from various sources such as images, languages, and brain data [144, 145, 146], which are usually more powerful than hand-crafted ones. The main idea of this work is to learn high-level features using neural networks that are specially designed to extract spatial and temporal patterns from the input data. In general, artificial neural networks can be divided into two categories, feed-forward neural networks and recurrent neural networks (RNNs). Feed-forward neural networks, such as the convolutional neural networks (CNNs), have shown powerful feature abstraction capability for extracting spatial and temporal dependencies from brain data [66, 67, 147]. RNNs, such as Long short-term memory (LSTM) and Echo State Networks (ESN), have shown to be very effective in extracting temporal patterns from time-series data [148, 149, 150, 151, 152, 153, 154]. Moreover, re-

searchers have explored the combination of feedback-forward neural networks and RNNs for time-series data classification. For example, multivariate Long Short Term Memory Fully Convolutional Networks (LSTM-FCNs) has received a lot of attention from the time series classification community due to its superior performance over other models [155]. However, research to date has not explored the application of RNN-based models or for fNIRS feature extraction. In this work, we set out to employ both feed-forward neural networks and RNNs-based architectures for fNIRS feature extraction. Particularly, we employ CNNs, multivariate LSTM-FCNs, and ESNs and compare their results on classifying fNIRS as an estimator of driver cognitive load.

In this paper, we report on a study that involved the collection of fNIRS data in a simulated driving environment. Drivers completed an n-back task to impart additional structured cognitive load during driving, as a proxy for real-world tasks that increase cognitive load during driving. Because the collected data are represented as multi-channel time-series signals, we propose to apply CNNs, multivariate LSTM-FCNs, and ESNs for driver cognitive load classification. Moreover, to fully capture the global temporal information and to be trained on a larger dataset, we build group-level models across all participants' data without selecting particular windows. The results show that CNNs, multivariate LSTM-FCNs, and ESNs are suitable for fNIRS feature extraction, while the proposed ESN method achieved greater classification accuracy than CNNs and multivariate LSTM-FCNs for differentiating different levels of driver cognitive load using fNIRS signals.

The main contributions of this paper can be summarized as:

- We propose a machine learning framework for driver cognitive load classification using fNIRS data.
- We describe the application CNNs, multivariate LSTM-FCNs, and ESNs for fNIRS data classification.

- We show that the proposed ESN method yields state-of-the-art classification accuracy for group-level models without window selection for fNIRS-based driver cognitive load classification.

5.2 Background

In this section, we first review previous work in using secondary task and psychophysiological data for driver cognitive load analysis, which motivates our work in investigating fNIRS for driver cognitive load classification. We then discuss previous work and challenges for fNIRS data classification.

5.2.1 Driver Cognitive Load Assessment

Secondary Task Paradigms During Driving

As a driver's cognitive demand often includes competition between the driving task and non-driving related activities, driver cognitive load studies often utilize controlled and repeatable secondary task paradigms. Recent studies have adopted many types of secondary tasks and collected a variety of psychophysiological data for driver cognitive load analysis. Tsunashima *et al.* used mental calculation tasks, which consisted of a low-demand task (one digit addition), a medium-demand task (one digit addition of three numbers), and a high-demand task (subtraction and division with a decimal fraction), and evaluated the effectiveness of fNIRS for measuring differences in driver cognitive load [42]. In addition to steering and maintaining a set speed in a driving simulator, during secondary task periods designed to model increased cognitive load, Wu *et al.* asked participants to press one of the buttons on a panel when prompted by a command on the display screen [156]. Zhang *et al.* employed a verbal task and a spatial-imagery task as secondary tasks. The verbal task required drivers to name words starting with a designated letter while

the spatial-imagery task asked them to respond letters from A to Z under five rules that they predefined. During the task, eye tracker and head tracker were applied to obtain corresponding physiological data [157]. Putze *et al.* asked participants to perform a visual search task and a mathematical cognitive task, while multiple biosignal streams (skin conductance, pulse, respiration, EEG) were collected [139].

Besides the aforementioned secondary tasks, recent studies have frequently adopted a type of secondary task called an n-back. A version of the n-back task was developed by the MIT AgeLab [136, 158] for the context of driving and later incorporated into ISO 14198 [159] as a standardized method to calibrate or otherwise characterize reference levels of demand placed upon a driver. In the standardized presentation of this form of the n-back, a series of single-digit numbers are presented via audio. Participants are asked to respond with the corresponding number n positions before the current number. As a result, the parameter n can easily adjust the level of cognitive load. For example, using the n-back task as the secondary task, Solovey *et al.* collected heart rate and skin conductance data while participants were driving on the highway [135]. Li *et al.* collected fNIRS and heart rate data while implementing an alternate n-back task in a simulated driving experiment [44]. The latter is an example of a study using a form of n-back task that presents a series of single letters. As each letter appears, the participant responds if the new letter matches a letter presented n-places back in the sequence (see Owen *et al.* [160] for a review). This matching form is arguably more difficult for a given value of n [158].

Driver Cognitive Load Analysis

To analyze driver cognitive load using physiological data, researchers have proposed various data analysis methods. Tsunashima *et al.* proposed a signal processing method based on multi-resolution analysis (MRA) using a discrete wavelet transform. The results on nine participants suggested that fNIRS data were effective for driver cognitive load eval-

uation [42]. However, they only conducted statistical analysis in this work, and did not apply machine learning for driver cognitive load classification. Wu *et al.* proposed a queuing network based on the theory of human performance and neuroscience, and explored the cognitive characteristics of drivers' cognitive load caused by their actions with the vehicle information system [156]. Kim *et al.* extracted EEG variation rates in five different driving situations, including left and right-turn, rapid-acceleration, rapid-deceleration, and lane-change [138].

In recent years, due to its success in classification tasks, machine learning has become a popular tool for driver cognitive load classification. Yang *et al.* applied SVM and extreme learning machine (ELM) as the classifiers for eye gaze data, and the results show that the ELM-based method achieved better performance, with an accuracy of 76.4% for classifying high driver mental cognitive load from low driver mental cognitive load [161]. Solovey *et al.* evaluated different machine learning classifiers for driver cognitive load by using heart rate data. They achieved a high accuracy of 89% for classifying consecutive 2-back elevated periods from normal driving, when using logistic regression with window selection [135]. Fridman et al [162] considered classification using 3D convolutional neural networks leveraging visual-only attributes alone to achieve 86% accuracy over a 3-class problem. Le *et al.* trained and tested multiple classifiers for classifying driver cognitive load using fNIRS. They show that the decision trees achieved the best results with an accuracy of 82% for classifying different cognitive load elevated by the n-back task during driving. However, it is unclear which tasks and time window their classification was based on [43].

Results from previous work suggest that driving cognitive load is predictable by machine learning techniques using visual behavior and physiological data. However, researchers also pointed out that other factors rather than cognitive load, such as physical exertion and emotional state, can also influence physiological signals, which could result

in conflicting or unreliable results [163]. fNIRS measures changes in cerebral hemodynamic activity and can be used to infer information on drivers' underlying cognitive activity directly. Moreover, it is safe, portable, easy to use, and quick to set up - characteristics that show promise for use in real-world settings. As such, fNIRS could provide an alternative for measuring driver cognitive load levels objectively. However, an fNIRS-based system using state-of-the-art using machine learning techniques for driver cognitive load classification is not fully explored. Therefore, it would be valuable to explore the potential of such approaches and develop a solid framework.

5.2.2 fNIRS Feature Extraction and Classification

In addition to being used for driver cognitive load assessment, fNIRS data has been widely explored for classifying cognitive load levels in other circumstances, often through employing a range of variations on the ISO standardized version of the n-back task. Due to the high dimensionality and redundancy, the raw signal of fNIRS data is not suitable for being used as features for classification. Therefore, feature extraction is an important process in fNIRS-based classification.

Hand-crafted Features vs. Deep Learned Features

Before CNN-based methods became the superior approach for feature extraction, the hand-crafted feature approach was used in most previous work. As fNIRS data are time-series data, statistics obtained by specific time windows were often calculated as features. Aghajani *et al.* classified different cognitive load levels elevated by the n-back task (n from 0 to 3), using the calculated slope, standard deviation, skewness, and kurtosis of each HbO and HbR signal, and the zero lagged correlation between HbO and HbR as features. These features were then selected based on their sensitivity to the changes in cognitive load. By using SVM and the moving window method, they achieved a mean

accuracy of 74.8% for binary classification [164]. Similarly, Liu *et al.* extracted the average HbO and HbR amplitude changes as features for classifying cognitive load elevated by the n-back task (n = 0, 2, 3). By using LDA, they achieved a mean accuracy of 53.9% for three-class classification [83].

Besides using statistical features, regression techniques were also employed to extract features from fNIRS data. In the work of Herff *et al.* , features were extracted by fitting the slope of a straight line to the data in a specific window using linear regression during the n-back task. Their results show that classifying 3-back, 2-back, 1-back against a relaxed state achieved an accuracy of 81%, 80%, and 72%, respectively, while the accuracy for four-class classification is 45% [69].

With the advances in deep learning, more recent work has investigated using deep learning methods to automatically extract features from fNIRS data. For example, Trakoolwilaiwan *et al.* [147], utilized four different CNNs to extract fNIRS features. The results show that CNNs achieved higher accuracy than the combination of SVM/ANN and handcrafted features (mean, variance, kurtosis, skewness, peak, slope from HbO and HbR). Similarly, combined with the moving window method, Saadati *et al.* showed that the CNN approach can improve the accuracy for cognitive load classification using fNIRS data, with an average accuracy of 82% [71].

These studies have demonstrated the advantages of advanced machine learning methods for automatic fNIRS feature extraction and classification. However, challenges remain, including the fact that researchers need to take the sizes of fNIRS datasets into consideration when applying deep learning models. Brain datasets are usually small due to the costly and time-consuming data collection process. At the same time, deep learning techniques require a large number of training data to achieve satisfactory results [165]. Also, since fNIRS data are time-series data, researchers need to take the spatial and temporal dynamics of fNIRS data into consideration when applying these models. In the next

section, we outline these considerations and possible approaches.

Considerations

There are two important considerations when applying machine learning techniques on fNIRS data: 1) the selection of sample windows and 2) the choice between individual models and group models.

Research has shown that due to the latency of the underlying physiological processes, fNIRS cognitive load classification may require a minimum window length of 10 seconds [69, 70]. Therefore, to capture the global temporal information, in this work, we will regard each complete trial (30 seconds without window selection) as one sample for classification.

There are only a few studies that have investigated building group models for fNIRS-based cognitive load classification. Putze *et al.* implemented the n-back task in a virtual environment, and extracted the signal mean for all HbO and HbR channels, as well as the resulting slope and coefficient of each channel through linear regression as features. By pooling the data of all participants together and using shrinkage LDA as the classifier, they achieved a mean accuracy of 66% for classifying the 3-back period from the 1-back period, a mean accuracy of 64% for classifying the 2-back period from the 1-back period, and a mean accuracy of 42% for three-classes classification (1-, 2-, or 3-back) [70]. Liu *et al.* also investigated fNIRS-based cognitive load classification accuracy by learning from the data of other participants. They extracted the average HbO and HbR amplitude change between different windows from n-back tasks, and achieved a mean accuracy of 53.9% for three classes classification (0-, 2-, or 3-back) [83].

From these studies, we can see that while it is beneficial to build group-level models using fNIRS data from a complete trial without window selection, it is difficult to achieve high accuracy for cognitive load classification. Thus, it would be valuable to research

more advanced machine learning methods to extract temporal dynamics from fNIRS data without window selection and enable higher performance for group-level models. Considering the relatively small sample sizes of most fNIRS datasets, it could be difficult for the CNN-based method to fully extract temporal information from the data without overfitting [147]. Therefore, in this work, in addition to CNNs, we also investigate the application of multivariate LSTM-FCNs and ESNs for extracting temporal patterns from fNIRS data.

5.3 Data Collection

The goal of our study is to build a dataset of fNIRS data associated with different levels of working memory demands that come from secondary tasks during driving. While there is a wide range of tasks that a driver may perform, we use a variant of the n-back task as the secondary task, which has established capacity for eliciting scaled levels of working memory demand [135]. This task serves as a structured proxy for cognitively loading auditory-verbal working memory tasks that a driver may perform. The study was approved by the relevant institutional review board and informed consent was obtained for all participants.

5.3.1 Driving Simulator

Our study was conducted in a driving simulator equipped with fNIRS. The driving simulator consisted of a fixed-base, full-cab Volkswagen New Beetle in front of an 8×8 ft projection screen (Figure 5.1) with established validity for assessing changes in cognitive demand using the n-back [141] and visual manual based tasks [166]. Participants had an approximately 40-degree view of a virtual environment at a resolution of 1024×768 pixels. Graphical updates to the virtual world were computed by using Systems Technology

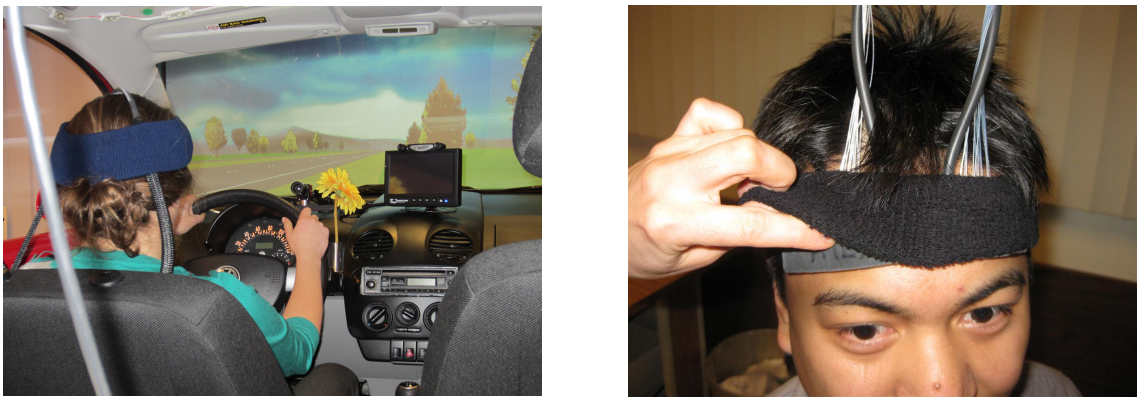

Figure 5.1: Driving simulation environment (left). The participants sit in the car and are instrumented with fNIRS (right). The screen in the front presents the simulated driving environment.

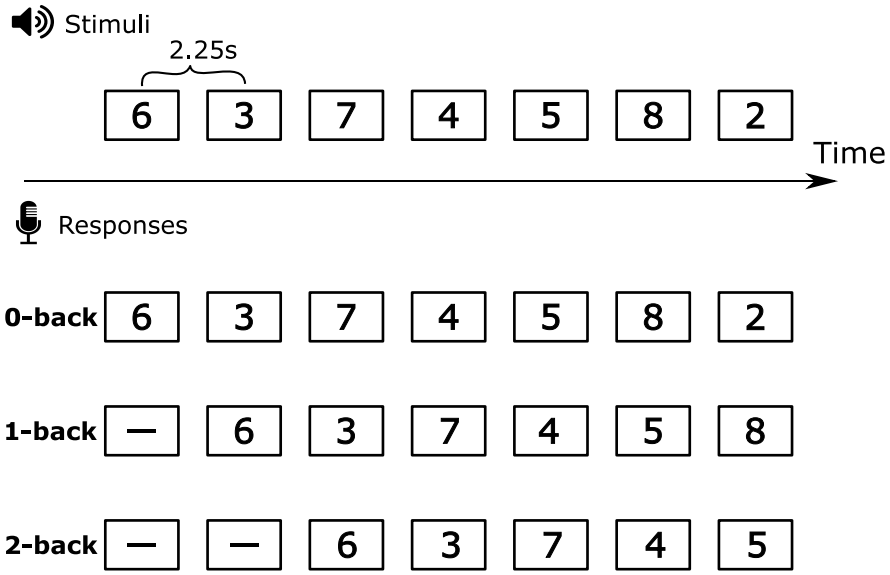

Figure 5.2: Example task block of auditory stimuli and the appropriate verbal responses for a 0-back task, a 1-back task, and a 2-back task.

Inc. STISIM Drive and STISIM Open Module based upon a driver's interaction with the wheel, brake, and accelerator. Additional feedback to the driver was provided through the wheel's force feedback system and auditory cues. The time-based triggering of visual and auditory stimuli was supported by custom data acquisition software and used to present

prerecorded instructions for the cognitive task.

5.3.2 fNIRS Recording and Body Sensing

The fNIRS data were acquired using was a multichannel frequency domain Imagent from ISS Inc. Two probes were placed on the forehead to measure the two hemispheres of the anterior prefrontal cortex (Figure 5.1). Each source emits two near-infrared wavelengths (690 nm and 830 nm) to detect and differentiate between oxygenated and deoxygenated hemoglobin. Each source corresponds to four detectors, with the source-detector distances being 1.5, 2, 2.5, and 3cm. The sampling rate was 11.8 Hz. The sensors were kept in place using headbands, which can also reduce light interference.

Physiological data was obtained from a MEDAC System/3 instrumentation unit (NeuroDyne Medical Corporation). A modified lead II configuration was employed for electrocardiograph (ECG) recording in which the negative lead was placed just under the right clavicle (collar bone), ground just under the left clavicle, and the positive lead on the left side over the lower rib.

5.3.3 Driving Task and Secondary task

Participants sat in a stationary car and drove a divided, multi-lane interstate highway consisting largely of straight roadway with occasional gradual curves in the simulated environment.

While driving, an auditory presentation - verbal response n-back task was employed to impose additional cognitive load while driving [158, 159]. In each 30-second task block, a series of single digits (0-9) were presented in random order (one at a time) at 2.25 seconds intervals. As each new digit was presented, participants were to say out loud the digit n items back in the current sequence - the difficulty of the task increases

as n increases. Three levels of difficulty were employed to present drivers with a low, moderate, and high level of secondary cognitive load. At the lowest cognitive load level (0-back), participants simply repeat each number as it is presented. At the moderate level (1-back), participants were required to respond with the number one item back in the sequence. In the most difficult level (2-back), participants responded with the number two item back in the sequence. Figure 5.2 describes an example set for the 0-back, 1-back and 2-back task.

5.3.4 Participants

Thirty individuals driving more than three times a week and having a valid driver's license for at least three years were recruited. Participants had to report a driving record free of accidents for the past year. Due to recording issues, only 18 of the participants (between the ages of 20 and 33) had reliable fNIRS signal recording.

5.3.5 Design and Procedure

Participants were given instructions on how to complete the n-back task and practiced the task following training standards detailed in Appendix A of [158] prior to entering the simulator. During the experiment, blocks were formed with a random ordering of each with three load levels (0-back, 1-back, and 2-back), a 30-second period in which participants were asked to 'just drive,' (which we refer to as the *single-task driving* task) and a *blank-back* [167] where digits of the n-back were played with participants instructed to listen but not to respond. The blank-back condition is not considered in this analysis. Participants completed three blocks separated by a 90-second cooldown.

5.4 Dataset Curation

Based on the fNIRS data collected during the study, we built the dataset for investigating feature extraction and classification for different levels of cognitive load.

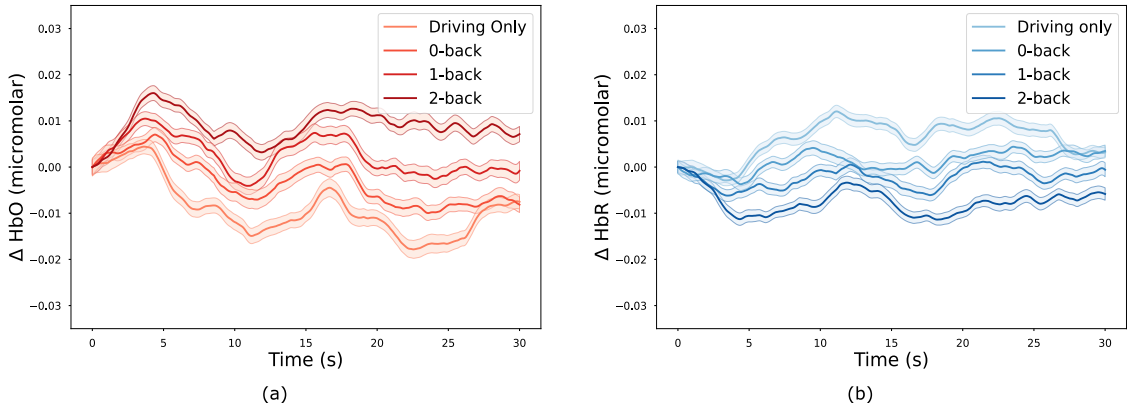

Figure 5.3: Variation of the changes in HbO and HbR concentration for different conditions. The figures show the mean (averaged across all channels and all individuals) and standard error over each condition. Shaded areas represent the standard error of the mean for each condition.

5.4.1 Behavioral Data and Heart Rate

We analyzed the participants' performance during the n-back conditions, as well as participants' heart rate data during the experiment. Participants performed well on the secondary task, with an average accuracy of 100% on the 0-back task, 98.72% on the 1-back task, and 96.44% on the 2-back task. A One-Way ANOVA shows significant differences between the three n-back levels in the number of errors ($F = 6.85$; $p < 0.001$). Furthermore, Tukey's post hoc tests showed that participants made significantly more errors during the 2-back task than the 0-back task ($p < 0.005$). For the heart rate data, we employed a QRS detection algorithm to identify heartbeats in the EKG signal [168, 169]. The results of heartbeat detection were manually reviewed and edited. In general, the

average heart rate during the driving only conditions is 75.82 beats per minute (bpm) (SD = 1.96), while the average heart rate during the 0-back task, 1-back task, and 2-back are 80.13 bpm (SD = 2.03), 82.19 bpm (SD = 2.13), and 85.62 bpm (SD = 1.98), respectively. A One-Way ANOVA shows there are significant differences between different conditions in the average heart rate ($F = 4.50$; $p < 0.005$). Tukey's post hoc tests showed that participants' heart rate is significantly higher during the 2-back task than the driving only condition ($p < 0.001$). These results are consistent with prior findings and indicate that the different n-back conditions can induce different levels of workload during the experiment [69, 135].

5.4.2 General Dataset Description

The dataset consists of fNIRS data of 8 channels, from 18 participants. Each sample consists of data in a 30-second period. There are a total of 54 samples for each class (*single-task driving*, 0-back, 1-back, 2-back).

5.4.3 Dataset Preprocessing

Since signals measured by fNIRS may suffer from biological and technical artifacts, preprocessing is usually employed to enhance signal quality [114]. Following typical preprocessing techniques [115], we used a band-pass filter with a high pass value of 0.02 Hz and a low pass value of 0.5 Hz to remove the physiological noise (e.g., heart rate, respiration) and the instrumental noise. Raw light intensity data was then converted to HbO and HbR values using the Modified Beer-Lambert Law. Then, the correlation-based signal improvement (CBSI) is introduced to reduce motion artifacts. It has been shown that the CBSI method can effectively remove large spikes brought by head movements as well as enhance signal quality and spatial specificity [170]. All preprocessing was completed in

MATLAB using HomER [116].

