9-2021

Advancing Proper Dataset Partitioning and Classification of Visual Search and the Vigilance Decrement Using EEG Deep Learning Algorithms

Alexander J. Kamrud

Advancing Proper Dataset Partitioning and Classification of Visual Search and the Vigilance Decrement Using EEG Deep Learning Algorithms

DISSERTATION

Alexander J. Kamrud, Major, USAF

AFIT-ENG-DS-21-S-011

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This is an academic work and should not be used to imply or infer actual mission capability or limitations.

AFIT-ENG-DS-21-S-011

ADVANCING PROPER DATASET PARTITIONING AND CLASSIFICATION OF VISUAL SEARCH AND THE VIGILANCE DECREMENT USING EEG DEEP LEARNING ALGORITHMS

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Computer Science

Alexander J. Kamrud, B.S.E.E., M.S.E.E.
Major, USAF

September 2021

AFIT-ENG-DS-21-S-011

ADVANCING PROPER DATASET PARTITIONING AND CLASSIFICATION OF VISUAL SEARCH AND THE VIGILANCE DECREMENT USING EEG DEEP LEARNING ALGORITHMS

Alexander J. Kamrud, B.S.E.E., M.S.E.E.
Major, USAF

Committee Membership:

Brett J. Borghetti, PhD
Chairman

Christine M. Schubert Kabban, PhD
Member

Michael E. Miller, PhD
Member

John J. Elshaw, PhD
Dean's Representative

Adedeji B. Badiru
Dean, Graduate School of Engineering and Management

AFIT-ENG-DS-21-S-011

Abstract

Electroencephalography (EEG) classification of visual search and vigilance tasks has vast potential in its benefits. In future human-machine teaming systems, EEG could act as the tool for operator state assessment, enabling AI teammates to know when to assist the operator in these tasks. These future augmented cognition systems have the potential to lead to increased safety of operations, better training systems for our operators, and improved operational effectiveness. This dissertation investigates EEG models built to utilize any individual's EEG signals, i.e. cross-participant models, in the areas of:

1. Dataset partitioning for proper training and validation
2. Classification of the efficiency of an operator's search
3. Classification of whether an operator is in a vigilance decrement during a vigilance type task.

First, the necessity of proper dataset partitioning for EEG cross-participant models is demonstrated both mathematically and empirically using publicly available datasets, with empirical results demonstrating that improper partitioning of datasets can lead to error rates underestimated between 35% and 3900%. Next, the results of a conducted visual search experiment are presented, in which EEG signals were captured while participants performed a visual search task, and various techniques were tested to mitigate inefficient search to efficient search. Efficient search was found to be on average faster than inefficient search, resulting in a 13% speed up, and also more accurate, with a 61% reduction in error rate. Two techniques (the *nudge* and *hint*) were found to be effective in mitigation of inefficient search, resulting in a 169% increase of efficient searches. The collected EEG signals were utilized to build deep learning models for detection of whether or not a participant was performing an inefficient or efficient search, with models overall found to perform no better than random chance (50% accuracy). However, this may be attributed to the *nudge* technique altering fundamental aspects of gaze fixation and saccade intervals, suggesting the need for further investigation with a modified experiment design. Lastly, EEG cross-participant models are presented to classify whether or not a participant was in a vigilance decrement during an unseen vigilance type task: a multilayer perceptron neural network (MLPNN) which employed spectral features extracted from the five traditional EEG frequency bands, a temporal convolutional network (TCN), and a TCN autoencoder (TCN-AE), with these two TCN models being time-domain based, i.e. using raw EEG time-series voltage values. The MLPNN and TCN models both achieved accuracy greater than random chance (50%), with the MLPNN performing best with a 7-fold CV balanced accuracy of 64% (95% CI: 0.59, 0.69), and validation accuracies greater than random chance for 9 of the 14 participants. The MLPNN outperforming both the TCN and TCN-AE models suggests that frequency-domain models may outperform time-domain based models for the purpose of vigilance decrement detection due to the salient information contained within spectral features.

Table of Contents

Page

Abstract iv

List of Figures ix

List of Tables xi

I. Introduction 1

1.1 Summary of Research Objectives and Contributions 6
1.2 Structure 11

II. Background 13

2.1 EEG 13
2.1.1 Non-Stationarity and Individual Differences 16
2.2 Deep Learning 19
2.2.1 Recurrent Neural Networks 21
2.2.2 Temporal Convolutional Networks 26
2.2.3 Autoencoders 28

III. The Effects of Individual Differences, Non-Stationarity, and the Importance of Data Partitioning Decisions for Training and Testing of EEG Cross-Participant Models 31

3.1 Paper Overview 31
3.2 Abstract 31
3.3 Introduction 32
3.4 Background 34
3.4.1 Covariate Shift 34
3.4.2 Non-Stationarity and Individual Differences 39
3.4.3 Approaches to Data and Problem Formulation 39
3.5 Initial Demonstration 43
3.5.1 Defining and Estimating the Effects of Covariate Shift 43
3.5.2 Covariate Shift in EEG 51
3.5.3 Reducing Inter-Participant Variability 53
3.6 Empirical Demonstrations in Diverse EEG Case Studies 60
3.6.1 Driver Fatigue 63
3.6.2 Confused Students 65
3.6.3 Alcoholism 68
3.6.4 Post-Traumatic Stress Disorder (PTSD) 70
3.6.5 Schizophrenia 72
3.7 Discussion 75

3.8 Conclusions . . . 77
3.9 Appendix . . . 79
3.9.1 Appendix A.1. Driver Fatigue Data . . . 79
3.9.2 Appendix A.2. Confused Students Data . . . 79
3.9.3 Appendix A.3. Alcoholism Data . . . 80
3.9.4 Appendix A.4. PTSD Data . . . 80
3.9.5 Appendix A.5. Schizophrenia Data . . . 80

IV. Detection and Mitigation of Inefficient Visual Searching . . . 81

4.1 Paper Overview . . . 81
4.2 Abstract . . . 81
4.3 Introduction . . . 82
4.4 Method . . . 84
4.4.1 Participants . . . 84
4.4.2 Appartus . . . 84
4.4.3 Procedure . . . 86
4.5 Results . . . 91
4.6 Discussion . . . 94
4.7 Appendix - Methods . . . 96
4.7.1 Participants . . . 96
4.7.2 Stimuli . . . 96
4.7.3 Apparatus . . . 97
4.7.4 Questionnaires . . . 107
4.7.5 Signals Collected . . . 111
4.7.6 Preprocessing Pipeline . . . 112
4.7.7 Datasets . . . 114
4.7.8 Model Creation . . . 116
4.7.9 Mitigation Analysis . . . 119

V. Detection of the Vigilance Decrement in any Sustained Attention Task using EEG-based Deep Learning . . . 120

5.1 Paper Overview . . . 120
5.2 Abstract . . . 120
5.3 Introduction . . . 121
5.4 Related Work . . . 123
5.4.1 Vigilance Decrement . . . 124
5.4.2 EEG . . . 128
5.5 Methods . . . 129
5.5.1 Datasets . . . 129
5.5.2 Vigilance Decrement Tasks . . . 130
5.5.3 Preprocessing and Epoching of EEG signals . . . 133

5.5.4 Model Creation 136
5.5.5 Temporal Convolutional Networks 137
5.5.6 Autoencoders 139
5.5.7 Frequency-domain Model 140
5.5.8 Time-domain Models 141
5.6 Results 146
5.6.1 Frequency-domain Model 146
5.6.2 Time-domain Model - TCN-AE 147
5.6.3 Time-domain Model - TCN 148
5.7 Discussion 150
5.8 Conclusions and Future Work 153
5.9 Appendix A - EEG Preprocessing 156

VI. Conclusions and Future Work 158

6.1 Contributions and Findings 158
6.2 Future Work 163
Bibliography 167

List of Figures

Figure Page

1 EEG DL Related Published Papers 15

2 Visual Illustration of ANN 20

3 Visual Representation of an RNN 22

4 Visual Illustration of an LSTM 24

5 Visual Illustration of a GRU 25

6 Visual Illustration of a TCN 27

7 Visual Illustration of a stacked TCN 28

8 Visual Representation of an Autoencoder 29

9 Simple example of covariate shift 36

10 t-SNE on MNIST dataset 38

11 Proper Dataset Partitioning 42

12 PCA for Improper and Proper partitioning - Driver Fatigue dataset 48

13 Heat maps for log-transformed weight ratio values 50

14 t-SNE for 2D dimensionality reduction of datasets 53

15 t-SNE visualization of transformed entropy data 57

16 t-SNE visualization of transformed spectral data 59

17 Example of experiment trial. 85

18 General architectures for ESE models. 90

19 ESE Color Combinations 97

20 ESE instructions screen 99

21 ESE varying proportions of colors 101

Figure Page

22 ESE nudge trial 103

23 ESE block instructions 103

24 ESE Hint mitigation 104

25 ESE Explanation mitigation 105

26 ESE Block instructions for instructions mitigation 105

27 ESE trial instructions for instructions mitigation 106

28 ESE block design 107

29 ESE equipment setup 111

30 Hitchcock Air Traffic Controller Task 131

31 3-Stimulus Task 132

32 Line Task 133

33 BIS measures for each task 135

34 Visual Illustration of a TCN 138

35 Visual Illustration of a stacked TCN 139

36 Visual Representation of an Autoencoder 140

37 Visual representation of MLPNN classifier 141

38 Visual representation of the TCN-AE architecture 143

39 Visual representation of the TCN-AE classifier 144

40 Visual representation of the TCN classifier 145

41 Participant validation accuracies for MLPNN model 147

42 Participant validation accuracies for TCN model 148

List of Tables

Table Page

1 Study A Objectives and Contributions 7

2 Study B Objectives and Contributions 9

3 Study C Objectives and Contributions 10

4 Classification accuracies for transformed data. 60

5 Details for the publicly available datasets. 61

6 Validation results for the five case studies. 76

7 Mean balanced accuracy scores of the within-participant ESE models. 92

8 Mean balanced accuracy scores of the cross-participant ESE models. 93

9 Vigilance Decrement Model Performance by participant 149

10 Oddball task BIS measures and MLPNN model performance by participant 152

ADVANCING PROPER DATASET PARTITIONING AND CLASSIFICATION OF VISUAL SEARCH AND THE VIGILANCE DECREMENT USING EEG DEEP LEARNING ALGORITHMS

I. Introduction

Many professions within the military routinely perform duties which require visual search (e.g. pilots or sensors operators scanning instruments). Visual search requires an operator to perform a scan of the environment to locate a target, while ignoring other distracting objects or features, with the order of the targets searched determining their visual search pattern (VSP) [1]. Visual search is susceptible to confirmation bias, resulting in confirmatory search which on average leads to inefficient search, and inefficient search on average is longer and less accurate than efficient search [2, 3]. This confirmation bias in search can be especially detrimental for military operations, as military operations commonly occur in stressful and time sensitive environments, and additionally, the consequences of suboptimal decision making can be disastrous. An example of confirmation bias in a military operation occurred in 1988, where the commander of the USS Vincennes erroneously shot down a commercial Iranian Airliner, resulting in the death of all 290 passengers on board. The commander's erroneous decision was partially attributed to confirmation bias, as the commander overly relied on incorrect information [4].

These search tasks also often coincide with maintaining sustained attention over a long period of time (i.e. vigilance), and it is vital that vigilance is maintained as it correlates to detection of critical events within these tasks and overall task performance [5]. These vigilance tasks require operators to remain focused and alert to stimulus during a task, and in the control and surveillance of today's automated systems, vigilance typically suffers either due to the low level of workload and stimulus associated with this task [6], or due to the mental demands vigilance requires over a lengthy task [7]. As automated systems are becoming more and more prevalent across the military, low levels of sustained attention in operators is a growing concern.

A solution to inefficient search and maintaining vigilance is to have artificial intelligence (AI) systems assist operators during these tasks, known as human-machine teaming (HMT). Through enhanced decision aids, biased and inefficient search could be mitigated and reduce the likelihood of similar accidents of the USS Vincennes, and also lead to more efficient searches, which are on average both faster and more accurate than biased inefficient search [3]. By detecting a vigilance decrement in operators, AI systems can vary levels of stimulus to aid in sustained attention [8], plan breaks in the task to keep an operator alert [9], and also detect levels of fatigue that require the operator to be relieved. However, in order for HMT to be done effectively, the system requires the state of the operator to know how and when to assist, just as humans know how and when to assist in effective human-human teams. For operator state assessment, this can be performed using a number of different physiological signals, and typically uses some combination of the following: Electroencephalography (EEG) to measure activity in the brain, electrooculography (EOG) for recording eye movement such as saccades and blink rate, electrocardiography (ECG) for measuring activity in the heart such as heart rate, and electrical conductivity of the skin such as the galvanic skin response (GSR).

This research focuses on using EEG, as the goal is to inform the system of the operator's cognitive state, and the electrical activity of the brain measured through the scalp can contain aspects of cognitive processes, mental activities, and cognitive load [10]. Specifically, this research leverages machine learning and deep learning approaches using EEG signal inputs to investigate models capable of detecting when an operator is performing inefficient search, and when an operator is in a vigilance decrement, and to perform this classification on any individual (a.k.a. a cross-participant model).

The previous paragraphs delve into two different applications of EEG machine learning models which could aid military operators, namely in detecting a vigilance decrement in vigilance type tasks, and in detecting inefficient visual search. However, an additional challenge in building EEG models is in their ability to generalize. Ideally, EEG cross-participant models are desired to have the following characteristics:

- Task Generalization - The ability to generalize: to perform well in tasks other than the task they were specifically trained to perform well in.
- Invariance to Inter-person Differences - The model is invariant to the variability in EEG signals which differ uniquely for every individual due to unique characteristics of each person's brain, their genetics, and other variables specific to that one individual. With invariance to inter-person differences, the model is able to generalize and perform classification on any participant.

However, machine learning models typically fall short in both of these areas. Models typically lack in task generalization as they are usually built using data from one task or experiment [11, 12], meaning the model is only trained and evaluated for that one specific task, with unknown generalization on other tasks. While the model is useful for that one specific task, it could be more useful to have a model which applies to all related tasks which fall under the umbrella of the phenomena of interest. For detection of a vigilance decrement, there are numerous vigilance tasks, however, the vast majority of machine learning models in the literature are trained to perform well on a single task, and thus, cannot detect vigilance decrements in other tasks [13, 14], despite the neural correlates of the vigilance decrement being similar across different tasks [15, 16]. In this research, we advance the field of vigilance decrement classification by building a model which is capable of classifying the vigilance decrement across different vigilance tasks, as well as across unseen participants.

With invariance to inter-person differences, the core challenge is that EEG is both non-stationary for individuals [17] and that there are inter-person differences in EEG signals across individuals that result in inter-participant variability [18]. EEG non-stationarity is due to a variety of internal and external causes, such as brain activity causing continual changes in states of neuronal assemblies [19], user attention levels, user fatigue, sensor equipment used, and scalp placement of electrodes [20]. Similar to non-stationarity, the inter-person differences in EEG signals are also due to a variety of causes, such as differences in variability in frequency peaks for individuals due to differences in personality traits [21], genetic variations [22–24], gamma–aminobutyric acid concentrations in the brain [25, 26], and memory task performance [27]. These characteristics which differ between individuals are unknown covariates for the machine learning models, and these covariates result in different input distributions of the EEG signals for each participant, while the conditional distribution of the output class given the EEG input feature vector stays the same, resulting in a covariate shift for machine learning models when they are tested upon EEG from participants that the model has not seen [28, 29]. Covariate shift in machine learning is a difference in the input distributions of the training and testing datasets [30]. When there is covariate shift and a model is trained using one distribution and tested on a different (shifted) distribution, it will likely have worse performance on the test distribution, as a general guideline and assumption that is used in supervised machine learning is that all of the data is independently sampled from the same population and identically distributed (i.i.d. assumption). Without this assumption, many theoretical guarantees and bounds on minimizing the test error are lost. For EEG cross-participant classification models, this covariate shift and its effects will typically be present when the model classifies EEG data belonging to a participant that the model has not seen. However, models should be tested with data which is representative of the data they will predict upon in the real world, and thus, EEG cross-participant models should be tested with unseen participants, especially if it is likely that there are distributional differences among the EEG signal-to-classification mappings in the participants. Therefore, as a best practice in reporting accurate model performance for models intended to classify any individual's EEG, EEG cross-participant models should always be validated and tested using EEG data that comes from participants the model has not trained upon.

Despite previous work showing that EEG has inter-participant variability [18], and that this inter-participant variability leads to covariate shift when EEG models are tested with an unseen participant [28, 29, 31–33], the majority of EEG studies built to classify any individual's EEG do not follow this best practice of testing the model with unseen participants. In a recent literature review of deep learning-based EEG models by Roy et al., only 23 out of 108 cross-participant models utilized some method of proper dataset partitioning to ensure the model was tested with a participant that was not used for training [11]. This same literature review also compared the number of studies exploring models built for a specific individual (within-participant) versus cross-participant, and they found that since 2016, the growing trend has shifted toward building cross-participant models, with the latest ratio of studies researching cross-participant models to within-participant models being over 5:1 [11]. With this ever-growing popularity in EEG cross-participant models, it is critical that the research community understands that solely training an EEG cross-participant model doesn't guarantee generalizability of the model, and that proper evaluation of the cross-participant model using EEG from unseen participants is necessary to avoid significant underestimation of the model error rate. This means that the data from entire individuals must be sequestered and only be used for testing of the model, with no data from those individuals used for training of the cross-participant model. By not following this best practice, the research pool may become increasingly diluted with studies reporting model performance metrics that are unrealistic and unrepresentative of the model's true ability. This research seeks to address this issue by demonstrating the necessity of proper partitioning of EEG datasets for EEG cross-participant models through both mathematical and empirical methods.

1.1 Summary of Research Objectives and Contributions

This section reviews the research objectives and contributions for each research study in this dissertation, mapping studies A - C to the results presented in Chapters III - V respectively. For clarity, a table outlining the objectives and contributions is presented for each study.

Study A demonstrated both mathematically and empirically the effects of proper and improper methods of dataset partitioning for EEG cross-participant models. Proper dataset partitioning for EEG cross-participant models means that data from participants used for model training must not be used for model validation or testing, and participants that are utilized for validation must not be used for testing; this ensures the cross-participant model is tested with unseen participants and reflects its intended purpose of estimating the model's performance on individuals in the future. Empirically this research found that models which do not follow proper methods of dataset partitioning are sub-optimal in their hyperparameter selection and learned model parameters, and also underestimate their test error rate between 35% to 3900% for unseen participants. Mathematically this research demonstrated the presence of covariate shift in improper dataset partitioning using Shimodaira's loss rescaling equation [34]. This equation was utilized in combination with two-dimensional histogram estimators to generate a heatmap for visualization of the difference in the loss rescaling weight ratio values between the two methods, which indicated the presence of covariate shift for proper methods of dataset partitioning. How inter-participant variability affects the presence of covariate shift was also demonstrated. A novel method of data transformation was applied to real data and was shown to reduce inter-participant variability within the data, which was also then shown to reduce the presence of covariate shift, and thus increase model performance, demonstrating the link between inter-participant variability and covariate shift in EEG data.

Table 1. Summary of objectives and contributions from Study A discussed in Chapter III

	Objectives
AO1	Demonstrate mathematically how inter-participant variability in EEG affects the presence of covariate shift in the data
AO2	Evaluate five publicly available datasets using proper and improper methods of dataset partitioning for comparison of model classification performance
	Findings and Contributions
A1	Utilized Shimodaira's loss rescaling on real data in order to demonstrate the presence of covariate shift between proper and improper methods of dataset partitioning using a novel heat map visualization of the loss rescaling weight ratio values, providing the first known method and use of Shimodaira's equation on real data to mathematically demonstrate the presence of a covariate shift.
A2	Developed a EEG data transformation technique to reduce inter-participant variability in the data and utilized this transformation to demonstrate how inter-participant variability affects covariate shift and resulting model performance.
A3	Empirical results show that not following the proper method of dataset partitioning can result in underestimation of error rates in models between 35% and 3900% for unseen participants, and also results in sub-optimal model creation for its intended purpose.

Study B investigated how to detect and mitigate inefficient visual search, as prior research has hinted that human search is naturally inefficient, with inefficient search being slower and less accurate than efficient search [2]. An experiment was designed and conducted for natural inefficient search to occur, and utilized various techniques in order to mitigate this inefficient search. The experiment was successful in creating a search environment where inefficient search was naturally the default behavior [35, 36], with 80.86% of participant searches being inefficient prior to mitigation (with 7.18% of those being circular inefficient). Mitigation techniques that curbed the default behavior and encouraged efficient search were also found to be significant, with a 169% increase of efficient searches, resulting in a total of 51.41% of all searches being efficient, with the *nudge* and *hint* techniques being most effective, and the *explanation* and *instructions* techniques not found effective. Efficient search was also found to be faster than inefficient search, resulting in a 13% speed up, and also more accurate, with a 61% reduction in error rate. Physiological measures of EEG, ECG, EOG, GSR, and gaze tracking data, were also collected throughout the experiment. EEG was utilized to develop within and cross-participant models to evaluate the effectiveness of EEG for classification of inefficient and efficient search. Model types evaluated included Linear Discriminant Analysis (LDA), Random Forest Classifier (RFC), Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), and Temporal Convolutional Network (TCN). Due to the nature of the *nudge* mitigation altering the fundamental aspects of the search, such as gaze fixation duration and saccade intervals, four different methods of partitioning the data were utilized in order to isolate these effects for training and testing of the different model types. Overall, models designed to detect inefficient and efficient search using EEG did not perform better than chance accuracy (chance accuracy defined as 50% for this binary classification task). However, this may be attributed to the *nudge* technique altering fundamental aspects of gaze fixation and

Table 2. Summary of objectives and contributions from Study B discussed in Chapter IV

	Objectives
BO1	Design and execute an experiment which elicits a natural inefficient visual search pattern, and also limits this inefficient search through various mitigation techniques.
BO2	Capture behavioral and physiological measures during the experiment. Behavioral measures include the participant's VSP, the duration of each search, the accuracy of a participant's search, and the effect of mitigation techniques on their VSP. Physiological measures include EEG, ECG, EOG, GSR, and gaze tracking data.
BO3	Determine which techniques are effective in mitigating inefficient search.
BO4	Evaluate the effectiveness of using EEG for classification of efficiency of search.
	Findings and Contributions
B1	Without a mitigation technique or training of how to perform an efficient search, the majority of participant searches were inefficient.
B2	Efficient search was found to be faster and more accurate than inefficient search.
B3	The *nudge* technique was the most effective mitigation technique, followed by the *hint* technique. The *explanation* and *instructions* techniques were not found to be effective.
B4	The effectiveness of EEG for classification of efficient search was also investigated, with multiple dataset partitioning methods and model classifiers being investigated for both within and cross-participant models. Overall, the models did not perform better than chance.

saccade intervals, suggesting the need for further investigation with a modified experiment design.

Study C investigated the ability to detect the vigilance decrement using EEG across participants (cross-participant) and across tasks (cross-task). This is significant because the vigilance decrement is typically measured using sustained attention tasks which fall into different taxonomies based on their information-processing demands and thus, generalized behavior [5, 37]. Thus far, a generalized model which can classify the vigilance decrement across any sustained attention task using independent measures, such as physiological signals, has not been achieved. Additionally, models which can generalize and classify on

Table 3. Summary of objectives and contributions from Study C discussed in Chapter V

	Objectives
CO1	Evaluate the effectiveness of machine learning models in both the frequency-domain and time-domain for classification of the vigilance decrement in unseen participants performing a vigilance task different than the tasks used for training.
CO2	Evaluate the effectiveness of latent space representation for classification of the vigilance decrement in unseen participants performing a vigilance task different than the tasks used for training.
	Findings and Contributions
C1	The vigilance decrement can be classified in an unseen task using EEG from unseen participants, as evidenced by both the frequency-domain MLPNN and time-domain TCN achieving cross-validation (CV) accuracies greater than random chance.
C2	Frequency-domain models may outperform time-domain models for classification of the vigilance decrement due to the salient information contained within spectral features, as evidenced by the frequency-domain MLPNN having a significantly greater CV balanced accuracy of 64% (95% confidence interval (CI): 0.59, 0.69), and significantly more participants with validation accuracies greater than random chance (9 out of 14 participants).

any participant are desirable, as there is no additional time needed to either train a new model or update an existing model for every new user. Three different EEG cross-task cross-participant models were evaluated to investigate this capability: a frequency-domain MLPNN which employed spectral features extracted from the five traditional EEG frequency bands, and two time-domain models utilizing the raw EEG time-series voltage values, specifically a TCN and a TCN autoencoder (TCN-AE). The MLPNN and TCN models both achieved accuracy greater than random chance (50%), with the MLPNN performing best with a 7-fold cross-validation (CV) balanced accuracy of 64% (95% confidence interval (CI): 0.59, 0.69), and validation accuracies greater than random chance for 9 of the 14 participants. The TCN model resulted in 7-fold CV balanced accuracy of 56% (95% CI: 0.51, 0.61), with validation accuracies greater than random chance for 3 of the 14 participants. The MLPNN model was found to be statistically greater in CV accuracy than the TCN model, as evidenced by the 95% confidence interval of the model accuracy difference between the two classifiers not containing 0, i.e. model accuracy difference of 8% (95% CI: 0.01, 0.15), and the MLPNN also had statistically more participants with validation accuracies greater than random chance, as evidenced by the McNemar's test statistic of $4.5 \geq 3.84$ ($p < .034$, $\alpha = 0.05$). The novel TCN-AE was explored to evaluate the effectiveness of utilizing the latent space for classification of the vigilance decrement, with the rationale being that the reduced dimensionality of the distilled latent space would contain only the most salient features of the EEG signal. However, the TCN-AE did not achieve accuracy greater than random chance, with 7-fold CV balanced accuracy of 52% (95% CI: 0.47, 0.57), and no participants with validation accuracies greater than random chance. The MLPNN outperforming both the TCN and TCN-AE models suggests that frequency-domain models may outperform time-domain based models for the purpose of vigilance decrement detection due to the salient information contained within spectral features. Overall, these results demonstrate that it is possible to classify a vigilance decrement using EEG, even with EEG from an unseen individual and unseen task.

1.2 Structure

This dissertation has the following structure: Chapter II provides necessary background on EEG and deep learning that is relevant across all subsequent chapters. Chapters III - V are presented in scholarly article format using publications resulting from this dissertation research. Chapter III demonstrates both mathematically and empirically the effects of proper and improper partitioning of datasets when building EEG cross-participant models, and demonstrates why proper partitioning is necessary in order to avoid underestimation of model error rates. Mathematically these effects are demonstrated both using Shimodaira's loss rescaling equation to demonstrate the presence of covariate shift in improper partitioning, as well as by using a novel method of data transformation which reduces inter-participant variability within EEG and subsequently also reduces the presence of covariate shift and thus increases model performance. Empirically these effects were demonstrated using five publicly available datasets to train models using both improper and proper methods of dataset partitioning, with results between each method compared for each dataset. Chapter IV investigates detection of visual search using machine learning and EEG signals, and also mitigation of inefficient visual search through the application of various techniques. The design and execution of a visual search experiment is presented which collected EEG and other physiological measures during both inefficient and efficient search as mitigation techniques were applied, and machine learning models are presented which utilized the EEG signals for classification of inefficient and efficient visual search. Chapter V explores the ability to detect the vigilance decrement in EEG from unseen participants performing an unseen task, with both frequency-domain and time-domain models presented which are capable of performing this classification with CV accuracies greater than random chance. Finally in Chapter VI, conclusions, contribution and findings, and future work are presented for each of the three studies.

II. Background

2.1 EEG

EEG analysis has been a useful tool in neuroscience for decades in both clinical settings and the medical research community. It has been demonstrated to be useful for numerous applications such as classifying sleep patterns, epilepsy, identifying patterns of attention deficit hyperactivity disorder (ADHD), levels of mental workload [38, 39], and emotion recognition [11]. EEG has also been useful for neural engineering with Brain–Machine Interfaces (BMIs), primarily due to EEG being used in combination with machine learning. Over the past decade, deep learning (DL) has been increasingly used to improve performance within models, allowing for automatic end-to-end processing and classification of the data, to include feature extraction using sequence models.

Feature selection for machine learning is an important aspect of EEG preprocessing, with three common types of input features: calculated features, images, and signal values [12]. Calculated features are the result of methods that extract various features from the EEG signals. These could be statistical measures from the signal such as mean, standard deviation, variance, kurtosis, and entropy. They could also be measures in the time-frequency domain, such as the power spectral density (PSD), the mean power of the frequency bands and their ratios, wavelet decomposition, and functional coherence values. Images for EEG feature selection are typically the result of some form of post-processing that is also commonly used for data visualization, such as spectograms, fast Fourier transform (FFT) maps, and color scale topography; however these can also be hand-crafted solely for the purpose of deep learning. Lastly, the EEG signal values can be utilized for deep learning, allowing the network to extract its own meaningful features from the voltage values themselves. Raw voltage values are the most commonly used features, but averaged voltage values are also used. A recent literature review of 90 EEG deep learning studies found that 41% of their reviewed studies used some kind of calculated feature from the EEG signal, 39% used the EEG signal values, and 20% used images [12]. PSD, wavelet decomposition, and statistical measures were found to be the three most common input types of calculated features, spectograms and FFT maps the most common of images, and raw values the most common of signal values. Similarly, another recent EEG deep learning literature review of 154 studies found that out of the 98% of studies which identified their model's input type, half used raw EEG data, while the other half used hand-crafted features, with the majority of those being frequency domain features [11].