5.4.4 Dataset Overview

For an overview of the dataset, we calculated the folded average of HbO and HbR change across all participants for each condition. Specifically, we calculated the changes in HbO and HbR by subtracting the corresponding value of the starting point for each trial. Fig. 6.5 shows the block averages of changes in HbO (red) and HbR (blue) for all participants across all channels and all n-back conditions. From Fig. 6.5 (a), we can see that for all conditions, at the beginning of each trial, following neural activation, there is an increase in HbO, which is followed by a decrease in HbO due to the metabolic consumption of oxygen. Moreover, it is clear that the peak value of HbO increases as the difficulty of the task increase. The peak value of HbO is higher in the 1-back condition than the 0-back condition and driving only, with the highest value during the 2-back condition. From Fig. 6.5 (b), similarly, we can see that there is a decrease in HbR at the beginning of each trial, and followed by an increase. Also, the value of HbR is lower in the 1-back condition than the 0-back condition and driving only, with the lowest value during the 2-back condition. Moreover, we tested the effect of n-back condition and channels using two-way repeated-measures ANOVA and determined the main effects using Tukey's post hoc tests. we calculated the mean values for the driving only condition and three n-back conditions (0-back, 1-back, 2-back). The mean HbO and HbR values were then analyzed by a $4(condition) \times 8(channel)$ repeated measures ANOVAs. Both the n-back condition and channels showed a significant effect on HbO and HbR ($p < 0.001$), while the interaction effect was not statistically significant. Furthermore, post hoc analyses showed that the 2-back task elicited higher HbO increases than the 0-back and the driving only condition ($p < 0.01$). our results are consistent with prior research and suggest heterogeneous activation at the prefrontal area as the difficulty of the task increase [44, 69, 164, 171].

Furthermore, this lays the foundation for our feature extraction and classification techniques.

5.5 Classification Methods

We investigate the application of CNNs, multivariate LSTM-FCNs, and ESNs for fNIRS data classification.

5.5.1 Input

For each sample, the HbO and HbR from eight channels in the 30-second period are used as the input for all feature extraction methods. Since the sampling rate was 11.8 Hz, the length of the data is 354. Data from each channel is normalized using the Min-Max normalization technique. In addition, considering that the corrected HbO and HbR signals using the CBSI methods are highly correlated, we evaluate the effect of using only HbO, using only HbR, and using the combination of HbO and HbR as input on model performance when comparing different feature extraction methods.

5.5.2 CNNs

When using CNNs for classification tasks, researchers have shown that unsupervised pre-training can improve the model's performance [172]. Therefore, in this work, we chose to use a CNN with unsupervised pre-training for fNIRS data classification.

An autoencoder neural network is often used to pre-training neural networks [173]. An autoencoder neural network is an unsupervised learning algorithm that aims to minimize reconstruction error between the input data and the output data. Autoencoders consist of three main parts: the encoder, the bottleneck, and the decoder. The encoder

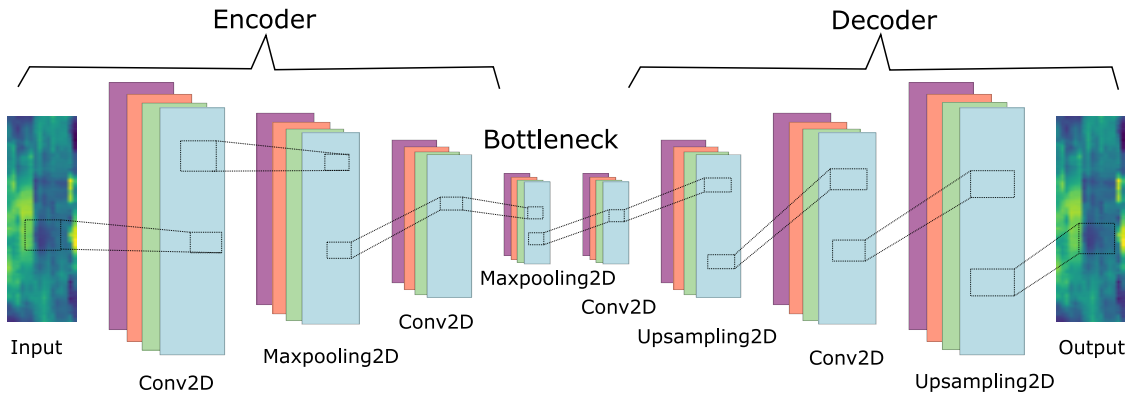

Figure 5.4: The architecture of convolutional autoencoders, which include the encoder, the bottleneck, and the decoder. After unspervised training, the bottleneck layer becomes the learned features for the input.

learns how to compress the input data into a low-dimensional representation. The bottleneck is the layer containing the compressed representation of the data. The decoder part learns how to reconstruct the compressed data to be as close to the original input as possible. By minimizing the reconstruction loss through backpropagation, the compressed representation of the input becomes learned features that contain meaningful information of the input and are useful for future tasks. Convolutional autoencoder (CAE) uses convolutional layers in the encoder and decoder, which inherit the powerful feature abstraction ability of traditional CNNs and have been widely applied for extracting spatial and temporal dependencies from data. Particularly, it can preserve spatial locality by receptive field and parameter sharing. Additionally, convolutional layers can be followed by pooling layers for downsampling in the encoder part, while convolutional layers in the decoder are followed by unpooling layers for upsampling. Figure 5.4 shows the overview of applying CAE for pretraining the CNNs.

Specifically, in this work, to fully capture the spatial information contained by fNIRS signals collected by different channels and the time-series behavior of fNIRS data, fNIRS data was constructed as a set of 2D images, with the length of the image equal to the

number of samples in the time window, and the width of image equal to the number of channels. For a given multi-channel fNIRS data input matrix X, and a set of n convolutional filters $\{F_1^{(1)}, \cdots, F_N^{(1)}\}$, the encoder computes:

$$e_m = \sigma(X * F_m^{(1)} + b_m^{(1)}) \tag{5.1}$$

where σ denotes activation function, $*$ represents 2D convolution. F_m is m_{th} 2D convolutional filter, and b_m denotes encoder bias. Then, the reconstruction can be obtained using of feature maps $E = \{e_{m=1,\cdots,n}\}$ and convolutional filters $F^{(2)}$ in the decoder:

$$\tilde{X} = \sigma(E * F_m^{(2)} + b_m^{(2)}) \tag{5.2}$$

The mean square error between the original input data of and the reconstructed data can be used as the cost function:

$$L_\varepsilon(X, \tilde{X}) = \frac{1}{2}||X - \tilde{X}||^2 \tag{5.3}$$

During pre-training, the reconstruction error is minimized through optimizing the network weights, and the bottleneck layer becomes the learned representation for the input and can be used for classification.

Considering that the architecture of CAE can affect the resulting performance, we determine the best architecture of CAE for classifying driver cognitive load using fNIRS by investigating the effect of filter sizes, as well as depth and width on the classification accuracy. CNNs can be constructed by removing the decoder part and adding fully connected layers. Specifically, we add two fully connected layers and output neurons with the rectified linear unit (ReLU) activation function. Each layer has 200 units, and 100 units, respectively. We implemented an optimizer using RMSprop with a learning rate of

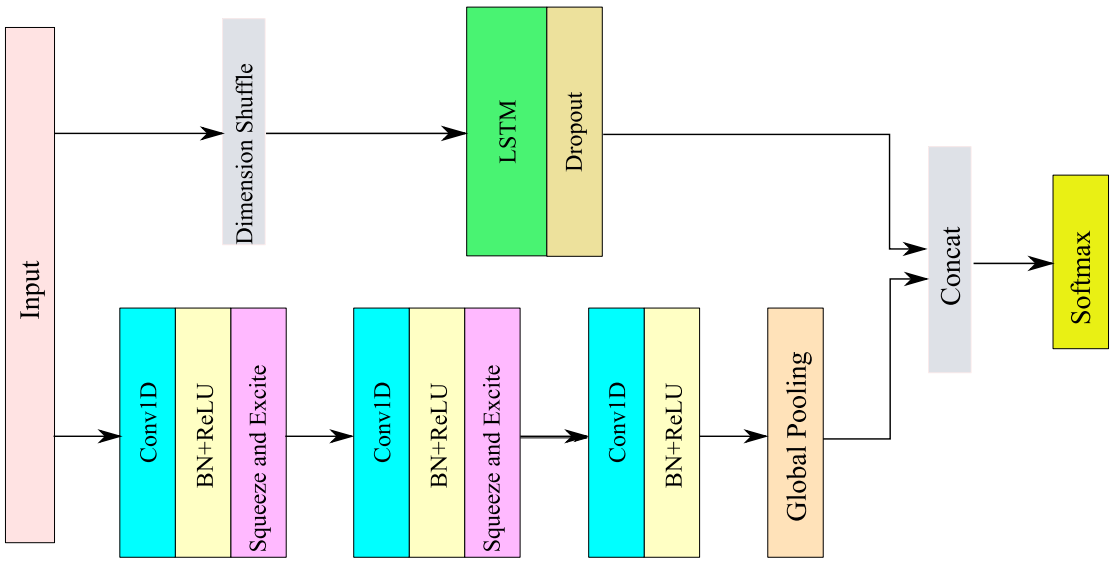

Figure 5.5: The architecture of multivariate LSTM-FCN.

0.01. The parameters of the CNNs including the pre-trained weights are then fine-tuned through optimizing.

5.5.3 Multivariate LSTM-FCNs

Multivariate LSTM-FCNs have received a lot of attention due to their advantage over other models on time series classification. The model achieved state-of-the-art classification accuracy on many multivariate time series datasets while requiring minimal preprocessing of the data [155]. The model augments a convolutional neural network with long short term memory recurrent neural network (LSTM RNN). We see promises of applying the models for fNIRS data classification. CNNs are well suited for capturing the temporal dependency between different channels of fNIRS data, while LSTM can strengthen the model's ability to capture the temporal patterns of the data.

Figure 5.5 shows the architecture of Multivariate LSTM-FCNs. The model comprises a fully convolutional block and an LSTM block. The fully convolutional block contains

three stacked temporal convolutional blocks. The convolutional blocks contain a convolutional layer, which is succeeded by batch normalization and the ReLU activation function. Then, to incorporate inter-correlations between multiple variables at each time step, the first two convolutional blocks end with a *squeeze and excite block*. The *squeeze and excite* block can adaptively recalibrate the weights of each channel based on its importance, by rescaling the output feature maps of prior layers [174]. Specifically, for input X, after the convolution operation, we represent the feature maps as U. For each channel of U, a *squeeze* operation is performed to calculate the channel-wise statistics z over the temporal dimension T:

$$z_c = F_{sq}(u_c) = \frac{1}{T}\sum_{t=1}^{T} u_c(t) \tag{5.4}$$

After that, an excite operation is performed to capture the channel-wise dependencies:

$$s = F_{ex}(z,W) = \sigma(g(z,W) = \sigma(W_2\delta(W_1 z)) \tag{5.5}$$

where, δ refers to the ReLU activation operation, σ refers to the Sigmoid activaion operation, $W1$ is a fully-connected layer for dimensionality reduction by a ratio r, $W2$ is a second fully-connected layers for dimensionality increasing, and z is the output from the squeeze block. Then, the final output of the block can be represented as:

$$x_c = F_{scale}(u_c, s_c) = s_c \times u_c \tag{5.6}$$

where, $F_{scale}(u_c, s_c)$ denotes the channel-wise multiplication between the feature map u_c and scale s_c.

After the final convolution block, a global average pooling is applied. The LSTM block comprises an LSTM layer, which is followed by a dropout layer. Also, a dimension shuffle process is applied to the input, which transfers the temporal dimensions of the

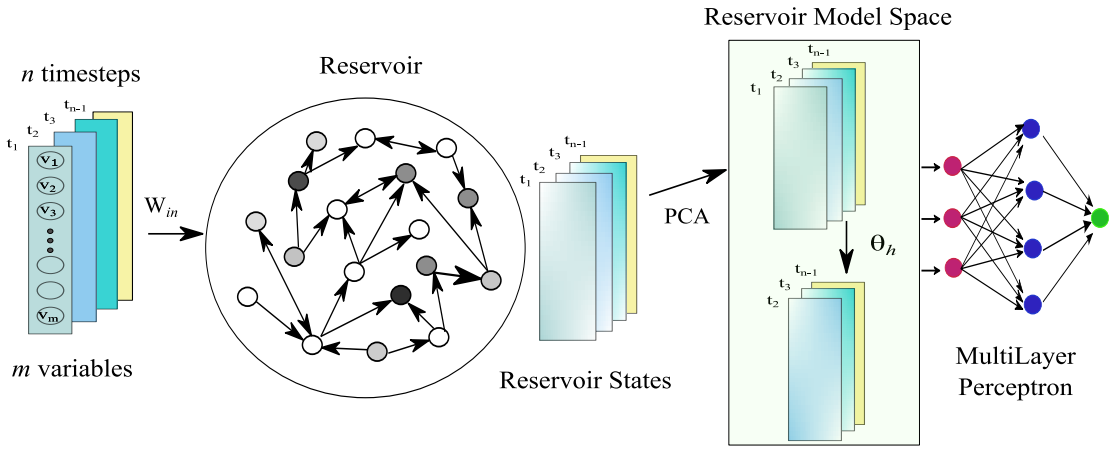

Figure 5.6: The overview of using Echo State Network for fNIRS data classification.

data. For example, a multivariate time series with Q time steps and M distinct variables per time step, after transformation, would be viewed as multivariate time series (having Q variables) with M time steps. Therefore, when the number of variables M is less than the number of time steps Q, dimension shuffle improves the efficiency of the model by requiring an order of magnitude less time to train. Previous work shows the dimension shuffle operation does not affect the performance of a model. Finally, the output of the global pooling layer after the convolution block and the output of the LSTM block are concatenated and passed to a softmax layer for classification.

Considering the size of our dataset, the optimal number of LSTM cells for our dataset was found via grid search over 4 distinct choices: 8, 16, 32, 64; while the number of filters of the FCN block was found via grid search over 16-32-16, 32-64-32, and 64-128-64, 128-256-128, with kernel sizes of 8, 5, and 3, respectively. Following previous work, we use the Adam optimizer, with an initial learning rate set to 1e-3 and the final learning rate set to 1e-4 to train all models. The models are trained for 50 epochs.

5.5.4 ESNs

The Echo State Network (ESN) is a family of recurrent neural network models with a strong architectural simplification. The connectivity and weights of hidden neurons in the recurrent neural network (called "reservoir") are kept fixed and randomly assigned. Only output weights are learned during training so that the network can produce specific temporal patterns. As such, ESN has an unrivaled training speed compared to other recurrent neural networks. Previous work has shown that ESNs can achieve excellent performance in many fields, and are an efficient solution for multivariate time-series classification [175, 176, 177, 178].

To improve classification accuracy by learning more powerful representations from the sequence of reservoir states, Chen *et al.* proposed a "model space" feature extraction approach by training a model for one-step-ahead prediction of the inputs, and then using the model parameters as features for classification. This approach has been successfully applied for multivariate time series classification and unsupervised EEG feature extraction [150, 151, 152]. Moreover, Bianchi *et al.* proposed a "reservoir model space" feature extraction approach, which consists of parameters from a model trained for one-step-ahead prediction of the future reservoir state, instead of the input. Their results show this approach can achieve superior classification accuracy on many multivariate time series datasets when comparing to state-of-the-art recurrent networks and time series kernels [179].

Therefore, in this work, we investigate the "reservoir model space" approaches for fNIRS data feature extraction. Figure 5.6 shows the overview of using this approach for feature extraction and classification. Specifically, we consider classification of fNIRS data consisting of M channels and observed for T time steps. The observation at time t is denoted as $x(t) \in \mathbb{R}^M$. We represent the multi-channel fNIRS data as a $T \times M$ matrix: $X = [x(1), ..., x(T)]^T$. For a echo state network with input weights W_{in} and recurrent

connections W_r (randomly generated and left untrained), the state-update equation is:

$$h(t) = f(W_{in}x(t) + W_r h(t-1)) \tag{5.7}$$

where $h(t)$ is the reservoir state at time t, which depends on its previous value $h(t-1)$ and the current input $x(t)$. $f(\cdot)$ is a nonlinear activation function. For input X, the sequence of the reservoir states generated over time is denoted as H ($H = [h(1), ..., h(T)]$). Therefore, reservoir states are a nonlinear high-dimensional representation of the input time series. Input data not linearly separable in the original space often become separable in the expanded space [180]. Moreover, due to the sparsely connected neurons, ESN can retain the historical information of a time series such that input time series with similar short-term history will produce similar reservoir states [181].

Then, the ESN is trained to perform one step-ahead prediction of each reservoir state:

$$h(t+1) = V_h h(t) + v_h \tag{5.8}$$

The parameters $\theta_h = \{V_h, v_h\}$ are learned by minimizing a ridge regression loss function. These parameters then becomes the representations for the input and used for classification. Also, since dimensionality reduction applied on top of $h(t)$ can enhance the representaions' generalization capability, we applied Principle Component Analysis (PCA) on $h(t)$ [179].

The performance of ESN can be influenced by the number of hidden neurons and the internal connectivity of the reservoir [152]. Therefore, in this work, we determine the optimal parameters for ESN for classifying driver cognitive load using fNIRS by investigating the effect of the number of hidden neurons and the internal connectivity of the reservoir on the classification accuracy. For features extracted using the ESN, we choose Multilayer Perceptron (MLP) as the classifier. MLPs have been widely used in

previous work and have shown high performance for fNIRS data classification. MLP is a feed-forward neural network with multiple fully-connected layers. Similarly, we use an MLP consisting of two hidden layers with the ReLU activation function. Each hidden layer has 200 units, and 100 units, respectively. We also implemented an optimizer using RMSprop with a learning rate of 0.01. The flow of using ESNs for fNIRS data classification is described in Algorithm 2.

Algorithm 2 ESNs for fNIRS data classification

1: **Input** Training data X_1, Test data X_2, reservoir weights W_{in} and recurrent connections W_r
2: Construct an ESN according to W_{in} and W_r
3: Get reservoir embedding H_1 of X_1 and obtain principle component H_p of H_1 by fitting a PCA P
4: Minimize the loss function of one step-head prediction of H_p using the reservoir space approach, and obtain learned parameters θ_h
5: Train a classifier F with learned parameters θ_h
6: Get reservoir embedding H_2 of X_2, and obtain principle component H_p^2 of H_2 by applying P
7: Minimize the loss function of one step-head prediction of H_p^2 using the reservoir space approach, and obtain learned parameters $\theta_h^{'}$.
8: Get the test accuracy on classifier F with learned parameters $\theta_h^{'}$

5.6 Statistical Comparisons of Machine Learning Models

The resampled paired t-test procedure is a popular method for comparing the performance of two machine learning models, however, this method has many drawbacks [182]. To correct the paired Student's t-test for the violation of the independence assumption from repeated k-fold cross-validation, Claude Nadeau and Yoshua Bengio proposed a "corrected resampled t-test" method for statistical comparisons between machine learning models [183]. This test is associated with a repeated estimation method (for example

holdout): in $i-th$ of the m iterations, a random data partition is conducted and the values for the scores $A(i)k1$ and $A(i)k2$ of compared classifiers $k1$ and $k2$, are obtained. This method has also been popular for comparing the performance of two models.

Moreover, in the work by Bouckaert and Frank [184], the authors argue that a test should have not only acceptable type 1 error and low type 2 error, but also high replicability. They recommend using 10×10-fold cross-validation with the Nadeau and Bengio correction to the paired Student t-test in order to achieve good replicability. Therefore, we decided to use 10×10-fold cross-validation with the corrected paired Student t-test in our work. Specifically, for test based on r-times of k-fold cross-validation, the test statistic is calculated by:

$$t = \frac{\frac{1}{k \times r} \sum_{i=1}^{k} \sum_{j=1}^{r} x_{ij}}{\sqrt{(\frac{1}{k \times r} + \frac{n_2}{n_1})\hat{\sigma}^2}}$$

where $\hat{\sigma}^2$ is the estimate of the variance: $\hat{\sigma}^2 = \frac{1}{k \times r - 1} \sum_{i=1}^{k} \sum_{j=1}^{r} (x_{ij} - m)^2$, m is the estimate of the mean: $m = \frac{1}{k \times r} \sum_{i=1}^{k} \sum_{j=1}^{r} x_{ij}$, n_1 is the number of instances used for training, and n_2 is the number of instanced used for testing.

5.7 Classification Results

We report the classification results achieved using CNNs, multivariate LSTM-FCNs, and ESNs. Moreover, to evaluate the effectiveness of these approaches, we also extract commonly-used hand-crafted features from fNIRS data and use MLP for classification (we report the classification results with other traditional classifiers in Appendix A). The average values of HbR and HbO and the slope over the whole window of all channels are used as hand-crafted features. Specifically, the classification results of using CNN were pre-trained using CAE and then fined-tuned. In addition, we compare the classification

Table 5.1: Parameter optimization table for CNNs for driver cognitive load classification. *SD* refers to the *single-task driving* condition.

Filter sizes	Width	*SD* vs. 2-back	*SD* vs. 1-back	*SD* vs. 0-back	Four-classes classification
7×2, 5×2	32, 16	71.61 ± 1.22	67.23 ± 2.03	65.80 ± 1.43	44.64 ± 1.82
7×3, 5×3	32, 16	70.25 ± 2.23	67.42 ± 1.45	63.23 ± 1.23	43.75 ± 2.06
7×2, 5×2, 3×2	**16, 16, 8**	$\mathbf{73.25\pm1.59}$	$\mathbf{68.75\pm1.04}$	$\mathbf{65.71\pm1.87}$	$\mathbf{47.21\pm3.52}$
7×3, 5×3, 3×3	16, 16, 8	71.92 ± 1.76	67.73 ± 1.66	64.67 ± 1.73	45.33 ± 2.47
7×2, 5×2, 5×2, 3×2	16, 16, 8, 8	70.20 ± 1.34	67.19 ± 1.59	63.08 ± 1.32	43.46 ± 2.28
7×3, 5×3, 5×3, 3×3	16, 16, 8, 8	68.35 ± 1.46	66.13 ± 1.86	62.33 ± 1.67	42.78 ± 2.05

results achieved when using only HbO, using only HbR, and using the combination of HbO and HbR as input with different classification methods.

We use 10-fold cross-validation to evaluate the classifiers' performance. Moreover, for features extracted using the ESN model, since the reservoir networks are randomly created, we take the impact of the reservoir's randomness into account by implementing each ESN 10 times according to specified parameters and comparing the results.

5.7.1 CNNs Results

Table 5.1 shows the classification accuracy for differentiating different cognitive load levels from fNIRS data with the fine-tuned CNN with unsupervised pre-training using the CAE. To determine the optimal architecture for the CNNs, Table 5.1 compares the classification accuracy achieved with CNNs consisting of different filter sizes and widths (all convolutional layers are followed by a max-pooling layer with filters of size 2×2). The accuracies are the mean accuracies of 10×10 cross-validation. We can see that the architecture of the CNNs can slightly affect the classification accuracy. Specifically, when the depth is 3, and the filter sizes are 7×2, 5×2, 3×2 with a width of 16, 16, 8, we achieved the highest classification accuracy for differentiating different cognitive load with fNIRS data. As expected, classifying 2-back against *single-task driving* achieved the best re-

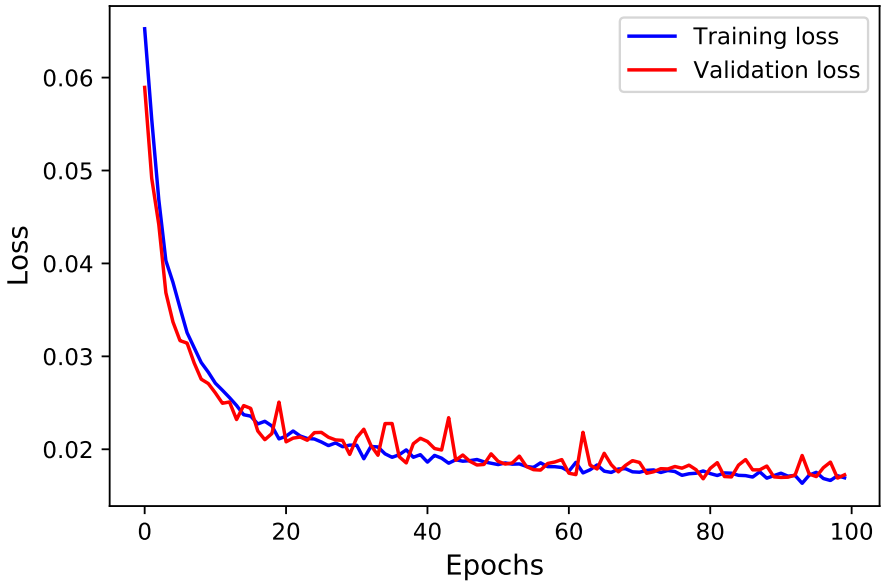

Figure 5.7: The mean squared error loss for training and validation sets of the CAE network with the optimal architecture, when classifying 2-back against *single-task driving*.

sults of 73.25% accuracy (precision = 74.16%, recall = 68.53%, F1-score = 71.14%), while classifying 1-back and 0-back against *single-task driving* achieved an accuracy of 68.75% (precision = 70.75%, recall = 62.90%, F1-score = 66.56%) and 65.71% (precision = 69.39%, recall = 59.26%, F1-score = 63.92%), respectively. For the four-class classification task (single-task driving vs. zero-back vs. one-back vs. two-back), we achieved an accuracy of 47.21% (chance accuracy 25%).

Furthermore, Figure 5.7 shows the training loss and validation loss for the CAE with the optimal architecture across 100 epochs for the task of classifying 2-back against *single-task driving*. It is clear that the validation loss and training loss were converged at around the 80th epoch. More importantly, they almost dropped simultaneously, indicating that the proposed training approach allows the model to learn good generalization capability without overfitting.

5.7.2 Multivariate LSTM-FCNs Results

Table 5.2 shows the classification accuracies for classifying different driver cognitive load using multivariate LSTM-FCNs. To determine the optimal parameters of the architecture, Table 5.2 compares the classification accuracy achieved with multivariate LSTM-FCNs consisting of different FCN filter sizes and different number of LSTM cells. The accuracies are the mean accuracies of 10×10 cross-validation. We can see that these parameters can slightly affect the classification results. Specifically, when the filter sizes are 32, 64, 32 for the FCN block, and the number of LSTM cells is 8, we achieved the highest classification accuracy for differentiating different cognitive load with fNIRS data. Classifying 2-back against *single-task driving* achieved the best results of 71.81% accuracy (precision = 74.05%, recall = 69.78%, F1-score = 69.52%), while classifying 1-back and 0-back against *single-task driving* achieved an accuracy of 67.87% (precision = 71.55%, recall = 64.33%, F1-score = 65.58%) and 66.76% (precision = 69.89%, recall = 62.23%, F1-score = 64.75%), respectively. For the four-class classification task (single-task driving vs. zero-back vs. one-back vs. two-back), we achieved an accuracy of 44.16%.

5.7.3 ESNs Results

Figure 5.8 shows the comparison results of fNIRS data classification accuracy when using ESNs for feature extraction, with different reservoir internal connectivity. For simplicity, we only show the classification accuracy for differentiating *2-back* vs. *single-task driving* here. The accuracy reported is the mean accuracy of 10-fold cross-validation with 10 repetitions, and the standard deviation of each point reflects the variation of the accuracy caused by the reservoir's randomness. We can see that the reservoir's internal connectivity only slightly changes the classification results, with the best classification accuracy achieved when the connectivity is around 0.3. Moreover, we can see that the variance

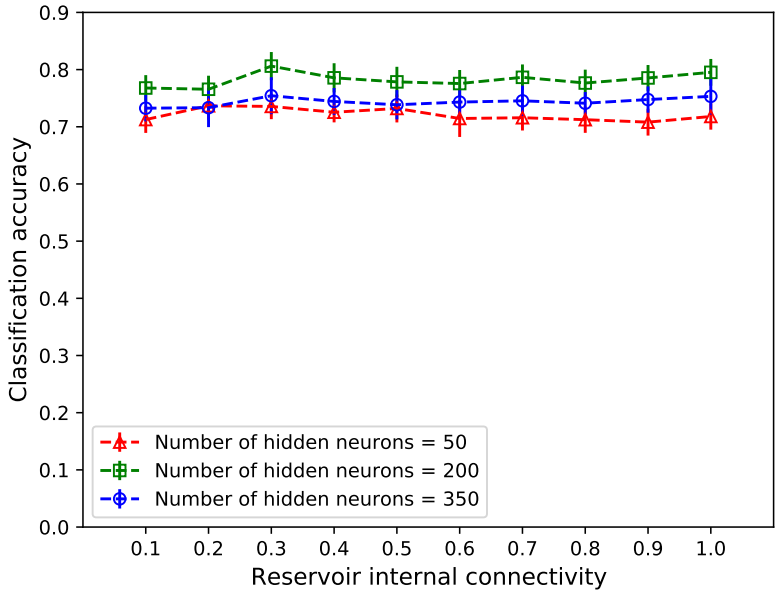

Figure 5.8: fNIRS data classification accuracy for *2-back* vs. *single-task driving* when using ESNs, with different reservoir internal connectivity. The accuracy reported represents the mean accuracy of the 10-fold cross-validation with 10 times repetitions.

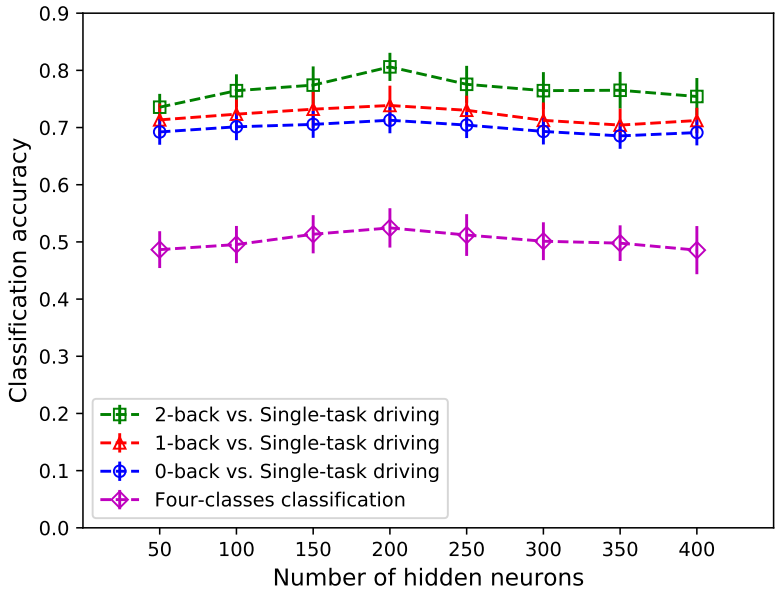

Figure 5.9: The impact of number of hidden neurons in ESNs on fNIRS data classification accuracies, when the internal connectivity is set to 0.3. The accuracies represent the mean accuracy of 10-fold cross-validation with 10 times repetition.

Table 5.2: Parameter optimization table for multivariate LSTM-FCNs for driver cognitive load classification.

FCN Filter Sizes	The N. of LSTM Cells	*SD* vs. 2-back	*SD* vs. 1-back	*SD* vs. 0-back	Four-classes classification
16, 32, 16	8	69.54 ± 3.01	65.78 ± 2.34	64.54 ± 2.76	41.25 ± 2.18
16, 32, 16	16	68.66 ± 2.75	66.91 ± 3.46	65.17 ± 2.38	42.50 ± 2.97
16, 32, 16	32	67.46 ± 3.27	66.01 ± 2.65	64.83 ± 2.64	43.28 ± 2.16
16, 32, 16	64	69.32 ± 2.62	65.67 ± 3.27	64.12 ± 3.73	43.76 ± 2.05
32, 64, 32	8	$\mathbf{71.81 \pm 2.39}$	$\mathbf{67.87 \pm 3.83}$	$\mathbf{66.76 \pm 3.96}$	$\mathbf{44.16 \pm 2.89}$
32,64, 32	16	70.23 ± 3.19	66.28 ± 3.72	64.92 ± 2.85	43.33 ± 1.97
32, 64, 32	32	70.54 ± 3.36	67.32 ± 2.63	65.29 ± 3.32	41.57 ± 2.38
32, 64, 32	64	68.67 ± 3.20	65.18 ± 3.02	65.67 ± 2.49	43.18 ± 2.09
64, 128, 64	8	69.77 ± 2.78	65.65 ± 3.28	66.26 ± 2.63	42.72 ± 2.21
64, 128, 32	16	68.59 ± 3.24	66.28 ± 2.94	65.82 ± 2.74	41.18 ± 2.04
64, 128, 64	32	68.42 ± 2.31	67.07 ± 2.63	66.56 ± 3.27	43.33 ± 2.16
64, 128, 64	64	67.32 ± 3.22	66.15 ± 3.61	64.18 ± 2.56	42.64 ± 1.94

of accuracy due to the randomness of echo state network randomness is small (around 3.0%), which is consistent with prior work [152]. As such, we can conclude that fNIRS data classification results based on ESNs are robust against the reservoir's randomness.

Figure 5.9 shows the impact of the number of hidden neurons in ESNs on fNIRS data classification accuracies, when the internal connectivity is set to 0.3. We can see that the classification accuracy first increases as the number of hidden neurons in the ESNs increase, and then decreases. The best classification results are achieved when the number of hidden neurons is 200. Specifically, classifying *2-back* against *single-task driving* achieved a mean accuracy of 80.61% (precision = 79.08%, recall = 81.80%, F1-score = 80.38%), while classifying *1-back* and *0-back* against *single-task driving* achieved a mean accuracy of 73.86% (precision = 74.16%, recall = 72.70%, F1-score = 73.26%) and 71.28% (precision = 72.54%, recall = 67.26%, F1-score = 69.60%), respectively. For the four-class classification task, we achieved an accuracy of 52.45%.

Table 5.3: Comparison of classification accuracy, precision, recall, and F1 score achieved by using hand-crafted features, while using only HbO, using only HbR, and using the combination of HbO and HbR.

		HbO	HbR	HbO+HbR
2-back v.s *SD*	Accuracy	64.85	63.89	62.94
	Precision	66.66	66.04	65.45
	Recall	56.72	57.45	58.18
	F1-score	61.26	61.36	61.45
1-back v.s *SD*	Accuracy	58.31	57.40	60.21
	Precision	58.99	58.24	60.45
	Recall	59.93	58.18	63.63
	F1-score	59.38	58.11	61.97
0-back v.s *SD*	Accuracy	59.26	56.49	55.58
	Precision	59.22	57.48	56.72
	Recall	59.99	56.36	54.54
	F1-score	59.43	56.85	55.58
Four-classes	Accuracy	37.32	36.67	37.94

5.7.4 Comparison Results with Different Inputs

Table 5.3, Table 5.4, Table 5.5, and Table 5.6 shows the classification accuracy, precision, recall, and F1-score for classifying different levels of driver cognitive load when using hand-crafted features, CNNs, multivariate LSTM-FCNs, and the ESNs respectively, while using only HbO, using only HbR, and using the combination of HbO and HbR as the input. From these tables, we can see that, in general, when using CNNs, multivariate LSTM-FCNs, and the ESNs, using the combination of HbO and HbR as the input achieved slightly better classification results than using only HbO or only HbR. However, when using hand-crafted features, for classifying *2-back* against *single-task driving* and *0-back* against *single-task driving*, using only HbO as the input achieved slightly better classification results than using only HbR or using the combination of HbO and HbR; while for classifying *1-back* against *single-task driving* and four-classes classification, using the combination of HbO and HbR as the input achieved slightly better classification

Table 5.4: Comparison of classification accuracy, precision, recall, and F1 score achieved by using CNNs, while using only HbO, using only HbR, and using the combination of HbO and HbR.

		HbO	HbR	HbO+HbR
2-back v.s *SD*	Accuracy	71.30	69.48	73.25
	Precision	74.17	73.08	74.16
	Recall	67.26	63.63	68.53
	F1-score	70.40	67.96	71.14
1-back v.s *SD*	Accuracy	66.57	65.75	68.75
	Precision	68.40	66.80	70.75
	Recall	62.18	58.54	62.90
	F1-score	65.16	62.52	66.56
0-back v.s *SD*	Accuracy	65.08	64.84	65.71
	Precision	68.71	66.36	69.39
	Recall	57.44	56.72	59.26
	F1-score	62.60	61.20	63.92
Four-classes	Accuracy	44.57	45.67	47.21

results than using only HbO or only HbR. These results suggest that deep learning models can effectively extract useful information from the combination of HbO and HbR, while the hand-crafted features from HbO and HbR could contain redundant information and reduce the model performance.

Moreover, Fig 6.9 shows the comparison of the best accuracies achieved by these approaches. We can see the ESNs achieved superior classification results for fNIRS-based driver cognitive load classification. Specifically, compared to the highest classification accuracy achieved using hand-crafted features, the ESN model improved the classification accuracy by 15.76%, 12.85% and 11.17% for classifying *2-back* against *single-task driving*, *1-back* against *single-task driving*, and *0-back* against *single-task driving*, respectively; while the classification accuracy for four-classes classification was improved by 14.51%. When compare to using CNNs for feature extraction, the ESN model improved the classification accuracy by 7.36%, 5.11% and 5.55% for classifying *2-back*

Table 5.5: Comparison of classification accuracy, precision, recall, and F1 score achieved by using multivariate LSTM-FCNs, while using only HbO, using only HbR, and using the combination of HbO and HbR.

		HbO	HbR	HbO+HbR
2-back v.s *SD*	Accuracy	69.24	68.18	71.81
	Precision	73.23	71.39	74.05
	Recall	69.33	71.33	69.78
	F1-score	68.60	68.29	69.52
1-back v.s *SD*	Accuracy	66.97	67.27	67.87
	Precision	70.88	70.69	71.55
	Recall	63.33	63.92	64.33
	F1-score	64.91	65.21	65.58
0-back v.s *SD*	Accuracy	66.06	65.45	66.76
	Precision	70.88	70.13	69.89
	Recall	59.33	58.28	62.23
	F1-score	62.64	61.91	64.75
Four-classes	Accuracy	43.33	43.76	44.16

against *single-task driving*, *1-back* against *single-task driving*, and *0-back* against *single-task driving*, respectively; while the classification accuracy for four-classes classification was improved by 5.24%. Compared to the highest classification accuracy achieved using multivariate LSTM-FCNs, the ESN model improved the classification accuracy by 8.80%, 5.99% and 5.52% for classifying *2-back* against *single-task driving*, *1-back* against *single-task driving*, and *0-back* against *single-task driving*, respectively; while the classification accuracy for four-classes classification was improved by 8.29%.

Furthermore, statistical tests results on the best classification accuracy achieved by different methods show that the ESN model outperformed CNNs for classifying *2-back* against *single-task driving* and *1-back* against *single-task driving* ($p < 0.05$, 10×10 cross-validation with a corrected paired Student t-test [184]), while there are no significant differences between the classification accuracy for classifying *0-back* against *single-task driving* and four-classes classification. Similarly, the ESN model outper-

Table 5.6: Comparison of classification accuracy, precision, recall, and F1 score achieved by using ESNs, while using only HbO, using only HbR, and using the combination of HbO and HbR.

		HbO	HbR	HbO+HbR
2-back v.s *SD*	Accuracy	78.70	77.80	80.61
	Precision	77.72	76.36	79.08
	Recall	81.81	81.58	81.67
	F1-score	79.68	78.97	80.38
1-back v.s *SD*	Accuracy	72.21	71.30	73.86
	Precision	74.70	74.16	74.82
	Recall	69.07	67.26	72.70
	F1-score	71.62	70.40	73.26
0-back v.s *SD*	Accuracy	69.48	68.52	71.28
	Precision	73.08	70.74	72.54
	Recall	63.63	62.90	67.26
	F1-score	67.96	66.59	69.60
Four-classes	Accuracy	50.12	49.78	52.45

formed multivariate LSTM-FCNs for classifying *2-back* against *single-task driving* and *1-back* against *single-task driving*, as well as four-classes classification ($p < 0.05$, 10×10 cross-validation with a corrected paired Student t-test [184]), while there are no significant differences between the classification accuracy for classifying *0-back* against *single-task driving*. When comparing to using hand-crafted features, CNNs, multivariate LSTM-FCNs, and ESNs all achieved significantly higher accuracy for all classification tasks ($p < 0.01$, 10×10 cross-validation with a corrected paired Student t-test [184]). These results suggest that the proposed ESN model can effectively extract useful temporal information for fNIRS data classification.

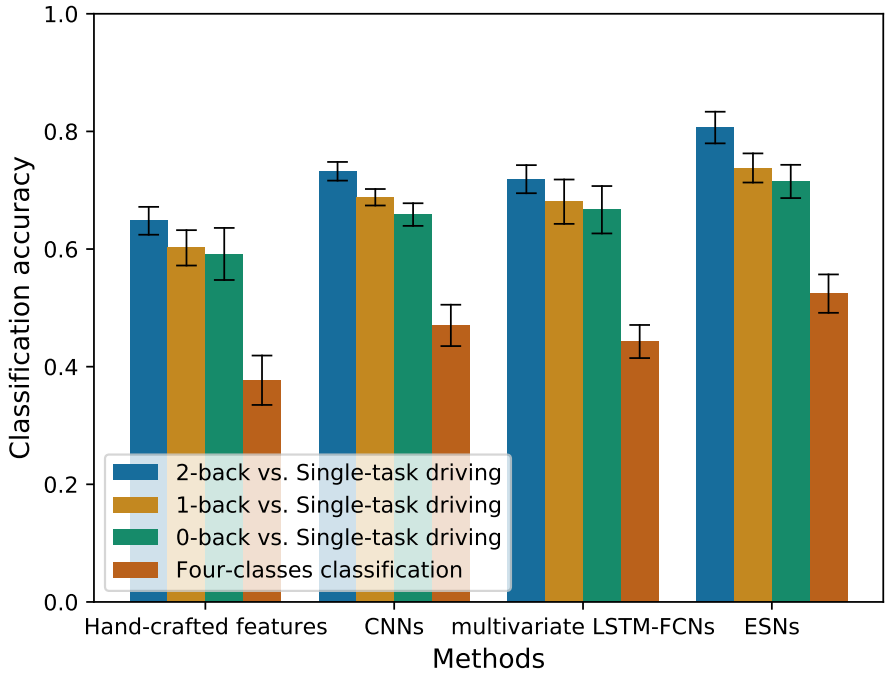

Figure 5.10: Comparison of classification accuracy achieved by using different methods.

5.8 Discussion

Physiological data has shown to be useful for measuring driver cognitive load non-intrusively and continuously. However, physiological data are not always entirely reliable [134, 163]. To improve robustness, brain-sensing can provide an additional objective measure of driver cognitive load level. In this work, we describe an advanced machine learning framework for driver cognitive load classification using fNIRS data. To collect an fNIRS data set with different driver cognitive load levels, we conducted a study in a driving simulator where participants were asked to perform an auditory-vocal working memory secondary-task (n-back). We then investigate advanced machine learning methods to extract useful features from fNIRS data for classification.

Previous research has shown the superiority of CNNs-based approaches for automatically extracting features from fNIRS data comparing to hand-crafted features. However,

a moving window method was often used in previous work to carefully pick a small segment from the original data as the input. While using the moving window method could result in better classification accuracy, this approach ignores the global temporal information and makes the results over-optimistic for deploying in real-world applications. Particularly, a small segment of the fNIRS data has limited capability to represent the cognitive process for measuring driver cognitive load. Therefore, we set out to investigate feature extraction methods from a long period of fNIRS data without window selection. Nevertheless, due to overfitting, the small sample sizes of fNIRS datasets make it challenging for the CNN-based method to fully extract temporal information from a long time series data [147].

As such, in this work, we investigate the application of CNN-based models and RNN-based models for extracting patterns from fNIRS data. Specifically, we compare the classification results achieved using CNNs, multivariate LSTM-FCNs, and ESNs. CNNs are pre-trained using CAE, which learns a compressed representation of the input by reconstructing the original input. After unsupervised pre-training, CAE can then be used for fine-tuning CNN in classification tasks. On the other hand, the multivariate LSTM-FCN mode is specially designed for multivariate time-series data and has received a lot of attention due to its superior performance over other models. At the same time, ESN has been proven an efficient solution for many multivariate time series data classification problems. Both multivariate LSTM-FCNs, and ESNs have not been explored for applying on fNIRS data. To the best of our knowledge, this is the first work to explore the application of multivariate LSTM-FCNs and ESNs for extracting temporal patterns from fNIRS data. Specifically, the multivariate LSTM-FCN model comprises a fully convolutional block and an LSTM block, which can strengthen the model's ability to capture the temporal patterns in data. Especially, a *squeeze and excite block* is applied after the first two convolutional blocks to incorporate the inter-correlation between multiple channels at each

time step [155]. The ESN model aims to perform a one-step-ahead prediction for each reservoir state, and learned output weights become the features. Our results show that CNNs, multivariate LSTM-FCNs, and ESNs are all suitable for fNIRS feature extraction, while ESNs achieved higher classification accuracy than CNNs and multivariate LSTM-FCNs for fNIRS-based driver cognitive load classification. Furthermore, the ESN model can be used in various fNIRS-based machine learning problems. Apart from the higher performance, compared to other RNNs, ESN is computationally efficient and has a fast training speed, which makes it useful for real-time fNIRS data classification. For future work, we will explore the application of ESNs in other fNIRS data classification tasks.

Our findings have important implications for building Advanced Driver Assistance Systems that can automatically measure drivers' cognitive load. For real-time applications, a classifier would be trained first with features extracted from the ESN model using labeled fNIRS data. Then, real-time fNIRS data from the driver would be processed and fed into the ESN model for feature extraction, which can then be used to predict the label of real-time data by the classifier. Furthermore, the predicted driver's cognitive load level can enable appropriate adaptive behavior of the in-vehicle technology and autonomy mechanisms, as well as adaptive user experiences. Moreover, our proposed approach can be used together with other non-invasive brain and body sensing techniques to improve the accuracy of assessing drivers' cognitive load. For example, we see promises for integrating fNIRS signals and EEG signals for a more accurate estimation of drivers' cognitive load, by building a deep ESN model that can extract both hemodynamic features from fNIRS signals and neuronal features from EEG signals.