Figure 1 shows some of the recent trends in the number of EEG DL published papers, categorized by various domains, and worth noting is the substantial increase in the number of published papers per year since 2016. This use of DL has lead to models capable of classifying EEG signals for the applications of those listed above, as well as additional applications such as motor imagery for the use of BMIs. The main advantage to using EEG is that there is evidence that the electrical activity of the brain picked up through the scalp can contain aspects of cognitive processes, mental activities, and cognitive load [10]. Another advantage is that EEG provides excellent temporal resolution versus other physiological metrics [40], which is a boon for low latency augmented cognition, and scenarios where near real-time adaptive assistance is needed.

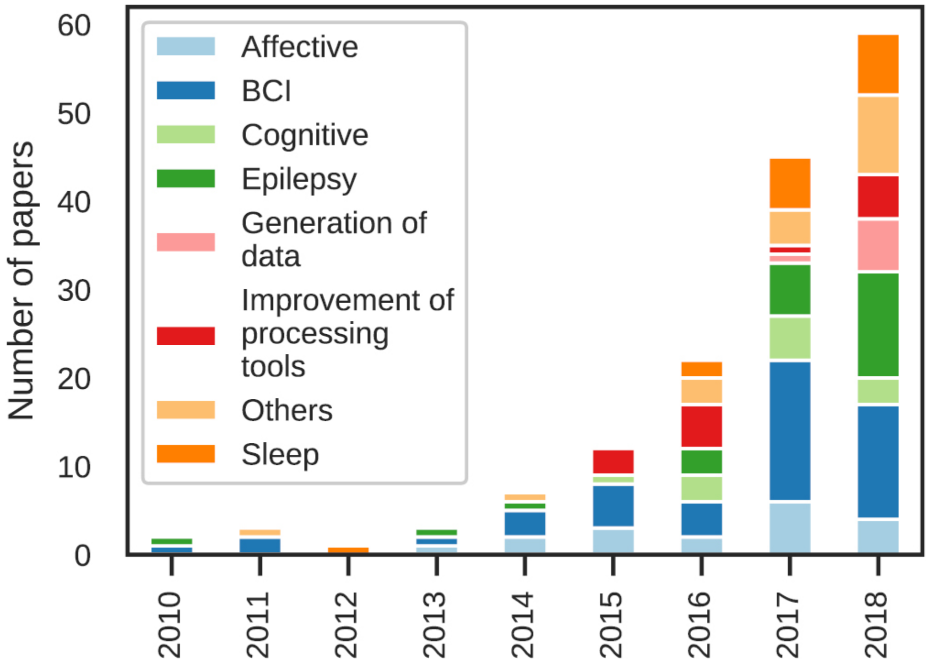


Figure 1. Trend of the number of EEG DL related published papers per domain per year, from 2010 through 2018 [11].

A disadvantage to using EEG as a feature for machine learning is that it has low spatial resolution, meaning that the electrical activity picked up at the scalp cannot be well correlated to a specific location in the brain, due to smearing from the tissue surrounding the brain, such as the skull, scalp, etc [11]. An alternative physiological measure to EEG that does not have this drawback is functional near-infrared spectroscopy (fNIRS). While EEG is based upon the electrical activity of the brain, fNIRS instead uses the physiological property of NeuroVascular Coupling (NVC), which is the connection between neurons and their vascular supply, meaning there is a connection between the change in regional cerebral blood flow, oxygenated hemoglobin (Hboxy), and deoxygenated hemoglobin (Hbdeoxy) induced by neuronal activation [41]. Using this property, fNIRS can be used to infer brain activity, as it measures blood oxygenation levels within the brain. This measure has good spatial resolution as it measures the location of the recorded activity, however it has poor temporal resolution, as there is a lag time inherent in hemodynamics. Due to these properties, EEG and fNIRS can be seen as complementary, as one has good temporal resolution, and the other good spatial resolution [42, 43]. Recently, the integration of EEG and fNIRS data has also led to improvements upon state of the art models for BMI interfaces [44], and this data fusion holds potential for other applications as well.

2.1.1 Non-Stationarity and Individual Differences.

One of the significant challenges associated with EEG analysis and classification is that EEG is both non-stationary [17] and that there are individual differences in EEG signals across individuals that result in inter-participant variability [18]. EEG non-stationarity is due to a variety of internal and external causes, such as brain activity causing continual changes in states of neuronal assemblies [19], user attention levels, user fatigue, sensor equipment used, and scalp placement of electrodes [20]. Similar to non-stationarity, the individual differences in EEG signals are also due to a variety of factors, such as differences in variability in frequency peaks for individuals due to differences in personality traits [21], genetic variations [22–24], gamma–aminobutyric acid concentrations in the brain [25, 26], and memory task performance [27].

These individual differences are underlying shifting covariates across participants, and they result in a change in the input distributions across all participants, while the conditional distribution of the output class, y, given the input feature vector, x, stays the same, resulting in a covariate shift for cross-participant machine learning models when they are tested upon EEG from participants that the model has not seen [28, 29]. Thus, because of this inherent inter-participant variability in EEG signals, different strategies need to be used when performing EEG analysis [18] and training of cross-participant models [45].

There are two main categories of approaches for dealing with non-stationary environments in machine learning, namely passive and active approaches [46]. Passive approaches assume that the input distribution will be shifting over time due to the non-stationary environment of the selected domain. With this assumption, passive schemes adapt to shifts in the data through either careful architecture selection, finding invariant representation between the training and testing datasets through domain adaptation with conditional transferable components [47], dimensionality reduction [48], or a number of other adaptation techniques [20]. Recent passive approaches in EEG-based classifiers that have seen improvements in classification performance over previous approaches are ensemble classifiers (bagging, boosting, random subspace [45, 49–51] and dynamically weighted ensemble classification [52]) and latent factor decoding of autoencoder networks [53]. Ensemble methods provide for better generalization over a single classifier [54], which is especially useful when the training data is insufficient in covering the scope of the domain, such as in non-stationary environments. Similarly, latent factor decoding of autoencoders also provides for improved generalization of non-stationary environments due to the autoencoder's ability to encode the high-dimensional input and its distribution to a compressed representation [53]. While both ensemble classifiers and autoencoder networks are state of the art passive approaches, ensemble classifiers still have the significant challenge of selecting the optimal number of classifiers, as well as the significant computational costs of increasing the number of classifiers [45].

Active approaches differ from passive approaches in that they wait to utilize a domain adaptation technique until a shift in the input distribution has been detected through a covariate shift detection technique [45, 55]. Recent domain adaptation techniques within active approaches that have had improved performance over passive approaches are knowledge base reconfiguration techniques using forms of semi-supervised learning (SSL) [56]. In knowledge base reconfiguration, the knowledge base is continually updated as data is streamed in using SSL techniques such as transductive learning [50], and once a covariate shift has been detected, the classifier is then retrained on the updated knowledge base. In transductive learning, once a covariate shift has been detected, then a probabilistic K-nearest neighbor (KNN) model (built upon the known labels of the training data) is used to determine a confidence ratio in whether the new data point streamed in belongs to a new class or an existing class; if the confidence ratio is greater than some threshold, then the data point is added to the new knowledge base, otherwise the data point is discarded. A recent advancement to this technique has been unsupervised adaptive ensemble learning (UAEL), in which the knowledge base is continually updated using the transductive learning knowledge base reconfiguration technique, but instead of retraining the single classifier on the updated knowledge base whenever there is a covariate shift detection, there is instead a dynamic ensemble of classifiers, where classifiers are newly added to the ensemble and trained on the updated knowledge base whenever there is a covariate shift detection [45].

While Raza et al. claim that these recent advances in active approaches outperform passive approaches in single-trial EEG classification [45, 50], there are some disadvantages as well. Active approaches which make use of unlabeled data for the knowledge base update must make some assumptions about the underlying distribution of the data, and the SSL approach must satisfy at least one of the following assumptions such as smoothness, cluster, or manifold assumptions [45]. For transductive learning, the smoothness assumption is made, which makes the assumption that points which are close to one another are more likely to share the same label (hence its use of KNN). Another difference which can be a disadvantage is that active approaches are relying on the accuracy of the covariate shift detection technique, while passive approaches always assume that the input distribution is shifting over time. Additionally, active approaches are also relying on the accuracy of the SSL technique used to determine when to update the knowledge base. Furthermore, Raza et al. used the Brain-Computer Interface (BCI) competition IV dataset [57] to make these claims, which is comprised of short duration (<10 minutes) within-session data, for their recent research in [45, 50]. The models were also within-participant models. This means their results apply to covariate shift in short duration within-session within-participant data, but not necessarily covariate shift in cross-session or cross-participant data.

2.2 Deep Learning

In recent years, deep learning has allowed humans to enjoy computer systems that are capable of achieving and even exceeding human-levels of accuracy for tasks that the machine learning community has been unable to solve for decades. Most notable of these tasks are ones that we as humans perform on a daily basis, such as speech recognition, image recognition, and natural language processing [58]. Deep learning has been able to achieve these breakthroughs primarily due to more availability in data, more availability in computational power through the use of powerful graphics processing units (GPUs) that are better suited for the matrix multiplication calculations needed for training a neural network, and advances in algorithms and architectures within deep learning. At its core, an artificial neural network is able to extract from the data its own concepts/features, and does this in a way that allows it to extract complex concepts out of simpler concepts, using layers of perceptrons to form the multilayer perceptron (MLP). These perceptrons are simple mathematical functions themselves, and a single layer of combining many of them allows for a mapping of some set of input values to some set of output values. When these layers are combined and backpropagation is utilized for learning, then we have the common artificial neural network that can learn its own concepts, and extract complex concepts from simpler concepts. Figure 2 provides a visual example of this for the application of image recognition [58].

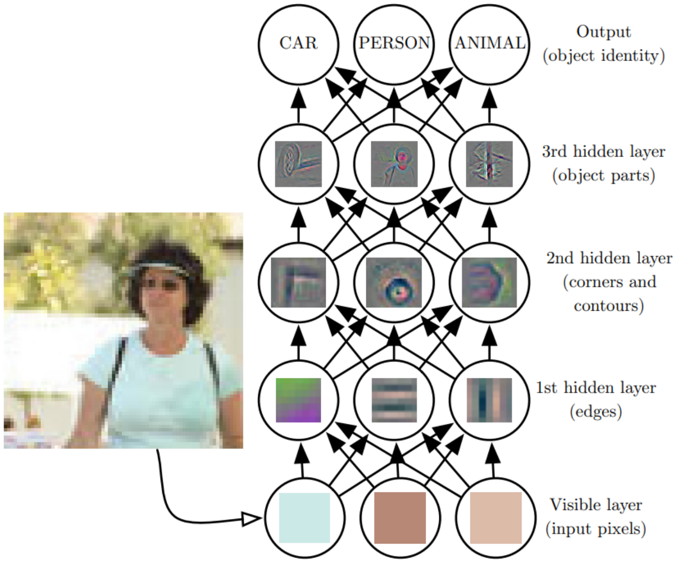


Figure 2. Visual illustration of an artificial neural network extracting complex concepts from simpler concepts for image recognition [58].

The recent trend of deep learning has also applied to very recent advancements in EEG signal classification, with the numbers of published papers increasing significantly in the early 2010s, similar to Figure 1 which shows this for a number of EEG domains [11, 12]. As mentioned in Section 2.2, numerous tasks have been classified using EEG in recent years. While deep learning is prized on its ability to extract and learn features from the data itself, many applications using EEG signals for deep learning still perform some form of feature extraction first. Approximately 41% still use calculated features for input to the neural network, with 20% using images (such as spectograms, FFT mappings, etc.), and 39% using the raw signal values [12]. Of the portion that use extracted features, the most common features extracted are power spectral density (PSD), wavelet decomposition, and statistical measures of the signal such as mean and standard deviation, with almost 50% using power spectral density. Initial feature extraction in deep learning allows for a reduction in complexity of the data, and is helpful if there is information known about the data a priori that lends to why those features were extracted, and for EEG specifically, frequency domain analysis has been correlated with behavioral patterns [59].

2.2.1 Recurrent Neural Networks.

Recurrent Neural Networks (RNNS) are a family of neural network architectures that are used to process sequential data. In a traditional ANN, we assume that all inputs are independent of one another and don't share some type of relationship with the other inputs. However in an RNN, we instead know that the data is sequential in nature and thus, these sequential inputs can contain information for preceding/succeeding inputs (e.g. words, sentences, etc.) [58]. To take advantage of the information contained within the sequence, a hidden state is utilized which represents preceding information and allows the input at one timestep to affect the output at a later timestep, propagating the sequence information across time through the hidden state. Figure 3 gives a visual example of both a recursive computational graph for an RNN (left) and an unfolded computational graph for an RNN (right). This unfolded computational graph illustrates how the input $x^{(t)}$, hidden state $h^{(t)}$, and output $o^{(t)}$, are characterized across time steps t. Corresponding to each of these is a weight matrix, U, W, and V, respectively, that leverages the concept of parameter sharing by having these weight matrices shared across all time steps, allowing the network to generalize across time and sequences not seen during training.

A significant issue with traditional RNN's however is the challenge of learning long-term dependencies. Gradients back-propagated over many time steps either vanish or explode. Exploding gradients can be primarily solved through clipping [60], however vanishing gradients are more difficult to solve, with a traditional RNN having significant issues for sequences of length of 10-20, where gradients typically trend towards 0 at those lengths [58]. Proper initialization of the W matrix, use of regularization, and using the Rectified Linear Unit (ReLU) activation function as opposed to a tanh or sigmoid activation function, all can help reduce the effect of vanishing gradients. While the ReLU has become the popular default activaition for neural networks and isn't as likely to see the effects of vanishing

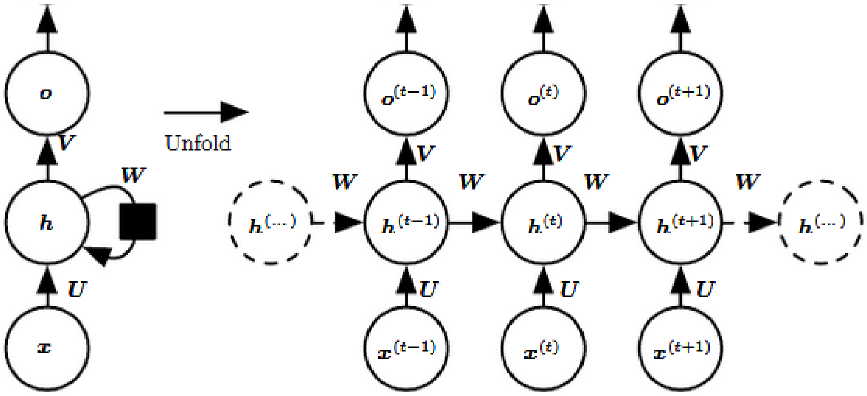


Figure 3. A visual representation of an RNN using computational graphs. Here $x^{(t)}$ represents the input, $h^{(t)}$ the hidden state, and $o^{(t)}$ the output. Corresponding to each of these is a weight matrix, U, W, and V, respectively. Left: The computational graph with cyclical connections. Right: The same computational graph but unfolded to show each individual time step [58].

gradients due to its derivative being either 0 or 1 [58], advanced RNN architectures have better solved these issues.

2.2.1.1 Gated RNNs: Long-Short Term Memory models & Gated Recurrent Units.

The most recent and effective solution to the issue of long-term dependencies in RNNs is to use gated RNNs [58]. A Gated RNN refers to adding gates into the architecture which allow/disallow the flow of information from one time step to another through a memory bus, either preserving the information in memory, or forgetting it if it is no longer needed. Two of the most popular and effective Gated RNNs as of late are Long-Short Term Memory (LSTM) models [61] and Gated Recurrent Units (GRUs) [62]. While both have similar performance and are both effective in combating long-term dependencies, they each have slight advantages and disadvantages that can lead one to superior performance over the other, depending upon the domain and the amount of data [63, 64].

A visual illustration of the inner workings of an LSTM can be seen in Figure 4, with C_i^t denoting the memory state, h_t denoting the current state, and x_t denoting the current input [61]. The gates that an LSTM utilizes for its memory are a forget gate, an input gate, and an output gate [61], and as mentioned above, these gates allow information to flow from one time step to another (or conversely, not flow). The forget gate is a mechanism to decide whether or not to keep information from the previous time step, and it is determined by a sigmoid layer (σ) which outputs a number between 0 and 1. The sigmoid takes in the concatenated current state and previous output and uses that to determine its output of a value between 0 and 1, with 0 representing to forget all previous state information, and 1 to retain all previous state information. With the biases, input weights, and recurrent weights denoted as b, U, and W, respectively, the forget gate equation is given by:

$$f_t = \sigma\Bigg(b^f + U^f x_t + W^f h_{t-1}\Bigg). \tag{2.1}$$

Next, is the input gate i_t shown in Equation 2.2 which uses another sigmoid, and it determines whether or not to retain the information of the current input. This is then used in combination with the candidate update vector $\tilde{C}_t$ (Equation 2.3) to modify the memory state.

$$i_t = \sigma\Bigg(b^i + U^i x_t + W^i h_{t-1}\Bigg). \tag{2.2}$$

$$\tilde{C}_t = tanh\Bigg(b^g + U^g x_t + W^g h_{t-1}\Bigg). \tag{2.3}$$

Finally, the output gate shown in Equation 2.4 uses a final sigmoid which determines what will be output from the current cell state, and this is then used in combination with the update vector C_t (Equation 2.7) to determine the state h_t (Equation 2.6).

$$o_t = \sigma\Bigg(b^o + U^o x_t + W^o h_{t-1}\Bigg). \tag{2.4}$$

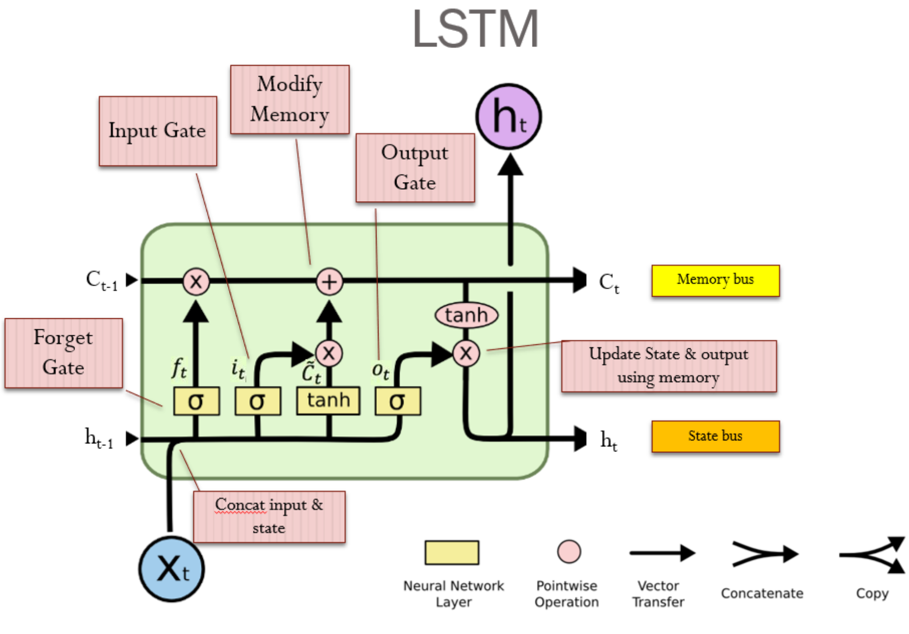


Figure 4. Visual illustration of the internal structure for an LSTM. $C_{(t)}$ represents the current memory state, $h_{(t)}$ represents the current state, and $x_{(t)}$ represents the current input. Note the labelled gates, which are all sigmoid activation functions [65].

$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right). \tag{2.5}$$

$$h_t = tanh(C_t) * o_t. \tag{2.6}$$

GRUs are similar to LSTMs albeit with a relatively more simple architecture, which can be seen in Figure 5 [62]. GRUs differ from LSTMs in that they only have one gating unit for both forgetting and updating the state unit, as opposed to LSTMs which have both a forget and input gate for a separate memory bus. This results in the update gate, z_t, following Equation 2.7 and the reset gate, r_t, following Equation 2.8. As can be seen, both the reset and update gates can each determine how much of the state vector to drop.

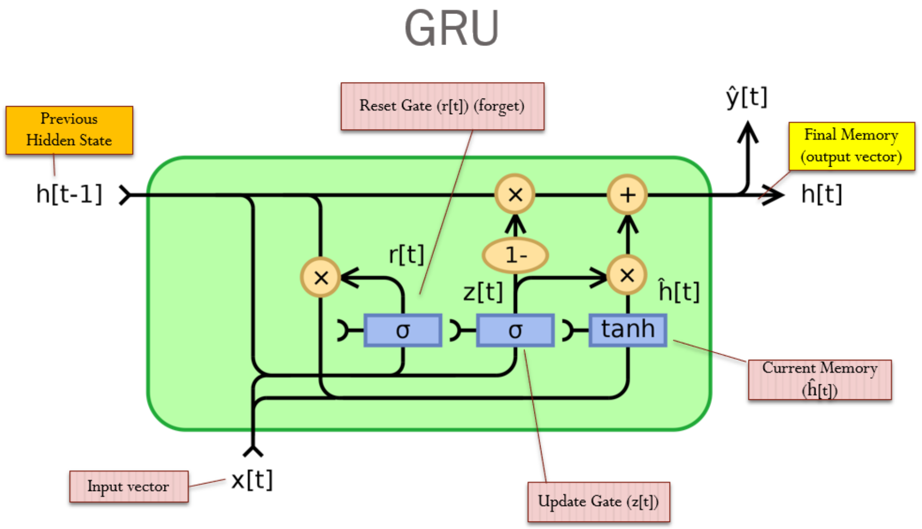


Figure 5. Visual illustration of the internal structure for a GRU.

$$z_t = \sigma\left(b^z + U^z x_t + W^z h_{t-1}\right). \tag{2.7}$$

$$r_t = \sigma\left(b^r + U^r x_t + W^r h_{t-1}\right). \tag{2.8}$$

First the update gate, z_t, determines what information is to be passed along to the future. Next, the reset gate, r_t, determines the amount of past information to forget. The candidate update vector, $\tilde{h}_t$, shown in Equation 2.9 is then determined from the input vector x_t, and the decision of the reset gate, r_t, for how much past information is still relevant enough to store. Lastly, the final state to be passed on is determined using the result from the update gate, z_t, and the candidate update vector, $\tilde{h}_t$, as shown in Equation 2.10.

$$\tilde{h}_t = tanh\left(b^h + r_t * U^h h_{t-1} + W^h x_t\right). \tag{2.9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t. \tag{2.10}$$

Due to the differences in their architectures, there are different advantages to both LSTMs and GRUs, and some datasets will perform better using one over the other. One advantage GRUs have over LSTMs is that they are more efficient and converge faster than LSTMs [63, 64]. GRUs also, in general, require less data to generalize, so depending on the amount of data, GRUs may perform better than LSTMs. Alternatively, with adequate amounts of data, LSTM's greater expressive power through its ability to remember longer sequences, could result in superior performance over GRUs. Additionally, if the data has lengthy long-term dependencies within its sequences, LSTMs may also perform better.

While Chung et al. found empirically that GRUs can have these advantages over LSTMs [63], Craik et al. found that papers comparing LSTMs to GRUs for EEG applications typically found LSTMs to have superior performance [12]. However, GRUs are still relatively new and the vast majority of EEG RNN research utilizes LSTMs over GRUs [12]; this is possibly by default given LSTMs precedent in the literature, as the vast majority of LSTM research papers are absent a comparison between the two architectures, or fail to mention why LSTMs were selected over GRUs for their RNN architecture. Given these findings, its important to consider and at least preliminarily test both architectures for a given dataset, as it would be difficult to know which will perform better for the data at hand through a priori selection criteria.

2.2.2 Temporal Convolutional Networks.

A temporal convolutional network (TCN) is a type of CNN for 1D sequence data and was recently developed by Bai et al. [66]. A TCN utilizes dilated convolutions in order to process a sequence of any length, without having a lengthy memory history such as the case with LSTMs. TCNs are typically causal, meaning there is no information leakage from the future to the past, however they can be non-causal as well. The primary elements of a TCN consist of the dilation factor d, the number of filters n, and the kernel size k.

The dilation factor controls how deep the network is, with dilations typically consisting of a list of multiples of two. Figure 6 provides a visual example of a non-causal TCN and aids in understanding the dilated convolutions on a sequence, with the dilation list in the figure being [1,2,4,8]. The kernel size controls the volume of the sequence to be considered within the convolutions, with Figure 6 showing a kernel size of 2. Finally, the filters are similar as they are in a standard CNN, and can be thought of as the number of features to extract from the sequence.

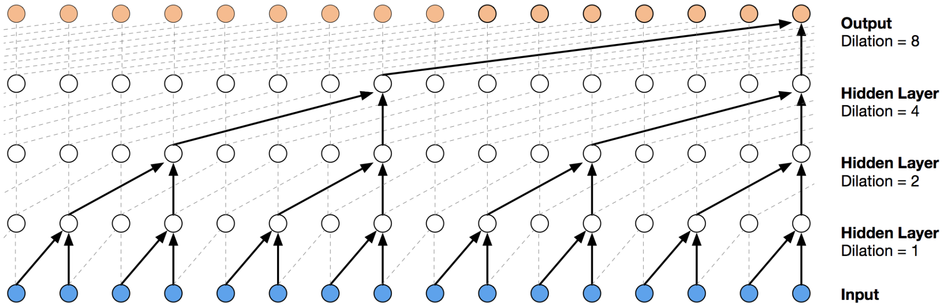


Figure 6. Visual illustration of a causal TCN. This TCN has a block size of 1, a dilation list [1,2,4,8] (i.e. dilation factor 8), and a kernel size of 2 [67].

These elements combined form a block as in Figure 6, and blocks can be stacked as they are in Figure 7. This increases what is called the receptive field, which is the total length that the TCN captures in processing, and is a function of the number of TCN blocks, the kernel size, and the final dilation, as shown in Equation 2.11. It is common to have a receptive field which matches the input sequence length, however the receptive field is flexible and can be designed to process any length, which is a primary advantage to TCNs. Other advantages include that they are able to train faster than LSTMs/GRUs of similar length, have a longer memory than LSTMs/GRUs when capacity of the networks is equivalent, and have been shown to have similar or better performance than LSTMs/GRUs on a number of sequence related datasets [66, 68]

$$R_{field} = 1 + 2(K_{size} - 1) * N_{blocks} * d_{final}. \tag{2.11}$$

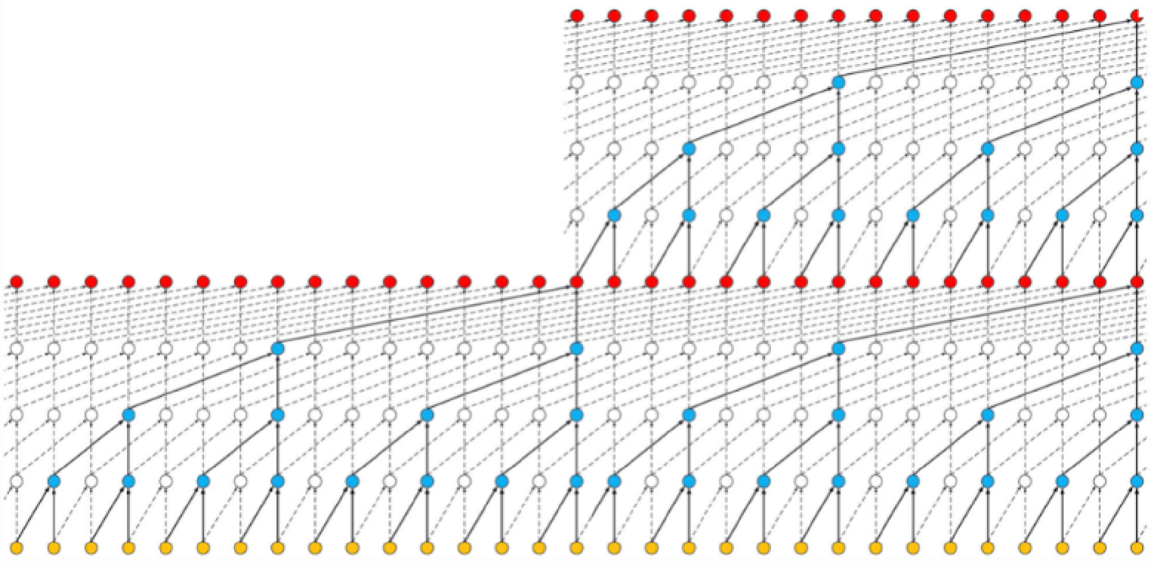


Figure 7. Visual illustration of a causal TCN with stacked blocks. This TCN has a block size of 2, a dilation list [1,2,4,8] (i.e. dilation factor 8), and a kernel size of 2 [68].

2.2.3 Autoencoders.

An autoencoder is a type of neural network architecture for unsupervised learning that is primarily used for reproduction of what is input into the network [58]. This is done through the use of two separate networks. One network named the *encoder* $f(\mathbf{x})$ compresses the input into a lower-dimensional representation called the *code* or the *latent-space* $\mathbf{h} = \mathbf{f}(\mathbf{x})$, and another network named the *decoder* reconstructs the input from the code $\mathbf{r} = g(\mathbf{h})$. An example of a standard autoencoder architecture can be seen in Figure 8. Because of the nature of the encoder, autoencoders are useful for dimensionality reduction, are powerful feature detectors, and can also be used for unsupervised pretraining of deep neural networks [69].

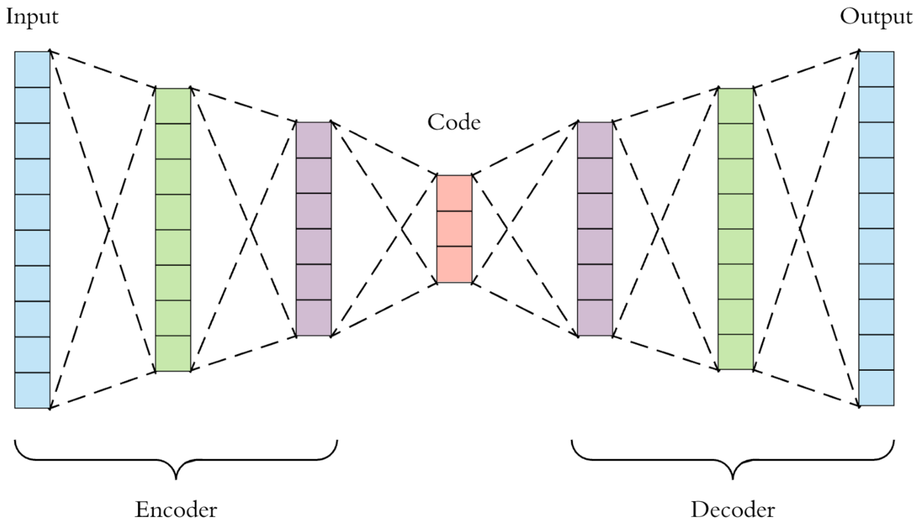


Figure 8. Visual representation of a standard autoencoder architecture.

In Figure 8 the code **h** is constrained to have smaller dimension than the input **x**. This is called being *undercomplete* and is typical of an autoencoder, as it forces the autoencoder to capture the most salient features of the training data, and thus the autoencoder doesn't overfit the training data and copy it perfectly, as this would not be useful [58]. A standard undercomplete autoencoder uses the loss function found in Equation 2.12. If the decoder is linear and L is the mean squared error, then the results of this undercomplete autoencoder are equivalent to the results of Principal Component Analysis (PCA). However, if the encoder and decoder are nonlinear functions, then the autoencoder can learn more powerful nonlinear generalizations of PCA.

$$L(\mathbf{x}, g(f(\mathbf{x}))) \tag{2.12}$$

Regularization is often used to prevent the autoencoder from having too much capacity and perfectly replicating the input. In the case of the sparse autoencoder, this takes the form of a sparsity penalty $\Omega(\mathbf{h})$ added to the reconstruction error, resulting in a generic expression given in 2.13 [58]. This sparsity penalty can be constructed in a number of ways, with two common methods being L1 regularization and KL-divergence. The penalty of Ω can also take different forms, resulting in different architectures. An example of this is the contractive autoencoder (CAE), in which the model is resistant to large changes when $\mathbf{x}$ changes slightly, due to the penalty Ω taking the form of Equation 2.14 [70]. This penalty is the squared Frobenius norm (sum of squared elements) of the Jacobian matrix of partial derivatives, and it encourages the derivatives of f to be as small as possible. Denoising autoencoders are another form of regularized autoencoder. Instead of being trained to reconstruct the given input, these autoencoders reconstruct the input minus a noise component that is added into the original inputs. By training the autoencoder in this manner, the network can learn a representation that removes noise from the input, leading to applications of denoising images, signals, etc [58].

$$L(\mathbf{x}, g(f(\mathbf{x}))) + \Omega(\mathbf{h}). \tag{2.13}$$

$$\Omega(\mathbf{h}) = \lambda \left\| \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right\|_F^2 . \tag{2.14}$$

The following chapters utilize this background material to develop deep learning models which use EEG signals. The next chapter explores EEG inter-participant variability and its effects on the presence of covariate shift, as well as develops machine learning models for both improper and proper methods of dataset partitioning for EEG cross-participant models. The following chapters III-V are also presented in scholarly article format, presenting papers which have already been published or are already submitted to various sources of publication.

III. The Effects of Individual Differences, Non-Stationarity, and the Importance of Data Partitioning Decisions for Training and Testing of EEG Cross-Participant Models

3.1 Paper Overview

This paper was published in the MDPI Sensors journal special issue Intelligent Biosignal Analysis Methods on 6 May, 2021 [71]. It demonstrated the importance of following proper dataset partitioning guidelines for training and evaluation of EEG-based cross-participant models. This is significant because without proper dataset partitioning, EEG-based cross-participant models boast unrealistic performance metrics, with empirical results showing that error rates of EEG-based cross-participant models can be underestimated between 35% to 3900%, pointing to a significant and widespread issue in the body of research surrounding model creation using EEG signals.

3.2 Abstract

EEG-based deep learning models have trended toward models that are designed to perform classification on any individual (cross-participant models). However, because EEG varies across participants due to non-stationarity and individual differences, certain guidelines must be followed for partitioning data into training, validation, and testing sets, in order for cross-participant models to avoid overestimation of model accuracy. Despite this necessity, the majority of EEG-based cross-participant models have not adopted such guidelines. Furthermore, some data repositories may unwittingly contribute to the problem by providing partitioned test and non-test datasets for reasons such as competition support. In this study, we demonstrate how improper dataset partitioning and the resulting improper training, validation, and testing of a cross-participant model leads to overestimated model

accuracy. We demonstrate this mathematically, and empirically, using five publicly available datasets. To build the cross-participant models for these datasets, we replicate published results and demonstrate how the model accuracies are significantly reduced when proper EEG cross-participant model guidelines are followed. Our empirical results show that by not following these guidelines, error rates of cross-participant models can be underestimated between 35% and 3900%. This misrepresentation of model performance for the general population potentially slows scientific progress toward truly high-performing classification models.

3.3 Introduction

EEG analysis has been a useful tool in neuroscience for decades in both clinical settings and the medical research community, proving to be useful for numerous applications such as classifying sleep patterns, epilepsy, identifying patterns of attention deficit hyperactivity disorder (ADHD), levels of mental workload [38, 39], and emotion recognition [11]. EEG has also been useful for neural engineering with Brain–Machine Interfaces (BMIs), primarily due to EEG being used in combination with machine learning. Over the past decade, deep learning (DL) has been increasingly used to improve performance within models, allowing for automatic end-to-end processing and classification of the data, to include feature extraction using sequence models. Despite these improvements in model selection, the challenges of EEG's non-stationarity and inter-participant variability are still present [17, 18].

One of the most significant challenges in building EEG classification models that are intended for use on any individual's EEG (cross-participant model) is accounting for the covariate shift that occurs due to EEG's non-stationarity and inter-participant variability [28, 29, 31–33]. Covariate shift in machine learning is a difference in the input distributions of the training and testing datasets [30]. This difference can significantly affect model performance, as a general guideline and assumption that is used in supervised machine learning is that these two input distributions are independent and identically distributed (i.i.d. assumption). Without this assumption, many theoretical guarantees and bounds on minimizing the test error are lost. For EEG cross-participant classification models, this covariate shift and its effects will always be present when the model classifies EEG data belonging to a participant that the model has not seen. However, models should be tested with data, which is representative of the data they will predict upon in the real world, and thus, EEG cross-participant models should be tested with unseen participants. Therefore, as a best practice in reporting accurate model performance for models intended to classify any individual's EEG, EEG cross-participant models should always be validated and tested using EEG data that comes from participants the model has not trained upon.

Despite previous work showing that EEG has inter-participant variability [18], and that this inter-participant variability leads to covariate shift when EEG models are tested with an unseen participant [28, 29, 31–33], the majority of EEG studies built to classify any individual's EEG do not follow this best practice of testing the model with unseen participants. In a recent literature review of deep learning-based EEG models by Roy et al., only 23 out of 108 cross-participant models utilized some method of proper dataset partitioning to ensure the model was tested with a participant that was not used for training [11]. This same literature review also compared the number of studies exploring models built for a specific individual (within-participant) versus cross-participant, and they found that since 2016, the growing trend has shifted toward building cross-participant models, with the latest ratio of studies researching cross-participant models to within-participant models being over 5:1 [11]. With this ever-growing popularity in EEG cross-participant models, it is critical that the body of research corrects its trend by properly using EEG data from unseen participants for validation and testing. By not following this best practice, the research pool may become increasingly diluted with studies reporting model performance metrics that are unrealistic and unrepresentative of the model's true ability. Additionally, data repositories that split data into training and testing datasets prior to being made available for download, such as Kaggle [72] and the University of California, Irvine (UCI) machine learning data repository [73], should also take this best practice into account. In this paper, we aim to present to the reader the importance of proper dataset partitioning.

This paper has the following structure. First, in Section 3.4, a well-established background is presented to ground the reader in regard to covariate shift and inter-participant variability within EEG; then, we fully articulate the problem of improper dataset partitioning using this background knowledge. Next, in Section 3.5, we demonstrate the effects of covariate shift and inter-participant variability both mathematically and in simple models, presenting evidence for the effects of these phenomena at a fundamental level. Finally, in Section 3.6, we utilize five publicly available datasets to present empirically the difference in model performance when following and not following this best practice of proper model validation and testing. We close with discussion in Section 3.7 and conclusions and future work in Section 3.8.

3.4 Background

3.4.1 Covariate Shift.

For supervised machine learning, a standard guideline is that the training input distribution $P_{TR}(x)$ is equivalent to the test input distribution $P_{TE}(x)$ [30]. However, when these two distributions are not equivalent $P_{TR}(x) \neq P_{TE}(x)$, then there is typically a decrease in performance for most machine learning models. This form of dataset shift is referred to as covariate shift. This can happen for a number of reasons, such as the training and testing data being drawn from different populations, a lack of randomness in the number of trials/observations, an inadequate amount of them, or other biased sampling measures;

in the case of EEG, covariate shift is due to individual differences and non-stationarity [33, 51].

Below, in Figure 9, we see a simple example of covariate shift. Here, there is a classification boundary between two different classes, one represented by circles, and the other represented by triangles, with the classification boundary following the function $y = -x^3$ The training dataset is marked in red and the test dataset is marked in blue. If we train a machine learning algorithm on only the training dataset and then test it on similar data such that $P_{TR}(x) = P_{TE}(x)$, then the model will be able to perform very well when tested, since the classification boundary is well defined between the two classes. In fact, many functions could easily define a reasonable boundary in this case; for example, $y = \frac{x^2}{3}$ or $y = 2|x|$ would yield good performance at discriminating the two classes of the training set shown in Figure 9. However, if we trained the model using only the red training data and tested with the blue testing data, the machine learning algorithm would have been trained with different data than it would be tested with ($P_{TR}(x) \neq P_{TE}(x)$), and it is unlikely that during training, the machine learning algorithm would have been able to discover the more complicated underlying discriminator function $y = -x^3$ having used only the red training data. Thus, the model trained only on the training data would perform poorly for classification of the test data, because the data distribution of the features from the training data and the distribution of the features from the test data are different.

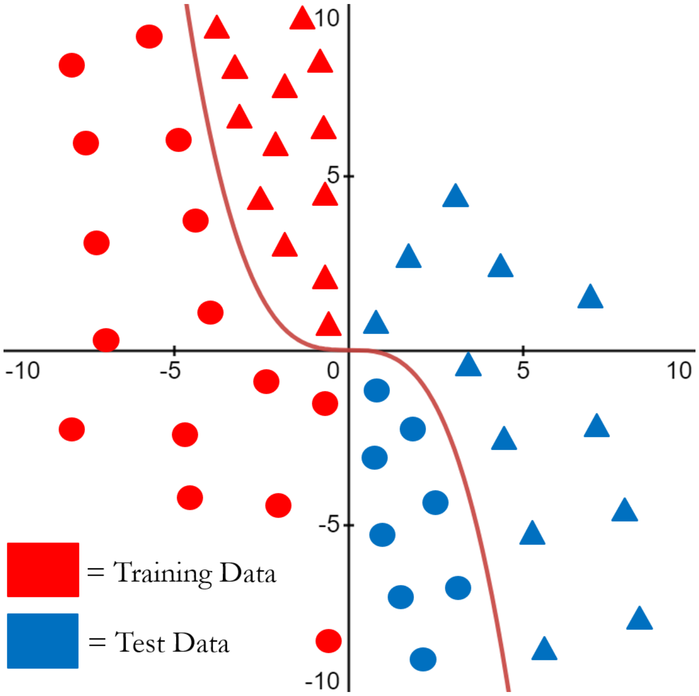


Figure 9. Simple example of covariate shift in classification data. Two classes of data are represented by circles and triangles, with the training dataset marked in red and the test dataset marked in blue. The true decision boundary between the two classes follows the function $y = -x^3$.

There are a number of different methods that can be used to detect if covariate shift is present due to the input distributions from two datasets being different. Given two datasets, $P_{TR}(x)$ and $P_{TE}(x)$, one method is to calculate how different the two probability distributions of the two datasets are,

$$D_{KL}(P_{TR}||P_{TE}) = \mathbb{E}_{x\sim P_{TR}}[log\frac{P_{TR}(X)}{P_{TE}(X)}] = \mathbb{E}_{x\sim P_{TR}}[log(P_{TR}(X)) - log(P_{TE}(X))] \quad (3.1)$$

Another method for covariate shift detection is through visualization of the distributions in low-dimensional space using dimensionality reduction techniques. Manifold learning techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE), multidimensional scaling (MDS), IsoMap, and others, are useful for this as they capture nonlinear information in the data [69]. t-SNE is an unsupervised machine learning algorithm that is widely used for data visualization as it is particularly sensitive to local structure and reduces the tendency to crowd points toward the center of low-dimensional space [74]. As an unsupervised machine learning algorithm, t-SNE does not use labels of data for its learning, and it solely uses the features of each observation to perform its algorithm. It does this by first constructing a probability distribution for all pairs of observations in high-dimensional space such that similar observations (observations that are closer to one another in feature space) are assigned a higher probability of being neighbors, and dissimilar observations (observations that are further apart in feature space) are assigned a lower probability of being neighbors. Then, a new dataset is created with the same number of observations, but it is now spread randomly in low-dimensional feature space. It uses a Student's *t*-distribution to compute the similarity between all pairs of observations in low-dimensional space to create a second probability distribution and then uses gradient descent to iteratively shift the observations such that the KL divergence between the two different distributions is minimized. The main limitations of t-SNE are that it is computationally expensive and that the algorithm uses a non-convex objective function (KL divergence minimized using gradient descent, but initiated randomly), meaning multiple executions of the algorithm can lead to different embeddings (mappings of high-dimensional space to low-dimensional space). The dimensions of t-SNE are also difficult to interpret, as they are arbitrary distances that represent that closer neighboring points in low-dimensional space are likely to be neighbors in high-dimensional space [75].

Figure 10 shows an example of previous work utilizing t-SNE for high-dimensional data visualization outside of the EEG domain, with t-SNE performed on the well-known MNIST dataset, with the clusters corresponding to different input distributions within the data, and the colors corresponding to different classes [74, 76].

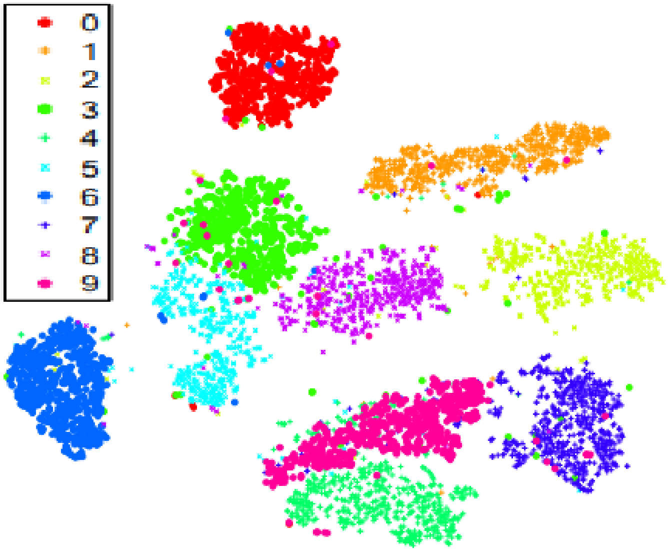


Figure 10. Example of 2D visualization using t-SNE on the MNIST dataset [74, 76]. The dimensions of t-SNE are arbitrary distances that represent that closer neighboring points in low-dimensional space are likely to be neighbors in high-dimensional space.

t-SNE can also be used to visually detect covariate shift. A common example of covariate shift is when the testing data is partitioned from a subset of the clusters (i.e., participants for EEG), and the training data is partitioned from a different and separate subset of clusters; e.g., if in Figure 10 class 0 (red) was selected as the test data and classes 1–9 were selected as the training data. Cluster analysis algorithms such as k-means clustering or fuzzy c-means clustering can be utilized to identify if the training and testing data belong to separate clusters [77]; however, a simpler method to detect this is through visual inspection of the t-SNE graph. One can separately label the training and test data in the graph (e.g., with different colors) and then visually inspect to see if the training and test data correspond to separate clusters within the graph (covariate shift). Visual inspection for clusters involves identifying that for the majority of observations in one class, the majority of the nearest neighbors for those observations also belong to the same class, with a clear boundary between its class (cluster) and another class, meaning there is little to no overlap. This simple method of visual inspection also provides the benefit of visualizing the high-dimensional data in 2D.

3.4.2 Non-Stationarity and Individual Differences.

One of the significant challenges associated with EEG analysis and classification is that EEG is both non-stationary [17] and that there are individual differences in EEG signals across individuals that result in inter-participant variability [18]. EEG non-stationarity is due to a variety of internal and external causes, such as brain activity causing continual changes in states of neuronal assemblies [19], user attention levels, user fatigue, sensor equipment used, and scalp placement of electrodes [20]. Similar to non-stationarity, the individual differences in EEG signals are also due to a variety of factors, such as differences in variability in frequency peaks for individuals due to differences in personality traits [21], genetic variations [22–24], gamma–aminobutyric acid concentrations in the brain [25, 26], and memory task performance [27].

These individual differences are underlying shifting covariates across participants, and they result in a change in the input distributions across all participants, while the conditional distribution of the output class y given the input feature vector x stays the same, resulting in a covariate shift for cross-participant machine learning models when they are tested upon EEG from participants that the model has not seen [28, 29]. Thus, because of this inherent inter-participant variability in EEG signals, different strategies need to be used when performing EEG analysis [18] and training of cross-participant models [45].

3.4.3 Approaches to Data and Problem Formulation.

When developing an EEG classification model, it is likely that it will belong to one of two main types of EEG models, either within-participant (a.k.a. intra-subject) or cross-participant (a.k.a. inter-subject) [11]. A within-participant model is one that intends to perform accurate classification of EEG for one individual and is thus built using only data from one participant. A cross-participant model is one that intends to perform classification on multiple individuals and is thus built using data from multiple participants. By training on data from multiple individuals, the goal is that the model becomes invariant to inter-participant variability, learning a function that accurately maps EEG input to the desired output label for most people. Additionally, cross-participant models can be built for different purposes and goals, such as for specific populations or for the general population. For example, the goal of a cross-participant BMI model could be to perform classification on only those specific individuals that use that specific BMI machinery. However, a more typical cross-participant model is one in which results are reported as though they are indicative of the model's ability to perform classification on the general population and thus any individual.

Each of these model types require different approaches to data partitioning across participants in order to report results that are accurate for their intended goal and target population. The within-participant model is more straightforward, as there is only one participant for both training, validation, and testing. However in cross-participant models, there are data from multiple participants, and because of the inter-participant variability that is inherent in EEG from individual to individual, how participants are used in cross-participant models for training, validation, and testing can have significant effects on model performance due to the differences in input distributions from individual to individual [28, 29]. For example, if a cross-participant model is tested using data from an unseen participant, then the model's classification performance will be reduced due to the resulting covariate shift of this individual's unseen data. If a cross-participant model is only intended to perform classification on the same population that it is training upon and not also unseen individuals, as is in some BMI models, then ensuring the model is tested with unseen individuals is not necessary. However, for cross-participant models in which the model is intended for the general population and therefore unseen individuals, data should be prepared such that *participants* that are used for training are not also used for validation or testing, and *participants* used for validation are not also used for testing; otherwise, the model's performance will not accurately reflect its intended purpose of classification upon unseen individuals. This means that if participant A is used for training, then not even a single observation from participant A should be used for validation or testing, and if participant B is used for validation, then not even a single observation from participant B should be used for testing. An example of this method of proper vs. improper dataset partitioning for general population cross-participant models is depicted in Figure 11. It is also worth noting that proper validation of general population cross-participant models does not exclude the use of cross-validation (CV) as a performance evaluation technique. Instead, CV merely needs to be modified so that for each fold, *participants* used in training are not also used for validation, such as a Leave-One-Participant-Out approach or a Leave- N-Participants-Out approach.

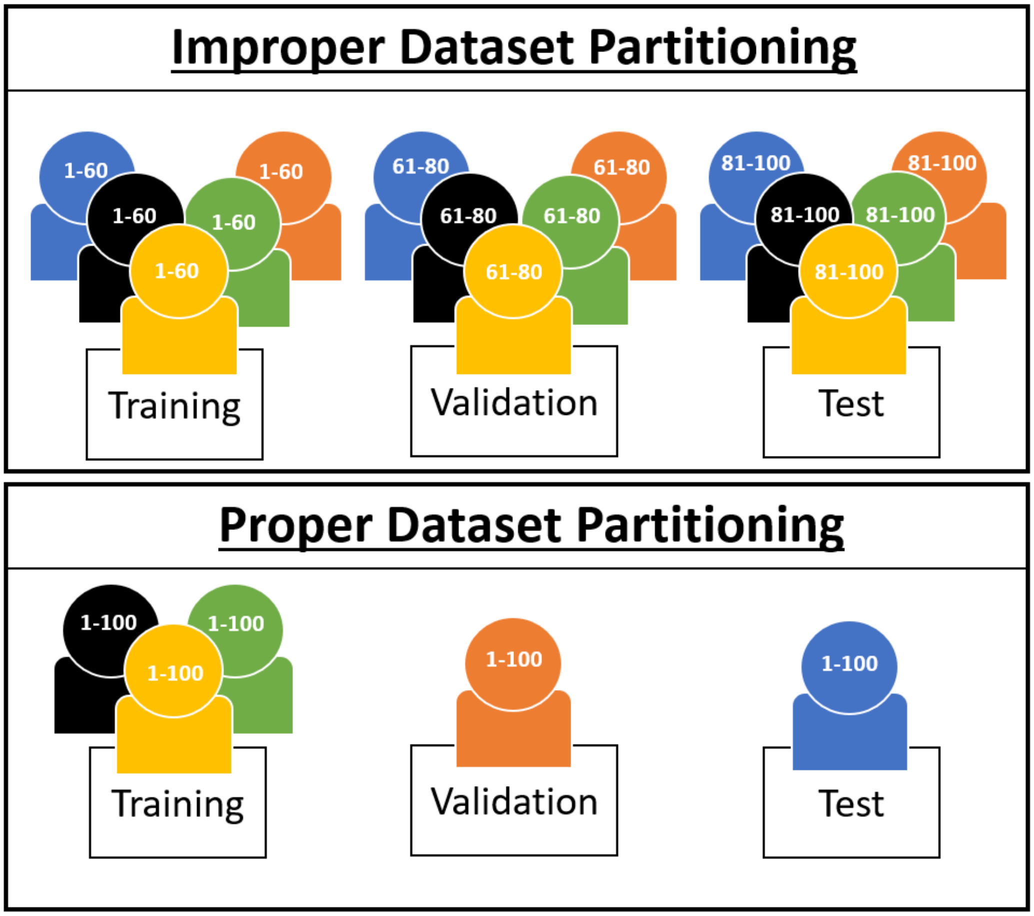


Figure 11. Two examples of creating the training, validation, and testing datasets with data from five participants. Numbers correspond to unique observations within each participant's dataset, with "1–60" referring to observations #1 through #60, "61–80" referring to observations #61 through #80, etc. The top illustrates improper dataset partitioning: data from each participant are used for all three datasets. In the top panel, while no unique observation is in more than one subset, each participants' data is still present in each subset. The bottom illustrates proper dataset partitioning: each participant's data are present in no more than one of the subsets.

Cross-participant models have significantly grown in popularity in recent years [11]; however, the majority of studies using cross-participant models do not follow this proper method of dataset partitioning. In Roy et al.'s literature review of deep learning-based EEG models, out of 108 studies using cross-participant models, only 23 utilized some method of proper dataset partitioning with a Leave-N-Participants-Out approach or a Leave-One-Participant-Out approach [11]. This results in the majority of studies having overestimated performance metrics—suggesting readers use models which, when used in scenarios involving the general population, may not perform as well as they were reported to have performed in the research. To obtain meaningful estimates of performance in the general population, cross-participant models need to follow proper dataset partitioning, as shown in Figure 11. Alternately, if the intent is not to use the model in the general population and is instead a tailored model designed for a specific population subset, the study should specifically state that the model's intended goal is only to perform classification upon the individuals it has been trained upon, to prevent the reader from incorrectly believing its efficacy would be similar in the general population.

In the following sections, we demonstrate in greater detail how covariate shift occurs, as well as its effects, in both simple model examples (Section 3.5), and in real-world, publicly available datasets (Section 3.6).

3.5 Initial Demonstration

To build understanding for how covariate shift manifests in any data, we utilize an initial demonstration of its effects in three settings: (1) first, we define covariate shift mathematically and illustrate how its effects on the expected loss of the test distribution can be accounted for; (2) next, we depict covariate shift using t-SNE, specifically using EEG data; and (3) finally, we demonstrate how we can affect covariate shift in EEG data by reducing the inter-participant variability through data transformations, thus increasing the model accuracy of properly validated EEG cross-participant models.

3.5.1 Defining and Estimating the Effects of Covariate Shift.

In order to understand covariate shift at its fundamental level, we first define supervised learning. Supervised learning is the task of learning a function $f(x)$, which maps an input vector x to a labeled output y, typically done by estimating the conditional probability $p(y|x)$ [58]. In order to estimate this function $f(x)$, a loss function $\ell(f(x),y)$ provides a measure of the difference between the true output y and the estimated $\hat{y}$ for the input vector x, with the loss function producing smaller values if $\hat{y}$ is correct and larger values if $\hat{y}$ is incorrect. Thus, the task of learning involves minimizing the expected loss of $\ell(f(\mathrm{x}),\mathrm{y})$ over the probability density $p(x,y|\lambda)$ (parameterized by λ), i.e., minimizing the loss $\ell(f(x),y)$ over all possible inputs x [78],

$$E_{(x,y)\sim p(x,y|\lambda)}[\ell(f(x),y)] = \iint \ell(f(x),y)p(x,y|\lambda)dxdy. \tag{3.2}$$

However, in practice, the distribution $p(x,y|\lambda)$ is unknown and thus replaced by the empirical distribution, which can be estimated from training samples. If there is the set of samples L drawn from $p(x,y|\lambda)$, then Equation 3.2 becomes the objective of minimizing the empirical loss [78],

$$E_{(x,y)\sim L}[\ell(f(x),y)] = \frac{1}{|L|}\sum_{(x,y)\varepsilon L}\ell(f(x),y). \tag{3.3}$$

After minimizing the empirical loss and a prediction model is learned, the model is tested with the set of test samples T drawn from $p(x,y|\lambda)$, where T does not contain any samples from L that were used to minimize the empirical loss.

If the training data and testing data are independently and identically distributed (i.i.d.), meaning that every single observation of training and testing data are sampled independently and from the same distribution of $p(x,y|\lambda)$, then we expect that minimizing the expected training loss will in general also minimize the expected test loss [78]. This is an assumption that is common for many predictive models and is referred to as the i.i.d. assumption. However, many models are developed under conditions such as non-stationary signals or covariate shift. In these conditions, the i.i.d. assumption no longer holds, as the training and testing data come from different distributions, e.g., $p(x,y|\lambda)$ (parameterized by λ) for the training data, and $p(x,y|\theta)$ (parameterized by θ) for the testing data. As we no longer have the assumption of i.i.d. data, then we can no longer expect that minimizing the expected training loss also in general minimizes the expected test loss,

$$\overset{argmin}{f} E_{(x,y)\sim p(x,y|\lambda)}[\ell(f(x),y)] \neq \overset{argmin}{f} E_{(x,y)\sim p(x,y|\theta)}[\ell(f(x),y)]. \tag{3.4}$$

One method to address this lack of minimizing the expected test loss under covariate shift is through loss rescaling. Shimodaira proposed that if the training and test distributions are known, that the expected test loss could be minimized by appropriately weighting the training loss for each x with instance-specific weights $\frac{p(x|\theta)}{p(x|\lambda)}$ [34, 78],

$$E_{(x,y)\sim\theta}[\ell(f(x),y)] = E_{(x,y)\sim\lambda}\left[\frac{p(x|\theta)}{p(x|\lambda)}\ell(f(x),y)\right]. \tag{3.5}$$

This loss rescaling results in larger loss values for instances of x where there are fewer training samples than test samples (weight ratio > 1), and smaller loss values for instances of x where there are more training samples than test samples (weight ratio < 1). Thus, in a dataset without covariate shift between training and test, more weight ratios' magnitudes would be close to unity because the features of the training data have a similar distribution to the features of the test data. Conversely, in a dataset with covariate shift between training and test, fewer ratios would be closer to 1.0, and more weight ratios would have magnitudes differing further from 1.0 because the training distribution and test distribution differ in their feature distributions.