5.9 Conclusion

In this paper, we investigated feature extraction methods for classifying driver cognitive load using fNIRS. The proposed ESN method can effectively extract temporal patterns from fNIRS data, and enables more accurate classification of driver cognitive load. This work builds a foundation for using fNIRS to measure driver cognitive load in real-world applications. Furthermore, the proposed ESN model method can be useful for other fNIRS-based machine learning tasks.

Chapter 6

Classifying Successful and Unsuccessful Rule Learning Processes Using fNIRS with CNNs, Multivariate LSTM-FCNs, and ESNs

In the previous chapter, we explored driver cognitive load classification and applied advanced machine learning approaches to extract spatial and temporal patterns from fNIRS data. The results suggested that Echo State Network (ESN) is particularly suitable for fNIRS data classification, and can achieve superior results than Convolutional Neural Networks (CNNs), and Short Term Memory Fully Convolutional Networks (LSTM-FCNs). In this chapter, we explore using fNIRS to classify successful and unsuccessful rule learning processes. We also investigate whether the proposed ESN model is generalizable across different tasks.

6.1 Introduction

Previous work in brain-computer interfaces for learning has been focusing on using brain data to measure learners' cognitive load and attention level [185, 186, 187, 188]. While

these cognitive states have shown to play an important role in success during learning tasks, only a few research has explored the underlying cognitive mechanisms of the induction processes [189, 190, 191]. Therefore, in this paper, we focus on detecting the induction process during learning using brain sensing and investigating the relationship between learners' brain data and learning outcomes. Previous research has detected states similar to induction using neuroimaging techniques. Using fMRI, Strange *et al.* concluded that the aPFC is highly activated during abstract rule learning and less activated as task performance improves [190]. Savage *et al.* used PET and reported activation in the PFC during new learning but not during automatic performances or expert behavior [192]. Recently, the use of functional near-infrared spectroscopy (fNIRS) has received a focus from researchers in brain-computer interfaces because of its promise for detecting a user's cognitive state in more ecologically valid studies. fNIRS devices are relatively inexpensive, portable, and comfortable [47], and thus we anticipate them becoming realistic for use in learning environments in the future. fNIRS emits near-infrared light into the brain, and the light returned to the surface is measured and used to calculate oxygenation in the blood. This calculation reflects brain activity in that particular area. Moreover, prior work has shown the potential of using fNIRS data to identify brain activation related to expertise development [193]. These studies indicate that fNIRS brain data collected from the aPFC would be valuable in identifying cognitive states that are predictive of robust learning.

The goal of this work is to explore the feasibility of using fNIRS to detect cognitive states during the induction processes associated with positive and negative learning outcomes. Specifically, we collect an fNIRS dataset during a rule learning task described in the work of Strange *et al.* [190]. We chose the rule learning task to build a parallel between low-level cognitive structures that are well-studied within cognitive science research and the induction and refinement processes in the learning domain. The task

utilizes a modified artificial grammar paradigm to elicit an explicit abstract rule induction process, which allows for investigating the processing of rule-based regularities in a controlled way. During the task, participants were required to categorize letter strings as 'grammatical' or 'ungrammatical' according to a pre-defined rule through trial-by-trial feedback. As such, both attention and working memory processes can affect participants' performance on the task. Moreover, cognitive processes that are likely involved during the task include: (1) pattern extraction; (2) model building; and (3) retrieval or recognition processes [191]. These cognitive processes also appear during the induction and refinement processes in the learning domain, such as learning from examples, generalization, discrimination, categorization, and schema induction [194]. Then, to unravel the underlying brain activation pattern that leads to positive and negative outcomes, we explore advanced machine learning techniques to differentiate between successful and unsuccessful rule learning processes using fNIRS data.

We compare the classification results of CNNs, multivariate LSTM-FCNs, and ESNs on classifying successful and unsuccessful rule learning processes using fNIRS data. The experimental results confirm that both ESNs and multivariate LSTM-FCNs are suitable for fNIRS data classification, while ESNs achieved superior classification accuracy than multivariate LSTM-FCNs and CNNs. Furthermore, to improve the transparency of the ENS models, we visualize the heat maps of hidden neuron activations of the ESN model for successful and unsuccessful rule learning sessions. The visualization verifies that the ESN model can obtain abundant discriminative features and significant temporal patterns during successful and unsuccessful rule learning processes.

The main contributions of this work can be summarized as:

- We propose to use fNIRS for identifying brain activation patterns during induction processes that lead to positive and negative learning outcomes.
- We describe a study in which we collected fNIRS brain data during a rule-learning

task. The dataset contains instances of successful and unsuccessful rule learning sessions. We show that there are differences in frontal lobe blood oxygenation patterns between successful and unsuccessful rule learning sessions.

- We describe the application of CNNs, ESNs, and multivariate LSTM-FCNs for fNIRS data classification and compare their results. We show that ESNs achieved superior classification accuracy than multivariate LSTM-FCNs and CNNs for classifying successful and unsuccessful rule learning sessions using fNIRS data.
- We further validate the ability of ESNs for extracting discriminate temporal patterns from fNIRS through model visualization.

6.2 Background

6.2.1 Brain Sensing During Learning

Previous work has explored using brain-sensing techniques to identify cognitive states that are important for learning, including working memory load, attention levels, and emotional change. These measured cognitive states can then be used to build learner models, which can adapt the learning system to answer the needs, objectives, and interests of the learner. Spuler *et al.* explored predicting math-related cognitive workload by using EEG data. They achieved an average classification accuracy of 56% for differentiating arithmetic problems into three difficulty categories [186]. Walter *et al.* presented an EEG-based arithmetic learning environment, which can detect the users' workload and adapt the learning materials accordingly [187]. Szafir *et al.* designed adaptive agents that measure student attention in real-time by using EEG, and re-engage the student following drops in their attention by using verbal and nonverbal cues [195]. Their results show an adaptive robotic agent can improve student performance compared to the nonadaptive and random adaptive conditions. Further, Szafir *et al.* presented a novel computer-based

education system based on the technique of adaptive content review. By measuring students' attention level by EEG and students will provide students with the contexts that they had the lowest average attention levels, their results show the adaptive review technology can improve student recall ability [196]. Shen et al developed a learning system using EEG that can recognize students' emotional change and provide emotion-aware content recommendations. By applying the Support Vector Machine, they achieved an accuracy of 86.3% for classifying learners' four emotional states by using the EEG brainwaves together with other peripheral physiological data. Moreover, their results show the emotional-aware content recommendation could greatly improve the performance of the e-Learning system and lead to enhanced user satisfaction [197]. Mills *et al.* assessed the relationship between sessions' difficulty level in an intelligent system and EEG-based estimate of students' cognitive load. Their results show that the EEG-based measure of students' cognitive load is correlated with the difficulty level of the learning task, as well as the learning performance [198].

These studies suggest that it is feasible to use brain data as an unobtrusive measure of learners' cognitive states for adapting learning content. However, there is little we know about the underlying cognitive mechanisms during the induction process and their relationship with the learning outcome. In this paper, we aim to explore the feasibility of using brain-sensing for identifying brain activation patterns during the induction process that are associated with positive and negative learning outcomes.

6.2.2 Induction During Learning

Strange *et al.* used fMRI to measure learning-dependent neural responses during an explicit rule learning task [190]. Their results show that the anterior prefrontal cortex is highly activated during abstract rule learning and less activated as task performance improves. Savage *et al.* used PET and reported activation in the prefrontal cortex during

new learning but not during automatic performances, or expert behavior [192]. Skrandies *et al.* investigated the change of evoked EEG frequencies while participants were learning mathematical rules. Their results show that there a significant relationship between successful learning divisibility rules and the changes in EEG frequencies over frontal and centro-parietal scalp areas of the right hemisphere [199]. However, fMRI and PET are often prohibitively expensive and require restrictions on the study participant that are not reasonable for use in real-world learning environments. EEG has been the main technology used in education research due to its low cost, portability, and high temporal resolution. However, EEG has a limited spatial resolution. fNIRS is non-invasive, affordable, and portable [47]. Compare to fMRI, fNIRS is a more convenient and more affordable technology. Compared to EEG, it has a higher spatial resolution, is easy to set up, and robust to noise [78]. Moreover, most fNIRS research focuses on the anterior prefrontal cortex, including Brodmann area 10 (BA10), which lies behind the forehead. Indeed, Leff *et al.* found significant changes in prefrontal cortex activation with expertise development using fNIRS [193]. These studies indicate that fNIRS brain data collected from the anterior prefrontal cortex during learning would be valuable in identifying induction during learning.

6.3 Data collection

In this experiment, we aim to collect brain signals as a participant learns a new rule in a highly controlled, validated task. The changes that occur during this abstract task have relevance to the process a student would go through as they learn a new method or topic.

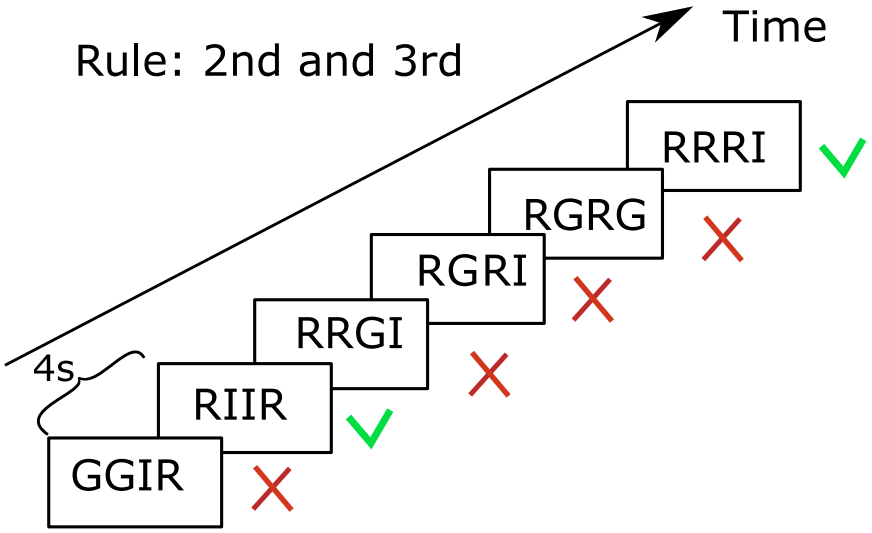

Figure 6.1: Illustration of the abstract rule learning task. A sample rule and sample stimuli are shown. The sample rule refers to the presence of a repeated letter in the second and third position of each string. The tick or cross next to the string indicates if it follows the current rule.

6.3.1 fNIRS Recording

The fNIRS data are acquired using was a multichannel frequency domain Imagent from ISS Inc. Two probes are placed on the forehead to measure the two hemispheres of the anterior prefrontal cortex. The source-detector distances are 3 cm or 0.8 cm. Each source emits two near-infrared wavelengths (690 nm and 830 nm) to detect and differentiate between oxygenated and deoxygenated hemoglobin. The sampling rate is 6.649 Hz. The sensors are kept in place using headbands, which can also reduce light interference.

6.3.2 Abstract Rule Learning Task

We adopted the rule learning task designed by Strange *et al.* [190]. During the task, participants are required to perform explicit abstract rule induction. In each section, there is a pre-specified rule, which is based on the position of a repeated letter in four-letter

strings. Participants are instructed to learn the rule over the course of 20 trials. In each trial, participants are presented with a string of four letters in upper case on the screen and asked to press 'A' on the keyboard if the string follows the current abstract rule, and press 'L' on the keyboard if the string does not follow the current abstract rule (Figure 6.1). Each trial lasts for 4 seconds. Feedback will be presented on the screen after the participant's response in each trial to indicate if their answer is correct or wrong. Different from the experiment in Strange *et al.* [190], where rules from all sessions are easy to learn over the 20 trials, we include sessions where the rules are more difficult to learn in our experiment. For example, as illustrated in Fig 6.1, in this session, in order to acquire the rule of a repeated letter in the second and third position of a string, participants need to remember the feedback they receive at multiple strings (e.g., the feedback they receive for the second string and sixth string) to induct the actual rule. We conducted a pilot study with four participants to ensure that the task can obtain variance in learning success.

6.3.3 Participants

The study included 14 healthy volunteers (nine males) between the ages of 18 and 41. One participant's data are removed from analysis due to unstable fNIRS signals.

6.3.4 Design and Procedure

Before brain-sensing, participants are trained on the task by giving two practice sections. Researchers will then confirm with participants whether they understand the task. Ensuring they fully understand the task, participants are then equipped with the fNIRS sensors on their forehead and start the experiment. The experiment consisted of 11 sections. Each section may or may not follow the same rule or use the same characters as the previous section. In between each section, there is a rest period during which the strings 'AAAA'

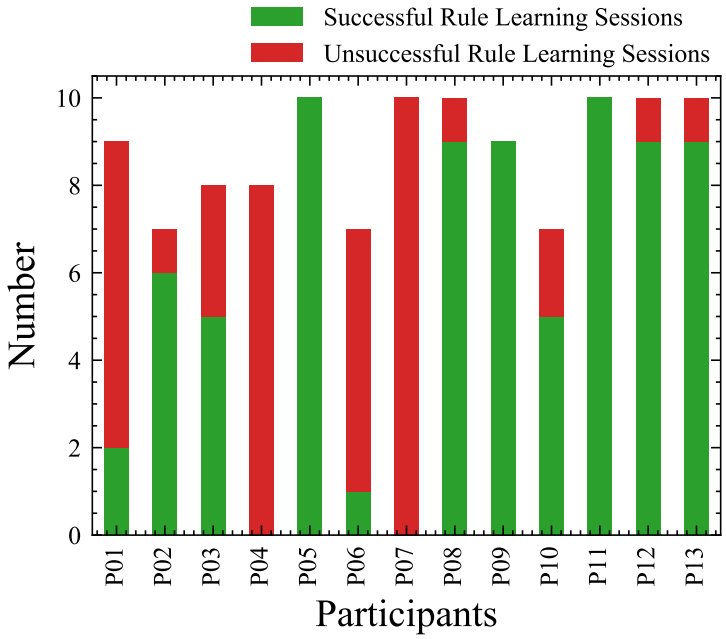

Figure 6.2: The number of successful rule learning sessions and unsuccessful rule learning sessions from each participant.

or 'LLLL' are presented (five of each). Participants are required to respond by pressing 'A' and 'L' on the keyboard, respectively.

6.4 Dataset Curation

Based on the fNIRS data collected during the study and participants' performance, we built the dataset for investigating the classification of successful and unsuccessful rule learning processes.

6.4.1 Dataset Labeling

The number of correct responses during the rule learning session can reflect if the participant learned the rule or not. In addition to the total number of correct responses across the

whole session, the responses to the trials near the end are also important to determine the success of rule learning. Therefore, to ensure that we accurately provide the ground truth for data labeling, We calculated the total number of correct responses across the 20 trials as well as the number of correct responses in the last 5 trials for each session. We then label sessions as successful rule learning sessions only if participants achieved at least 75% accuracy across the 20 trials and achieved 100% accuracy for the last 5 trials. Similarly, Sessions were only labeled as unsuccessful rule learning session if participants achieved less than 75% accuracy during the 20 trials and made at least two errors in the last 5 trials (less than 80% accuracy). All other sessions that do not belong to these two classes were ignored for this analysis. Fig 6.2 shows the number of successful rule learning sessions and unsuccessful rule learning sessions from each participant. Due to the nature of the task, the number of successful and unsuccessful rule learning sessions varied across participants. From Fig 6.2, we can see that some participants successfully learned the rule during most sessions in the experiment (e.g., P05 successfully learned the rule for 10 out of 11 sessions), while some other participants did not learn the rule for most sessions (e.g., P07 did not learn the rule for 10 out of 11 sessions). 8 out of 13 participants experienced a mix of successful and unsuccessful rule learning sessions. Across all participants, the dataset contains 75 successful rule learning sessions and 40 unsuccessful rule learning sessions.

6.4.2 Behavioral Data

Figure 6.3 shows the average percentage of correct response over the 20 exemplars for all successful rule learning sessions and all unsuccessful rule learning sessions. As expected, the performance of successful rule learning sessions improves over trials, reaching 100% correct by the end of each session; while the performance of unsuccessful rule learning sessions fluctuates over the 20 trials, indicating participants did not learn the current rule.

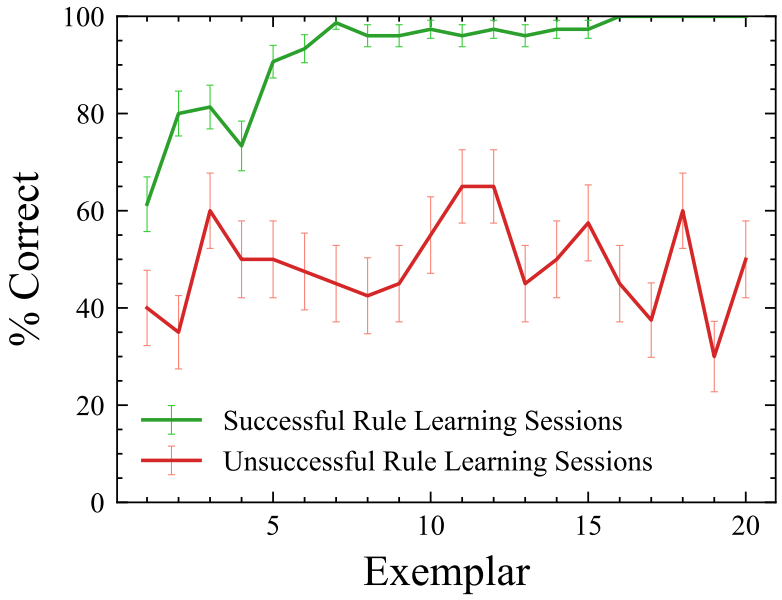

Figure 6.3: The average performance for all successful rule learning sessions (in green) and for all unsuccessful rule learning sessions (in red). The figure shows the average percentage of correct responses and standard error over the 20 exemplars.

Furthermore, Figure 6.4 shows the average response time over the 20 exemplars for all successful rule learning sessions and all unsuccessful rule learning sessions. We can see that the response time for the first exemplar for both successful and unsuccessful rule learning sessions is longer than the others. This is understandable since participants were forced to guess the rule at the beginning of each session. Over the trials, the response time for the successful rule learning sessions decreases, while the response time of unsuccessful rule learning sessions fluctuates over the 20 trials. In general, the response time of each trial during the unsuccessful rule learning sessions is longer than successful rule learning sessions. These confirm that during the successful rule learning session, participants engaged in successful pattern extraction and model building process, and were following the rule after figuring out the rule; while during the unsuccessful rule learning session, participants were struggling with extracting the rule.

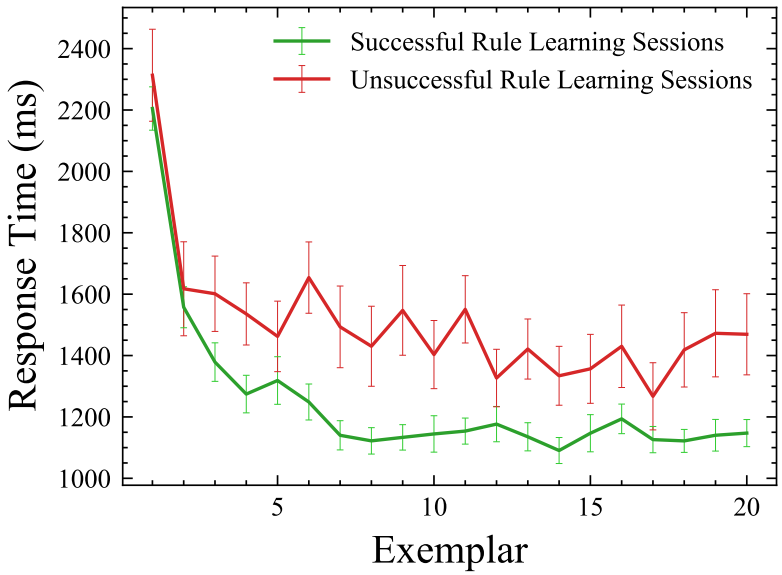

Figure 6.4: The average response time for all successful rule learning sessions (in green) and for all unsuccessful rule learning sessions (in red). The figure shows the average response time and standard error over the 20 exemplars.

6.4.3 fNIRS Dataset

The dataset consists of fNIRS data of six channels, from 13 participants. Since the two short-separation channels (0.8cm) contain mostly noise, we only analyze fNIRS signals from the six long-separation channels. Each sample consists of data in an 80-second session (4 seconds for each trial, there are 20 trials for each session). There are a total of 75 samples for successful rule learning sessions and 40 samples for unsuccessful rule learning sessions.

Dataset Preprocessing

To enhance signal quality, preprocessing is usually required to remove biological and technical artifacts from fNIRS data [114]. We followed typical preprocessing techniques [115]: we first used a band-pass filter with a high pass value of 0.02 Hz and a low pass value of 0.5 Hz to remove the physiological noise and the instrumental noise; we then

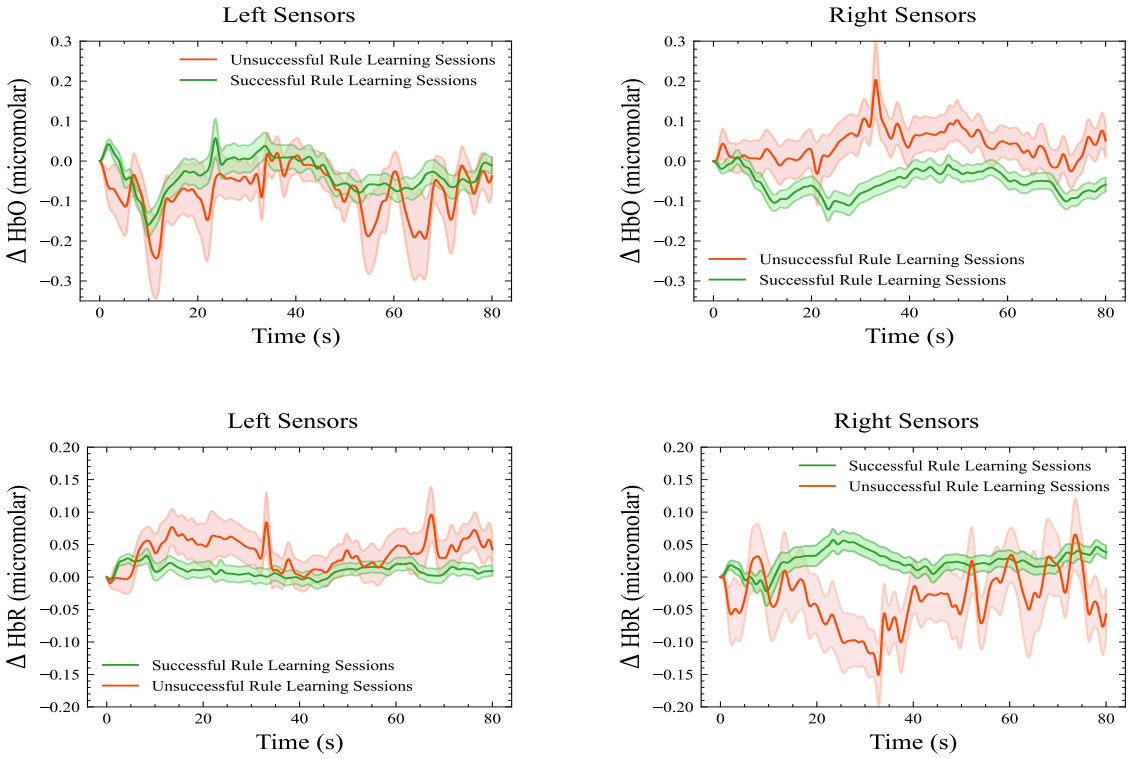

Figure 6.5: Variation of the HbO and HbR concentration for successful and unsuccessful rule learning sessions. The figures show the mean (averaged across all long channels and all participants) and standard error over each condition. Shaded areas represent the standard error.

converted the raw light intensity data to HbO and HbR values using the Modified Beer-Lambert Law; finally, we applied the correlation-based signal improvement (CBSI) to reduce the motion artifacts [170]. All preprocessing was completed in MATLAB using HomER [116].