While loss rescaling could be used to adjust machine learning performance outcomes, implementing loss rescaling can be difficult to achieve. As can be seen in Equation 3.5, for each instance of x with positive $p(x,y|\theta)$, there must also be a positive $p(x,y|\lambda)$; otherwise, there is a zero denominator, meaning this loss rescaling can only occur if the training distribution covers the entire support of the test distribution [78]. In high-dimensional data, it is more difficult to have this coverage due to the curse of dimensionality, i.e., that the sparsity of the data increases exponentially as the number of dimensions (e.g., number of features) increase. High-dimensional data are common in EEG datasets due to the nature of recording brain activity with high numbers of channels (i.e., scalp electrodes), and additionally, if spectral features are utilized, there are multiple frequency bands that could be extracted for each channel; it is not uncommon to collect spectral energy from five frequency bands across 64 electrodes for a total of 320 features in x.

While loss rescaling is unlikely to be useful for determining better estimates of performance in real-world EEG machine learning models, it can be useful for exploring effects of covariate shift in low-dimensional spaces. Next, we present a low-dimensional transformation of EEG datasets using Principal Components Analysis (PCA) in order to explore the performance differences between improper and proper partitioning of datasets for machine learning models.

We demonstrate the effects of these loss-rescaling weight ratio values $\frac{p(x|\theta)}{p(x|\lambda)}$ [34, 78] using the spectral features of the Driver Fatigue dataset [79] described in Section 3.6. First, the input vectors are log transformed to reduce skew, and the dataset is partitioned into two separate training and test datasets using the proper and improper methods:

- For improper dataset partitioning, all participant data were shuffled together and one-twelfth of the data were randomly selected for the test set, with the remaining data selected for the training set.

- For proper dataset partitioning, one participant was selected for the test set, and the remaining 11 participants were selected for the training set.

Then, PCA was applied separately to the improper and proper datasets in order to reduce the dimensionality of the data to its first two principal components,with the amount of variance explained by the first two components being 0.72 for improper and 0.73 for proper. PCA dimensionality reduction is applied to both datasets so that the training distribution is more likely to cover the entire support of the test distribution [78]. Figure 12 depicts the graphs for improper and proper dataset portioning: red dots representing training data observations, and blue dots representing test data observations. Note that in Figure 12 improper (top), the test distribution is more uniformly spread throughout the training distribution, as all 12 participants are included in the test distribution, while in Figure 12 proper (bottom), the test distribution is more clustered due to the entire test distribution belonging to a single participant.

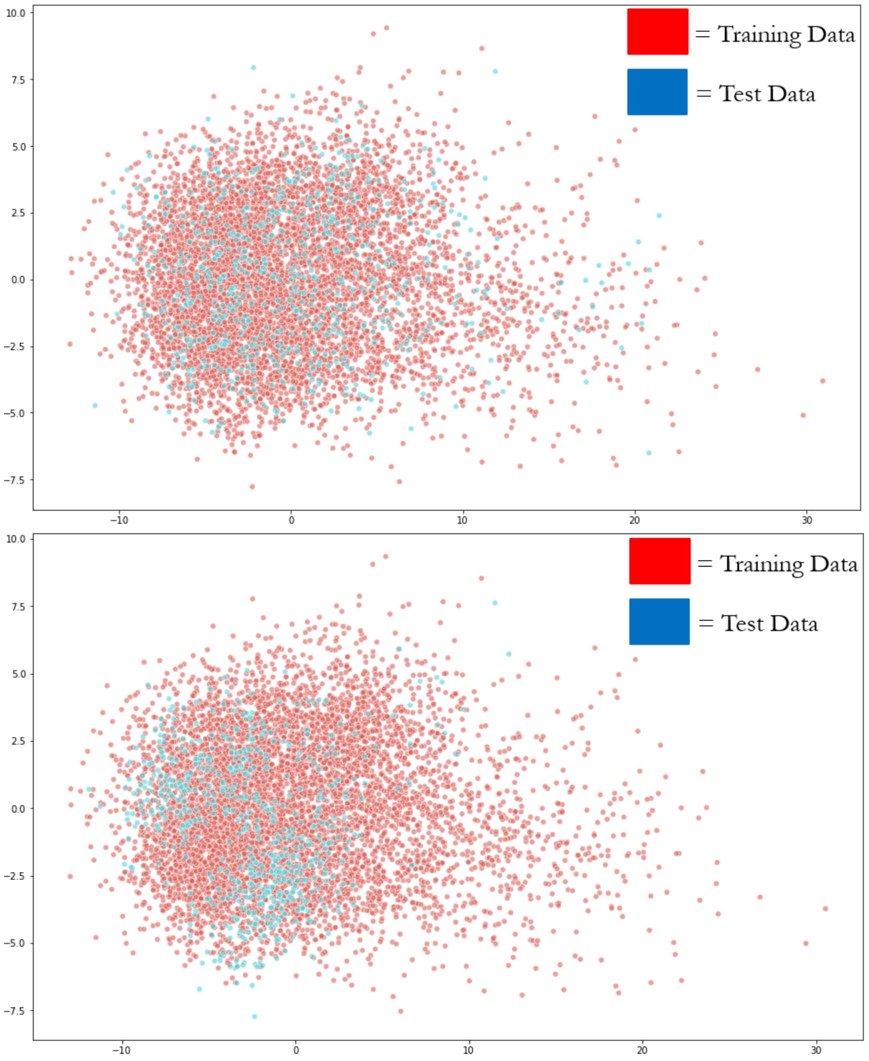


Figure 12. PCA projection of the first two principal components for Improper (top) and Proper (bottom) methods of dataset partitioning for spectral features of the Driver Fatigue dataset [79]. Red dots represent training data observations, and blue dots represent test data observations. Note that in the improper (top) that the test distribution is more uniformly spread throughout the training distribution, as all 12 participants are in the test distribution, while in the proper (bottom), the test distribution is more clustered due to the entire test distribution belonging to a single participant. These graphs are newly generated from the data obtained in the Driver Fatigue dataset [79]

Recall that the loss rescaling weight ratios represent a multiplier on the loss function in order to better estimate the expected real loss function from the loss estimate produced during evaluation of a model when there was a covariate difference between the test $p(x,y|\theta)$ and training sets $p(x,y|\lambda)$ used for machine learning. Ratios with values higher than 1 imply that there are more test data than training in this region; thus, the importance of the loss value in this region needs to be magnified; conversely, in regions with ratios smaller than 1, there is less test data than training data, meaning the loss values in this region are less important and their contribution to overall performance should be suppressed.

To calculate the loss rescaling weight ratio values $\frac{p(x|\theta)}{p(x|\lambda)}$ [34, 78] within these datasets, some method of density estimation of the marginal input distributions is required; for the purposes of visualization and discussion, we utilize two-dimensional histogram estimators generated across a 7 × 7 grid of bins for each dataset (# of bins = 49). To help visualize this, imagine a 7 × 7 grid placed over the observations in each graph of Figure 12, with the grid extending from the minimum values within the dataset, to the maximum values within the dataset, for both the X and Y axes. The number of training and testing observations within each histogram bin are calculated and normalized, providing our density estimation for the marginal input distributions, and subsequently the weight ratio values $\frac{p(x|\theta)}{p(x|\lambda)}$ for each bin. To better display the magnitude of difference in these weight ratio values, we display them in log scale, with a small value ($\varepsilon = 1.0 \times 10^{-5}$) added to the ratio values to avoid undefined values of log(0). This results in the log-transformed heat maps seen in Figure 13, with the top-left being the log-transformed weight ratio values for the proper dataset ($\log(proper + \varepsilon)$), top-right being the log-transformed weight ratio values for the improper dataset ($\log(improper + \varepsilon)$), and bottom being the difference between the log-transformed weight ratio values for proper minus improper ($\log(proper + \varepsilon) - \log(improper + \varepsilon)$).

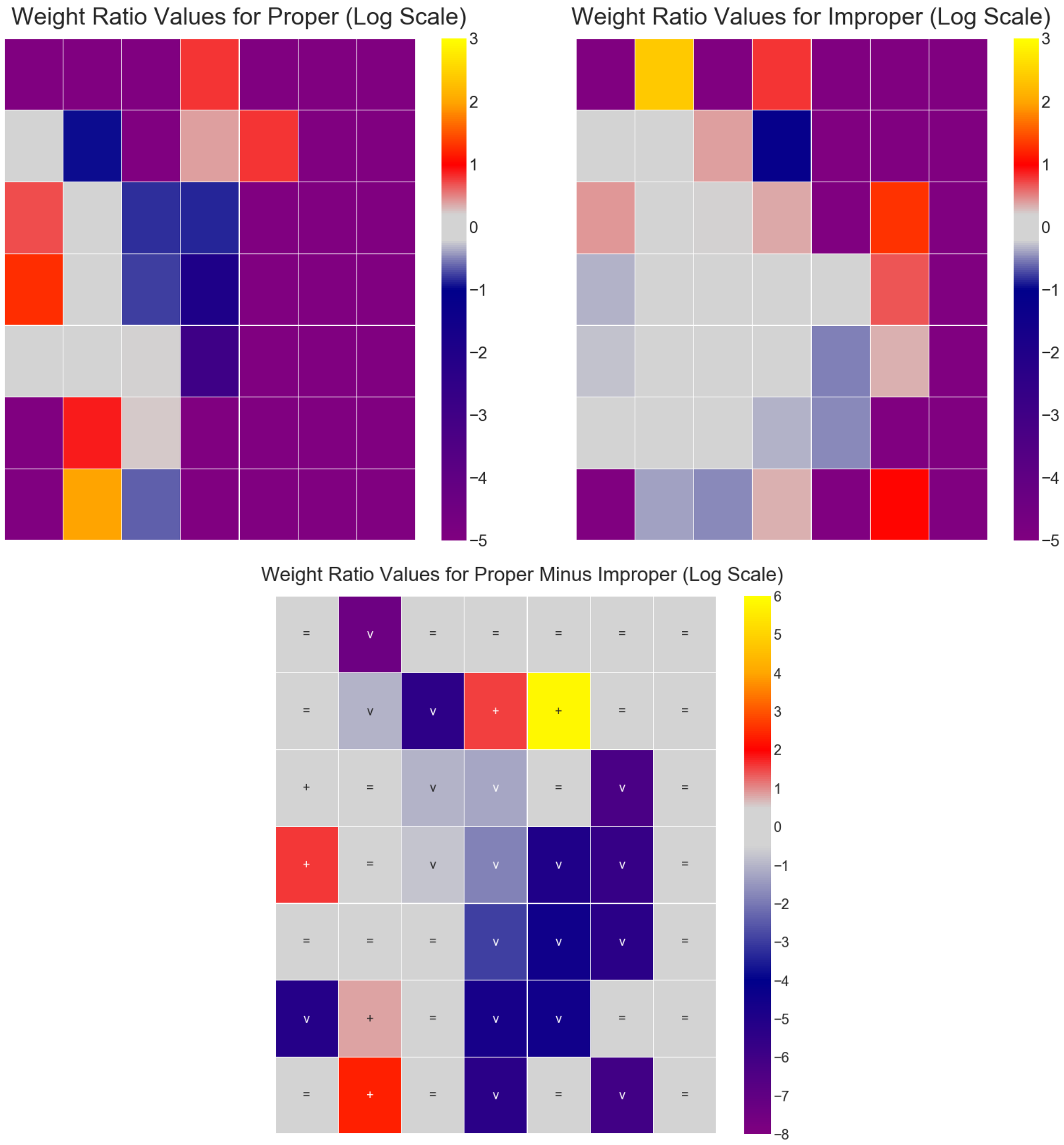


Figure 13. Heat maps for the log-transformed weight ratio values generated using two-dimensional histograms for Proper (top-left) (log($proper + \varepsilon$)) and Improper (top-right) (log($improper + \varepsilon$)) ($\varepsilon = 1.0 \times 10^{-5}$) methods of dataset partitioning for spectral features of the Driver Fatigue dataset [79]. The bottom graph depicts the difference in log-transformed weight ratio values between the proper and improper methods (log($proper + \varepsilon$) $-$ log($improper + \varepsilon$)), with labels for each bin indicating approximately equal weights (=), a significant negative delta (v), or a significant positive delta (+). These heat maps are newly generated from the data obtained in the Driver Fatigue dataset [79]

Figure 13 (top-left) depicts that for the proper dataset partition, there are few bins ($\approx$ 7) with a weight ratio close to 0, and many bins that are less than 0 (with many equal to -5, i.e., $\log(\varepsilon)$) or greater than 0. In contrast, Figure 13 (top-right) depicts that for the improper dataset partition, there are more bins ($\approx$14) with a weight ratio close to 0, and fewer bins that are less than 0 or greater than 0. Bins that are less than 0 for proper are also darker blue than bins that are less than 0 for improper, indicating the training data have a more similar distribution to the test for improper vs. proper. In Figure 13 (bottom), the difference of the log-transformed weight ratio values between the two heat maps (proper minus improper) indicates that approximately half of the bins have a delta of 0, and the other half of the bins have a delta that is significantly less than 1.0 or significantly greater than 1.0. This signifies that there can be significant differences in the weights required to rescale the loss depending on how the data are partitioned, with significantly more loss rescaling being required for the proper method of dataset partitioning vs. the improper method. This significant difference in loss rescaling between the two methods is indicative of proper dataset partitioning resulting in a covariate shift, and because the only difference in partitioning between the two methods is how participants are distributed, it is also indicative of an unseen participant resulting in covariate shift.

3.5.2 Covariate Shift in EEG.

In Section 3.4.1, we discussed how t-SNE can be utilized in order to detect covariate shift in data, and in Section 3.4.2, we discussed how covariate shift is inherent in EEG models due to the nature of EEG's non-stationarity and the individual differences that result in inter-participant variability. Here, we utilize t-SNE to visually showcase why this inter-participant variability leads to the effect of covariate shift in EEG cross-participant models. As mentioned previously, t-SNE allows one to inspect for covariate shift in the data by first applying the unsupervised technique and then visually exploring the data in 2D space, examining it to see if the clusters of training data and testing data are isolated from one another through visual inspection.

We perform t-SNE on spectral features of the PTSD [80], Schizophrenia [81], and Driver Fatigue datasets [79], as well as entropy features for the Driver Fatigue data [79], with results shown in Figure 14. This is done to showcase that inter-participant variability is present across many tasks and participant populations and demonstrates it visually to complement the quantitative empirical results within Section 3.6. For each of the graphs in Figure 14, we see that the majority of the data are clustered by participant, meaning that most of the participant data belong to its own unique input distribution, with some overlap and similarity between participants. However, there are some limitations of t-SNE that are worth noting and that are not obvious, and without their understanding, they can lead to incorrect assumptions about the underlying structure of the data. One limitation is that the cluster sizes in a t-SNE plot do not relate to distance between points of the cluster, as the algorithm adapts "distance" to each of the local clusters in the dataset, meaning dense clusters are expanded and sparse clusters are contracted [75]. This means that the sparsity of the cluster cannot be implied to have meaning. Another limitation is that the global geometry of the plot is not reliable as a source of information, meaning that the distances between clusters may or may not be accurate methods of interpreting the high-dimensional data in 2D space. While it is possible to dial in the hyperparameters to the correct values so that the 2D space does accurately represent the global geometry of the data in high-dimensional space, this requires a priori knowledge of the underlying structure of the high-dimensional data, which is unavailable. The implication of these limitations is that when interpreting t-SNE plots, the focus should be on simply the number of clusters present in the data and how they relate to the training dataset and the testing dataset. Any other information within the plot should not be taken as evidence of the underlying structure of the data in high-dimensional space. These limitations are important in understanding the data presented in the next section.

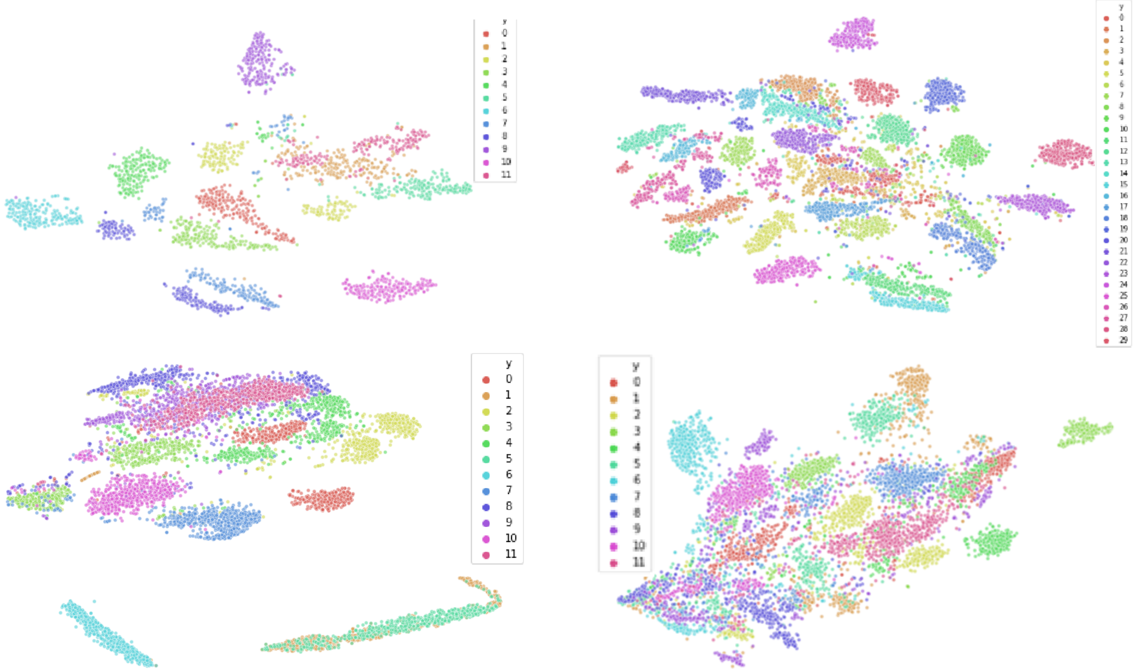


Figure 14. Example of using t-SNE for 2D dimensionality reduction and visualization of datasets utilized within this research, with colors corresponding to participants within the datasets, showcasing that inter-participant variability is present across different tasks and participant populations. Datasets depicted here are spectral features of the (a) PTSD, (b) Schizophrenia, (c) and Driver Fatigue datasets; and (d) Entropy features of the Driver Fatigue dataset. The dimensions of t-SNE are arbitrary distances that represent that closer neighboring points in low-dimensional space are likely to be neighbors in high-dimensional space.

3.5.3 Reducing Inter-Participant Variability.

As mentioned in Section 3.4.2, current approaches to EEG modeling are classified as either within-participant or cross-participant. Due to inter-participant variability, cross-participant models tend to always have lower classification accuracies than within-participant models, despite the fact that more participants typically also result in a larger training dataset for the model.

In order to demonstrate these effects of inter-participant variability within cross-participant models, we study the phenomenon with synthetically altered data through transformation. To generate the data, we utilize two mutually exclusive, independent applied data transformations named *shifted Heaviside* (our own naming for the transformation for the purpose of discussion) and *shift to median*. The goal of these transformations is to reduce the inter-participant variability of the data while still preserving the local structure of each participant's EEG data. In this manner, it can then be seen that as inter-participant variability is reduced and participants become more similar and no longer have different input distributions, classification performance improves because the effect of the covariate shift has been reduced. The purpose of this exploration is to demonstrate this performance-affecting relationship of inter-participant variability and covariate shift; we do not recommend utilizing these transformations in practice for the purpose of improving model performance.

The apparent performance improvement that occurs when data are transformed to reduce inter-participant variability implies that there will likely be overestimated classification performance in cross-participant models that are improperly validated and tested. When a model uses the same participants for both training and validation or testing, the higher measured performance is due to the reduced inter-participant variability between the training dataset and the validation or testing dataset—essentially masking the true differences that would exist between the people the model was trained on and the people the model was intended to be used on in the future. Similarly, when we apply transformations to reduce inter-participant variability, the goal is to transform the data in a manner such that multiple participants appear as if they belong to a single participant, and we can induce the effect of masking the true differences.

The transformation shifted Heaviside is both participant-based and feature-based. As mentioned at the beginning of this section, the name shifted Heaviside is the name we use in this paper to refer to this transformation proposed by Arevalillo-Herraez et al. in [82], based on the Heaviside function, as this transformation was not named by its originators. It was proposed by Arevalillo-Herraez et al. specifically for the use of reducing inter-participant variability in EEG data, and it does so by using the median value for each feature of each participant in order to map the original feature vector into a binary feature vector of the same size [82]. The effect of this transformation can be thought of as having the effect of shifting the data to the different corners of a hypercube. To create the mapping, first, the median value of each feature of each participant is calculated. Then, the original feature vector data are converted to a binary encoded vector where each feature value is transformed to a 1 if the value is greater than the median of the feature vector, or a 0 if less than or equal to the median (akin to a shifted Heaviside function). Specifically, they formulate their algorithm as follows: for pth participant, for all feature vectors $x_{p,j}, j = 1, 2, ..., n_p$ in the set of training samples X_p, compute the median vector x'_p. Then, transform all feature vectors u for the same participant p according to Equation 3.6, where $[k]$ denotes the kth element (feature) of the corresponding vector.

$$\bar{u}[k] = \begin{cases} 1 & u[k] > x'_p[k], \\ 0 & u[k] \leq x'_p[k], \end{cases} \tag{3.6}$$

The *shift to median* transformation involves calculating a center point for each output class y across all participants in feature space and then shifting by class y each participant's data closer to those class center points so that each participant's data distribution moves closer together (toward the calculated class centers), while still preserving differences within each participant's individual data observations. The goal is to reduce inter-participant variability by shifting all participants to a similar range in feature space, while still preserving local structure within each participant, including class effect. The effect of this transformation can be thought of as shifting each participant's entire cluster of data by a certain amount so that it is re-centered on a new point (performed by class y). Using the same symbols in the previous paragraph, we have the following algorithm.

Shift to Median—Variables are defined as follows: y represents class, j represents the observation, p represents the participant, and N represents the total number of training samples.

1. $\forall y$ Calculate median vector $\tilde{C}_y$ across all feature vectors $x_{p,y,j}$ of all participants $p = 1,...,P$

   - $\tilde{C}_y = \begin{cases} x_{y,\frac{N+1}{2}} & \text{N odd} \\ \frac{1}{2}(x_{y,\frac{N}{2}} + x_{y,\frac{N+1}{2}}) & \text{N even} \end{cases}$

2. $\forall p$ $\forall y$ $\forall x_{p,j}$ Calculate median centroid $\tilde{c}_{p,y}$ of p

   - $\tilde{c}_{p,y} = \begin{cases} x_{p,y,\frac{N+1}{2}} & \text{N odd} \\ \frac{1}{2}(x_{p,y,\frac{N}{2}} + x_{p,y,\frac{N+1}{2}}) & \text{N even} \end{cases}$

3. $\forall y$ $\forall x_{p,j}$ Compute shifted vector $x'_{p,y,j} = x_{p,y,j}$ + $(\tilde{C}_y - \tilde{c}_{p,y})$

This results in three different datasets: original dataset, *shifted Heaviside* transformation, and *shift to median* transformation. Employing t-SNE on the datasets allows us to view the local clusters within the data. For EEG specifically, this typically allows us to identify clustering by participant, showcasing the inter-participant variability inherent across participants. To demonstrate this clustering as well as the EEG data transformations described above, we utilize the Driver Fatigue dataset [79] described in Section 3.6.1.

This dataset contains both entropy and spectral features. In information theory, the entropy of a time series quantifies its regularity and predictability over time [83], and the entropy features extracted for use include approximate entropy (AE), sample entropy (SE), and fuzzy entropy (FE) features [84]. The spectral features were extracted using Morlet wavelet transforms in MATLAB to determine the frequency-domain mean power of two of the five clinical frequency EEG bands: alpha (12–15 Hz) and beta (16–22 Hz) ([18]). Two frequency-spectral-power features extracted from EEG were computed for each of the 30 channels. This results in 60 features for the spectral feature space and 90 features for the entropy feature space (three entropy measures across all 30 channels).

Figures 15 (top-left) and 16 (top) illustrate the results of applying t-SNE to the untransformed Drive Fatigue datasets for the entropy and spectral feature spaces, respectively. It can be seen that in these high-dimensional data spaces of 90 and 60 features each that there is significant clustering by participant, with coloring corresponding to a participant's data. Note that this coloring has no effect on the t-SNE algorithm itself and is applied afterwards for visualization. As mentioned earlier in Section 3.5.2, due to the limitations of t-SNE, we cannot reliably interpret any information from the 2D plot outside of the number of clusters. Clusters found within t-SNE should only be treated as such: that they are localized clusters that exist within the high-dimensional data. After a data transformation, if t-SNE is unable to find local clustering despite hyperparameter tuning, then local clustering does not exist [75]. For these datasets, a lack of local clustering means that the inter-participant variability has been reduced to the point that t-SNE can no longer distinguish between participants in the feature space.

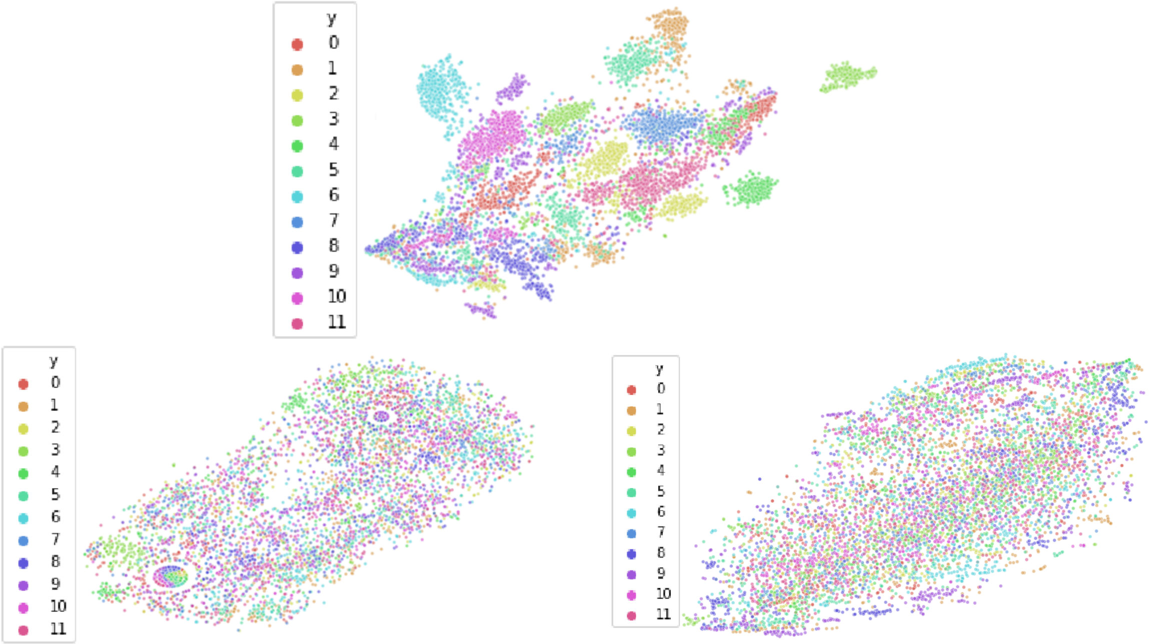


Figure 15. Results of visualizing the data using t-SNE for the entropy feature space before and after various data transformations: (top) Before any transformations; (bottom-left) After applying *shifted Heaviside* transformation; (bottom-right) After applying *shift to median* transformation. Colors correspond to different participants, with the same color applied to the same participant in each figure. Note in in the bottom graphs that there is a lack of local clustering, implying that inter-participant variability has been reduced due to the transformations. The dimensions of t-SNE are arbitrary distances which represent that closer neighboring points in low-dimensional space are likely to be neighbors in high-dimensional space.

Figure 15 (bottom-left and bottom-right) reveal the different data transformation's effects on local clustering within the entropy feature spaces and Figure 16 (bottom-left and bottom-right) show the transformation's effects on the spectral feature spaces. For the entropy feature space, we see that each transformation has reduced the inter-participant variability to the point where t-SNE no longer finds local clustering within the data. Similarly, for the spectral feature space, we see that the *shifted Heaviside* transformation has the same result, while the *shift to median* transformation largely reduces local clustering within t-SNE, but not to the same effect as the *shifted Heaviside* transformation.

To demonstrate the effects of reducing inter-participant variability on classification accuracy in cross-participant models, cross-participant models were also built using each of these three datasets of data within both of the feature spaces (entropy and spectral). As this is the Driver Fatigue dataset, models were trained according to the methodology specified in Section 3.6.1. For each of the three subsets of data within both of the feature spaces of entropy and spectral features, separate models were trained and tested according to both the improper and proper methods of cross-participant model generation. For proper model generation, we follow the guidelines specified in Section 3.4.3, resulting in 12-fold LOPO CV. As mentioned in Section 3.6.1, for improper model generation, in order to match the number of folds (and data per fold) in LOPO CV, 12-fold CV was used with all participant data shuffled together and split across 12-folds. Together, this results in 12 models generated for each method.

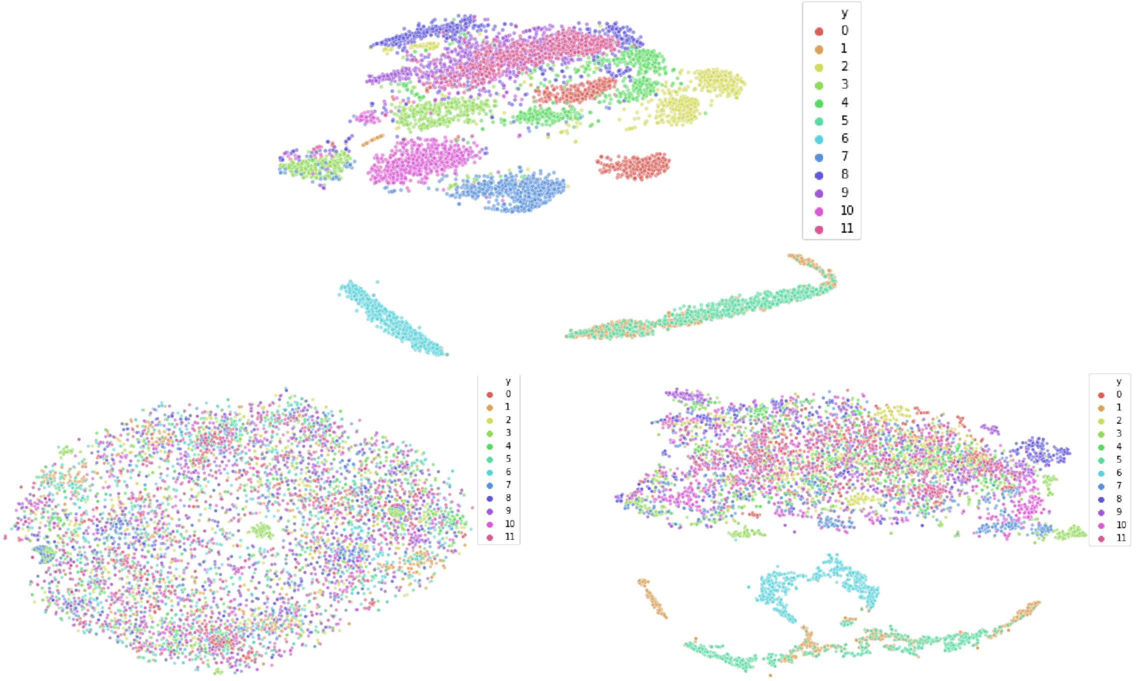


Figure 16. Results of visualizing the data using t-SNE for the spectral feature space before and after various data transformations: (a) Before any transformations; (b) After applying shifted *Heaviside transformation*; (c) After applying *shift to median* transformation. Colors correspond to different participants, with the same color applied to the same participant in each figure. The dimensions of t-SNE are arbitrary distances that represent that closer neighboring points in low-dimensional space are likely to be neighbors in high-dimensional space.

Table 4 contains the classification accuracy results for each of the 12 models. It can be seen that for both the entropy and spectral feature spaces that improper model testing did not benefit from the data transformations. Intuitively, this makes sense, as these models are tested improperly; thus, the model has seen each participant's input distribution, and therefore, a reduction of inter-participant variability is not impactful to the model. However, for proper model testing, we see that for both the entropy and spectral feature spaces that the *shift to median* transformation results in a dominance in accuracy of the 95% confidence interval (CI) from the transformation in comparison of the 95% CI's. While the *shifted Heaviside* transformation did result in a reduction of inter-participant variability for both feature spaces as shown in Figures 15 (bottom-left) and Figure 16 (bottom-left), this reduced inter-participant variability did not result in any significant effects on cross-participant model performance, suggesting that this transformation may be best suited for only certain datasets which its developers Arevalillo-Herraez et al. work with.

Table 4. Classification accuracies for the 12 models generated from transformed and non-transformed driver fatigue data. Improper models were generated with the improper method of cross-participant model generation utilizing 12-fold CV with all participant data shuffled together and split across 12 folds, and proper model generation utilized 12-fold LOPO CV. The purpose of this table is two-fold. One is to depict that improper model generation typically results in overestimated model accuracy as can be seen with increased accuracies for improper vs. proper. The other is to depict the results of the proper method on untransformed data versus the proper method on the two transformed datasets. Bold signifies dominance in accuracy of the 95% confidence interval from the transformation in comparison of the 95% confidence intervals.

	Entropy	**Spectral**
Improper		
Untransformed	0.91 (0.89, 0.93)	0.82 (0.79, 0.85)
Shifted Heaviside	0.72 (0.68, 0.76)	0.66 (0.62, 0.70)
Shift to Median	0.91 (0.89, 0.93)	0.82 (0.79, 0.85)
Proper		
Untransformed	0.50 (0.46, 0.54)	0.50 (0.46, 0.54)
Shifted Heaviside	0.50 (0.46, 0.54)	0.47 (0.43, 0.51)
Shift to Median	**0.80 (0.77, 0.83)**	**0.72 (0.68, 0.76)**

3.6 Empirical Demonstrations in Diverse EEG Case Studies

In this section, we utilize five publicly available datasets to empirically demonstrate the difference in machine learning performance results of using proper versus improper methods of dataset partitioning during training, validation and testing. These five datasets were selected to encompass diversity across the research activities using machine learning and EEG, to demonstrate the importance of following the proper methodology in many situations. The domains of the five datasets differ substantially in both tasks performed during data collection and subsequent classification using EEG, including both classification of different mental states within an individual: mental fatigue (Driver Fatigue), emotions (Confused Students), as well as determining of the existence of longer-term chronic conditions in individuals: mental disease (Alcoholism), psychological conditions (PTSD), and mental disorders (Schizophrenia). In the chronic condition datasets, each participant (and all of the observations corresponding to that participant) are either in the chronic condition class or the class representing normal. Summary details of these datasets can be seen in Table 2.

Table 5. Details for the publicly available datasets. All datasets are binary classification tasks, and all datasets are balanced except for the Alcoholism dataset. This gives chance accuracy for Alcoholism defined as 0.64 and 0.50 for all other datasets.

Dataset	Year Collected	Binary Classification Task	# of Participants
Driver Fatigue[79]	2017	Normal vs Fatigue	12
Confused Students[72]	2013	Confused vs Not Confused	10
Alcoholism[73]	1999	Alcoholic vs Non-Alcoholic	122
PTSD[80]	2018	Pre-Treatment vs Post-Treament	12
Schizophrenia[81]	2014	Schizophrenia vs Healthy Control	30

Model architectures used are selected based on research papers with top performance in their respective dataset and/or domain, with replication performed as closely as possible. In some cases, research papers were missing details about hyperparameters and other model details, and these details had to be selected using best practices of machine learning.

With architecture and hyperparameters selected, two models are then created and evaluated separately using the same architecture and hyperparameter sweep (grid search utilized):

- Improper: trained, tuned, and evaluated during tests using all participant data.
- Proper: trained and tuned using data from a subset of the participants, then, during the test, evaluated using only data from participant(s) that were not used to train or tune the model.

Then, results of the two methods are contrasted and compared, with error rates displayed in a summary table in Section 3.7. It is also worth noting that the amount of data used for training and validation/testing is kept consistent across both the proper and improper methods, meaning that both models have the same quantity of observations to train upon, and additionally, both models are validated and tested with the same number of observations. This ensures that there is minimal difference between the two models in terms of architecture, hyperparameter sweeps, or the amount of data used for training, validation, or testing, and that the only difference between the models is the restriction surrounding which participants are used for training, validation, and testing for the proper method vs. the improper method.

The next five subsections are structured as case studies for each of the five datasets, and they are in the following order: *Driver Fatigue*, *Confused Students*, *Alcoholism*, *PTSD*, and *Schizophrenia*. Each case study first discusses the purpose of the experiment, how it was conducted, and what EEG data were collected (pre-processing details are provided in the Appendix in Section 3.8). Then, information on the model architecture and its methodology are provided, as well as the results previous researchers had achieved using that methodology. Then, we detail our own methodology to include having to fill any gaps missing from their architecture or hyperparameter selection, as well as how we perform both improper and proper training, validation, and testing for the two different models. Finally, we state results achieved with both models and compare them.

3.6.1 Driver Fatigue.

This dataset is available on Figshare [79] through a link provided in Min et al.'s paper, which details both the experiment and the subsequent deep learning performed [84]. Their experiment consisted of collecting EEG recordings during a driving simulator for the purpose of using these signals to develop a model that could detect driver fatigue using EEG signals. Twelve participants used the driving simulator for 1–2 h in a highway setting with low traffic density, with EEG recorded in two phases during the session. The first phase consisted of 20 min of continuous driving, with the last 5 min of this 20-min segment recorded and labeled as the *normal* state. The second phase consisted of driving that lasted for 40–100 min until the participant's self-reported questionnaire indicated that they were fatigued (surveys used were Lee's Subjective Fatigue Scale [85] and the Chalder Fatigue Scale [86]), in which the last 5 min of driving were recorded in the EEG and labeled as the *fatigue* state. EEG was recorded using a 32-channel electrode cap, with two of the channels being reference channels linked to mastoid electrodes. The 5 min of EEG from each phase were epoched into 1 s segments for 300 epochs per phase per participant, resulting in a total of 3600 trials for the normal state and 3600 trials for the fatigue state. Then, the data were randomly split into training and testing datasets at a 50/50 ratio, without participants taken into account, thus resulting in improperly created datasets for cross-participant models. Feature extraction included several entropy measures, which were extracted for each trial and then normalized. In information theory, the entropy of a time series quantifies its regularity and predictability over time [83], with the measures extracted including approximate entropy, sample entropy, and fuzzy entropy [84].

In Min et al.'s work, these entropy features were then utilized for multiple classifiers, with the classifier that achieved the highest accuracy being an artificial neural network (ANN) [84]. The ANN had three layers, each with 20 hidden units and sigmoid activation functions. Gradient descent was used with mean squared error (MSE) for the loss the function, and the Levenberg–Marquardt function was used as the optimization function [87]. Leave-One-Out Cross-Validation (LOOCV) was utilized to report test classification accuracy, with their reported test accuracy being 0.968 or an error rate of 0.032.

The architecture above was followed for training both of our models; however, 12-fold CV was utilized, as there are 12 participants and Leave-One-Participant-Out (LOPO) CV results in 12-fold CV. Thus, for improper training and validation, 12-fold CV was used with all participant data shuffled together and split across 12-folds, and for proper training and validation, LOPO CV was used. Using this configuration, for improper testing of the cross-participant model, the best accuracy we obtained was 0.83, which was much lower than Min et al.'s reported test accuracy of 0.968 with their 50/50 training/testing split. In an effort to improve upon this, a hyperparameter sweep was conducted across hidden units (20, 30, 40, and 50), dropout rate (0.0, 0.1, 0.2), different learning rates (0.01, 0.001, 0.0001), and the *reduce_lr* callback of reducing the learning rate based on the number of epochs trained. The configuration with the highest classification accuracy for the improper method was one of 50 hidden units, 0.2 dropout rate, 0.001 learning rate, and *reduce_lr* callback was utilized. This hyperparameter sweep was also conducted for the proper method, with the configuration with the highest classification accuracy for the proper method being 40 hidden units, 0.2 dropout rate, 0.001 learning rate, and *reduce_lr* callback being utilized. Then, these configurations were used for improper and proper training and validation of the cross-participant models, respectively.

For improper training and validation of the cross-participant model using our configuration above, the reported classification accuracy using 12-fold CV was 0.91 (95% CI: 0.903, 0.917) or an error rate of 0.09 (95% CI: 0.083, 0.097). While this result is significantly lower than Min et al.'s error rate (0.09 vs. 0.032 [84]), our accuracy is still similar enough in magnitude for our goal of contrasting proper and improper methods of model evaluation. As such, when we built the model properly and trained and validated it using LOPOCV, the resulting accuracy was 0.540 (95% CI: 0.528, 0.552) or an error rate of 0.46 (95% CI: 0.448, 0.472). This error rate is over five times as that of the error rate of the improper method, illustrating how difficult classification of unseen participants is, and how significantly overestimated test accuracies can become by following an improper methodology, which does not account for the significance of inter-participant variability.

3.6.2 Confused Students.

Participant data for this dataset are available on Kaggle [72] and come from an experiment involving college students. The purpose of the experiment was to collect EEG from college students while they were in a confused state and a not confused state and then build a model that could determine if the student was *confused* or *not confused* using the EEG signals. Researchers collected EEG while the students watched online education videos in a *confused* state and a *not confused* state [88]. Ten young adult college students watched two-minute online education videos (lectures) on various topics, which were assumed to not confuse an average college student, such as basic algebra and geometry, as well as topics that would be confusing, such as quantum mechanics and stem cell research. Each student watched five randomly selected videos from each category, and after each video, students self-rated their confusion on a scale of 1 (least confused) to 7 (most confused). EEG was recorded at a sampling rate of 512 Hz using a single-channel NeuroSky MindSet device, which has a single electrode that rests over the middle of the forehead, and two electrodes for ground and reference, each in contact with an ear. The first 30 s and last 30 s of each session's EEG recording were removed in case the student was not ready; the middle 60 s was available for analysis. Then, NeuroSky software was used to extract features from the signal at 2 Hz to include the mean of the raw signal, mean power for the five traditional frequency bands (to include alpha low/high, beta low/high, and gamma low/high), and MindSet's proprietary "attention" and "meditation" signals.

In addition to the experiment data, Kaggle also lists references with some of the latest classification results to use these data for the purpose of binary classification of whether a student is *confused* or *not confused*. The two references with the greatest classification accuracies both use bidirectional Long Short-Term Memory (LSTM) models as their neural network architecture [89, 90], with the one we selected for replication being work from Ni et al., as their work provided the most detail for replication [90]. For Ni et al.'s work, each session consisted of a single trial as to provide sequence data for the recurrent neural network (RNN). Sessions from all nine participants were merged together for a cross-participant model, and 5-fold cross-validation was used across all participants (improper method). EEG features used consisted of proprietary measures from the MindSet EEG device labeled *Attention* (measure of mental focus) and *Meditation* (measure of calmness), the raw EEG signal values, and mean values of eight different frequency regions in the power spectrum. In addition to EEG signals, Ni et al. also opted to use the "Predefined Label" of whether a session was confusing or not as a feature. The bidirectional LSTM had 50 hidden units and used a tanh activation function, and it was followed by a fully connected layer with a sigmoid activation function. Before the bidirectional LSTM, batch normalization was used. No other architecture or hyperparameter methodology was provided. The CV test accuracy varied between 0.71 and 0.74 for their work, with an average 5-fold CV accuracy of 0.733.

To reproduce Ni et al.'s results for the improper model, the architecture above was followed along with the hyperparameters provided, and all of the EEG features were utilized, resulting in 11 total features used for training (in our replication of the research, the non-EEG "Predefined Label" feature was omitted; we do not recommend including a class label as a feature per standard practices of machine learning). In an effort to replicate their methodology of hyperparameter selection for the proper model, a hyperparameter grid search was performed across hidden units (40, 50, 60), dropout rate (0.0, 0.1, 0.2), and learning rate (0.001, 0.0001), with the highest performing proper model having hyperparameters of 50 hidden units, 0.0 dropout rate, and 0.0001 learning rate; the same hyperparameters Ni et al. and our improper model design used. This hyperparameter sweep ensures both the improper and proper models have selected their best hyperparameters for their input data. In an effort to increase the amount of training samples for the models, the EEG data were also segmented using a sliding sequence window of 15 samples in length and slides by 12 samples. Then, we built two separate cross-participant models using improper training and testing for one and proper training and testing for the other. The improper model utilized 5-fold CV for training and validation with all participant data shuffled together, resulting in every fold including some data from every participant (exact same method used by Ni et al.). The proper model also utilized 5-fold CV; however, it was Leave-Two-Participants-Out CV. Outside of this change in how the folds were formed for features used, hyperparameters, and the number of observations used for both training and validation.

For improper training and validation of the cross-participant model using 5-fold CV, our replication of Ni et al.'s configuration [90] resulted in a test accuracy of 0.69 (95% CI: 0.654, 0.726), which was close to their reported test accuracy of 0.733, with a difference in error rates of 0.31 (95% CI: 0.274, 0.346) vs. 0.267. However, our proper training and validation of the proper cross-participant model using Leave-Two-Participants-Out CV resulted in an accuracy of 0.584 (95% CI: 0.552, 0.628) or an error rate of 0.416 (95% CI: 0.372, 0.448). The error rate of the proper method is over 33% greater than the error rate of the improper method, suggesting that improper training and testing of EEG data can lead to overestimation of model performance on unseen participants and thus the human population in general.

3.6.3 Alcoholism.

Participant data for this dataset are available from both Kaggle [72] and the University of California, Irvine (UCI) machine learning data repository [73], with this research utilizing the Full Dataset from the UCI repository. The source of the data comes from one of a number of experiments sponsored by the National Institute on Alcohol Abuse and Alcoholism (NIAAA) in the early 1990s, which were conducted with the purpose of recording brain activity during a task that was expected to elicit differences in the neural activity of healthy participants and alcoholic participants [91, 92]. In the control group, there were 45 male participants, and in the experimental group, there were 77 alcoholic male participants. The task used was a visual object recognition task: the participant was presented with a sequence of two images and had to determine whether the second image was the same as the first. Signals were recorded from 64 scalp EEG electrodes and 2 electrooculography (EOG) electrodes, at a sampling rate of 256 Hz, and were referenced to node site Cz during EEG measurement. This resulted in a sequence dataset with 64 features $\times$ 256 μV values for each of the (approximately) 100 observations per participant.

Recently, the *Full Dataset* from the UCI repository was utilized by Farsi et al. to train both ANN and LSTM classifiers, with their LSTM architecture having the best performance with a reported test accuracy of 0.93 [93]. They used improper dataset partitioning, mixing the participants data and selecting 80% of the data for training and 20% for testing. For improper training and validation of the cross-participant model, we used 5-fold CV to align with Farsi et al.'s 80% training 20% testing dataset preparation. For proper cross-participant model evaluation, we utilized 5-fold Leave-N-Participants-Out CV, with N equal to 24 or 25 depending on the fold. Although the paper provided an architecture, it did not explicitly identify their choice of best hyperparameters that were selected for their final LSTM model—they only provided a list of what hyperparameters were explored. Therefore, in an effort to recreate their work, we utilized the architecture they specified and performed a hyperparameter sweep across all of the hyperparameters that were explored by the authors. This resulted in a 3-layer LSTM with layers and hidden units as follows (100-(Dropout Layer)-32-1), and a hyperparameter sweep performed for activation function (Relu, tanh, Sigmoid), dropout rate (0.2, 0.4), optimizer (Adam, SGD), batch size (50, 150), learning rate (0.1, 0.0001), epochs (50, 100), and loss function (MSE, Binary Cross Entropy). The resulting models from these hyperparameter sweeps performed poorly for both improper and proper models, so we instead used a 3-layer LSTM architecture with descending hidden units (H) across the three layers (H, H-50, H-100), dropout and recurrent dropout activated for all three layers, with activation function tanh, recurrent activation function sigmoid, batch size 256, optimizer Adam, learning rate 0.0001, and loss function Binary Cross Entropy. Then, we performed a hyperparameter sweep for this architecture across hidden units (200, 250, 300, 350), dropout rate (0.2, 0.3, 0.4), and epochs (200, 300, 400, 500). This architecture and its hyperparameter sweep had better performance, so we opted to use it as our final architecture for both the improper and proper methods of model creation. The best configuration for the improper model had hyperparameters of hidden units 350, dropout rate 0.4, and epochs 500. The best configuration for the proper model had hyperparameters hidden units 300, dropout rate 0.4, and epochs 400.

The resulting improper model had a test accuracy of 0.84 (95% CI: 0.82, 0.86) or an error rate of 0.16 (95% CI: 0.12, 0.18). While this result is significantly lower than Farsi et al.'s error rate (0.16 vs. 0.07 [93]), our accuracy is still similar enough in magnitude for our goal of contrasting proper and improper methods of model evaluation. The resulting proper model had a test accuracy of 0.69 (95% CI: 0.67, 0.71) or an error rate of 0.31 (95% CI: 0.29, 0.33), which is close to chance accuracy of 0.64 or a chance error rate of 0.36, as this dataset was imbalanced with a majority class of alcoholics. The error rate of the properly data-partitioned model is almost twice as large as the error rate of the improper model, again suggesting that if the goal is to build a model that can be used to make accurate estimates on unseen individuals, then the EEG cross-participant model must be evaluated properly by evaluating it only using data from participants not used during training or validating the model.

3.6.4 Post-Traumatic Stress Disorder (PTSD).

This publicly available PTSD dataset can be found on Figshare [80] through an appendix and link provided in Rahmani et al.'s paper, which details the experiment used and their subsequent EEG analysis [80] (unrelated to machine learning). Researchers captured resting-state EEG from six healthy control (HC) participants and six combat-related PTSD participants, while they had an MRI taken, with the goal being to find differences between HCs and PTSD participants through analysis of the EEG. For this dataset, there were 33 channels of EEG recorded, with two of the 33 channels being used for ground and reference, and at a sampling rate of 5000 Hz. EEG preprocessing was performed in both the proprietary software BrainVision Analyzer2 and within EEGLAB. ICA was used to remove blink and saccade artifacts, and time periods containing motion artifacts from observed participant head motion were also removed. After artifact removal, the EEG was down-sampled to 250 Hz. Scans lasted 526 s, and the first 6 s were removed for steady-state signals, resulting in 520 s of raw voltage value data per participant. However, only the first continuous 50,000 data points without participant motion were used within Rahmani et al.'s analysis, and this was subsequently the case with the data uploaded and made available to the public, resulting in 200 s of raw EEG per participant being available for machine learning. Then, EEG signals were segmented into 1-s non-overlapping epochs, resulting in 200 observations per participant. For feature selection, spectral features were extracted for the 31 EEG channels using Morlet wavelet transforms in MATLAB to determine the frequency-domain mean power of the five traditional frequency bands: delta (2–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (15–30 Hz), and gamma (30–80 Hz) [18]. The mean power of these five bands for all 31 channels results in a total of 155 features (31 × 5) for each of the 260 observations for each of the 12 participants.

Since this dataset has not yet been used for published research in the area of machine learning, there is no machine learning workflow we are attempting to replicate; instead, we utilize a standard fully connected multi-layer perceptron neural network (MLPNN) for our architecture, which is a common and most fundamental ANN.

A hyperparameter sweep was performed to find a good model. The sweep was conducted across the following hyperparameters: hidden layers (1, 2), hidden units (20, 30, 40, 50), dropout rate (0.0, 0.1, 0.2), and learning rates (0.01, 0.001, 0.0001) for both the improper and proper methods of model evaluation, and the hyperparameter configuration that resulted in the highest validation accuracy was selected for each method. The architecture used ReLU activation functions for dense layers, a Sigmoid activation function for the output layer, and 'Adam' for the optimizer; training was conducted for 50 epochs. For training and validation of the improper model, 12-fold CV was used with all participant data shuffled together and split across the 12-folds, and for training and validation of the proper model, 12-fold LOPO CV was used.

The best configuration for the improper model consisted of 1 hidden layer, 50 hidden units, a learning rate of 0.001, and a dropout rate of 0.2. This configuration resulted in a 12-fold CV accuracy of 0.995 (95% CI: 0.9922, 0.9978) or an error rate of 0.005 (95% CI: 0.0022, 0.0078). The best configuration for the proper model was similar in that it consisted of the same parameters for everything except the hidden units being 40 instead of 50. This configuration resulted in a 12-fold LOPO CV of 0.803 (95% CI: 0.7871, 0.8189) or an error rate of 0.197 (95% CI: 0.1811, 0.2129). This results in an error rate that is over 39 times larger for the proper method versus the improper method of training and validation, which is the 2nd largest difference between proper and improper partitioning within these case studies. Relying on the overly optimistic, extremely low error rate measured in the performance of the model trained using the improper training method would falsely drive overconfidence in the model's performance in future use. Once again, the evidence suggests that if the intent is to estimate performance on new people, proper segregation of participants in the partitioning of the training, validation, and test datasets is paramount.

3.6.5 Schizophrenia.

This dataset is available on Kaggle [81] and was collected in an effort to study the difference in corollary discharge between participants with schizophrenia and those without schizophrenia (HCs) [94]. The participant's task was to either (1) press a button every 1–2 s to deliver an 80 dB tone, (2) passively listen to that same tone, or (3) press a button that did not produce a tone or any other effect other than the tactile response of depressing the button. Each event condition occurred a total of 100 times for each participant, resulting in 300 trials per participant. In total, in the dataset there were 32 HCs and 49 patients with schizophrenia; however, data from only 40 participants were available online (25 HCs and 15 diagnosed with schizophrenia).

Data were collected using a BioSemi ActiveTwo 64 + 2 electrode cap, with 64 scalp sites and 2 references electrodes placed over the mastoids [94]. Data were sampled at 1024 Hz and epoched at 3 s for each trial, with the start of each epoch being time-locked to 1.5 s before button press. The EEG data were uploaded to Kaggle [81] in two different formats, one in time-series as raw EEG voltage values, and the other with event-related potential (ERP) features. In order to generate richer features for machine learning, spectral features were extracted from the raw EEG voltage values for all 64 channels. This was done similarly as done in the PTSD dataset, using Morlet wavelet transforms in MATLAB to determine the frequency-domain mean power of the five traditional frequency bands: delta (2–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (15–30 Hz), and gamma (30–80 Hz) ([18]). This resulted in 320 features for each observation (64 channels × 5 frequency bands

= 320), with participants having between 280 and 290 observations each. Unfortunately, the dataset was heavily imbalanced, with 25 participants being HCs, and only 15 participants being diagnosed with schizophrenia. To alleviate this imbalance, only 15 of the 25 HCs were randomly selected to be used for machine learning.

For our architecture selection for this dataset, we use both a neural network, as well as a more traditional machine learning model—the random forest classifier. Buettner et al. achieved high levels of accuracy for EEG classification of HCs vs. participants with schizophrenia using an RFC [95] (albeit on a different EEG dataset), so they are a proven model type for this domain, with the neural network architecture implemented for additional investigation. The spectral features generated were utilized for both the MLPNN and the RFC, and both architectures followed both the proper and improper methods of model evaluation, resulting in four separate models generated. For improper training and validation, 30-fold CV was utilized with all participant data shuffled together and split across the 30-folds, and for proper training and validation, 30-fold LOPO CV was used. As with the PTSD dataset, we did not have a published neural network methodology to replicate for this dataset.

For the MLPNN architecture, a hyperparameter sweep was conducted across the following hyperparameters: hidden layers (1, 2), hidden units (20, 30, 40, 50), dropout rate (0.0, 0.1, 0.2), and learning rates (0.01, 0.001, 0.0001). This hyperparameter sweep was conducted for both the improper and proper methods of model evaluation, and the hyperparameter configuration that resulted in the highest validation accuracy was selected for each method. Other parameters of the architecture include using the ReLU activation function for dense layers, a Sigmoid activation function for the output layer, and ‘Adam’ for the optimizer; the number of training epochs set to 50.

RFC hyperparameters selected for hyperparameter tuning included the maximum depth of the trees and the number of features to consider. The number of estimators (trees) was determined by incrementally increasing the number of estimators by 5 from a low value of 50 until validation accuracy no longer improved. For this, maximum depth was set to its default sklearn value of 'None' so that there was no limit to depth, and the maximum features set to its typical recommended amount of $m = \sqrt{p}$ where p equals the 320 features, and thus, $m = \sqrt{320} = 18$ [96]. By incrementally increasing the number of estimators by 5 from 50 to 750 as described above, 110 was found to result in the best validation accuracy, and this amount was used for both proper and improper methods of model evaluation for the RFCs. From here, a hyperparameter sweep for the number of features and the maximum depth was conducted, utilizing values from 1 to 25 for each. These values were determined by going far above and below the typical recommended values for these parameters (e.g., the square root of features for the number of max_features m) [50]. This resulted in a hyperparameter sweep of $25^2 = 625$ models for both the improper and proper methods of model evaluation, resulting in 1250 models in total generated during hyperparameter search.

The best configuration for the MLPNN improper model consisted of 1 hidden layer, 50 hidden units, a learning rate of 0.001, and a dropout rate of 0.2. This configuration resulted in a 30-fold CV accuracy of 0.992 (95% CI: 0.990, 0.9939) or an error rate of 0.008 (95% CI: 0.0061, 0.01). For the proper MLPNN model, there was no significant difference between any of the configurations, and no model was able to perform better than random chance (50%), illustrating how severe the effect of covariate shift can be in EEG data, depending on the participants used. The best configuration for the RFC improper model was maximum features set to 15 and maximum depth set to 24, resulting in a 30-fold CV accuracy of 0.941 (95% CI: 0.936, 0.946) or an error rate of 0.059 (95% CI: 0.054, 0.064). For the proper RFC model, similar to the proper MLPNN model, there was no significant difference between any of the configurations, and no model was able to perform better than random chance (50%). This final case study showcases the most significant effect of covariate shift, resulting in models that are unable to perform better than random chance due to the significant inter-participant variability that exists between the participants.

3.7 Discussion

Our empirical results show that improper dataset evaluation can lead to unrealistic and overestimated accuracies for general population EEG cross-participant models. Table 6 specifies the extent of these differences in error rates between improper and proper methods, ranging from a 35% increase in error rate for the *confused students* dataset, all the way up to a 3900% increase in error rate in the case of the *schizophrenia* dataset. As mentioned in Section 3.6, the diversity of these datasets and the methods used provide evidence that performance overestimation due to improper data partitioning is indeed a phenomenon of EEG that is not unique to any one subset of experiment, task, participant, or equipment used, nor is it merely an aspect of only certain EEG features or types of machine learning models. Instead, the risk of performance overestimation is an inherent phenomenon of individual differences in EEG that should always be considered when developing general population EEG cross-participant models.

Table 6. Validation results for the five case studies (95% CI). All datasets are binary classification tasks, and all datasets are balanced except for the Alcoholism dataset. This gives chance error rate for Alcoholism defined as 0.36, and 0.50 for all other datasets. Results should be compared within datasets (left to right) between the improper and proper method. The proper method always reveals a significantly greater error rate than the improper method, suggesting the risks of overestimation of performance, which can result from using the improper method.