Dataset Overview

Figure 6.5 shows the block averages of changes in HbO (red) and HbR (blue) across all participants for successful and unsuccessful rule learning sessions. Specifically, we calculated the folded average of all long-separation channels on the left side of the head and all long separation channels on the right side of the head separately. The changes in HbO

and HbR were calculated by subtracting the corresponding value of the starting point for each trial. From Fig. 6.5, we can see that for successful rule learning sessions, on both sides of the brain, following neural activation at the beginning, there is a decrease in HbO due to the metabolic consumption of oxygen around 3s to 10s. After that, there is a steady increase during 10s to 30s in HbO on the left side of the brain. For unsuccessful rule learning sessions, there is a significant increase in HbO around 20s to 35s on the right side of the brain, followed by a decrease. Similarly, we can see that there is a significant decrease in HbR around 15s to 35s on the right side of the brain, and followed by an increase. In general, the value of HbO on the right side of the brain is lower during successful rule learning sessions than unsuccessful rule learning sessions. Moreover, the changes of both HbO and HbR are comparatively more stable during successful rule learning sessions than unsuccessful rule learning sessions. We can see that for unsuccessful rule learning sessions, during the last 5 trials of the session (60s to 80s), there are frequent increases and decreases for both HbO on the left side of the brain and HbR on the right side of the brain. These indicate that during the successful rule learning session, participants were engaged in abstract rule learning processes in the beginning and then switched to follow the rule after successfully acquired the rule; while during unsuccessful rule learning sessions, participants were continuously engaging in abstract rule learning processes. This is consistent with previous findings, which suggest activation in the prefrontal area, especially on the right hemisphere, during abstract rule learning and less activated as task performance improves [190, 193, 199].

6.5 Classification Methods

We investigate the application of CNNs, ESNs, and multivariate LSTM-FCNs on fNIRS data for classifying successful and unsuccessful rule learning processes.

6.5.1 Input

For each sample, the HbO and HbR from six channels in the 80-second period are used as the input for both ESNs and multivariate LSTM-FCNs. Since the sampling rate was 6.648 Hz, the length of the data is 532. Data from each channel is normalized.

6.5.2 CNNs

The same as the previous chapter, we determine the best architecture of CNNs by investigating the effect of filter sizes, as well as depth and width on the classification accuracy. CNNs are pre-trained using CAE and fine-tuned in classification tasks. We add two fully connected layers and output neurons with the rectified linear unit (ReLU) activation function. Each layer has 200 units, and 100 units, respectively. We implemented an optimizer using RMSprop with a learning rate of 0.01.

6.5.3 Multivariate LSTM-FCNs

The same as the previous chapter, considering the size of our dataset, the optimal number of LSTM cells for our dataset was found via grid search over 4 distinct choices: 8, 16, 32, 64; while the number of filters of the FCN block was found via grid search over 16-32-16, 32-64-32, and 64-128-64, 128-256-128, with kernel sizes of 8, 5, and 3, respectively. Following previous work, we use the Adam optimizer, with an initial learning rate set to 1e-3 and the final learning rate set to 1e-4 to train all models. The models are trained for 50 epochs.

6.5.4 ESNs

The same as the previous chapter, we determine the optimal parameters for ESNs by investigating the effect of the number of hidden neurons and the internal connectivity of

Table 6.1: Parameter optimization table for CNNs for classifying successful rule learning sessions and unsuccessful rule learning sessions.

Filter sizes	Width	Accuracy	F1-score
7×2, 5×2	32, 16	72.72 ± 4.12	68.57 ± 3.75
7×3, 5×3	32, 16	71.46 ± 5.16	66.32 ± 4.46
7×2, 5×2, 3×2	16, 16, 8	74.27 ± 3.37	68.95 ± 3.78
7×3, 5×3, 3×3	**16, 16, 8**	**76.36 ± 4.09**	**70.86 ± 5.32**
7×2, 5×2, 5×2, 3×2	16, 16, 8, 8	71.21 ± 4.12	66.28 ± 4.56
7×3, 5×3, 5×3, 3×3	16, 16, 8, 8	70.03 ± 4.54	65.76 ± 5.51

the reservoir on the classification results. For the architecture of MLP, we use an MLP consisting of two hidden layers with the ReLU activation function. Each hidden layer has 200 units, and 100 units, respectively. We also implemented an optimizer using Adam with a learning rate of 1e-3. Furthermore, to explore the interpretability of the ESN models, we visualize the reservoir state sequences generated by successful rule learning sessions and unsuccessful rule learning sessions.

6.6 Classification Results

We report the classification results achieved using CNNs, multivariate LSTM-FCNs, and ESNs. Specifically, we use 10-fold cross-validation with 10 times of repetition to evaluate the models' performance. Moreover, since the dataset is unbalanced (there are more successful rule learning sessions than unsuccessful rule learning sessions), we used random oversampling during training to avoid biases towards one class. We report the classification accuracy as well as the F1 score for all models.

Table 6.2: Parameter optimization table for multivariate LSTM-FCNs for classifying successful rule learning sessions and unsuccessful rule learning sessions.

FCN Filter Sizes		The Number of LSTM Cells			
		8	16	32	64
16, 32, 16	Accuracy	76.74 ± 3.26	76.59 ± 3.78	78.86 ± 3.96	75.25 ± 4.42
	F1 Score	73.49 ± 3.65	75.08 ± 3.97	77.22 ± 4.09	78.10 ± 4.07
32, 64, 32	Accuracy	81.81 ± 3.34	78.33 ± 4.15	81.81 ± 4.66	80.75 ± 3.90
	F1 Score	79.84 ± 3.51	75.87 ± 4.60	80.58 ± 4.87	77.59 ± 4.39
64, 128, 64	Accuracy	82.65 ± 3.14	$\mathbf{83.48 \pm 1.92}$	82.24 ± 2.74	80.83 ± 2.39
	F1 Score	80.08 ± 3.47	$\mathbf{80.49 \pm 2.62}$	79.34 ± 3.63	78.46 ± 2.64
128, 256, 128	Accuracy	81.66 ± 2.70	82.65 ± 2.67	81.74 ± 2.83	82.50 ± 2.56
	F1 Score	78.44 ± 3.33	79.25 ± 3.47	78.50 ± 3.31	79.96 ± 2.71

6.6.1 CNNs Results

Table 6.1 shows the classification accuracy and F1-score for classifying successful and unsuccessful rule learning processes using fNIRS data with the fine-tuned CNN. To determine the optimal architecture for the CNNs, Table 6.1 compares the classification accuracy achieved with CNNs consisting of different filter sizes and widths (all convolutional layers are followed by a max-pooling layer with filters of size 2×2). The accuracies are the mean accuracies of 10×10 cross-validation. We can see that the architecture of the CNNs can greatly affect the classification results. Specifically, when the depth is 3, and the filter sizes are 7×3, 5×3, 3×3 with a width of 16, 16, 8, we achieved the highest classification accuracy with an average accuracy of 76.36% and an average F1-score of 70.86%.

6.6.2 Multivariate LSTM-FCNs Results

Table 6.2 shows the classification accuracies and F1-scores for classifying successful and unsuccessful rule learning processes using multivariate LSTM-FCNs. The accuracy and F1-score are the averages of 10×10 cross-validation. To determine the optimal parameters of the architecture, Table 6.2 compares the classification accuracy and F1-score

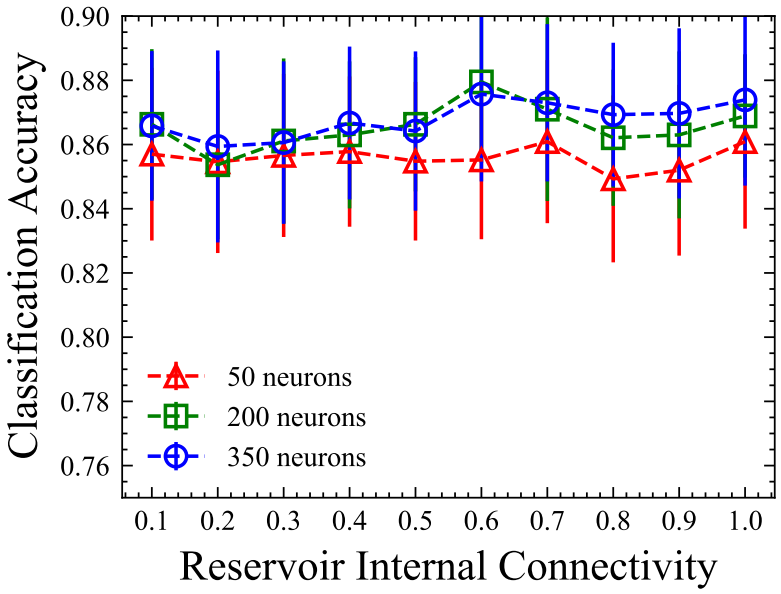

Figure 6.6: fNIRS data classification accuracy for classifying successful and unsuccessful rule learning processes using ESNs with different number of hidden neurons, and different reservoir internal connectivity. The accuracy reported represents the mean accuracy of the 10-fold cross-validation with 10 times repetitions.

achieved with multivariate LSTM-FCNs consisting of different FCN filter sizes and a different number of LSTM cells. We can see that these parameters can affect the classification results. In general, when the number of LSTM cells is fixed, models with FCN filter sizes 64-128-64 achieved better results than models with other filter sizes. Specifically, when the filter sizes are 64, 128, 64 for the FCN block, and the number of LSTM cells is 32, we achieved the best classification results with an average accuracy of 83.48% and an average F1-score of 80.49%.

6.6.3 ESNs Results

Figure 6.6 shows the impact of hidden neurons number and reservoir internal connectivity on fNIRS data classification accuracy when using ESNs. Similarly, Fig 6.7 show the F1-score for classifying successful and unsuccessful rule learning processes using ESNs

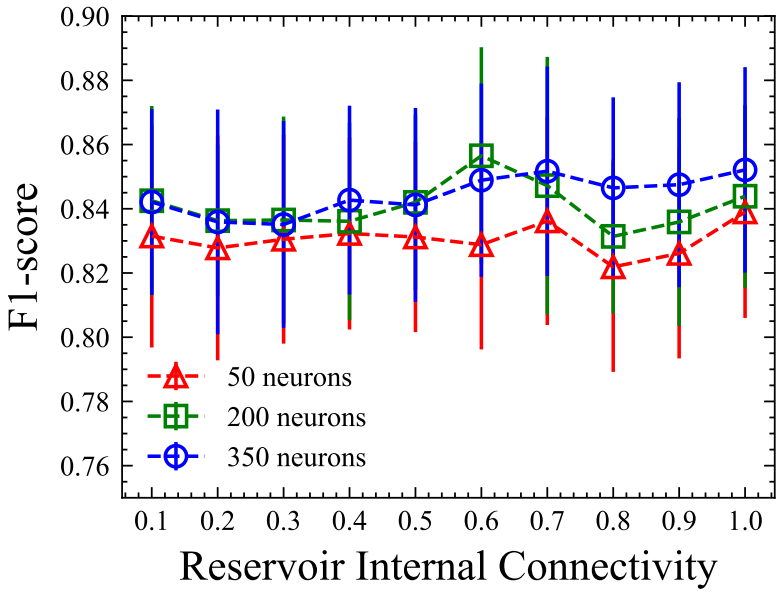

Figure 6.7: The F1-score for classifying successful and unsuccessful rule learning processes using ESNs with different number of hidden neurons, and different reservoir internal connectivity. The F1-score reported represents the mean accuracy of the 10-fold cross-validation with 10 times repetitions

with a different number of hidden neurons, and different reservoir internal connectivity. We can see that the hidden neurons number and the reservoir's internal connectivity can slightly change the classification results. In general, ESNs with 200 neurons and 350 neurons achieved better classification results than ESNs with 50 neurons. while ESNs with 200 neurons and ESNs with 350 neurons achieved similar classification accuracy. As shown in Fig 6.6, setting the number of hidden neurons to be 200 is enough to achieve good results on the dataset. When increasing the internal connectivity, the accuracy and F1-score are relatively stable, with a slight increase at the beginning followed by a slight decrease. Specifically when the internal connectivity is set to 0.6, and the number of hidden neurons is set to 200, we achieved an accuracy of 87.95% and an F1-score of 85.64%. When compared to the best classification results achieved using Multivariate LSTM-FCNs, the ESNs improved the classification accuracy by 4.47% and improved the

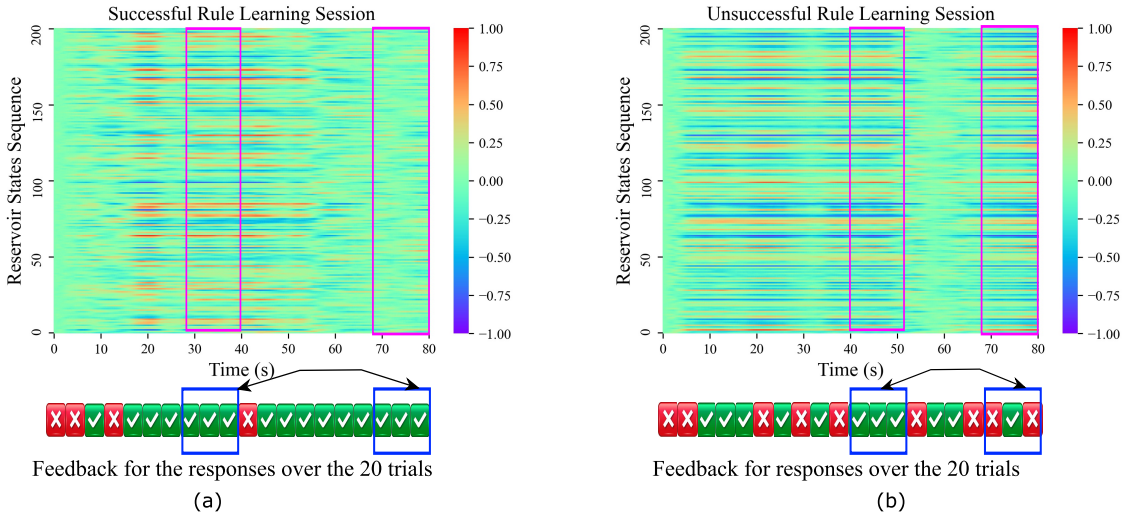

Figure 6.8: The heat map of the corresponding reservoir state sequence for a successful rule learning sessions and an unsuccessful rule learning session. There are three distinct phases for the successful rule learning session, while there are repetitive patterns for the unsuccessful rule learning session. Moreover, in (a), it shows that the reservoir sequence from 28s to 40s and the sequence from 68s to 80s are distinctively different, even though they correspond to the same feedback sequences; while in (b), it shows that the reservoir sequence from 40s to 52s and the sequence from 68s to 80s are similar, even though they correspond to different feedback sequences.

F1-score by 5.15%.

For the visualization analysis, Fig 6.8(a) shows the corresponding reservoir states sequence of a successful rule learning session, along with the feedback for the responses over the 20 trials. Similarly, Fig 6.8(b) hows the corresponding reservoir states sequence of an unsuccessful rule learning session, along with the feedback for the responses over the 20 trials. We can see that there are significant differences between these two reservoir state sequences. Especially, for the successful rule learning session, there are three distinctive phases during the whole session: approximately from 0s to 15s, then from 15s to 50s, and finally from 50s to 80s). However, there are no distinctive phases for the unsuccessful rule learning session, instead, there are similar patterns during the whole session. For example, reservoir states during 40s to 50s are similar to reservoir states during 70s

to 80s. This is in line with the cognitive processes that were likely involved during these two examples: during the successful rule learning session, the participant was likely first engaging in the pattern extraction process, then moving on to the model building process, and finally switching to the retrieval or recognition processes; however, during the successful rule learning session, the participant was likely repetitively engaging in the pattern extraction and model building process. Moreover, we can see that the temporal patterns extracted by ESN are not a direct mapping of the feedback participants receive about the correctness of their responses. For example, in (a), we can see that the reservoir sequence from 28s to 40s and the sequence from 68s to 80s are distinctively different, even though they correspond to the same feedback sequences; while in (b), we can see that the reservoir sequence from 40s to 52s and the sequence from 68s to 80s are similar, even though they correspond to different feedback sequences. These suggest that the ESN model can extract distinct temporal patterns for successful and unsuccessful rule learning sessions. Moreover, the temporal patterns extracted by ESN from the fNIRS data can reflect users' underlying cognitive states, rather than focusing on sequences of correct and incorrect responses.

6.6.4 Comparison Results

Fig 6.9 shows the comparison of classification accuracy and F1-score achieved by using CNNs, multivariate LSTM-FCNs, and ESNs. We can see the ESNs achieved superior classification results for classifying successful rule-learning sessions and unsuccessful rule-learning sessions. Specifically, when compared to CNNs, the ESN model improved the classification accuracy by 14.59% and the F1-score by 14.78%; compared to multivariate LSTM-FCNs, the ESN model improved the classification accuracy by 4.47% and the F1-score by 5.15%. Furthermore, statistical tests results on the best classification accuracy achieved by different methods show that both the ESN model and multivariate

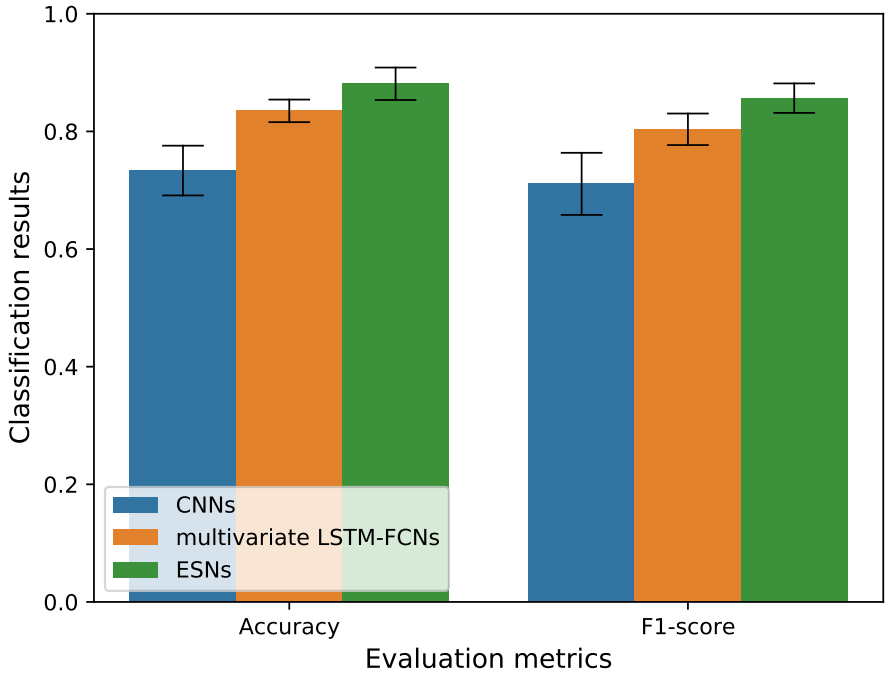

Figure 6.9: Comparison of classification accuracy and F1-score achieved by using CNNs, multivariate LSTM-FCNs and ESNs

LSTM-FCNs outperformed CNNs ($p < 0.01$, 10×10 cross-validation with a corrected paired Student t-test), while the ESN model achieved significantly higher accuracy than multivariate LSTM-FCNs ($p < 0.05$, 10×10 cross-validation with a corrected paired Student t-test).

However, the performance CNNs and multivariate LSTM-FCNs varies across the two datasets (see chapter 5). While CNNs achieved better classification results than multivariate LSTM-FCNs for driver workload classification, our results show that multivariate LSTM-FCNs are more suitable than CNNs for classifying successful and unsuccessful rule learning processes.

6.7 Discussion

The induction process is important for robust learning [200]. However, little is known about the underlying cognitive mechanisms during the induction process [194]. In this work, we explore using fNIRS to identify brain activation patterns that lead to positive and negative learning outcomes during the induction process. As a first step to understanding the induction process in more complex learning tasks, we adopt a well-understood rule-learning task in cognitive neuroscience [190]. The rule learning task can elicit explicit abstract rule induction, which consists of pattern extracting, model building, and retrieval or recognition process [191]. These cognitive processes are common in the induction and refinement process in the learning domain. We design the experiment to ensure variance in participant learning success. We then research advanced machine learning methods for classifying fNIRS data associated with successful rule learning sessions and unsuccessful rule learning sessions.

Our classification results show that both ESN and multivariate LSTM-FCN are suitable for fNIRS data classification, and outperformed CNNs for classifying successful rule learning sessions and unsuccessful rule learning sessions. However, when applying multivariate LSTM-FCN, the classification results are greatly affected by the parameter choices of the model. On the other hand, the fNIRS data classification results based on ESNs are more robust. In general, the ESN model achieved superior classification results for classifying successful rule learning sessions and unsuccessful rule learning sessions, with an accuracy of 87.95% and an F1-score of 85.64%. Furthermore, the visualization analysis of the ESN model shows that the reservoir state sequences can obtain abundant discriminative features for successful and unsuccessful rule learning sessions. It shows the temporal patterns extracted by ESN are in line with the cognitive processes involved during successful and unsuccessful rule learning sessions, rather than the sequences of

correct and incorrect responses.

Our findings have important implications for understanding the underlying cognitive mechanisms during learning activities and developing learning systems that can adapt to the user's cognitive states to support robust learning, as well as provide a better user experience. For example, if the system can detect when a user is struggling with learning, it can provide appropriate feedback and examples to better support the user. Particularly, we see promises of using fNIRS data with ESN for detecting brain activation patterns during the induction process. ESN is conceptually simple, easy to implement, and computationally inexpensive [180], which makes it practical for classifying fNIRS data in real-world applications. Furthermore, the visualization analysis of the ESN model show promises for explaining the classification results, which can help researchers better interpret the findings and provide insights for further investigation.