Dataset	Architecture Used	Error Rate - Improper Method	Error Rate - Proper Method
Driver Fatigue	MLPNN	0.09 (0.083, 0.097)	0.466 (0.448, 0.472)
Confused Students	Bi-LSTM	0.31 (0.274, 0.346)	0.416 (0.372, 0.448)
Alcoholism	LSTM	0.16 (0.12, 0.18)	0.31 (0.29, 0.33)
PTSD	MLPNN	0.005 (0.0022, 0.0078)	0.197 (0.1811, 0.2129)
Schizophrenia	MLPNN	0.008 (0.0061, 0.01)	0.50 (0.44, 0.56)
Schizophrenia	RFC	0.059 (0.054, 0.064)	0.50 (0.44, 0.56)

Proper care with EEG data preparation has been a subject of recent exploration by Li et al. as well [97]. Li et al. demonstrated that due to EEG's non-stationarity, proper guidelines for the design of the experiment much be followed in order obtain model results that are not overestimated, particularly in the block design of the experiment so that stimuli of different classes are intermixed. If not followed, models instead learn to classify through arbitrary temporal artifacts, giving the false appearance of high performance. Our findings are synergistic with Li's: we demonstrate the necessity of partitioning the data properly when performing machine learning on collected data *after* the experiment is complete; due to individual differences, proper care with EEG data partitioning by participant yields more accurate estimates of model results on future data. Together, both Li et al.'s guidelines for the design of the experiment and our guidelines for proper post-experiment dataset partitioning should be followed in order to obtain results for EEG cross-participant models that are representative of the model's performance on the general population.

In Section 3.5, we demonstrated how t-SNE can be used to visualize covariate shift between participants due to their inter-participant variability, and we also illustrated how the *shifted Heaviside* and the *shift to median* transformations could be utilized to reduce this inter-participant variability. Additionally, for the purpose of demonstrating the relationship between this inter-participant variability and covariate shift, we explored the effect of these transformations in improving cross-participant model accuracy for both improper and proper model creation across two different feature spaces (*entropy* and *spectral* features). As can be seen in Figures 15 and 16, both transformations were successful in reducing inter-participant variability for both feature spaces; however, only the *shift to median* transformation resulted in a dominant increase in accuracy of the 95% confidence intervals for both feature spaces for proper model creation, with the *shifted Heaviside* transformation having no improvement in model accuracy. In contrast to the *shifted Heaviside* results, Arevalillo-Herraez et al. (the originators of the *shifted Heaviside* transformation) had improvement of model accuracy in three different datasets they utilized, all of which were affect recognition-based datasets with arousal and valence features [82]. In their research, they also followed proper dataset partitioning guidelines and utilized LOPO CV. This suggests that a transformation that results in a reduction or elimination of inter-participant variability does not necessarily imply an improvement in cross-participant model accuracy.

3.8 Conclusions

As mentioned in Section 3.4, five out of six EEG deep learning models in research today are cross-participant models, with only one out of those five models following some method of proper dataset partitioning to ensure the model was tested with unseen participants [11]. Our empirical results show that models that utilize improper dataset evaluation have overestimated and unrealistic accuracies for the general population, with the difference in error rates for improper versus proper dataset evaluation ranging from a 35% increase in error rate up to a 3900% increase in error rate. These empirical findings suggest that if this trend continues, the body of research for EEG cross-participant models will become diluted with research that claims overestimated and unrealistic performance metrics, both downplaying the true difficulty in creating a high-performing EEG cross-participant model, and also slowing scientific progress of researching methodologies, which results in cross-participant models that are truly high performing for the general population. Thus, it is absolutely critical that the body of research corrects this trend and follows the proper dataset partitioning guidelines described in this research. Specifically, it means that:

- Data from participants used for model training must not be used for model validation or testing.
- Participants that are utilized for validation must not be used for testing.

This ensures the model is tested with unseen participants and reflects its intended purpose.

These findings extend beyond individual researchers. In addition, it is also important that data contributors, and the owners and maintainers of dataset repositories (e.g., Kaggle [72] and the UCI machine learning data repository [73]) managing human data ensure these guidelines are followed as well. Specifically, for these repositories, we recommend that:

- Any EEG data that are made available for download should always have (de-identified) participant labels available so that users may properly partition the data themselves.
- If the data contributors or maintainers decide to pre-partition the data into separate training and test datasets (as is sometimes done for competitions of machine learning models), then proper dataset partitioning guidelines should be followed for preparing those training and test datasets before they are made available for download by the general public.

We also recommend that the repository include these guidelines of proper dataset partitioning with all hosted EEG datasets, as this would help spread the word in regard to proper dataset partitioning and inform users who are unaware of inter-participant variability and its effects.

Lastly, we strongly recommend that the "Neurotechnologies for Brain–Machine Interfacing" group of the Institute of Electrical and Electronics Engineers Standards Association (IEEE SA) consider and adopt these guidelines for all future proposals of standards. In this group's most recent *Standards Roadmap* [98], stakeholders and experts across government, academia, and industry identified the existing gap in the standardization of performance assessment and benchmarking for BMI as a clear priority for standardization [99]. Specifically, the proposal should identify these guidelines as a minimal reporting requirement for performance evaluation of EEG cross-participant models, leading to standardization in reporting how the data are partitioned, identifying their limitations, and curbing performance claims accordingly.

3.9 Appendix

In this appendix, we include any pre-processing details that were provided by the originators of the datasets used within this research.

3.9.1 Appendix A.1. Driver Fatigue Data.

EEG was recorded using a 32-channel electrode cap, with two of the channels being reference channels linked to mastoid electrodes [84]. Scan 4.3 software of Neuroscan was used for preprocessing, with raw signals filtered by a 50 Hz notch filter and a 0.15 Hz to 45 Hz band pass filter in order to remove noise.

3.9.2 Appendix A.2. Confused Students Data.

No known preprocessing information was provided by the originators.

3.9.3 Appendix A.3. Alcoholism Data.

EEG correlates were sampled from 62 scalp electrodes and two EOG electrodes, at a sampling rate of 256 Hz [73]. Sampling started at 190 ms before onset of stimulus in order to record a pre-stimulus baseline, and EEG correlate durations provided in the dataset were 1 s in duration. Sensor values were provided in μV, resulting in a sequence of 256 temporally organized values for each EEG channel. Trials with excessive eye or body movements (>73.3 μV) were rejected online. Only artifact free EEG segments were used to include eye blink artifacts. EEG electrodes were referenced to node site Cz during EEG measurement.

3.9.4 Appendix A.4. PTSD Data.

For this dataset, there were 33 channels of EEG recorded, with two of the 33 channels being used for ground and reference, and at a sampling rate of 5000 Hz [80]. EEG preprocessing was performed in the proprietary software BrainVision Analyzer2. MRI gradient artifacts and cardio ballistic artifacts were removed using the template subtraction method. Then, the EEG was down-sampled to 250 Hz and filtered with a 40 Hz low-pass filter. Then, ICA was applied to remove residual cardioballistic artifacts as well as blink and saccade artifacts. Time periods of head motion were removed.

3.9.5 Appendix A.5. Schizophrenia Data.

Vertical EOG (VEOG) and Horizontal EOG (HEOG) were also collected for the purpose of capturing eye movement and blinks [81]. Due to the size of the raw EEG signals, preprocessing was already performed on the dataset prior to its upload for public use. This preprocessing included re-referencing to the averaged mastoid channels, applying a 0.1 Hz high-pass filter, interpolation of outlier channels, and rejection of outlier components and outlier trials due to EEG artifacts using the FASTER artifact rejection method [54].

IV. Detection and Mitigation of Inefficient Visual Searching

4.1 Paper Overview

This conference paper was presented at the 2020 Human Factors and Ergonomics Society (HFES) annual meeting on 5 October 2020, was awarded the Best Student Paper Award within the Augmented Cognition Technical Group, and was published in the conference proceedings [3].

Additional details for the methods can be found in the Appendix at the end of this Chapter in Section 4.7.

4.2 Abstract

A commonly known cognitive bias is a confirmation bias: the overweighting of evidence supporting a hypothesis and underweighting evidence countering that hypothesis. Due to high-stress and fast-paced operations, military decisions can be affected by confirmation bias. One military decision task prone to confirmation bias is a visual search. During a visual search, the operator scans an environment to locate a specific target. If confirmation bias causes the operator to scan the wrong portion of the environment first, the search is inefficient. This study has two primary goals: 1) detect inefficient visual search using machine learning and Electroencephalography (EEG) signals, and 2) apply various mitigation techniques in an effort to improve the efficiency of searches. Early findings are presented showing how machine learning models can use EEG signals to detect when a person might be performing an inefficient visual search. Four mitigation techniques were evaluated: a *nudge* which indirectly slows search speed, a *hint* on how to search efficiently, an *explanation* for why the participant was receiving a *nudge*, and *instructions* to instruct the participant to search efficiently. These mitigation techniques are evaluated, revealing the most effective mitigations found to be the *nudge* and *hint* techniques.

4.3 Introduction

Cognitive biases can act as helpful heuristics in decision making when one is in a known environment with known variables; however in unpredictable or unknown environments, they can lead one astray and result in suboptimal outcomes [100, 101]. Suboptimal outcomes can range from a slight delay in the decision-making process to drawing an erroneous conclusion. One form of cognitive bias is confirmation bias: the tendency for people to seek for, interpret, favor, and recall information in a way that confirms one's pre-existing beliefs and/or hypotheses [102].

In military operations, confirmation bias can influence errors in decision-making that lead to disastrous outcomes. In 1988, the USS Vincennes mistakenly shot down an Iranian commercial airliner. Existing tensions and recent attacks on the US military in the area primed the Captain to assume the presence of threats. Despite many indications that the airliner was not a military fighter jet, the Captain overweighted the few misleading indications which suggested the approaching aircraft might be an Iranian F-14. The accident was partially attributed to the Commander's over-reliance on information suggesting the aircraft was a military fighter jet. Confirmation bias was partly to blame in the incident which resulted in the loss of the 290 passengers onboard the aircraft [4].

With the ubiquity of sensors and growing data fusion capabilities in military operations, the ability to quickly and accurately assess information is paramount. Due to stress and fast-paced operations, military decisions in high-information environments are particularly susceptible to cognitive biases. A method to detect and mitigate biased sub-optimal decisions could help avert disasters like the Vincennes incident.

One of the military operators' tasks is visual search. A visual search requires the operator to perform a scan of an environment to locate a specific object or feature while ignoring other distracting objects or features, with the order of targets searched determining one's visual search pattern (VSP) [1]. In order to perform a search quickly, operators will often fall back upon prior knowledge of the situation [103]. Falling back upon prior knowledge can yield poor decisions if that prior knowledge is inaccurate; overweighting incorrect information can lead to incorrect or inefficient visual searches. Literature suggests that inefficient search takes longer and is less accurate than efficient search [2]. While there are successful mitigation techniques for confirmation bias, there exists little research into successfully and consistently mitigating confirmatory, inefficient visual search.

The ability to detect and mitigate inefficient visual searches would assist military operators to safely and effectively perform their jobs. This study explores both detecting inefficient searches, and mitigating them.

First, this study explores the efficacy of detecting inefficient search using machine learning models trained with Electroencephalography (EEG). Five machine learning models were evaluated for detection of inefficient search: Linear Discriminant Analysis (LDA), Random Forest Classifier (RFC), Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), and Temporal Convolutional Network (TCN), with LDA achieving the greatest mean balanced accuracy, detecting inefficient VSP in 58.1% of trials. While certain participants' machine learning models performed well, overall, the models for each dataset performed only marginally better than chance. However, these results do suggest that it is possible to classify a search as efficient or inefficient from EEG.

Second, four mitigation techniques for inefficient visual search are evaluated. This study extends a previous visual search experiment which focused on identifying and mitigating confirmatory search [2]. The experiment created an environment where inefficient search is naturally the default behavior due to inherent biases towards confirmatory search, which aren't simply a result of feature-priming [35, 36]. However, because it is possible to perform a confirmatory search while also being efficient, and also because it is possible to perform a non-confirmatory search while being inefficient, this research focuses on encouraging efficient searches rather than using Rajsic's approach of discouraging confirmatory searches. Results revealed that the most effective inefficient search mitigation techniques are the *nudge* and the *hint* technique. These techniques significantly increased the number of efficient visual searches performed.

4.4 Method

4.4.1 Participants.

Sixteen United States Air Force personnel participated in this experiment. Participant age ranged from 22 to 37 (M = 28.9; SD = 5.2). All participants had, at minimum, a Bachelor's degree and all used computers daily in their job and personal lives. All participants self-reported having a sleep quality of "fair" or better (M = 6.7 hours; SD = 0.9).

4.4.2 Appartus.

Our Efficient Search Experiment (ESE) detected inefficient visual search and dynamically applied various mitigation techniques based on the individual participants' search patterns over the course of the experiment. The two-fold goal of detection and mitigation for inefficient visual search is semi-independent in this task: the participant's actual VSP is collected using gaze tracking and then evaluated for efficiency. This efficiency information is both used for the online mitigation system and stored for the offline machine learning activity.

The ESE consisted of 24 blocks each with 20 trials for a total of 480 trials. In each block of trials, participants were presented with search stimuli consisting of 8 colored circles, arranged in a ring, with white letters in the center of each colored circle. The white letters were one of p, q, b, d, intentionally chosen to reduce the chance that the target letter was easily distinguishable from other letters on the screen. There were only two colors per block. An example of a single trial screen can be seen in Figure 17.

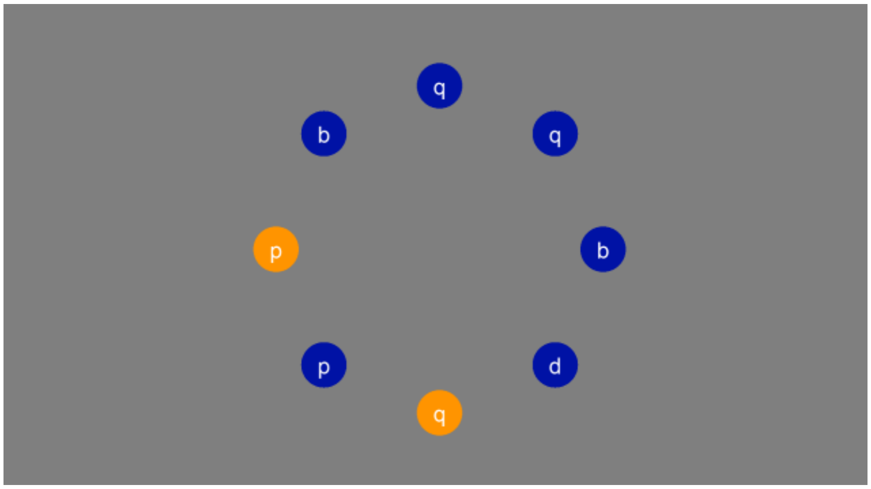


Figure 17. An example of a trial with target letter ‘d’.

Participants were instructed to indicate whether or not a specific target letter’s circle was a template color, called the target color. There is exactly one instance of the target letter present in each trial, and the template color would not change for the duration of the block. During the block, various proportions of the template color and non-template color appeared.

Rajsic’s research concluded that participants most often searched the target color first, leading to inefficient searches when the majority of circles matched the target color [2], such as in Figure 17, where the target was ‘d’ and the query asked the participant if the color of the target was blue.

The ordered search sequence of the participant’s gaze over the circles constituted the participant’s VSP; a Smart Eye Pro gaze tracking system (“SE PRO - Smart Eye”) was used to analyze the participants’ VSP and an algorithm marked trials as efficient/inefficient in real-time based on which circles were gazed at. In our experiment, a participant conducting an inefficient search gazes at extra circles beyond the minimum required circles for an efficient search. A trial was marked as an efficient trial if the participant searched only the required minimum circles to determine what color the target letter’s circle was, while the trial was marked as inefficient if the participant searched other circles.

The first eight blocks of the experiment were unmitigated. Starting in the ninth block, a mitigation technique known as the *nudge* was applied if the participant performed an inefficient search on more than half of the search trials of the previous block. The nudge consisted of hiding the letters on the colored circles until the gaze tracker determined the participant visually fixated upon the circle, which caused the letter to be revealed. Additionally, other mitigation techniques, (*hint*, *explanation*, and *instructions*) were presented to the participant during the ESE in later blocks. The *hint* consisted of showing the participant how to perform an efficient search, the *explanation* involved telling the participant that the nudge was occurring when they were performing mostly inefficient searches in the prior block, and the *instructions* screen explicitly instructed the participant to perform an efficient search. While the nudge was applied dynamically based on whether the participant was searching efficiently in the prior block, the other mitigation techniques were presented before pre-determined blocks. The hint was presented before starting the 11th block, the explanation was presented before the 14th block, and the instructions were presented before block 18 and again before each of the remaining blocks.

To reduce the likelihood of a learning effect, participants were trained and practiced 20 trials on a day prior to the experiment. Participants were also instructed what to expect from the gaze tracking equipment.

4.4.3 Procedure.

Independent variables. The independent variables in the ESE were: the number of template-color-matching circles and template color-mismatching circles, the presence of a nudge, whether the participant had received the hint, whether the participant had received the explanation, and whether the participant had been instructed to perform an efficient search.

Dependent variables. During the ESE, the participant's behavioral response and physiological measures were collected. Behavioral measures included the participants' gaze-tracked VSP, and their response time and accuracy. Physiological measures included EEG, Electrooculography (EOG), and Electrocardiography (ECG).

The collected behavioral data was evaluated to determine the distribution of VSPs participants initially used, as well as the effects of the mitigations on their VSPs in later trials. The physiological data was investigated to determine whether machine learning classification models could be trained to identify when a participant performed an efficient or inefficient search. Both within- and cross-participant models were considered.

Detection analysis. The EEG data that was used to conduct machine learning was obtained by using the Cognionics (CGX) Mobile-72 EEG system. The raw data from the 64 EEG electrodes was pre-processed in EEGLAB v2019.0 by following Makoto's preprocessing pipeline [104]).

Datasets. Two types of datasets were used for machine learning on the EEG signal: a frequency-domain dataset and a time-domain dataset. The frequency-domain dataset consisted of features extracted from an EEG signal segment recorded during the two seconds prior to the participant selecting which color the target letter's circle was. This dataset was obtained by using Morlet wavelet transforms in MATLAB to determine the frequency-domain mean power of the five clinical frequency bands: delta (1-6 Hz), theta (7-11 Hz), alpha (12-15 Hz), beta (16-22 Hz), and gamma (22-30 Hz) [18]. Five frequency-spectral-power features extracted from EEG were computed for each of the 64 channels. For each participant, this created 480 observations (1 per trial), each with 320 features. Each of the 480 observations in the time-domain dataset consisted of the time-series voltage values over each 2 second period for each of the 64 EEG electrodes.

Furthermore, because the nudge altered fundamental aspects of the search such as gaze fixation duration and saccade intervals, four datasets were created to isolate these effects for training and testing of the various models. Two datasets consisted of only trials which did not have the nudge (*Clean-Balanced* dataset & *Clean-Unbalanced* dataset), another dataset consisted of only trials which contained the nudge (*Nudge* dataset), and a final dataset consisted of all trials from the experiment (*Combined*). The *Clean* and *Nudge* datasets were created to determine if the nudge was having an effect on the models' performance. The *Clean-Balanced* dataset was created as the class distribution was widely imbalanced for all participants, with only between 7% to 24% of the trials being efficient trials (depending on the participant). The dataset was created by randomly selecting a subset of the majority class (inefficient) in order to reduce its size to that of the minority class (efficient). Both the *Balanced* and *Unbalanced* datasets were explored in order to investigate the effect of class imbalance on model performance. The *Combined* dataset was explored to determine if the model performed best when it has all data available for training, regardless of the effect of the nudge.

Machine learning models. Five machine learning model classifiers were evaluated: Linear Discriminant Analysis (LDA), Random Forest Classifier (RFC), Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), and Temporal Convolutional Network (TCN). The LDA, RFC, and ANN models used the frequency-feature dataset while the LSTM and TCN models used the time-series dataset. Python version 3.7.5 was used for machine learning. The LDA and RFC models were created using scikit-learn version 0.21.3 while the ANN, LSTM, and TCN models were created using Keras version 2.3.1. Using a grid search to choose hyperparameters, each model was tuned for each participant using a training-, validation-, and test-set approach.

An untuned LDA model was used as a baseline score because LDA models are stable even with a small number of observations [96]. Despite not being able to use feature selection, the high dimensionality of the dataset was accounted for by choosing the LDA's shrinkage parameter to be `auto` and its solver to `lsqr`. Using these settings helps improve the LDA's estimation of covariance matrices for datasets that have high dimensionality [105].

The RFC models were tuned to find the optimal number of trees, the number of features considered, and the maximum depth of the tree. The optimal number of trees was found by training an RFC using the square root of the max number of features ($\sqrt{320}$=$\sim$18). Using the optimal number of trees, a hyperparameter sweep was performed to find the optimal number of maximum features and the maximum depth.

A hyperparameter sweep was conducted on the ANN models to find the optimal number of hidden layers, the number of hidden nodes per layer, and the learning rate. The activation function for each layer was a rectified linear unit (`ReLU`) and the optimizer used was `Adam`.

The LSTM model was inspired by Kumar et al.'s LSTM model used in the OPTICAL predictor [106]. It used two CuDNN LSTM layers and the number of hidden units per layer was tuned.

The TCN model used was inspired by Bai et al.'s model and was created through hyperparameter testing of kernel size, dilations, filters, and stacks [66]. The ANN, LSTM, and TCN model architecture can be seen in Figure 18.

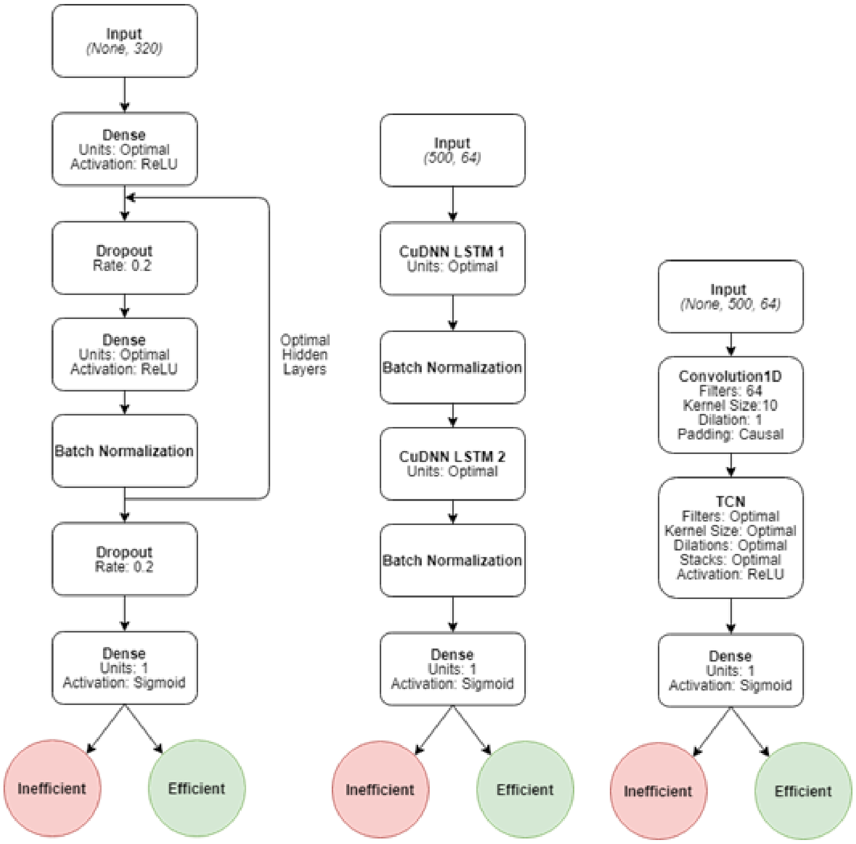


Figure 18. From left to right: the ANN, LSTM, and TCN model architectures.

Mitigation analysis. Based on the participants' VSPs, each trial was classified as either an inefficient, efficient, or as a circular search. Additionally, other behavioral measures such as trial search times and accuracy results were recorded. To understand what VSP participants naturally used, the VSPs from the first eight unmitigated blocks were determined for each participant. To determine whether the mitigations had the desired effect of increasing the number of efficient searches, these initial VSPs were compared to the VSPs during the final seven blocks of the ESE.

To determine the effect of each mitigation technique on the number of efficient searches, a linear regression was performed on each participant's number of efficient searches per block. The number of efficient searches per block was the response variable, while the presence or absence of a nudge, hint, explanation, or instruction were the independent variables. The p-value of the independent variables was considered and the log-worth of each independent variable was examined. The log worth is the $-log_{10}$ of the p-value and can indicate greater effect when the reported p-values are too small to differentiate. A higher log worth indicates that the variable is more significant and typically log worth values over two are considered significant.

Finally, a two-sample paired t-test was performed to determine whether there was a difference in the accuracies between efficient and inefficient searching. The data met all of the assumptions of the t-test. Additionally, the same test was performed to determine whether there was a difference in the search times between efficient and inefficient searching.

4.5 Results

Detection results. Five machine learning models were evaluated for each of the four datasets. The output of each model was a prediction of whether the EEG data was recorded during an inefficient or efficient search. Class distribution (inefficient vs. efficient) varied widely for each dataset depending on the participant. Scikit-learn's balanced-accuracy score (the average of recall obtained on each class) was used to report performance. The balanced accuracy score indicates a measure of each model's performance.

The within-participant classification results for each model for each dataset can be seen in Table 7. The average balanced accuracy across all participants is shown, as well as the number of participants with statistically significant results better than random chance included in parentheses next to the average (random chance being defined as a naïve classifier with accuracy of 50%). Overall, the *Combined* dataset resulted in the most participant models with statistically significant balanced accuracies for 11 LDA, 9 RFC, and 6 ANN participant models with significant results. In contrast, the *Nudge* dataset had the fewest of participant models with significant accuracies, and low mean accuracies close to random chance. For the *Clean* datasets, the *Balanced* dataset had a greater mean accuracy than the

Unbalanced dataset for both LDA and RFC classifiers, and also had a greater number of participant models with significant results for the RFC classifier.

Table 7. Mean balanced accuracy scores of the within-participant models. Parentheses indicate the number of participants w/ statistically significant accuracies of the 95% confidence interval in comparison of the 95% confidence intervals.

Within-Participant Dataset	**Mean Balanced Accuracy %**				
	LDA	**RFC**	**ANN**	**LSTM**	**TCN**
Nudge	49.2 (1)	50.5 (1)	51.7 (3)	51.4 (2)	50.7 (1)
Clean-Unbalanced	54.6 (5)	50.7 (1)	53.9 (4)	53.1 (5)	49.4 (1)
Clean-Balanced	59.2 (5)	58.2 (9)	52.5 (3)	53.7 (6)	50.8 (2)
Combined	58.1 (11)	56.2 (9)	53.2 (6)	55.3 (6)	49.7 (2)

The LDA model performed the best with a mean balanced accuracy on both the *Combined* (58.1%) and *Clean-Balanced* (59.2%) datasets. The participant models with the highest balanced accuracies were also from the *Combined* dataset, with accuracies of 74.8% (LDA), 66.7% (RFC), and 64.5% (ANN).

Cross-participant models were also explored to determine the models' performance when classifying an unseen participant. Because the frequency-feature models performed better than the time-series models, only LDA, RFC, and ANN models were explored. For cross-participant models, one participant's data was held out as the test set while the other 15 participants were used for training. This was repeated 16 times so that each participant's data was held out once.

Mean balanced accuracy scores of the cross-participant models for the four datasets can be seen in Table 8. Across all four datasets, the RFC classifier performed the best, achieving statistically significant accuracies for both of the *Clean* datasets. The *Clean-*

Balanced dataset had the highest mean balanced accuracies of 58.5% (±0.17%, 95% CI) for LDA and 59.0% (±0.17%, 95% CI) for RFC classifiers.

Table 8. Mean balanced accuracy scores of the cross-participant models. Bold indicates statistically significant accuracy of the 95% confidence interval in comparison of the 95% confidence intervals.

Cross-Participant Dataset	Mean Balanced Accuracy %		
	LDA	RFC	ANN
Nudge	50.2	53.2	49.9
Clean-Unbalanced	50.0	**57.3**	50.0
Clean-Balanced	**58.5**	**59.0**	50.0
Combined	51.0	54.0	50.1

Mitigation results. Initially, participants overwhelmingly performed inefficient searches. In the first eight blocks, in aggregate, 73.68% of participant searches were inefficient, 19.14% were efficient, and 7.18% were circular.

The average search time during an efficient search was faster (1.99 ± 0.37 seconds) compared to the average search time during an inefficient search (2.29 ± 0.50); there was a statistically significant decrease in average search times ($t(15) = 5.53$, $p = 0.00005$, $\alpha = 0.05$) of 0.30 seconds. The accuracy during an efficient search was higher (96.33% ± 2.16%) than the accuracy during an inefficient search (93.92% ± 2.57%) - a statistically significant 2.41% increase in accuracy ($t(15) = 5.59$, $p = 0.00005$, $\alpha = 0.05$).

Because efficient searches are both faster and more accurate, techniques which shift behavior toward efficient searches are desirable, and the techniques used in this experiment achieve that effect. In the last seven blocks, efficient searches were increased by 32.27% to 51.41%, inefficient searches were decreased by 26.15% to 47.53%, and circular searches were decreased by 6.12% to 1.06%. The mitigation techniques of the nudge and the hint had the largest effects on increasing the number of efficient searches, with the nudge having a log worth of 10.664, and the hint having a log worth of 8.493, and both techniques being found statistically significant ($p < 0.0001$, $\alpha = 0.05$).

4.6 Discussion

Existing methods of detecting inefficient searches will review search patterns once the task is complete and thus do not allow for a detection of an inefficient search in real-time. Furthermore, the majority of visual research has shown that most humans will naturally use an inefficient search method [2]. Our research findings support this hypothesis.

Military operators use visual searches every day in their job. This includes pilots scanning instrument gauges, intel analysts scanning satellite imagery, and doctors scanning patient x-rays. These military members would have lower performance if they used inefficient visual search. Therefore, our study investigated both mitigation of inefficient visual search, as well as how to detect it. For the goal of inefficient search detection, while certain participants' models performed well, overall, the models for each dataset did not perform statistically significantly better than chance. To classify an efficient or inefficient search from EEG signals will require much more future work.

For the goal of inefficient search mitigation, this work measured how humans perform a visual search, as well as the effectiveness of techniques to mitigate inefficient search, with the *nudge* technique being the most effective.

Future research. This work extended a visual search experiment to include a dynamic mitigation system based on each participant's own VSPs. As this was the first iteration of this experiment, there is room for improvement if a future experiment were to be conducted. The data used in this experiment depended on the way the EEG signals were epoched. In this experiment, the epochs started two seconds before the participants pushed the key to indicate their answer. Future work should include variations on epoching to determine the best epoch strategy for EEG signal capture.

Since there were 320 features and at most 480 trials per participant, detection model performance could also be hindered by overfitting, as there are almost as many features as observations. Future work should explore features selection and regularization – perhaps choosing the most salient features suggested by neuroscience literature, or through dimensionality reduction techniques such as Principal Component Analysis (PCA).

Although explained during training, some participants said they didn't know there was exactly one instance of a target letter appearing in each trial's stimuli. The participants who realized this during the experiment said that it changed their subsequent search patterns, adding a confound since this change was not due to a mitigation. A recommendation during training is to increase the emphasis that there is exactly one instance of the target letter present per trial – but it is important not to suggest any particular search pattern in order to maintain the ability to measure the effects of the mitigations.

Some participants felt as if they were "cheating" when using an efficient search pattern. A recommendation for a modification to the ESE is to emphasize to the participants during training that they are welcome to search the stimuli in whichever manner they feel is the most accurate and efficient, no matter what searching method they use.

Instead of using a gaze tracking system to detect inefficient patterns, a future ESE could use the detection of inefficient searching through the use of EEG signals. During training, initial data could be gathered on a participant to train a machine learning model. During the experiment, the model would then output the likelihood that the participant is conducting an inefficient search and apply appropriate measures.

4.7 Appendix - Methods

4.7.1 Participants.

Participants did not receive compensation for the experiment. Participants read an Informed Consent Document (ICD) before receiving training for the experiment, and were also allowed review of the ICD before start of the experiment. No participants were excluded, with exclusion criteria being the following:

- Unable to use a keyboard
- Visual impairment or inability to view information on a computer screen, to include color blindness (a simple test was administered during training to test for this)
- Specific motor, perceptual, or cognitive conditions that preclude one from operating a computer, reading small characters on a computer monitor, or hearing and comprehending verbal commands presented by the experimenter or through computer speakers
- Use of certain hair products (e.g. hair gel) which will interfere with the EEG electrodes
- Unusually thick hair which may prevent a proper fitting of the EEG cap
- Head size which is not coverable by the available EEG caps (too large or too small)

4.7.2 Stimuli.

There were only two colors per block, and circles could be any one of three different pairs of colors: purple and yellow, blue and orange, or green and red, with the first color listed for each of these pairs always being the target color for that pair, and grey always used as the background color. An example of a trial screen for each of these different color combinations can be seen in Figure 19. Stimuli were displayed on a 4K LCD monitor at a resolution of 1920x1080 and a refresh rate of 60 Hz, with circles being 100 pixels in diameter (2° of arc) and distanced 360 pixels from the center of the screen.

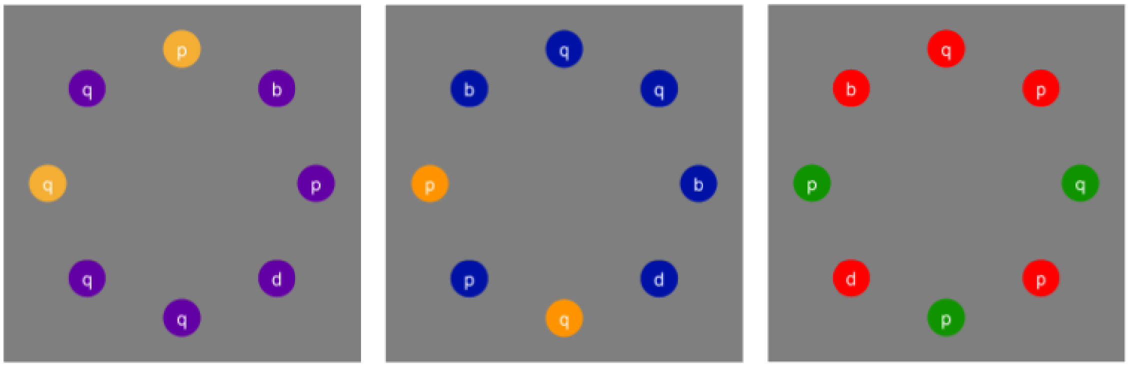


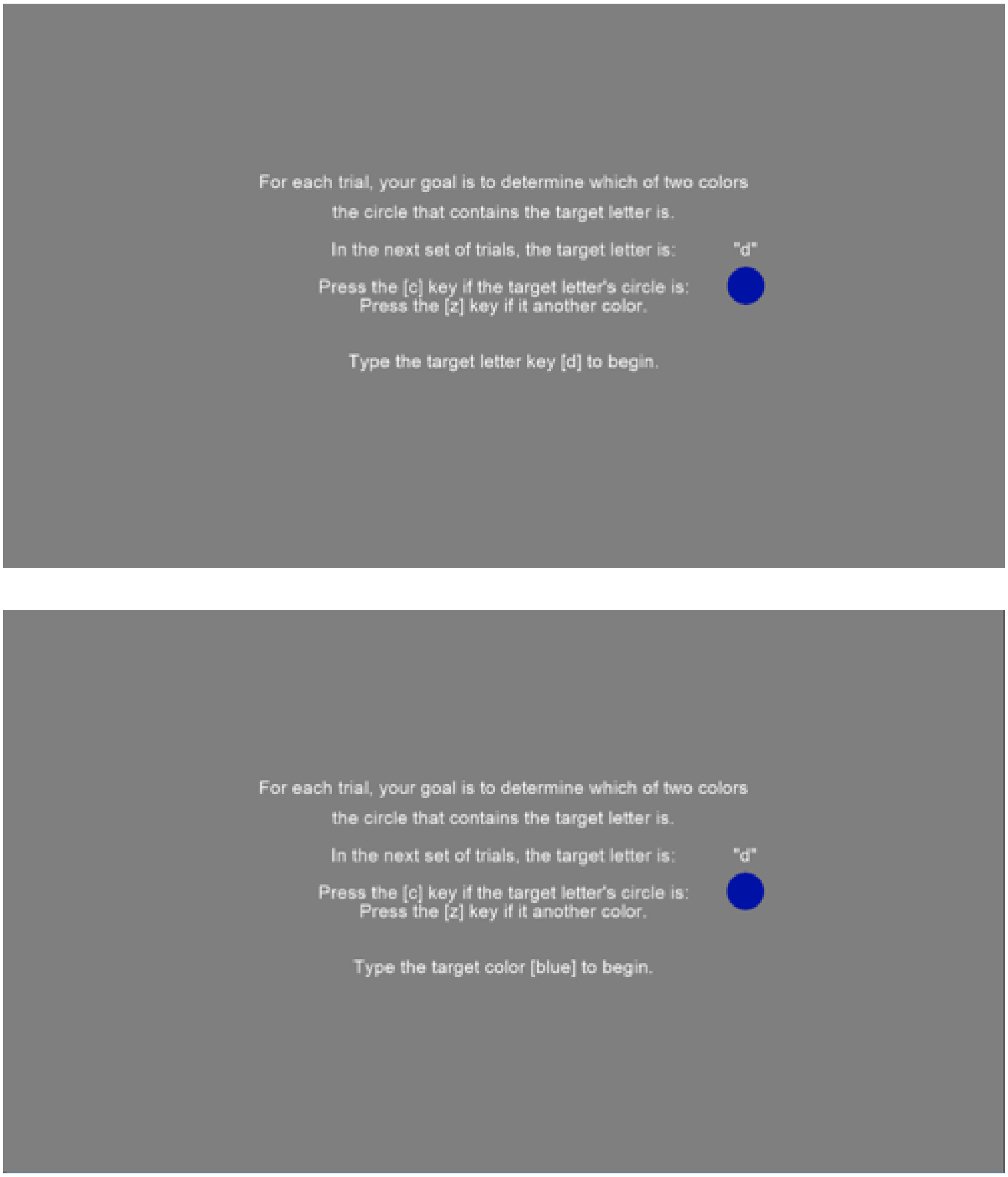
Figure 19. The three pairs of color combinations: purple and yellow (left), blue and orange (middle), green and red (right).

4.7.3 Apparatus.

Prior to the day of the experiment, participants had a training session in order to reduce the likelihood of a learning effect. During this training, participants reviewed the ICD and indicated their verbal consent. Participants were also administered a simple color-blindness test, which tested their ability to distinguish and count the number of colored circles shown on screen (with the test administered for each of the three color pairings). They also practiced a block's worth of trials (20), and were also instructed on what to expect from the gaze tracking equipment. Finally, head measurements were taken in order to know the correct EEG cap to have prepared for them on the day of the experiment.

On the day of the experiment, before starting the ESE, participants were given a pre-experiment questionnaire in order to obtain a self-assessment of how well rested the participant was, and if there were any medical abnormalities which may affect the results of the experiment (questionnaire details can be seen in Section 4.7.4).

The ESE consisted of 24 blocks each with 20 trials for a total of 480 trials. Participants were instructed before each block to indicate whether or not a specific target letter's circle is a template color, called the target color. There is exactly one instance of the target letter present in each trial, and the template color does not change for the duration of the block. An example of the block instructions screen presented at the start of each block can be seen in Figure 20. Prior to the block instructions screen, a two minute baseline of EEG was recorded.



For each trial, your goal is to determine which of two colors
the circle that contains the target letter is.

In the next set of trials, the target letter is: "d"

Press the [c] key if the target letter's circle is: ●
Press the [z] key if it another color.

Type the target letter key [d] to begin.

For each trial, your goal is to determine which of two colors
the circle that contains the target letter is.

In the next set of trials, the target letter is: "d"

Press the [c] key if the target letter's circle is: ●
Press the [z] key if it another color.

Type the target color [blue] to begin.

Figure 20. An example of an instructions screen. At the top is what is first presented to the participant. After typing the letter 'd', the bottom screen would be displayed, prompting the participant to type 'blue' in order to transition to the next screen.

During the block, proportions of the template color and non-template color vary, as well as the amount that each of these trials appears within a block. These proportions and how often each appeared within a block are listed below, as well as two additional properties of trials within blocks. An example of each of these trials and their differing proportions can also be seen in Figure 21.

- 35% of the trials had six target color matching circles
- 35% of the trials had five target color matching circles
- 15% of the trials had three target color matching circles
- 15% of the trials had two target color matching circles
- A trial does not repeat within a block
- The minority-colored circles are separated by at least one of the majority-colored circles. This ensures that a circular search can never also be an efficient search.

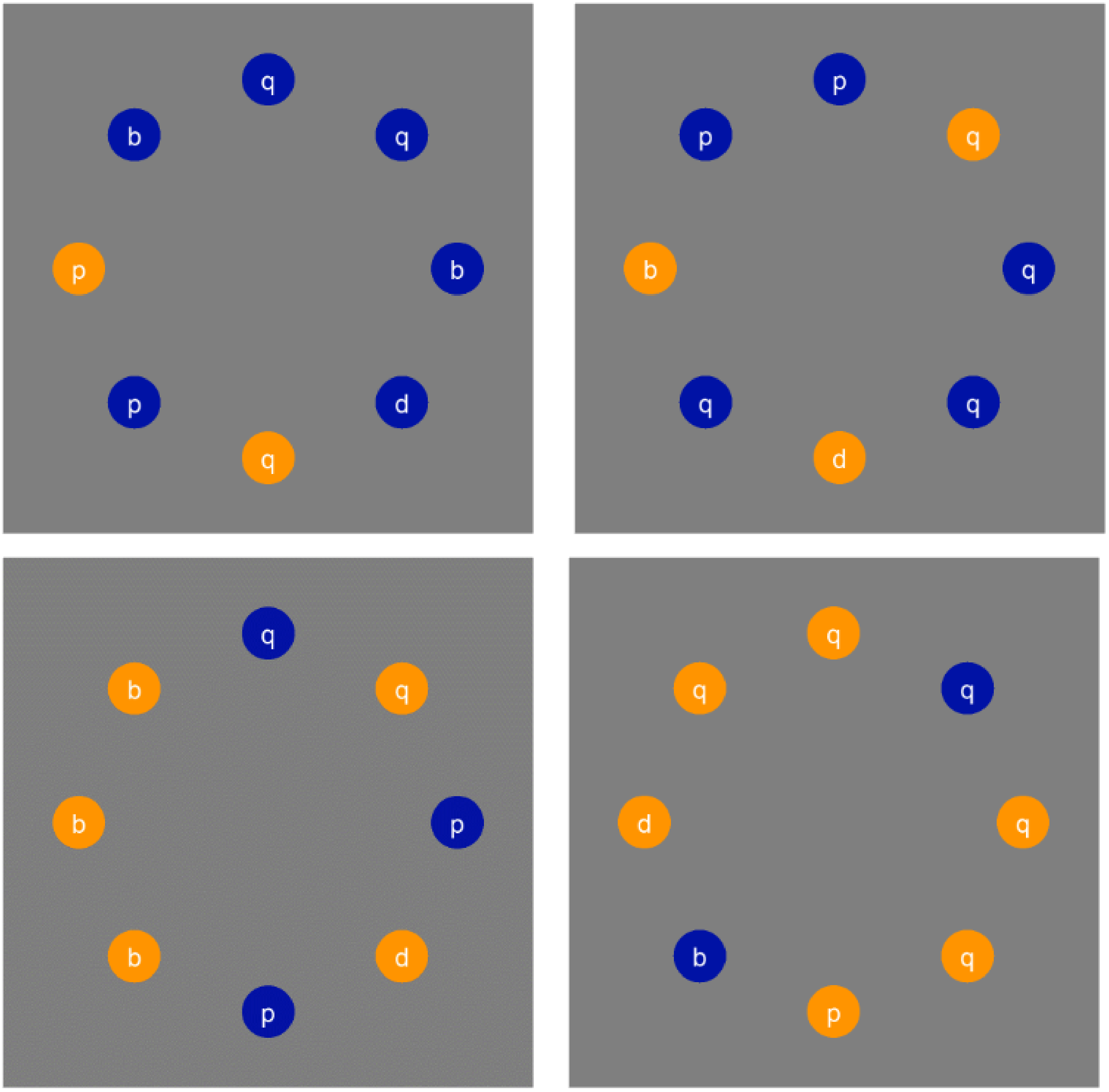


Figure 21. Examples of each of the varying proportions of colors, six (top-left), five (top-right), three (bottom left), and two (bottom-right) template color matching circles.

Searches would also be annotated as other types of searches if it met the criteria for those types, as not all of these types are mutually exclusive, with all types listed below:

- Efficient: Participant searched only the required minimum circles to determine what color target letter's circle was
- Circular: Participant searched the circles in a circular manner

- Miss: Participant gazed upon the target letter and then continued searching additional circles

- Non-Normal: Participant first searched all minority-colored circles, and then searched additional majority-colored circles

- Multiple-Minority-Only: Participant searched only the minority-colored circles and fixated upon one or more of the circles more than once

- Majority-then-Minority: Participant first fixates upon a majority-colored circle, but then searches only minority colored circles

Because the gaze tracker is not a perfect representation of a participant's gaze location and instead has a measure of error which differs from participant to participant, the fixation area for a letter was three times the area of the circle stimuli (6° of arc). Additionally, for a fixation to register within the algorithm (i.e. a circle to count as searched), the gaze tracker would need to be on the location for ten or more frames (60 Hz = 16.67ms per frame, so 166.67ms or more). Before each trial began, a fixation cross appeared which required participants to fixate upon the cross for five frames before the trial would begin. To reduce the likelihood of a learning effect, participants were trained and practiced 20 trials on a day prior to the experiment, and were also instructed what to expect from the gaze tracking equipment.

The nudge consisted of hiding the letters on the colored circles until the gaze tracker determined the participant visually fixated upon the circle (10 frames or more), which caused the letter to be revealed, but only while the participant continued to fixate upon that circle. An example of a trial during the nudge can be seen in Figure 22, showcasing both when the participant is not fixating upon a circle, and when he/she is fixating upon a circle. An example of the block instructions preceding a block containing nudge trials can be seen in Figure 23, with the instructions differing slightly by noting to the participant that letters will not appear until they fixate upon a circle.

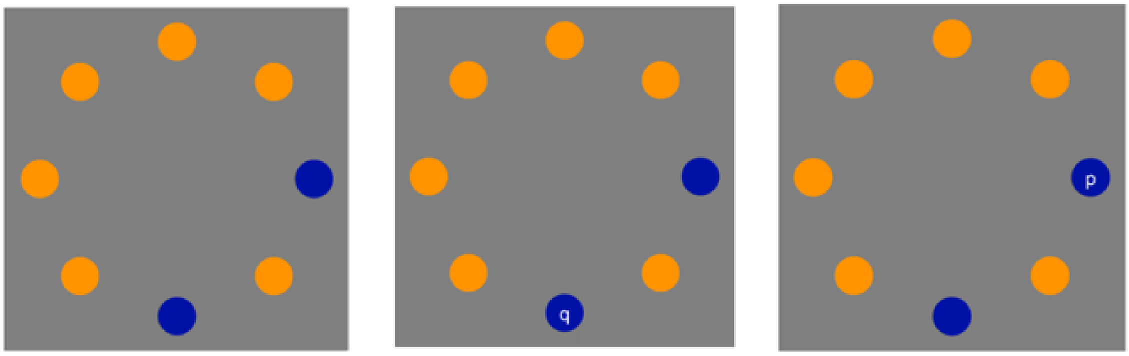
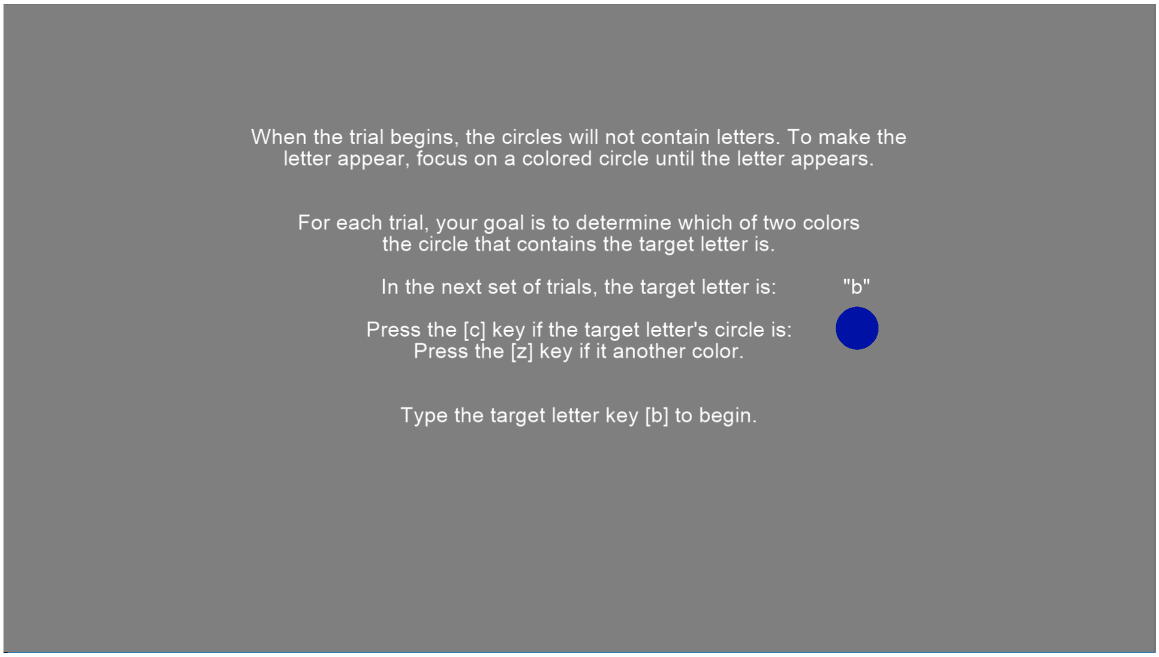


Figure 22. Example of a trial while the nudge is active. (Left) The participant is currently not fixating upon any of the circles. (Middle) The participant has fixated upon the bottom-middle circle, revealing the letter 'q'. (Right) The participant has fixated upon the right-middle circle, revealing the letter 'p'.



When the trial begins, the circles will not contain letters. To make the letter appear, focus on a colored circle until the letter appears.

For each trial, your goal is to determine which of two colors the circle that contains the target letter is.

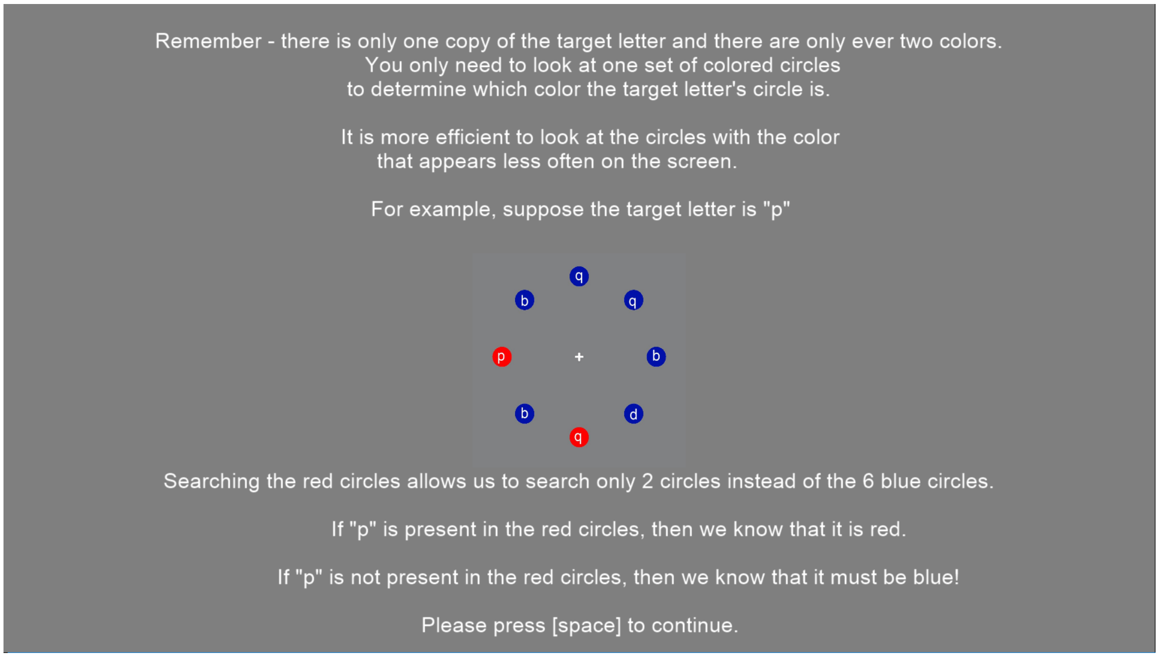
In the next set of trials, the target letter is: "b"

Press the [c] key if the target letter's circle is: ●
Press the [z] key if it another color.

Type the target letter key [b] to begin.

Figure 23. Block instructions preceding trials where the nudge will be active.

As mentioned in Section 4.4.2, other mitigation techniques such as the *hint*, *explanation*, and *instructions*, were presented to the participant during the ESE in later blocks. The *hint* consisted of showing the participant how to perform an efficient search and can be seen in Figure 24, the *explanation* involved telling the participant that the nudge was occurring when they were performing mostly inefficient searches in the prior block and can be seen in Figure 25, and the *instructions* screen explicitly instructed the participant to perform an efficient search and can be seen in Figures 26 and 27.



Remember - there is only one copy of the target letter and there are only ever two colors.
You only need to look at one set of colored circles
to determine which color the target letter's circle is.

It is more efficient to look at the circles with the color
that appears less often on the screen.

For example, suppose the target letter is "p"

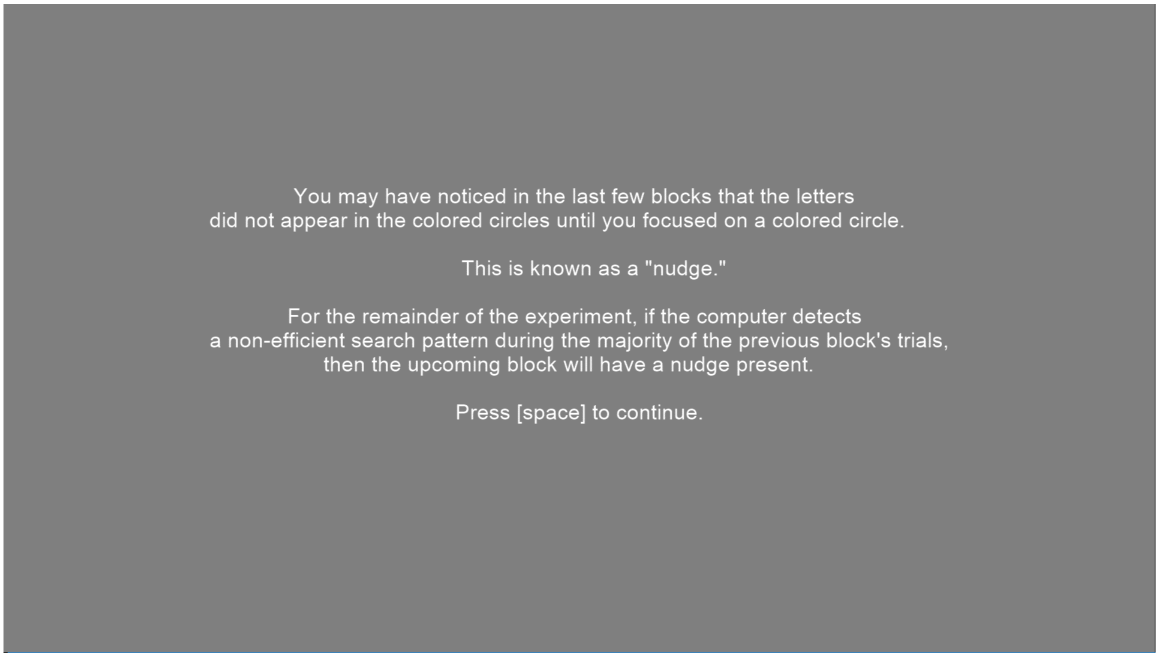
Searching the red circles allows us to search only 2 circles instead of the 6 blue circles.

If "p" is present in the red circles, then we know that it is red.

If "p" is not present in the red circles, then we know that it must be blue!

Please press [space] to continue.

Figure 24. Hint mitigation shown before block 11.



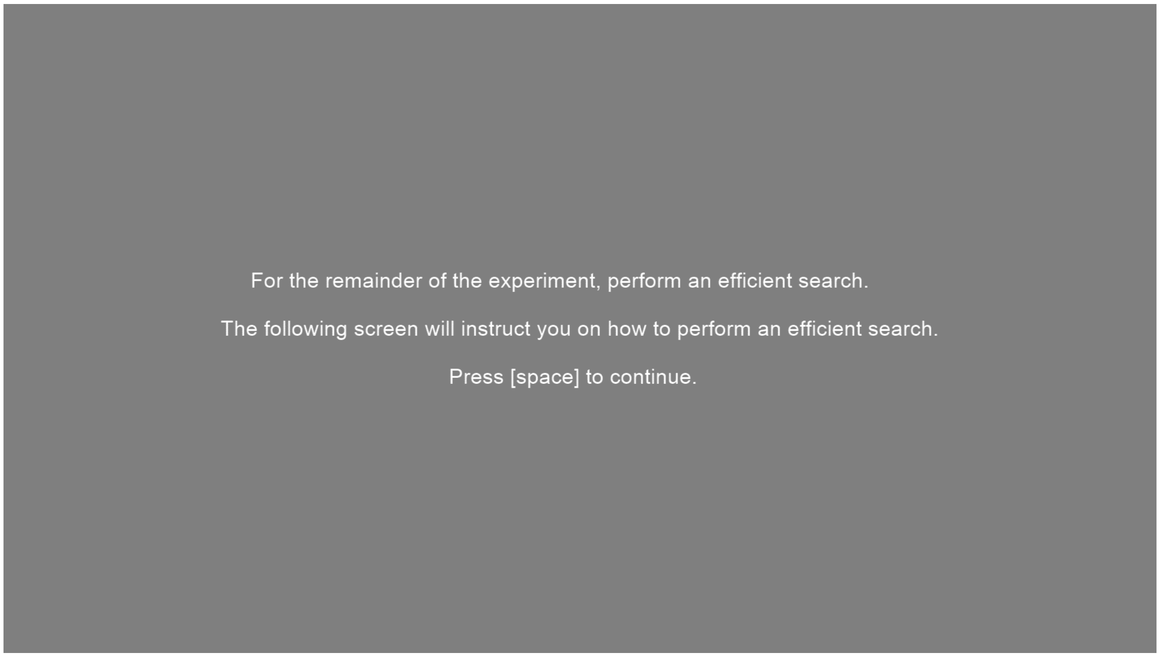
You may have noticed in the last few blocks that the letters
did not appear in the colored circles until you focused on a colored circle.