6.8 Conclusion

In this work, we investigated using fNIRS data to classify successful and unsuccessful rule learning processes, by applying CNNs, multivariate LSTM-FCNs, and ESNs. We show these models achieved satisfactory classification results, while the ESN model can extract distinctive temporal patterns for successful and unsuccessful rule learning processes and achieved better classification results. This work is the first step for using fNIRS to understand the underlying cognitive mechanisms during the induction process, and it builds a foundation for future adaptive learning systems based on fNIRS data.

Chapter 7

Discussion and Conclusion

7.1 Research Contributions

We summarize the research contributions of this dissertation, centering around the three high-level research questions.

7.1.1 Considering RQ1: Mind-wandering Detection by Incorporating Individuals' Differences in fNIRS Data

We collected fNIRS brain data during the SART task. This dataset provides examples of mind-wandering and on-task episodes, defined based on behavioral data, that can be used to investigate robust classification algorithms. We show individual-level classifiers can achieve better classification results when focusing on specific windows rather than those using the entire episodes. We propose a novel individual-based time window selection (ITWS) algorithm to incorporate individual differences in window selection when building group-level classifiers. We demonstrate that the ITWS algorithm can improve the group-level classification results by comparing with other methods that do not use the ITWS algorithm.

The proposed classification framework is data-driven and enables a more accurate detection of mind-wandering. The findings from this study also reveal individual differences in window selection for mind-wandering detection. This work could inform further research about the time course aspects of mind-wandering, and it builds a foundation for both evaluation of multimodal learning interfaces and future attention-aware systems based on fNIRS data.

7.1.2 Considering RQ2: Driver Cognitive Load Classification by Extract Spatial and Temporal Patterns from fNIRS Data

We collected fNIRS brain data in a simulated driving environment with the n-back task used as a secondary task to impart structured cognitive load on drivers. We investigate the application of Convolutional Neural Networks (CNNs), multivariate Long Short Term Memory Fully Convolutional Networks (LSTM-FCNs), and Echo State Networks (ESNs) for fNIRS-based driver cognitive load classification. We show that the proposed ESN method yields state-of-the-art classification accuracy for group-level models without window selection for fNIRS-based driver cognitive load classification.

To the best of our knowledge, this is the first work to explore the application of multivariate LSTM-FCNs and ESNs for extracting temporal patterns from fNIRS data. This work builds a foundation for using fNIRS to measure driver cognitive load in real-world applications. Furthermore, the proposed ESN model method is conceptually simple, easy to implement, and computationally inexpensive [180], which can be useful for other fNIRS-based machine learning tasks.

7.1.3 Considering RQ3: Positive and Negative Cognitive Processes Classification by Applying ESNs

We move beyond cognitive states that have been explored in previous work using fNIRS and investigate the feasibility of using fNIRS to differentiating cognitive processes that lead to positive and negative learning outcomes. We describe a study in which we collected fNIRS brain data during a rule-learning task. The dataset contains instances of successful and unsuccessful rule learning sessions. We show that there are differences in frontal lobe blood oxygenation patterns between successful and unsuccessful rule learning sessions.

We applied CNNs, multivariate LSTM-FCNs, and ESNs for classifying successful and unsuccessful rule learning processes using fNIRS and compare their results. We show that same as fNIRS-based driver cognitive load classification, ESNs achieved superior classification accuracy than multivariate LSTM-FCNs and CNNs for classifying successful and unsuccessful rule learning sessions using fNIRS data. This confirms the capability of the ESN model for extracting useful information from fNIRS data, and validate the robustness of the proposed ESN model for fNIRS-based classification. Furthermore, our findings have important implications for understanding the underlying cognitive mechanisms during learning activities and developing fNIRS-based learning systems that can adapt to the user's cognitive states to support robust learning, as well as provide a better user experience.

7.2 General Discussion and Future Opportunities

7.2.1 Improving fNIRS Data Quality and Building Large fNIRS Datasets

The acquired fNIRS signals from the device contain physiological noise (e.g., heart rate, respiration) and artifacts (e.g., instrumental noise, participant head movement) [114]. Even though multiple pre-processing methods have been developed to remove these noise sources and artifacts from the fNIRS signals, there are no standard steps established in the fNIRS research community. At the same time, noise and outliers in fNIRS data can lead to poor performances of machine learning models. Therefore, for real-world applications, there is a need to improve the fNIRS data quality and take the noises in fNIRS data into consideration when applying machine learning models.

Hardware improvements. Improvements in fNIRS hardware can greatly improve the quality of fNIRS data. For example, the ongoing effort for improving time-domain fNIRS devices show promises for providing more accurate values for neurally evoked HbO and HbR values [201].

Advancement in noise-removal methods and machine learning models. Advancement in noise-removal methods can reduce the impact of noise and outliers in fNIRS data on the machine learning models [202]. For example, Sato *et al.* al proposed a new method that combines short distance channels and a general linear model to extract scalp-hemodynamics and reduce artifacts [203]. Moreover, it would be interesting to explore advanced machine learning techniques that can deal with noisy time-series data. For example, Sheng *et al.* explored using ESNs to predict noisy nonlinear time-series data. They proposed an improved ESN model which can consider the uncertainties in internal states and outputs. Their results show that the ESN model with addictive noises is effective and robust for predicting noisy nonlinear time series [204]. Furthermore, developing

open-source fNIRS data processing infrastructure and user-friendly interfaces be helpful for researchers in this emerging research area.

Building large fNIRS datasets. One limitation for accurately decoding cognitive states from fNIRS data using machine learning is the size of the datasets. By collecting large fNIRS datasets integrating with behavioral data during different tasks, it offers new opportunities for building robust machine learning models that can be used in real-world applications.

7.2.2 Bridging the Gap Between Cognitive Neuroscience Tasks and Realistic Tasks

In order to classify different cognitive states in real-time, there is a need to collect fNIRS data that are correctly labeled with a variety of cognitive states. Researchers have used standardized tasks adapted from experimental psychology (e.g., the n-back task and the SART task), or calibration tasks, which are task manipulations with known consequences. These tasks provide researchers with correctly labeled brain data. However, there is still a gap between highly controlled tasks and complex interaction scenarios in real-word. For example, even though experimental manipulation can closely respond to real-life experience, it may be difficult to assess which cognitive construct is being captured due to the absence of experimental conditions [205].

Therefore, to establish the validity of the measuring target cognitive states using fNIRS in more complicated and realistic interaction tasks, there needs to be a strong and unambiguous linkage between the standardized/calibration task and the target cognitive resources. One possible solution is to combine standardized tasks and experimental manipulation. After carefully selecting target cognitive states and conducting corresponding standardized tasks that are well-understood in neuroscience, research can follow up with

a realistic task to investigate if similar patterns as shown in previous standardized tasks can be detected. Then, researchers can alternate both the properties of the standardized tasks and the design of the realistic tasks to make them more similar. For example, when developing an adaptive learning interface, if our target cognitive state is mind-wandering, the standardized task is the SART task, and the realistic task is a learning task, then modifications can be made both to the SART tasks and the learning task to ensure the validity of the mapping.

7.2.3 Improving Users' Cognitive Model Based on fNIRS Data

In most previous work, information derived from brain data is directly interpreted as a specific user intention or a change in cognitive or affective state. However, representing users' state from physiological data has been one of the limitations of physiological computing. For example, when the fNIRS signals indicate high cognitive workload of a user in a learning task, it could indicate the user is being pushed cognitively in a constructive way and is engaged in learning, or it could be indicative of a state where the user is overwhelmed and is not able to learn [206]. One-dimensional representation of the user state may restrict the range of adaptive options available to the system [205]. Again, we take the high cognitive states of the user in a learning task for example. If the adaptation happens when the user is engaged in learning, it may be disruptive. For simple systems, this may not be a problem, but for complex systems, more delicate adaptive responses are required.

Multiple-dimension representation of user's cognitive states. One straightforward solution is to increase the psychophysiological complexity of user representation [205]. Taking adaptive learning interfaces as an example, Yuksel *et al.* suggests if a learning system can detect both cognitive workload and affective state, it would make the system

more powerful [206]. For example, frustration often leads to giving up during learning. By combining cognitive workload and emotion, the learning system may make a distinction between two states of high cognitive workload and responds accordingly. Therefore, we can use fNIRS to measure different cognitive states at the same time, and provide an adaptive system with a multi-dimensional representation of the user.

Combining context information with users' cognitive states. Another solution is to combine the representation of user state with events and contexts that evoked them. Again, taking adaptive learning interfaces as an example, Zander *et al.* demonstrated the computer can build and continuously update a context-sensitive for a specific user by re-registering EEG data with context information [207]. In the learning domain, Anderson *et al.* illustrated the importance of integrating the bottom-up information from imaging data with the top-down information from a cognitive model [208]. They collected fMRI data while students working with a tutoring system. By combining information relating to mouse-clicks and the distribution of possible lengths for different states, they show they can predict which step the student is when solving a problem. Therefore, when developing adaptive learning interfaces, by combining educational data mining techniques with fNIRS brain imaging, we can explore critical moments in the use of learning systems and get a better understanding of what is occurring during individual use of a learning environment.

7.2.4 Personalizing fNIRS-based Machine Learning Models

Personalization is defined as a process of changing a system to increase its personal relevance [209]. Even though we have mainly explored building machine learning models at a group-level, in order to gain more training samples, it would be interesting to explore fNIRS-based machine learning models that are personalized for each individual.

The personalized model can take individual's difference in cognitive styles, ability, and experiences into consideration. Such models can optimize the systems' adaption for each user and improve user experiences. For example, Gevins *et al.* investigated individual differences in cognitive ability and cognitive style using EEG. They show that subjects that scored high on psychometric tests also tended to have a better performance on experimental tasks examining working memory. High-ability subjects were relatively quick to optimize task allocation on frontal and parietal brain regions. Moreover, EEG signals distinguished between individuals with a verbal cognitive style and those with a nonverbal style, and indicated their different utilization of brain regions [210]. Thus, evaluating and designing different user interfaces and adaption techniques according to different cognitive styles would be an interesting topic to explore further. In learning and cognitive learning theory, a phenomenon called the expertise reversal effect was also explored. The expertise reversal effect means instructional techniques that are beneficial to beginners can have inverse effects on more experienced learners [211, 212]. Recent research suggests that fNIRS-based adaptive interfaces can take the expertise reversal effect into consideration. Leff *et al.* showed that fNIRS can be used to distinguish participants' different levels of experience on a bimanual task [193]. In the work of Yuksel *et al.* , they found out that the adaption techniques designed for beginner were too easy and frustrating for intermediate-level learners [206]. Therefore, again, taking adaptive learning interfaces as an example, if we can personalize the learning environment and adaptation techniques according to the user's background and experiences, it can positively impact users' engagement and improve learning performance. Furthermore, with the advancement in the fNIRS device to make it wearable in everyday settings, it would be interesting to explore using reinforcement learning to personalize intelligent systems based on fNIRS data.

7.3 Closing Remarks

fNIRS is a brain-imaging tool that is safe, portable, and easy to use, it has the potential to change the way we interact with computers. However, it is challenging to accurately decoding cognitive states from fNIRS data.

In this dissertation, we investigate the feasibility of decoding important user states from fNIRS data, through developing and applying novel machine learning methods that are tuned to the characteristics of fNIRS data. We explore using fNIRS for detecting mind-wandering state, and propose an individual-based novel window selection algorithm to incorporate individuals' differences in fNIRS data. The proposed algorithm can significantly improve the results for mind-wandering detection. We investigate the feasibility of using fNIRS for classifying different levels of cognitive load and explore advanced deep learning techniques for extracting spatial and temporal features from fNIRS data, including CNNs, LSTM-FCs, and ESNs. The proposed ESN method achieved state-of-art classification results for driver cognitive load classification. We further investigate the feasibility of using fNIRS to differentiate rule learning processes that lead to positive and negative learning outcomes. We validate the proposed ESN model's generalizability across tasks. We show that the ESN model can effectively extract significant temporal patterns from fNIRS data during successful and unsuccessful rule learning processes.

Successful results and findings of the research have an impact on the emerging fNIRS research and build a foundation for developing adaptive Brain-Computer interfaces using fNIRS. The machine learning frameworks developed from this work can facilitate the research of using fNIRS to measure individuals' cognitive states, which can lead to future developments of using fNIRS to enable appropriate adaptation in intelligent systems.

Appendices

A Driver Cognitive Load Classification Results with Traditional Classifiers

We report the driver cognitive load classification results with traditional classifiers. The average values of HbR and HbO and the slope over the whole window of all channels are used as hand-crafted features.

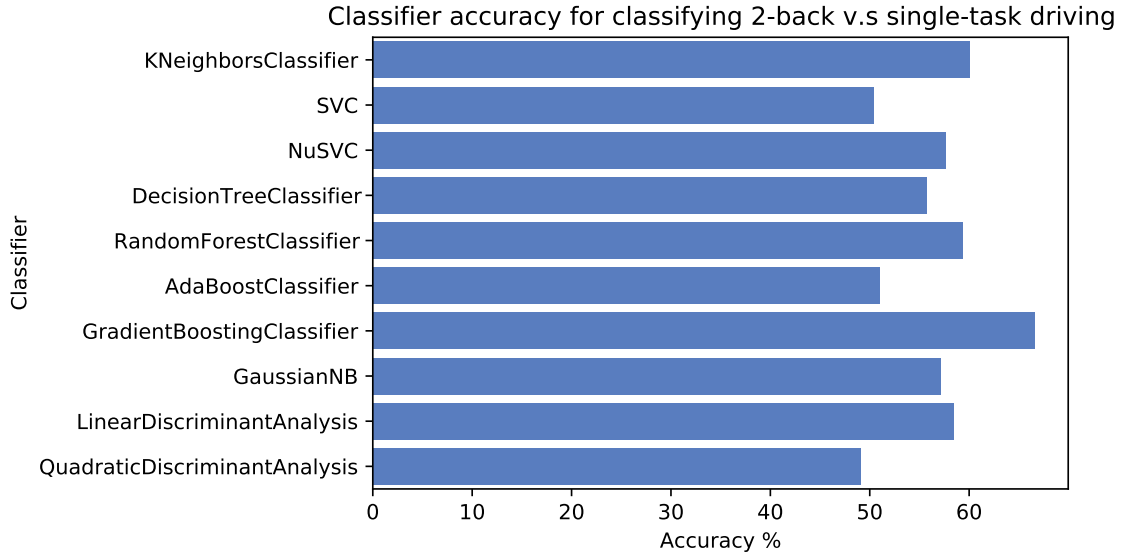

Figure 1: Accuracies for classifying 2-back v.s *single-task driving* with different classifiers, using 10-fold cross-validation.

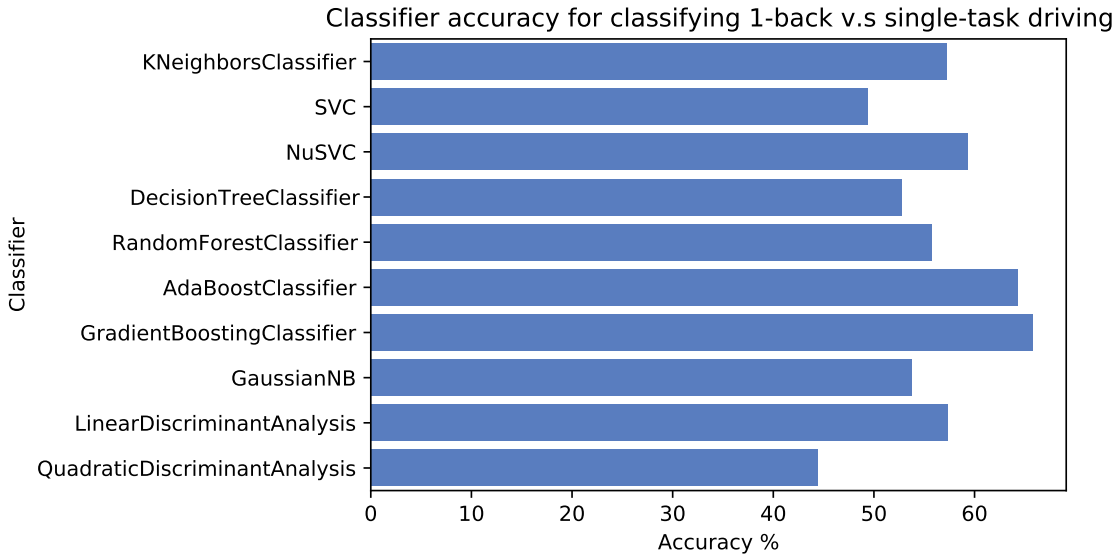

Figure 2: Accuracies for classifying 1-back v.s *single-task driving* with different classifiers, using 10-fold cross-validation.

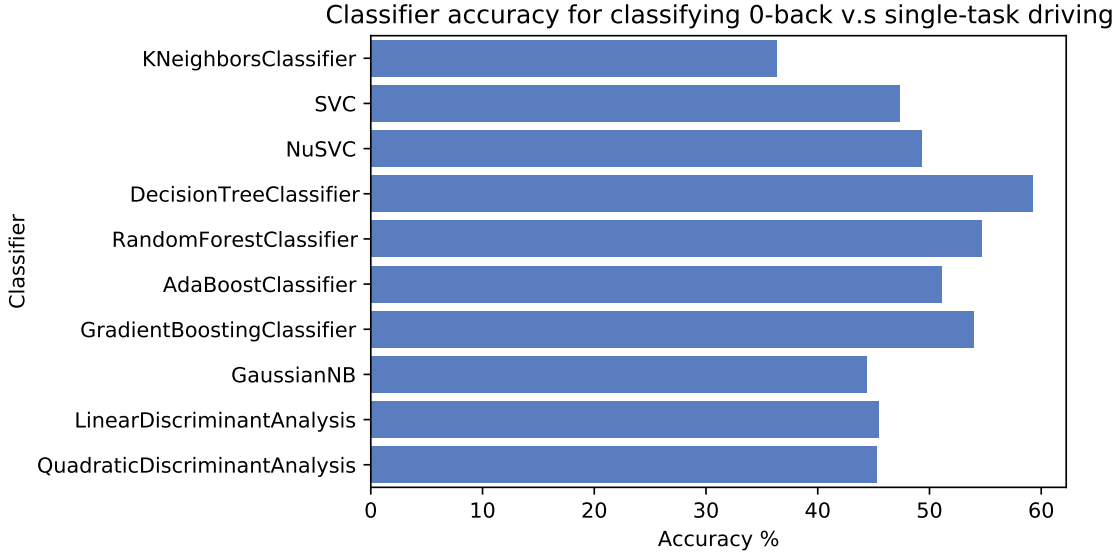

Figure 3: Accuracies for classifying 0-back v.s *single-task driving* with different classifiers, using 10-fold cross-validation.

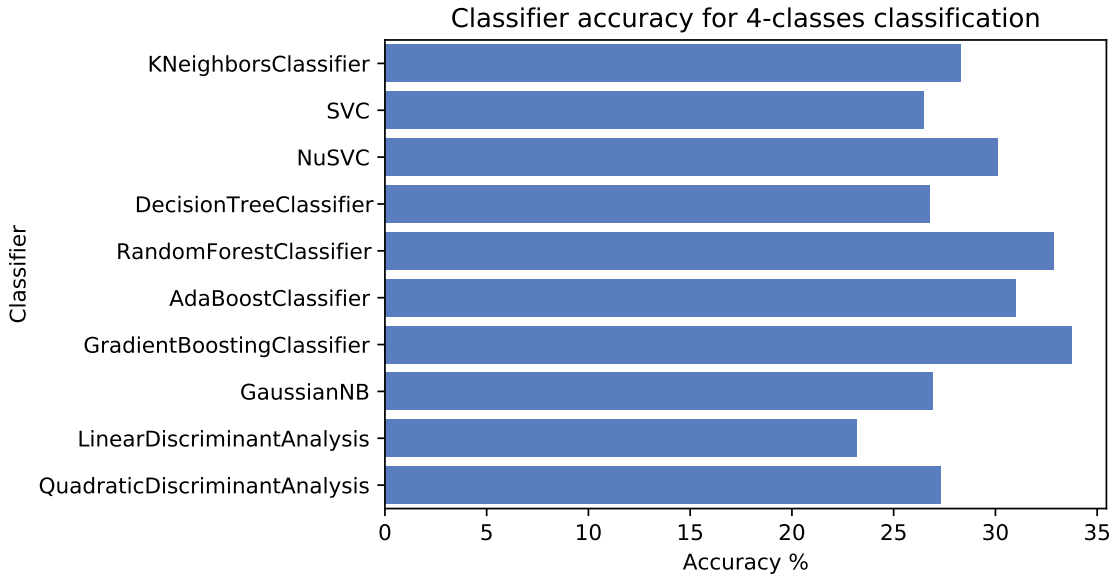

Figure 4: Accuracies for 4-classes classification with different classifiers, using 10-fold cross-validation.

Bibliography

[1] Christopher JD Patten, Albert Kircher, Joakim Östlund, and Lena Nilsson. Using mobile telephones: cognitive workload and attention resource allocation. *Accident analysis & prevention*, 36(3):341–350, 2004.

[2] Shamsi T Iqbal, Piotr D Adamczyk, Xianjun Sam Zheng, and Brian P Bailey. Towards an index of opportunity: understanding changes in mental workload during task execution. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 311–320. ACM, 2005.

[3] Brian P Bailey and Joseph A Konstan. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in human behavior*, 22(4):685–708, 2006.

[4] Mark S Young, Karel A Brookhuis, Christopher D Wickens, and Peter A Hancock. State of science: mental workload in ergonomics. *Ergonomics*, 58(1):1–17, 2015.

[5] Laura E Berk. Relationship of elementary school children's private speech to behavioral accompaniment to task, attention, and task performance. *Developmental Psychology*, 22(5):671, 1986.

[6] Angela DiDomenico and Maury A Nussbaum. Effects of different physical workload parameters on mental workload and performance. *International Journal of Industrial Ergonomics*, 41(3):255–260, 2011.

[7] Bin Xie and Gavriel Salvendy. Review and reappraisal of modelling and predicting mental workload in single-and multi-task environments. *Work & stress*, 14(1):74–99, 2000.

[8] Michael S Franklin, Jonathan Smallwood, and Jonathan W Schooler. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic bulletin & review*, 18 (5):992–997, 2011.

[9] Robert C Williges and Walter W Wierwille. Behavioral measures of aircrew mental workload. *Human Factors*, 21(5):549–574, 1979.

[10] Erwin R Boer. Behavioral entropy as an index of workload. In *Proceedings of the Human Factors*

and Ergonomics Society Annual Meeting, volume 44, pages 125–128. SAGE Publications Sage CA: Los Angeles, CA, 2000.

[11] Robert Bixler and Sidney D'Mello. Automatic gaze-based user-independent detection of mind wandering during computerized reading. *User Modeling and User-Adapted Interaction*, 26(1):33–68, 2016.

[12] Stephen Hutt, Caitlin Mills, Nigel Bosch, Kristina Krasich, James Brockmole, and Sidney D'mello. Out of the fr-eye-ing pan: Towards gaze-based models of attention during learning with technology in the classroom. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 94–103. ACM, 2017.

[13] Gerhard Marquart, Christopher Cabrall, and Joost de Winter. Review of eye-related measures of drivers' mental workload. *Procedia Manufacturing*, 3:2854–2861, 2015.

[14] Nigel Bosch and Sidney D'Mello. Automatic detection of mind wandering from video in the lab and in the classroom. *IEEE Transactions on Affective Computing*, 2019.

[15] Richard T Stone and Chen-Shuang Wei. Exploring the linkage between facial expression and mental workload for arithmetic tasks. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 55, pages 616–619. SAGE Publications Sage CA: Los Angeles, CA, 2011.

[16] Joseph Grafsgaard, Joseph B Wiggins, Kristy Elizabeth Boyer, Eric N Wiebe, and James Lester. Automatically recognizing facial expression: Predicting engagement and frustration. In *Educational Data Mining 2013*, 2013.

[17] Nathaniel Blanchard, Robert Bixler, Tera Joyce, and Sidney D'Mello. Automated physiological-based detection of mind wandering during learning. In *International Conference on Intelligent Tutoring Systems*, pages 55–60. Springer, 2014.

[18] Arthur F Kramer. Physiological metrics of mental workload: A review of recent progress. *Multiple-task performance*, pages 279–328, 1991.

[19] Karel A Brookhuis and Dick de Waard. Monitoring drivers' mental workload in driving simulators using physiological measures. *Accident Analysis & Prevention*, 42(3):898–903, 2010.

[20] Karim Moustafa, Saturnino Luz, and Luca Longo. Assessment of mental workload: a comparison of machine learning methods and subjective assessment techniques. In *International symposium on human mental workload: Models and applications*, pages 30–50. Springer, 2017.

[21] Dennis A Bertram, Donald A Opila, Jeffrey L Brown, Susan J Gallagher, Richard W Schifeling, Irene S Snow, and Charles O Hershey. Measuring physician mental workload: reliability and validity

assessment of a brief instrument. *Medical Care*, pages 95–104, 1992.

[22] Luca Longo. Human-computer interaction and human mental workload: Assessing cognitive engagement in the world wide web. In *IFIP Conference on Human-Computer Interaction*, pages 402–405. Springer, 2011.

[23] Peter A Hancock. Effects of control order, augmented feedback, input device and practice on tracking performance and perceived workload. *Ergonomics*, 39(9):1146–1162, 1996.

[24] M Myrtek, E Deutschmann-Janicke, H Strohmaier, W Zimmermann, S Lawerenz, G Brügner, and W Müller. Physical, mental, emotional, and subjective workload components in train drivers. *Ergonomics*, 37(7):1195–1203, 1994.

[25] Kevin Mandrick, Zarrin Chua, Mickaël Causse, Stéphane Perrey, and Frédéric Dehais. Why a comprehensive understanding of mental workload through the measurement of neurovascular coupling is a key issue for neuroergonomics? *Frontiers in human neuroscience*, 10:250, 2016.

[26] Erin Solovey, Paul Schermerhorn, Matthias Scheutz, Angelo Sassaroli, Sergio Fantini, and Robert Jacob. Brainput: enhancing interactive systems with streaming fnirs brain input. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 2193–2202. ACM, 2012.

[27] Daniel Afergan, Evan M Peck, Erin T Solovey, Andrew Jenkins, Samuel W Hincks, Eli T Brown, Remco Chang, and Robert JK Jacob. Dynamic difficulty using brain metrics of workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3797–3806. ACM, 2014.

[28] Evan M Peck, Emily Carlin, and Robert Jacob. Designing brain-computer interfaces for attention-aware systems. *Computer*, 48(10):34–42, 2015.

[29] Evan M Peck, Daniel Afergan, Beste F Yuksel, Francine Lalooses, and Robert JK Jacob. Using fnirs to measure mental workload in the real world. In *Advances in physiological computing*, pages 117–139. Springer, 2014.

[30] Thibault Gateau, Gautier Durantin, Francois Lancelot, Sebastien Scannella, and Frederic Dehais. Real-time state estimation in a flight simulator using fnirs. *PloS one*, 10(3):e0121279, 2015.

[31] Gautier Durantin, Frederic Dehais, and Arnaud Delorme. Characterization of mind wandering using fnirs. *Frontiers in systems neuroscience*, 9:45, 2015.

[32] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 16(5):051001, 2019.

[33] Keum-Shik Hong, Noman Naseer, and Yun-Hee Kim. Classification of prefrontal and motor cortex signals for three-class fnirs–bci. *Neuroscience letters*, 587:87–92, 2015.

[34] Danushka Bandara, Senem Velipasalar, Sarah Bratt, and Leanne Hirshfield. Building predictive models of emotion with functional near-infrared spectroscopy. *International Journal of Human-Computer Studies*, 110:75–85, 2018.

[35] Xuerui Wang, Rebecca Hutchinson, and Tom M Mitchell. Training fmri classifiers to detect cognitive states across multiple human subjects. In *Advances in neural information processing systems*, pages 709–716, 2004.

[36] Farzan Majeed Noori, Noman Naseer, Nauman Khalid Qureshi, Hammad Nazeer, and Rayyan Azam Khan. Optimal feature selection from fnirs signals using genetic algorithms for bci. *Neuroscience letters*, 647:61–66, 2017.

[37] Rahilsadat Hosseini, Bridget Walsh, Fenghua Tian, and Shouyi Wang. An fnirs-based feature learning and classification framework to distinguish hemodynamic patterns in children who stutter. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(6):1254–1263, 2018.

[38] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[39] Benjamin W Mooneyham and Jonathan W Schooler. The costs and benefits of mind-wandering: a review. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 67(1):11, 2013.

[40] Jibo He, Ensar Becic, Yi-Ching Lee, and Jason S McCarley. Mind wandering behind the wheel: performance and oculomotor correlates. *Human factors*, 53(1):13–21, 2011.

[41] Robert Bixler and Sidney D'Mello. Toward fully automated person-independent detection of mind wandering. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 37–48. Springer, 2014.

[42] Hitoshi Tsunashima and Kazuki Yanagisawa. Measurement of brain function of car driver using functional near-infrared spectroscopy (fnirs). *Computational intelligence and neuroscience*, 2009, 2009.

[43] Anh Son Le, Hirofumi Aoki, Fumihiko Murase, and Kenji Ishida. A novel method for classifying driver mental workload under naturalistic conditions with information from near-infrared spectroscopy. *Frontiers in human neuroscience*, 12:431, 2018.

[44] Lan-peng Li, Zhi-gang Liu, Hai-yan Zhu, Lin Zhu, and Yuan-chun Huang. Functional near-infrared

spectroscopy in the evaluation of urban rail transit drivers' mental workload under simulated driving conditions. *Ergonomics*, 62(3):406–419, 2019.

[45] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

[46] Britton Chance, Endla Anday, Shoko Nioka, Shuoming Zhou, Long Hong, Katherine Worden, C Li, T Murray, Y Ovetsky, D Pidikiti, and R Thomas. A novel method for fast imaging of brain function, non-invasively, with light. *Optics Express*, 2(10):411, 1998. ISSN 1094-4087. doi: 10.1364/oe.2.000411.

[47] Erin Treacy Solovey, Audrey Girouard, Krysta Chauncey, Leanne M Hirshfield, Angelo Sassaroli, Feng Zheng, Sergio Fantini, and Robert J K Jacob. Using fNIRS Brain Sensing in Realistic HCI Settings : Experiments and Guidelines. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 2009.

[48] F. Orihuela-Espina, D. R. Leff, D. R.C. James, A. W. Darzi, and G. Z. Yang. Quality control and assurance in functional near infrared spectroscopy (fNIRS) experimentation. *Physics in Medicine and Biology*, 55(13):3701–3724, 2010. ISSN 00319155. doi: 10.1088/0031-9155/55/13/009.

[49] Leanne M Hirshfield, Erin Treacy Solovey, Audrey Girouard, James Kebinger, Robert JK Jacob, Angelo Sassaroli, and Sergio Fantini. Brain measurement for usability testing and adaptive interfaces: an example of uncovering syntactic workload with functional near infrared spectroscopy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2185–2194. ACM, 2009.

[50] Meltem Izzetoglu, Kurtulus Izzetoglu, Scott Bunce, Hasan Ayaz, Ajit Devaraj, Banu Onaral, and Kambiz Pourrezaei. Functional near-infrared neuroimaging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13:153–159, 2005. ISSN 15344320. doi: 10.1109/TNSRE.2005.847377.

[51] Horia A. Maior, Matthew Pike, Sarah Sharples, and Max L. Wilson. Examining the reliability of using fNIRS in realistic HCI settings for spatial and verbal tasks. In *proceedings of the 33rd annual ACM conference on human factors in computing systems*, volume 2015-April, pages 3039–3042. ACM, 2015. ISBN 9781450331456. doi: 10.1145/2702123.2702315.

[52] Matthew F. Pike, Horia A. Maior, Martin Porcheron, Sarah C. Sharples, and Max L. Wilson. Measuring the effect of think aloud protocols on workload using fNIRS. In *Proceedings of the 32nd*

annual ACM conference on Human factors in computing systems, pages 3807–3816. ACM, 2014. ISBN 9781450324731. doi: 10.1145/2556288.2556974.

[53] Daniel Afergan, Evan M. Peck, Erin Treacy Solovey, Andrew Jenkins, Samuel W. Hincks, Eli T. Brown, Remco Chang, and Robert J.K. Jacob. Dynamic difficulty using brain metrics of workload. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pages 3797–3806, 2014. ISBN 9781450324731. doi: 10.1145/2556288.2557230.

[54] Dylan D. Schmorrow and Cali M. Fidopiastis. Phylter: a system for modulating notifications in wearables using physiological sensing. In *International conference on augmented cognition*, volume 9183, pages 167–177. Springer, Cham, 2015. ISBN 9783319208152. doi: 10.1007/978-3-319-20816-9.

[55] Tomoki Shibata, Evan M Peck, Daniel Afergan, Samuel W Hincks, Beste F Yuksel, and Robert J.K. Jacob. Building implicit interfaces for wearable computers with physiological inputs: Zero shutter camera and phylter. In *UIST 2014 - Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 89–90. ACM, 2014. ISBN 9781450330688. doi: 10.1145/2658779.2658790.

[56] Evan M. Peck, Emily Carlin, and Robert Jacob. Designing Brain-Computer Interfaces for Attention-Aware Systems. *Computer*, 48(10):34–42, 2015. ISSN 00189162. doi: 10.1109/MC.2015.315.

[57] Joy Hirsch, Xian Zhang, J Adam Noah, Yumie Ono, New Haven, New Haven, and New Haven. Frontal temporal and parietal systems synchronize within and across brains during live eye-to-eye contact. *Neuroimage*, 157:314–330, 2017. doi: 10.1016/j.neuroimage.2017.06.018.Frontal.

[58] Naama Mayseless, Grace Hawthorne, and Allan L. Reiss. Real-life creative problem solving in teams: fNIRS based hyperscanning study. *NeuroImage*, 203(August):116161, 2019. ISSN 10538119. doi: 10.1016/j.neuroimage.2019.116161. URL `https://doi.org/10.1016/j.neuroimage.2019.116161`.

[59] Ryan McKendrick, Raja Parasuraman, Rabia Murtza, Alice Formwalt, Wendy Baccus, Martin Paczynski, and Hasan Ayaz. Into the wild: Neuroergonomic differentiation of hand-held and augmented reality wearable displays during outdoor navigation with functional near infrared spectroscopy. *Frontiers in Human Neuroscience*, 10(MAY2016):216, 2016. ISSN 16625161. doi: 10.3389/fnhum.2016.00216.

[60] Ryan McKendrick, Raja Parasuraman, and Hasan Ayaz. Wearable functional near infrared spectroscopy (fNIRS) and transcranial direct current stimulation (tDCS): expanding vistas for neu-

rocognitive augmentation. *Frontiers in Systems Neuroscience*, 9, 2015. ISSN 1662-5137. doi: 10.3389/fnsys.2015.00027.

[61] Sophie K. Piper, Arne Krueger, Stefan P. Koch, Jan Mehnert, Christina Habermehl, Jens Steinbrink, Hellmuth Obrig, and Christoph H. Schmitz. A wearable multi-channel fNIRS system for brain imaging in freely moving subjects. *NeuroImage*, 85:64–71, 2014. ISSN 10538119. doi: 10.1016/j.neuroimage.2013.06.062.

[62] Miki Watanabe, Kohei Ogawa, and Hiroshi Ishiguro. Can androids be salespeople in the real world? In *Conference on Human Factors in Computing Systems - Proceedings*, volume 18, pages 781–788, 2015. ISBN 9781450331463. doi: 10.1145/2702613.2702967.

[63] Peter T Fox and Marcus E Raichle. Focal physiological uncoupling of cerebral blood flow and oxidative metabolism during somatosensory stimulation in human subjects. *Proceedings of the National Academy of Sciences*, 83(4):1140–1144, 1986.

[64] Noman Naseer and Keum-Shik Hong. fNIRS-based brain-computer interfaces: a review. *Frontiers in Human Neuroscience*, 9(January):1–15, 2015. ISSN 1662-5161. doi: 10.3389/fnhum.2015.00172. URL `http://www.frontiersin.org/Human_Neuroscience/10.3389/fnhum.2015.00172/full`.

[65] Danushka Sandaruwan Bandara. Machine learning methods for functional near infrared spectroscopy. *Dissertations - ALL.*, 953, 2018. URL `https://surface.syr.edu/etd/953`.

[66] Thi Kieu Khanh Ho, Jeonghwan Gwak, Chang Min Park, and Jong In Song. Discrimination of Mental Workload Levels from Multi-Channel fNIRS Using Deep Leaning-Based Approaches. *IEEE Access*, 7:24392–24403, 2019. ISSN 21693536. doi: 10.1109/ACCESS.2019.2900127.

[67] Johannes Hennrich, Christian Herff, Dominic Heger, and Tanja Schultz. Investigating deep learning for fnirs based bci. In *2015 37th Annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2844–2847. IEEE, 2015.

[68] Gauvain Huve, Kazuhiko Takahashi, and Masafumi Hashimoto. Brain activity recognition with a wearable fnirs using neural networks. In *2017 IEEE international conference on mechatronics and automation (ICMA)*, pages 1573–1578. IEEE, 2017.

[69] Christian Herff, Dominic Heger, Ole Fortmann, Johannes Hennrich, Felix Putze, and Tanja Schultz. Mental workload during n-back task—quantified in the prefrontal cortex using fnirs. *Frontiers in human neuroscience*, 7:935, 2014.

[70] Felix Putze, Christian Herff, Christoph Tremmel, Tanja Schultz, and Dean J Krusienski. Decoding

mental workload in virtual environments: a fnirs study using an immersive n-back task. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3103–3106. IEEE, 2019.

[71] Marjan Saadati, Jill Nelson, and Hasan Ayaz. Convolutional neural network for hybrid fnirs-eeg mental workload classification. In *International Conference on Applied Human Factors and Ergonomics*, pages 221–232. Springer, 2019.

[72] Etienne Combrisson and Karim Jerbi. Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy. *Journal of neuroscience methods*, 250:126–136, 2015.

[73] Sinem Burcu Erdoğan, Eran Özsarfati, Burcu Dilek, Kübra Soğukkanlı Kadak, Lütfü Hanoğlu, and Ata Akın. Classification of motor imagery and execution signals with population-level feature sets: implications for probe design in fnirs based bci. *Journal of neural engineering*, 16(2):026029, 2019.

[74] Jessica Gemignani, Eike Middell, Randall L Barbour, Harry L Graber, and Benjamin Blankertz. Improving the analysis of near-infrared spectroscopy data with multivariate classification of hemodynamic patterns: a theoretical formulation and validation. *Journal of neural engineering*, 15(4): 045001, 2018.

[75] Ruixue Liu, Erin Walker, Leah Friedman, Catherine M Arrington, and Erin T Solovey. fnirs-based classification of mind-wandering with personalized window selection for multimodal learning interfaces. *Journal on Multimodal User Interfaces*, 2020.

[76] Xiao-Su Hu, Keum-Shik Hong, and Shuzhi Sam Ge. fnirs-based online deception decoding. *Journal of Neural Engineering*, 9(2):026012, 2012.

[77] Alborz Rezazadeh Sereshkeh, Rozhin Yousefi, Andrew T Wong, and Tom Chau. Online classification of imagined speech using functional near-infrared spectroscopy signals. *Journal of neural engineering*, 16(1):016005, 2018.

[78] Xu Cui, Signe Bray, and Allan L Reiss. Speeded near infrared spectroscopy (nirs) response detection. *PLoS one*, 5(11):e15474, 2010.

[79] Noman Naseer and Keum-Shik Hong. Functional near-infrared spectroscopy based brain activity classification for development of a brain-computer interface. In *2012 International Conference of Robotics and Artificial Intelligence*, pages 174–178. IEEE, 2012.

[80] Audrey Girouard, Erin Treacy Solovey, Leanne M Hirshfield, Evan M Peck, Krysta Chauncey, Angelo Sassaroli, Sergio Fantini, and Robert JK Jacob. From brain signals to adaptive interfaces: using

fnirs in hci. In *Brain-Computer Interfaces*, pages 221–237. Springer, 2010.

[81] Erin Treacy Solovey, Paul Schermerhorn, Matthias Scheutz, Angelo Sassaroli, Sergio Fantini, and Robert J.K. Jacob. Brainput: Enhancing interactive systems with streaming fNIRS brain input. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 2193–2202, 2012. ISBN 9781450310154. doi: 10.1145/2207676.2208372.

[82] Erin Treacy Solovey, Daniel Afergan, Evan M Peck, Samuel W Hincks, and Robert JK Jacob. Designing implicit interfaces for physiological computing: Guidelines and lessons learned using fnirs. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(6):35, 2015.

[83] Yichuan Liu, Hasan Ayaz, and Patricia A Shewokis. Multisubject "learning" for mental workload classification using concurrent eeg, fnirs, and physiological measures. *Frontiers in human neuroscience*, 11:389, 2017.

[84] Kieran C.R. Fox, R. Nathan Spreng, Melissa Ellamil, Jessica R. Andrews-Hanna, and Kalina Christoff. The wandering brain: Meta-analysis of functional neuroimaging studies of mind-wandering and related spontaneous thought processes. *NeuroImage*, 111:611–621, 2015. ISSN 10959572. doi: 10.1016/j.neuroimage.2015.02.039. URL `http://dx.doi.org/10.1016/j.neuroimage.2015.02.039`.

[85] Matthew A. Killingsworth and Daniel T. Gilbert. A wandering mind is an unhappy mind. *Science*, 330(6006):932, 2010. ISSN 00368075. doi: 10.1126/science.1192439.

[86] Jonathan W. Schooler, Jonathan Smallwood, Kalina Christoff, Todd C. Handy, Erik D. Reichle, and Michael A. Sayette. Meta-awareness, perceptual decoupling and the wandering mind. *Trends in Cognitive Sciences*, 15(7):319–326, 2011. ISSN 13646613. doi: 10.1016/j.tics.2011.05.006. URL `http://dx.doi.org/10.1016/j.tics.2011.05.006`.

[87] Benjamin W. Mooneyham and Jonathan W. Schooler. The costs and benefits of mind-wandering: A review. *Canadian Journal of Experimental Psychology*, 67(1):11–18, 2013. ISSN 11961961. doi: 10.1037/a0031569.

[88] Pieter Wouters, Erik D Van der Spek, and Herre Van Oostendorp. Current practices in serious game research: A review from a learning outcomes perspective. In *Games-based learning advancements for multi-sensory human computer interfaces: techniques and effective practices*, pages 232–250. IGI Global, 2009.

[89] Brian C Nelson. Exploring the use of individualized, reflective guidance in an educational multi-user virtual environment. *Journal of Science Education and Technology*, 16(1):83–97, 2007.

[90] David N Rapp. The value of attention aware systems in educational settings. *Computers in Human Behavior*, 22(4):603–614, 2006.

[91] Kalina Christoff, Alan M Gordon, Jonathan Smallwood, Rachelle Smith, and Jonathan W Schooler. Experience sampling during fMRI reveals default network and executive system contributions to mind wandering. *Proceedings of the National Academy of Sciences*, pages 8719–8724, 2009. URL papers3://publication/uuid/F7FC47FD-5AB1-4FCE-8F30-A99EE1870E01.

[92] Xu Cui, Signe Bray, Daniel M Bryant, Gary H Glover, and Allan L Reiss. A quantitative comparison of nirs and fmri across multiple cognitive tasks. *Neuroimage*, 54(4):2808–2821, 2011.

[93] Noman Naseer and Keum Shik Hong. Classification of functional near-infrared spectroscopy signals corresponding to the right- and left-wrist motor imagery for development of a brain-computer interface. *Neuroscience Letters*, 553:84–89, 2013. ISSN 03043940. doi: 10.1016/j.neulet.2013.08.021. URL http://dx.doi.org/10.1016/j.neulet.2013.08.021.

[94] M. Jawad Khan, Xiaolong Liu, M. Raheel Bhutta, and Keum Shik Hong. Drowsiness detection using fNIRS in different time windows for a passive BCI. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 227–231. IEEE, 2016. ISBN 9781509032877. doi: 10.1109/BIOROB.2016.7523628.

[95] Jaeyoung Shin, Alexander Von Luhmann, Benjamin Blankertz, Do Won Kim, Jichai Jeong, Han Jeong Hwang, and Klaus Robert Muller. Open Access Dataset for EEG+NIRS Single-Trial Classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1735–1745, 2017. ISSN 15344320. doi: 10.1109/TNSRE.2016.2628057.

[96] Thomas M Connolly, Elizabeth A Boyle, Ewan MacArthur, Thomas Hainey, and James M Boyle. A systematic literature review of empirical evidence on computer games and serious games. *Computers & education*, 59(2):661–686, 2012.

[97] Sidney D'Mello, Andrew Olney, Claire Williams, and Patrick Hays. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of human-computer studies*, 70(5):377–398, 2012.

[98] Jonathan Smallwood, Daniel J. Fishman, and Jonathan W. Schooler. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic Bulletin and Review*, 14(2):230–236, 2007. ISSN 10699384. doi: 10.3758/BF03194057.

[99] Christina Yi Jin, Jelmer P Borst, and Marieke K Van Vugt. Predicting task-general mind-wandering with EEG. *Cognitive, Affective, & Behavioral Neuroscience,*, 19:1–15, 2019.

[100] Robert Bixler and Sidney D'Mello. Automatic gaze-based user-independent detection of mind wan-

dering during computerized reading. *User Modeling and User-Adapted Interaction*, 26(1):33–68, 2016. ISSN 15731391. doi: 10.1007/s11257-015-9167-1.

[101] Stephen Hutt, Caitlin Mills, Nigel Bosch, Kristina Krasich, James Brockmole, and Sidney D'mello. Out of the Fr-"Eye"-ing Pan: Towards gaze-based models of attention during learning with technology in the classroom. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization.ACM*, pages 94–103. ACM, 2017. ISBN 9781450346351. doi: 10.1145/3079628.3079669.

[102] Nathaniel Blanchard, Robert Bixler, Tera Joyce, and Sidney D'Mello. Automated physiological-based detection of mind wandering during learning. In *International Conference on Intelligent Tutoring Systems*, pages 55–60. Springer, Cham, 2014. ISBN 9783319072203. doi: 10.1007/978-3-319-07221-0{_}7.

[103] John Champaign and Gord McCalla. AttentiveLearner: improving mobile MOOC learning via implicit heart rate tracking. In *International Conference on Artificial Intelligence in Education*, pages 367–376. Springer, Cham, 2015. ISBN 9783319197722. doi: 10.1007/978-3-319-19773-9.

[104] Michael S. Franklin, Jonathan Smallwood, and Jonathan W. Schooler. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic Bulletin and Review*, 18(5):992–997, 2011. ISSN 10699384. doi: 10.3758/s13423-011-0109-6.

[105] Caitlin Mills and Sidney D Mello. Toward a Real-time (Day) Dreamcatcher : Sensor-Free Detection of Mind Wandering During Online Reading. In *International Educational Data Mining Society*, 2015.

[106] Nigel Bosch and Sidney Dmello. Automatic Detection of Mind Wandering from Video in the Lab and in the Classroom. *IEEE Transactions on Affective Computing*, PP(c):1–1, 2019. doi: 10.1109/taffc.2019.2908837.

[107] Issaku Kawashima and Hiroaki Kumano. Prediction of mind-wandering with electroencephalogram and non-linear regression modeling. *Frontiers in Human Neuroscience*, 11(July):1–10, 2017. ISSN 16625161. doi: 10.3389/fnhum.2017.00365.

[108] Alessio Paolo Buccino, Hasan Onur Keles, and Ahmet Omurtag. Hybrid EEG-fNIRS asynchronous brain-computer interface for multiple motor tasks. *PLoS ONE*, 11(1):1–16, 2016. ISSN 19326203. doi: 10.1371/journal.pone.0146610.

[109] Rahilsadat Hosseini, Bridget Walsh, Fenghua Tian, and Shouyi Wang. An fNIRS-based feature learning and classification framework to distinguish hemodynamic patterns in children who stut-

ter. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(6):1254–1263, 2018. ISSN 15344320. doi: 10.1109/TNSRE.2018.2829083.

[110] Farzan Majeed Noori, Noman Naseer, Nauman Khalid Qureshi, Hammad Nazeer, and Rayyan Azam Khan. Optimal feature selection from fNIRS signals using genetic algorithms for BCI. *Neuroscience Letters*, 647:61–66, 2017. ISSN 18727972. doi: 10.1016/j.neulet.2017.03.013. URL `http://dx.doi.org/10.1016/j.neulet.2017.03.013`.

[111] Tom Manly, Ian H Robertson, Maria Galloway, and Kari Hawkins. The absent mind:: further investigations of sustained attention to response. *Neuropsychologia*, 37(6):661–670, 1999.

[112] Amishi P Jha, Alexandra B Morrison, Justin Dainer-Best, Suzanne Parker, Nina Rostrup, and Elizabeth A Stanley. Minds "at attention": Mindfulness training curbs attentional lapses in military cohorts. *PloS one*, 10(2), 2015.

[113] Jonathan Smallwood, Emily Beach, Jonathan W Schooler, and Todd C Handy. Going awol in the brain: Mind wandering reduces cortical analysis of external events. *Journal of cognitive neuroscience*, 20(3):458–469, 2008.

[114] Noman Naseer and Keum-Shik Hong. fnirs-based brain-computer interfaces: a review. *Frontiers in human neuroscience*, 9:3, 2015.

[115] Paola Pinti, Felix Scholkmann, Antonia Hamilton, Paul Burgess, and Ilias Tachtsidis. Current status and issues regarding pre-processing of fnirs neuroimaging data: An investigation of diverse signal filtering methods within a general linear model framework. *Frontiers in human neuroscience*, 12: 505, 2018.

[116] Theodore J. Huppert, Solomon G. Diamond, Maria A. Franceschini, and David A. Boas. HomER: A review of time-series analysis methods for near-infrared spectroscopy of the brain. *Applied Optics*, 48(10):0–33, 2009. ISSN 15394522. doi: 10.1364/AO.48.00D280.

[117] Jaeyoung Shin, Alexander Von Lühmann, Do-Won Kim, Jan Mehnert, Han-Jeong Hwang, and Klaus-Robert Müller. Simultaneous acquisition of eeg and nirs during cognitive tasks for an open access dataset. *Scientific data*, 5:180003, 2018.

[118] Jaeyoung Shin, Klaus R. Müller, and Han Jeong Hwang. Near-infrared spectroscopy (NIRS)-based eyes-closed brain-computer interface (BCI) using prefrontal cortex activation due to mental arithmetic. *Scientific Reports*, 6(October):1–11, 2016. ISSN 20452322. doi: 10.1038/srep36203. URL `http://dx.doi.org/10.1038/srep36203`.

[119] Suresh Balakrishnama and Aravind Ganapathiraju. LINEAR DISCRIMINANT ANALYSIS - A

BRIEF TUTORIAL. *Institute for Signal and information Processing*, 18(4):1–8, 1998. ISSN 03749096.

[120] Konstantinos Makantasis, Anastasios Doulamis, Nikolaos Doulamis, Antonis Nikitakis, and Athanasios Voulodimos. Tensor-based nonlinear classifier for high-order data analysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2221–2225. IEEE, 2018.

[121] Stanford Linear Accelerator. Regularized Discriminant. *Journal of the American Statistical Association*, 1988(July), 1988.

[122] Nitesh V. Chawla Keven, Kevin W. Bowyer, Lawrence O. Hall, and W. Philp Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique Nitesh. *Journal of artificial intelligence research*, 16 (1):321–357, 2002. ISSN 10769757. doi: 10.1613/jair.953.

[123] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

[124] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[125] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu: 785–794, 2016. doi: 10.1145/2939672.2939785.

[126] Angela R. Harrivel, Chad L. Stephens, Robert J. Milletich, Christina M. Heinich, Mary Carolyn Last, Nicholas J. Napoli, Nijo A. Abraham, Lawrence J. Prinzel, Mark A. Motter, and Alan T. Pope. Prediction of cognitive states during flight simulation using multimodal psychophysiological sensing. *AIAA Information Systems-AIAA Infotech at Aerospace, 2017*, pages 1–10, 2017. doi: 10.2514/6.2017-1135.

[127] Jonathan Smallwood and Jonathan W. Schooler. The restless mind. *Psychological Bulletin*, 132(6): 946–958, 2006. ISSN 00332909. doi: 10.1037/0033-2909.132.6.946.

[128] Ruixue Liu, Bryan Reimer, Siyang Song, Bruce Mehler, and Erin T Solovey. Unsupervised fnirs feature extraction with cae and esn autoencoder for driver cognitive load classification. *Journal of Neural Engineering*, 2020.

[129] World Health Organization et al. Global status report on road safety 2018: Summary. Technical report, World Health Organization, 2018.

[130] National Highway Traffic Safety Administration. Distracted driving. https://www.nhtsa.gov/risky-driving/distracted-driving, 2018. Accessed: 2020-04-08.

[131] Bryan Reimer, Bruce Mehler, Jonathan Dobres, Hale McAnulty, Alea Mehler, Daniel Munger, and Adrian Rumpold. Effects of an 'expert mode' voice command system on task performance, glance behavior & driver physiology. In *Proceedings of the 6th international conference on automotive user interfaces and interactive vehicular applications*, pages 1–9, 2014.

[132] Patrick Tchankue, Janet Wesson, and Dieter Vogts. The impact of an adaptive user interface on reducing driver distraction. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 87–94, 2011.

[133] Johan Engström, Emma Johansson, and Joakim Östlund. Effects of visual and cognitive load in real and simulated motorway driving. *Transportation research part F: traffic psychology and behaviour*, 8(2):97–120, 2005.

[134] Julie Paxion, Edith Galy, and Catherine Berthelon. Mental workload and driving. *Frontiers in psychology*, 5:1344, 2014.

[135] Erin T Solovey, Marin Zec, Enrique Abdon Garcia Perez, Bryan Reimer, and Bruce Mehler. Classifying driver workload using physiological and driving performance data: two field studies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4057–4066. ACM, 2014.

[136] Bruce Mehler, Bryan Reimer, Joseph F Coughlin, and Jeffery A Dusek. Impact of incremental increases in cognitive workload on physiological arousal and performance in young adult drivers. *Transportation Research Record*, 2138(1):6–12, 2009.

[137] Bruce Mehler, Bryan Reimer, and Joseph F Coughlin. Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task: an on-road study across three age groups. *Human factors*, 54(3):396–412, 2012.

[138] Hyun Suk Kim, Yoonsook Hwang, Daesub Yoon, Wongeun Choi, and Cheong Hee Park. Driver workload characteristics analysis using eeg data from an urban road. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1844–1849, 2014.

[139] Felix Putze, Jan-Philip Jarvis, and Tanja Schultz. Multimodal recognition of cognitive workload for multitasking in the car. In *2010 20th International Conference on Pattern Recognition*, pages 3748–3751. IEEE, 2010.

[140] Paul van Gent, Timo Melman, Haneen Farah, Nicole van Nes, and Bart van Arem. Multi-level

driver workload prediction using machine learning and off-the-shelf sensors. *Transportation research record*, 2672(37):141–152, 2018.

[141] Bryan Reimer and Bruce Mehler. The impact of cognitive workload on physiological arousal in young adult drivers: a field study and simulation validation. *Ergonomics*, 54(10):932–942, 2011.

[142] Bryan Reimer, Bruce Mehler, Ying Wang, and Joseph F Coughlin. A field study on the impact of variations in short-term memory demands on drivers' visual attention and driving performance across three age groups. *Human factors*, 54(3):454–468, 2012.

[143] Yue Gu, Shuo Miao, Junxia Han, Zhenhu Liang, Gaoxiang Ouyang, Jian Yang, and Xiaoli Li. Identifying adhd children using hemodynamic responses during a working memory task measured by functional near-infrared spectroscopy. *Journal of neural engineering*, 15(3):035005, 2018.

[144] Heng Huang, Xintao Hu, Yu Zhao, Milad Makkie, Qinglin Dong, Shijie Zhao, Lei Guo, and Tianming Liu. Modeling task fmri data via deep convolutional autoencoder. *IEEE transactions on medical imaging*, 37(7):1551–1561, 2017.

[145] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

[146] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 241–246. IEEE, 2016.

[147] Thanawin Trakoolwilaiwan, Bahareh Behboodi, Jaeseok Lee, Kyungsoo Kim, and Ji-Woong Choi. Convolutional neural network for high-accuracy functional near-infrared spectroscopy in a brain–computer interface: three-class classification of rest, right-, and left-hand motor execution. *Neurophotonics*, 5(1):011008, 2017.

[148] Jia-Shu Zhang and Xian-Ci Xiao. Predicting chaotic time series using recurrent neural network. *Chinese Physics Letters*, 17(2):88, 2000.

[149] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7), 2017.

[150] Huanhuan Chen, Peter Tiňo, Ali Rodan, and Xin Yao. Learning in the model space for cognitive fault diagnosis. *IEEE transactions on neural networks and learning systems*, 25(1):124–136, 2013.

[151] Witali Aswolinskiy, René Felix Reinhart, and Jochen Steil. Time series classification in reservoir-and model-space. *Neural Processing Letters*, 48(2):789–809, 2018.

[152] Leilei Sun, Bo Jin, Haoyu Yang, Jianing Tong, Chuanren Liu, and Hui Xiong. Unsupervised eeg

feature extraction based on echo state network. *Information Sciences*, 475:1–17, 2019.

[153] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002.

[154] Alaa Sagheer and Mostafa Kotb. Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, 323:203–213, 2019.

[155] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *Neural Networks*, 116:237–245, 2019.

[156] Changxu Wu and Yili Liu. Queuing network modeling of driver workload and performance. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):528–537, 2007.

[157] Yilu Zhang, Yuri Owechko, and Jing Zhang. Driver cognitive workload estimation: A data-driven perspective. In *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*, pages 642–647. IEEE, 2004.

[158] Bruce Mehler, Bryan Reimer, and Jeffery A Dusek. Mit agelab delayed digit recall task (n-back). *Cambridge, MA: Massachusetts Institute of Technology*, page 17, 2011.

[159] ISO/TS 14198 (11.2012). Road vehicles-ergonomic aspects of transport information and control systems-calibration tasks for methods which assess driver demand due to the use of in-vehicle systems: Iso international organization for standardization. 2019.

[160] Adrian M Owen, Kathryn M McMillan, Angela R Laird, and Ed Bullmore. N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies. *Human brain mapping*, 25(1):46–59, 2005.

[161] Yan Yang, Haoqi Sun, Tianchi Liu, Guang-Bin Huang, and Olga Sourina. Driver workload detection in on-road driving environment using machine learning. In *Proceedings of ELM-2014 Volume 2*, pages 389–398. Springer, 2015.

[162] Lex Fridman, Bryan Reimer, Bruce Mehler, and William T Freeman. Cognitive load estimation in the wild. In *Proceedings of the 2018 chi conference on human factors in computing systems*, pages 1–9, 2018.

[163] Monika Lohani, Brennan R Payne, and David L Strayer. A review of psychophysiological measures to assess cognitive states in real-world driving. *Frontiers in human neuroscience*, 13, 2019.

[164] Haleh Aghajani, Marc Garbey, and Ahmet Omurtag. Measuring mental workload with eeg+ fnirs. *Frontiers in human neuroscience*, 11:359, 2017.

[165] Tomoyuki Nagasawa, Takanori Sato, Isao Nambu, and Yasuhiro Wada. fnirs-gans: data augmenta-

tion using generative adversarial networks for classifying motor tasks from functional near-infrared spectroscopy. *Journal of Neural Engineering*, 17(1):016068, 2020.

[166] Ying Wang, Bruce Mehler, Bryan Reimer, Vincent Lammers, Lisa A D'Ambrosio, and Joseph F Coughlin. The validity of driving simulation for assessing differences between in-vehicle informational interfaces: A comparison with field testing. *Ergonomics*, 53(3):404–420, 2010.

[167] Bruce Mehler and Bryan Reimer. An initial assessment of the significance of task pacing on self-report and physiological measures of workload while driving. In *Proceedings of the 7th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, pages 170–176, 2013.

[168] Valtino X Afonso, Willis J Tompkins, Truong Q Nguyen, and Shen Luo. Multirate processing of the ecg using filter banks. In *Computers in Cardiology 1996*, pages 245–248. IEEE, 1996.

[169] Yonina C Eldar and Alan V Oppenheim. Filter bank interpolation and reconstruction from generalized and recurrent nonuniform samples. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 1, pages 324–327. IEEE, 2000.

[170] Xu Cui, Signe Bray, and Allan L Reiss. Functional near infrared spectroscopy (nirs) signal improvement based on negative correlation between oxygenated and deoxygenated hemoglobin dynamics. *Neuroimage*, 49(4):3039–3046, 2010.

[171] Ting Li, Qingming Luo, and Hui Gong. Gender-specific hemodynamics in prefrontal cortex during a verbal working memory task by near-infrared spectroscopy. *Behavioural brain research*, 209(1): 148–153, 2010.

[172] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, pages 52–59. Springer, 2011.

[173] Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel. Extracting deep bottleneck features using stacked auto-encoders. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 3377–3381. IEEE, 2013.

[174] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[175] Filippo Maria Bianchi, Simone Scardapane, Aurelio Uncini, Antonello Rizzi, and Alireza Sadeghian. Prediction of telephone calls load using echo state network with exogenous variables. *Neural Net-*

works, 71:204–213, 2015.

[176] Decai Li, Min Han, and Jun Wang. Chaotic time series prediction based on a novel robust echo state network. *IEEE Transactions on Neural Networks and Learning Systems*, 23(5):787–799, 2012.

[177] Qianli Ma, Lifeng Shen, Weibiao Chen, Jiabin Wang, Jia Wei, and Zhiwen Yu. Functional echo state network for time series classification. *Information Sciences*, 373:1–20, 2016.

[178] Pattreeya Tanisaro and Gunther Heidemann. Time series classification using time warping invariant echo state networks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 831–836. IEEE, 2016.

[179] Filippo Maria Bianchi, Simone Scardapane, Sigurd Løkse, and Robert Jenssen. Reservoir computing approaches for representation and classification of multivariate time series. *arXiv preprint arXiv:1803.07870*, 2018.

[180] Mantas Lukoševičius. A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade*, pages 659–686. Springer, 2012.

[181] Qianli Ma, Enhuan Chen, Zhenxi Lin, Jiangyue Yan, Zhiwen Yu, and Wing WY Ng. Convolutional multitimescale echo state network. *IEEE Transactions on Cybernetics*, 2019.

[182] Thomas G Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

[183] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine learning*, 52(3): 239–281, 2003.

[184] Remco R Bouckaert and Eibe Frank. Evaluating the replicability of significance tests for comparing learning algorithms. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 3–12. Springer, 2004.

[185] Chris Berka, Daniel J Levendowski, Michelle N Lumicao, Alan Yau, Gene Davis, Vladimir T Zivkovic, Richard E Olmstead, Patrice D Tremoulet, and Patrick L Craven. Eeg correlates of task engagement and mental workload in vigilance, learning, and memory tasks. *Aviation, space, and environmental medicine*, 78(5):B231–B244, 2007.

[186] Martin Spüler, Carina Walter, Wolfgang Rosenstiel, Peter Gerjets, Korbinian Moeller, and Elise Klein. Eeg-based prediction of cognitive workload induced by arithmetic: a step towards online adaptation in numerical learning. *ZDM*, 48(3):267–278, 2016.

[187] Carina Walter, Wolfgang Rosenstiel, Martin Bogdan, Peter Gerjets, and Martin Spüler. Online eeg-based workload adaptation of an arithmetic learning environment. *Frontiers in human neuroscience*,

11:286, 2017.

[188] Ronald H Stevens, Trysha Galloway, and Chris Berka. Eeg-related changes in cognitive workload, engagement and distraction as students acquire problem solving skills. In *International Conference on User Modeling*, pages 187–196. Springer, 2007.

[189] Bertram Opitz and Angela D Friederici. Brain correlates of language learning: the neuronal dissociation of rule-based versus similarity-based learning. *Journal of Neuroscience*, 24(39):8436–8440, 2004.

[190] B.A. Strange. Anterior Prefrontal Cortex Mediates Rule Learning in Humans. *Cerebral Cortex*, 11 (11):1040–1046, 2001. doi: 10.1093/cercor/11.11.1040.

[191] Dariya Goranskaya, Jens Kreitewolf, Jutta L Mueller, Angela D Friederici, and Gesa Hartwigsen. Fronto-parietal contributions to phonological processes in successful artificial grammar learning. *Frontiers in human neuroscience*, 10:551, 2016.

[192] Cary R Savage, Thilo Deckersbach, Stephan Heckers, Anthony D Wagner, Daniel L Schacter, Nathaniel M Alpert, Alan J Fischman, and Scott L Rauch. Prefrontal regions supporting spontaneous and directed application of verbal learning strategies: evidence from pet. *Brain*, 124(1): 219–231, 2001.

[193] Daniel Richard Leff, Clare E Elwell, Felipe Orihuela-Espina, Louis Atallah, David T Delpy, Ara W Darzi, and Guang Zhong Yang. Changes in prefrontal cortical behaviour depend upon familiarity on a bimanual co-ordination task: an fnirs study. *Neuroimage*, 39(2):805–813, 2008.

[194] Deniz Sonmez Unal, Catherine M Arrington, Erin Solovey, and Erin Walker. Using thinkalouds to understand rule learning and cognitive control mechanisms within an intelligent tutoring system. In *International Conference on Artificial Intelligence in Education*, pages 500–511. Springer, 2020.

[195] Daniel Szafir and Bilge Mutlu. Pay attention! designing adaptive agents that monitor and improve user engagement. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 11–20, 2012.

[196] Daniel Szafir and Bilge Mutlu. Artful: adaptive review technology for flipped learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1001–1010, 2013.

[197] Liping Shen, Minjuan Wang, and Ruimin Shen. Affective e-learning: Using "emotional" data to improve learning in pervasive learning environment. *Journal of Educational Technology & Society*, 12(2):176–189, 2009.

[198] Caitlin Mills, Igor Fridman, Walid Soussou, Disha Waghray, Andrew M Olney, and Sidney K D'Mello. Put your thinking cap on: detecting cognitive load using eeg during learning. In *Proceedings of the seventh international learning analytics & knowledge conference*, pages 80–89, 2017.

[199] Wolfgang Skrandies and Alexander Klein. Brain activity and learning of mathematical rules—effects on the frequencies of eeg. *Brain research*, 1603:133–140, 2015.

[200] Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798, 2012.

[201] Heidrun Wabnitz, Davide Contini, Lorenzo Spinelli, Alessandro Torricelli, and Adam Liebert. Depth-selective data analysis for time-domain fnirs: moments vs. time windows. *Biomedical Optics Express*, 11(8):4224–4243, 2020.

[202] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45 (1):12, 2012.

[203] Takanori Sato, Isao Nambu, Kotaro Takeda, Takatsugu Aihara, Okito Yamashita, Yuko Isogaya, Yoshihiro Inoue, Yohei Otaka, Yasuhiro Wada, Mitsuo Kawato, et al. Reduction of global interference of scalp-hemodynamics in functional near-infrared spectroscopy using short distance probes. *NeuroImage*, 141:120–132, 2016.

[204] Chunyang Sheng, Jun Zhao, Ying Liu, and Wei Wang. Prediction for noisy nonlinear time series by echo state network based on dual estimation. *Neurocomputing*, 82:186–195, 2012.

[205] Stephen H Fairclough. Fundamentals of physiological computing. *Interacting with computers*, 21 (1-2):133–145, 2009.

[206] Beste F Yuksel, Kurt B Oleson, Lane Harrison, Evan M Peck, Daniel Afergan, Remco Chang, and Robert JK Jacob. Learn piano with bach: An adaptive learning interface that adjusts task difficulty based on brain state. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pages 5372–5384, 2016.

[207] Thorsten O Zander, Laurens R Krol, Niels P Birbaumer, and Klaus Gramann. Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity. *Proceedings of the National Academy of Sciences*, 113(52):14898–14903, 2016.

[208] John R Anderson, Shawn Betts, Jennifer L Ferris, and Jon M Fincham. Neural imaging to track mental states while using an intelligent tutoring system. *Proceedings of the National Academy of Sciences*, 107(15):7018–7023, 2010.

[209] Jan Blom. Personalization: a taxonomy. In *CHI'00 extended abstracts on Human factors in computing systems*, pages 313–314, 2000.

[210] Alan Gevins and Michael E Smith. Neurophysiological measures of working memory and individual differences in cognitive ability and cognitive style. *Cerebral cortex*, 10(9):829–839, 2000.

[211] Slava Kalyuga. The expertise reversal effect. In *Managing cognitive load in adaptive multimedia learning*, pages 58–80. IGI Global, 2009.

[212] Slava Kalyuga. Expertise reversal effect and its implications for learner-tailored instruction. *Educational psychology review*, 19(4):509–539, 2007.