This is known as a "nudge."

For the remainder of the experiment, if the computer detects
a non-efficient search pattern during the majority of the previous block's trials,
then the upcoming block will have a nudge present.

Press [space] to continue.

Figure 25. Explanation mitigation shown before block 14.

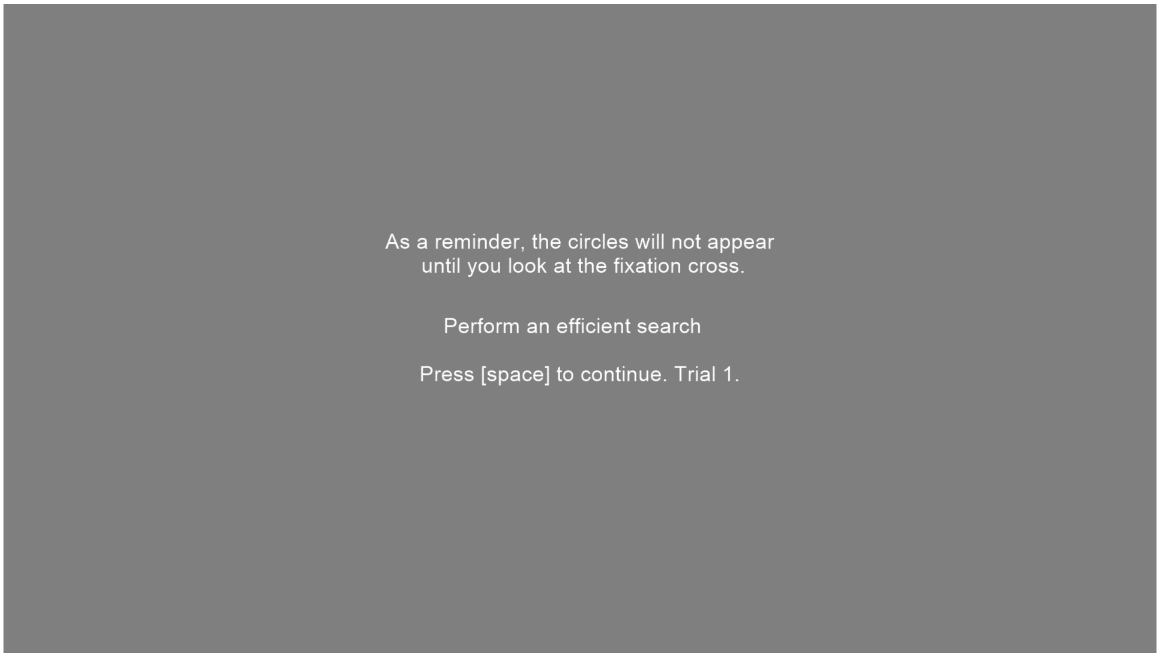


For the remainder of the experiment, perform an efficient search.

The following screen will instruct you on how to perform an efficient search.

Press [space] to continue.

Figure 26. Block instructions for the instructions mitigation.



As a reminder, the circles will not appear
until you look at the fixation cross.

Perform an efficient search

Press [space] to continue. Trial 1.

Figure 27. Trial instructions during the instructions mitigation.

While the nudge was applied dynamically based on whether the participant was searching efficiently in the prior block, the other mitigation techniques were presented before pre-determined blocks. The hint was presented before starting the 11th block, the explanation was presented before the 14th block, and the instructions were presented before block 18 and again before each of the remaining blocks. An example of the block design can be seen in Figure 28.

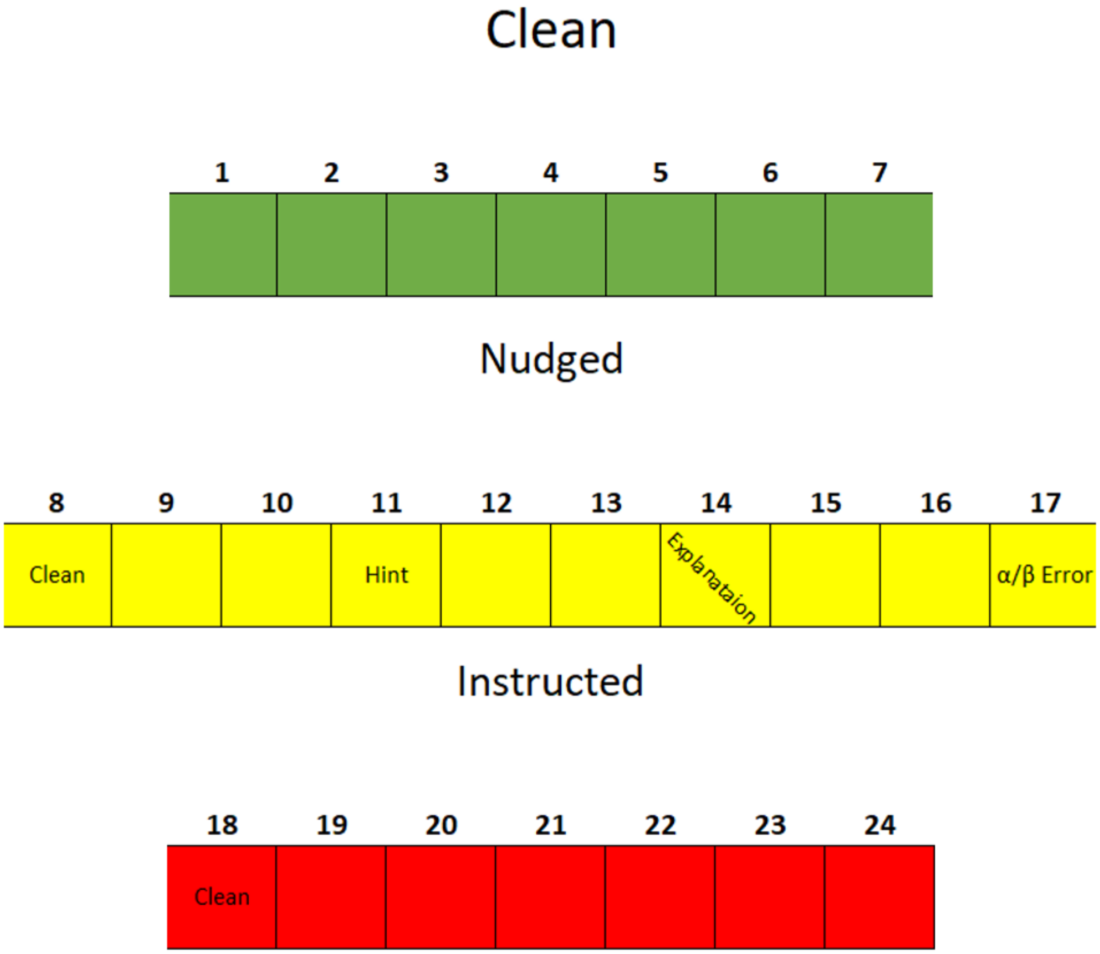


Figure 28. Block design for the ESE. Clean refers to no nudge mitigation being present for that block (blocks 1-8 and 18). Hint and Explanation refer to those mitigation techniques being introduced before blocks 11 and 14, respectively. α/β Error refers to including or removing the nudge for block 17 based on the opposite of the preceding block's results. Instructed refers to the instructed technique being introduced before every block and trial within blocks 18-24.

Upon conclusion of the experiment, another two minute baseline of EEG was recorded, and a post-experiment questionnaire was also given (see Section 4.7.4 for questionnaire details).

4.7.4 Questionnaires.

Below are the pre-experiment and post-experiment questionnaires given to participants.

ID: _______________ Date: _______________

Pre-Experiment Questionnaire (ONLY Experiment Day)

How many hours of sleep do you get on average? ____

How many hours of sleep did you have last night? ____

How would you characterize your sleep last night?
Circle one choice: Very Poor, Poor, Fair, Good, Very Good

Did you consume any products with caffeine today?
Circle one choice: yes or no
If yes:
What product(s) did you consume?
____
When did last consume this product?
____
Approximately how much (mg / ounces / cups) of this product have you consumed today? ____

Have you had seizures before? yes or no
Have you ever had brain surgery? yes or no
Do you have a history of brain tumors? yes or no
Do you have a history of head trauma? yes or no
Please list any other brain-related health issues that you may have (if any)?
____
____

Do you have any reason(s) to believe that your ability to accomplish tasks during this study today would be abnormal (for example: distracted, overly tired, hungry, stressed, injured)?
____
____

If yes:
Do you still want to participate in the cognitive study today? Circle one choice: Yes / No
If no:
Would you like to reschedule participation for another day?
____

ID: _______________ Date: _______________

Post-Experiment Questionnaire (ONLY Experiment Day)

Computer experience:

What sort of electronic devices do you use?
Circle all choices that apply:
Personal computer/Desktop/Laptop
TV/Game Console
Smartphone/Tablet
Enterprise Server
Other, ____

How often do you use electronic devices?
Circle one choice: Daily, A few times a week, Once a week, Never, Prefer not to answer
How often do you play video games?
Circle one choice: Daily, A few times a week, Once a week, Never, Prefer not to answer

Do you use electronic devices in your job?
Circle one choice: Yes, No, Prefer not to answer

Age: _______________

Are you male or female? Male___ Female____ Prefer not to answer ____

What's your highest education level?

A. Lower than high school
B. Graduated from high school
C. Some college, no degree
D. Associate's Degree
E. Bachelor's Degree
F. Master's degree
G. Ph.D. degree

Have you had pilot training or been trained in the scanning of instruments? yes or no

Psychological Knowledge:

On a scale of 1-5 (5: being you studied it extensively on your own, 4: you took a class which covered it, 3: read about it/looked it up, 2: heard the term used in discussion, 1: not familiar with the term), please rate the following:

How familiar are you with cognitive biases? __________

How familiar are you with confirmation bias and/or confirmatory search? __________

ID: _______________ Date: _______________

Performance:

Please explain how you searched for the target letter. If your search pattern changed or evolved throughout the experiment, please be sure to detail this change, when it occurred, and why you think that your search pattern changed.

____

____

____

____

____

____

On a scale of 1-5 (5: being you used it in 95% or more of trials, 4: 75% or more of trials, 3: 50% of more, 2: 25% or more, 1: less than 25%), please rate the following:

A confirmatory search pattern is one where you first search circles which match the color shown in the instructions.

- How often did you use confirmatory search overall in today's visual search experiment? ______
- How often did you use confirmatory search in the first 7 blocks of visual search trials (before the first break)? ___________
- How often did you use confirmatory search in the last 14 blocks of visual search trials (after the first break)? _____________

In the trials there were always two colors and one color had more circles than the other. An *efficient* search is one in which you only look at the lesser-represented-color circles to find the letter.

- How efficient do you believe you were overall in the first 7 blocks of visual search trials? _____
- How efficient do you believe you were overall in the last 14 blocks of visual search trials? _____

The covering of the letters so that they would not appear until gazed upon was a technique called a "nudge". This was used in order to encourage you to adopt an efficient search pattern, by adding a cost (i.e. time) to your search.

For each of the following questions – circle only one answer. Did the nudge…

Help reduce your tendency towards a biased visual search? Yes, No, Unsure

Annoy or irritate you when it was included in searching? Yes, No, Unsure

After the nudge was first introduced, if it was ever removed for a block of trials, do you believe your search for that block was efficient? Yes, No, Unsure

4.7.5 Signals Collected.

The Smart Eye Pro eye tracking system [107] was used to capture eye tracking data at 60 Hz with a calibrated accuracy of <3° for each eye. This provided gaze location data within this degree of error for the upper monitor of the experimental setup which can be seen in Figure 29. Only the upper monitor was used for the experiment. For this multi-camera system, six cameras with infrared (IR) filters were used in combination with four IR flashers.

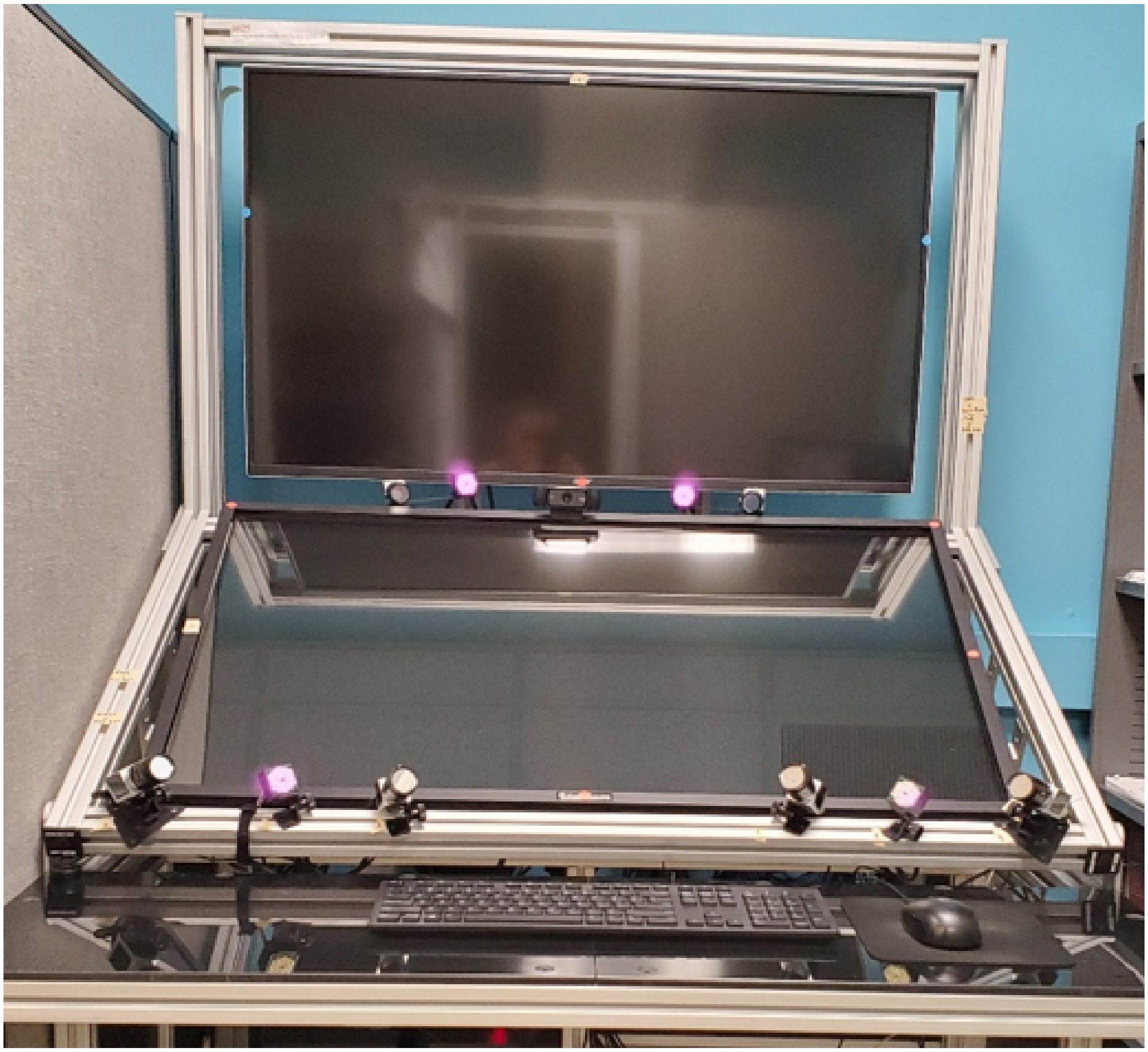


Figure 29. Equipment setup used for the experiment (bottom monitor was not used during the experiment).

EEG, EOG, ECG, and GSR, were collected using the Cognionics Mobile-72 system. The EEG was collected using 64 scalp electrodes plus reference, ground, and x-y-z motion, resulting in referenced 64 channel EEG at a sampling rate of 512 Hz. All scalp electrodes were gelled with an impedance of $<$100k ohms impedance. EOG, ECG, and GSR were collected through the Cognionics auxillary input module (AIM). VEOG and HEOG were both collected using electrodes attached underneathe the eyes and on the bone on the sides of the eyes (near the external canthi). ECG was collected through a 3-wire lead set using the Cognionics respiration electrode paddles, with one paddle attached beneath the middle of the right clavicle, another attached over middle of the left rib cage, and a ground electrode attached beneath the middle of the left clavicle. GSR was collected using electrodes attached to the right hand on the left and right sides of the palm (the right hand was not used during the experiment and participants were instructed to rest it and hold it still during the experiment).

4.7.6 Preprocessing Pipeline.

Below is a summary of the steps taken and rationale for initial preprocessing and cleaning of EEG data. These steps were performed through script batch processing using EEGLAB [108], and consisted of a combination of best practice steps from both Makoto's preprocessing pipeline [104] and the PREP pipeline [109].

1. Modify EEGLAB to use double precision, as single precision can destroy natural commutativity of the linear operations.

2. Import data into EEGLAB and include reference channels based on the equipment used (e.g. Biosemi's 64 scalp electrode cap uses channels 65 and 66 as reference channels, which are electrodes placed on the mastoids specifically for the purpose of referencing).

3. Down-sample to 250 Hz for purpose of improving ICA decomposition by cutting off unnecessary high-frequency information, and also to reduce data size.

4. High-pass filter the data at 1 Hz to reduce baseline drift, improve line-noise removal, and to improve ICA [110]. High-pass filter is done before line-noise removal and 1 Hz is used as we are not performing event-related potential (ERP) analysis, which could be affected by using a 1 Hz high-pass filter, and would require an alternate strategy.

5. Import channel info using International 10-20 system to allow for re-referencing.

6. Remove line noise using CleanLine plugin (default 60Hz notch filter) [111].

7. Remove bad channels using EEGLAB clean_rawdata plugin patented by Christian Kothe [112], which utilizes Artifact Subspace Reconstruction.

8. Interpolate all removed channels to minimize a potential bias in the average referencing step.

9. Re-reference data to the average. Mastoid referencing isn't always sufficient [109], and re-referencing the data to the average helps suppress line noise that wasn't rejected by CleanLine [104].

10. Independent Component Analysis (ICA) 'runica' "'infomax': (extended)" algorithm variant is run with the vertical EOG (VEOG) electrode used as input for the function.

11. ICA results from step 10 are used to remove artifact ICA components.

After initial cleaning and preprocessing of the data, trials are then epoched according to the desired epoching window. Epoching is done as EEG is recorded continuously and therefore, a trial must be sliced into an extractable segment which is then representative of that trial, and serves as the observation or sample for machine learning. This epoching is specific to the data and experiment that collected it, and thus, each study specifies these windows within their respective methodologies. In general, this window is desired to only include the segment of EEG that contains the phenomena that we want the machine learning model to learn. This differs from ERP analysis which typically includes a pre-stimulus baseline. In the case of machine learning, instead of helping the model, it could be that the pre-stimulus baseline would instead confuse it, as all trials would contain a portion of EEG that is relatively similar across all conditions, making it more difficult for the model to learn the features that differentiate the desired conditions.

For sequence based models, the raw time-series voltage values are saved into .csv files for each participant. For time-frequency domain models, feature extraction of the mean power of the five traditional brain frequency bands is performed using complex Morlet wavelets. To do this for each epoch, first the voltage values for each channel are convolved with complex Morlet wavelets. Wavelet parameters used are 30 frequencies spread in log-space (20-30 is recommended as a sufficient amount [18]) with a min frequency of 2 Hz and a max frequency of 80 Hz, and a time range -2s to +2s. Mean power is then extracted for each of the 30 frequencies by squaring the mean of the corresponding convolution, and then taking the mean of the frequencies which correspond to their respective frequency band (traditional bands being used, i.e. delta (2-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (15-30 Hz), Gamma (30-80 Hz)). These mean power values of each channel for each frequency band are then saved into .csv files for each participant, resulting in 320 features per trial.

4.7.7 Datasets.

Two different types of data were extracted from the EEG in order to evaluate two different types of machine learning classifiers: models which accept raw time series EEG voltage data as input, and models which accept EEG spectral features as input. Data for these two types of models was pre-processed, epoched, and extracted as outlined in Section 4.7.6, with the epoching window being two seconds prior to the participant selecting which color the target letter's circle was. This resulted in the raw time series dataset having trials with 64 features and sequence lengths of 500, and the spectral dataset having 320 features (mean power of the five traditional bands for 64 channels).

From these two types of data, four different datasets were created using various combinations of the experiment's trials. This was done because the nudge altered fundamental aspects of the search such as gaze fixation duration and saccade intervals, and thus, neural correlates that are associated with a VSP while a nudge is present may be different than neural correlates of a VSP when there isn't a nudge present. In order to isolate and investigate the effects of the nudge on model building, the following four datasets were created: Two datasets consisted of only trials which did not have the nudge (*Clean-Balanced* dataset & *Clean-Unbalanced* dataset), another dataset consisted of only trials which contained the nudge (*Nudge* dataset), and a final dataset consisted of all trials from the experiment (*Combined*). The *Clean* and *Nudge* datasets were created to determine if the nudge was having an effect on the models' performance. The *Clean-Balanced* dataset was created as the class distribution was widely imbalanced for all participants within the *Clean* dataset, with only between 7% to 24% of the trials being efficient trials (depending on the participant). This dataset was created by randomly selecting a subset of the majority class (inefficient) in order to reduce its size to that of the minority class (efficient). Both the *Clean-Balanced* and *Clean-Unbalanced* datasets were explored to investigate the effect of class imbalance on model performance. The *Combined* dataset was explored to determine if the model performed best when it has all data available for training, regardless of the effect of the nudge. All together, this resulted in eight datasets total, with each of the four listed above having both a separate raw time series EEG voltage dataset, and a spectral features dataset containing the mean power for each of the five traditional frequency bands.

4.7.8 Model Creation.

LDA and RFC models are selected as more traditional/baseline machine learning models for comparison of the neural network models. While LDA does not have hyperparameters to tune, RFCs have hyperparameters with significant effects. RFCs are an ensemble classifier utilizing a set of decision trees to collectively classify observations. RFC hyperparameters selected for tuning include the maximum depth of the trees, and the number of features to consider. The number of estimators (trees) is determined by incrementally increasing the number of estimators by 5 from a low value of 50 until validation accuracy no longer improves. For this, maximum depth is set to its default scikit-learn value of 'None' so that there is no limit to depth, and the maximum features set to its typical recommended amount of $m = \sqrt{p}$, where p equals the 320 features, and thus $m = \sqrt{320} = 18$ [113]. Once the number of estimators is determined, a hyperparameter sweep for the maximum depth of the trees and the number of features is performed, utilizing values 1-25 for each hyperparameter.

Deep, fully-connected Multi-Layer Perceptron Neural Networks (MLPNNs) are utilized for the spectral feature ANNs. For these models, hyperparameters tuned through sweeps included the number of hidden units (HUs), the number of hidden layers (HLs), the learning rate (LR), and the dropout rate (DR). Hyperparameter ranges and other hyperparameters that are selected based on literature include the following:

- # of HLs is either 2, 3, or 4.
- # of HUs for each layer is 128, 64, 32, 16, 8, or 4; with later layers decreasing in the number of HUs.
- LR is either 0.001 or 0.0001.
- DR is either 0.0, 0.2, or 0.4.

- Rectified Linear Units (ReLU) are selected as the activation function as they are commonly used in the literature [12].

- Batch normalization is used after each hidden layer as it can decrease time to convergence and lead to improved generalization [69].

- Batch size is set to 32 for a balance between performance and reduced training time [114].

- The ‘Adam’ optimizer [115] is selected as it is a great baseline optimizer, combines the advantages of both ‘RMSProp’ and momentum, and is overall fairly robust to the choice of hyperparameters [58].

- Data values are normalized by participant for within-participant models, and across participants for cross-participant models.

- As this is binary classification, the sigmoid activation function is used for the final activation function.

- ‘Early stopping’ and ‘ReduceLRonPlateau’ are both utilized with ‘val_loss’ monitored.

LSTMs and GRUs are utilized for the time-series based models. For these models, hyperparameters tuned through sweeps included the number of hidden units (HUs), the learning rate (LR), and the dropout rate (DR). Hyperparameter ranges and other hyperparameters that are selected based on literature include the following:

- # of HUs is 55 for first layer and 25, 40, 55 for 2nd layer.

- LR is either 0.001 or 0.0001.

- DR is either 0.0, 0.2, or 0.4.

- For the number of recurrent unit layers to use within the LSTM and GRU architectures, literature reviews of over 90 EEG classification studies concluded that using 2 recurrent unit layers was best, as 2 layers leads to significantly better accuracy than using a single layer, and that additional layers (3 or more) had little effect other than increasing model parameter size, and thus increasing training time [12].

- Batch size is set to 32 for a balance between performance and reduced training time [114].

- Literature reviews recommended one or two fully-connected layers for classification, as there wasn't a significant difference between them, and thus, one layer was selected [12].

- For an optimizer, 'RMSprop' was utilized as Chollet recommends it as being optimized for use in RNNs [116].

- As this is binary classification, the sigmoid activation function is used for the final activation function.

- 'Early stopping' is utilized with 'val_loss' monitored.

Scikit-learn's balanced test classification accuracy with confidence intervals are utilized to compare classifier performance across each of the five types of classifiers and four different dataset types. Balanced accuracy (the average of recall obtained on each class) is necessary due to the class imbalance across each of the four datasets. This results in comparing twenty within-participant models and twenty cross-participant models; with the four dataset types being *Combined*, *Clean-Balanced*, *Clean-Unbalanced*, and *Nudge*, and the five classifiers being LDA, RFC, ANN, LSTM, and GRU.

4.7.9 Mitigation Analysis.

The ESE utilizes the gaze tracking data and algorithms detailed within Section 4.7.3 (Apparatus) in order to determine the VSP of each trial in real-time, to include whether the search was efficient or inefficient, and what type of search it was (confirmatory, circular, etc.). For this analysis, we focus on the different mitigation techniques and their effect on searches being efficient. Some of the techniques are knowledge based, meaning once the participant has been exposed to the technique, any trials following the technique are included as the participant having had the mitigation technique. This results in trials being binned as to whether or not the trial does or does not have the presence of the nudge (X_1), and whether or not the participant has received the hint (X_2), explanation (X_3), or instructions (X_4). From here, a linear regression model is generated according to Equation 4.1 and the effects of each mitigation technique as well as the interaction effects of the nudge*hint, nudge*explanation, and nudge*instructions are determined.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_1 X_2 + \beta_6 X_1 X_3 + \beta_7 X_1 X_4 \tag{4.1}$$

LogWorth is used to compare effect sizes for the different mitigation techniques. LogWorth is defined as $-log_{10}$(P-Value) and is useful in comparing multiple small and significant p-values. In this experiment $\alpha = 0.05$ and thus a LogWorth >1.3 (i.e. $-log_{10}(0.05) = 1.301$) is considered significant.

V. Detection of the Vigilance Decrement in any Sustained Attention Task using EEG-based Deep Learning

5.1 Paper Overview

This paper was submitted to the MDPI Sensors journal special issue EEG Signal Processing for Biomedical Applications on 31 July, 2021.

5.2 Abstract

Tasks which require sustained attention over a lengthy period of time have been a focal point of cognitive fatigue research for decades, with these tasks including air traffic control, watchkeeping, baggage inspection, and many others. Recent research into physiological markers of mental fatigue indicate that markers exist which extend across all individuals and all types of vigilance tasks. This suggests that it would be possible to build an EEG model which detects these markers and the subsequent vigilance decrement in any task (i.e. a task-generic model), and in any person (i.e. a cross-participant model). However, thus far, no task-generic EEG cross-participant model has been built or tested. In this research, we explored creation and application of a task-generic EEG cross-participant model for detection of the vigilance decrement in an unseen task and unseen individuals. We utilized three different models to investigate this capability: a multilayer perceptron neural network (MLPNN) which employed spectral features extracted from the five traditional EEG frequency bands, a temporal convolutional network (TCN), and a TCN autoencoder (TCN-AE), with these two TCN models being time-domain based, i.e., using raw EEG time-series voltage values. The MLPNN and TCN models both achieved accuracy greater than random chance (50%), with the MLPNN performing best with a 7-fold CV balanced accuracy of 64% (95% CI: 0.59, 0.69), and validation accuracies greater than random chance for 9

of the 14 participants. This finding demonstrates that it is possible to classify a vigilance decrement using EEG, even with EEG from an unseen individual and unseen task.

5.3 Introduction

Mental fatigue is a significant contributor to a decline in performance for sustained attention type tasks [117, 118], a.k.a., vigilance tasks. Vigilance tasks require operators to remain focused and alert to stimulus during a task [5], and in the control and surveillance of today's automated systems, vigilance typically suffers either due to the low level of workload and stimulus associated with the task [6], or due to the mental demands vigilance requires over a lengthy task [7].

A decline in performance during these vigilance tasks is called a vigilance decrement, and it is defined as a decrease in probability of detecting rare but significant events within vigilance tasks [119]. Some form of mental fatigue is typically associated with a vigilance decrement, and this mental fatigue has been linked to increased human error rate [120–122]. If this mental fatigue could be detected using artificial intelligence (AI), then systems could be developed to regulate mental fatigue by varying levels of stimulus to aid in sustained attention [8, 123], or by providing recovery time [9].

Mental fatigue has also been linked to specific changes in physiological measures, such as specific increases and decreases in magnitude for the average spectral power of different frequency bands for electroencephalography (EEG) signals [124, 125]. Recent machine learning research has utilized EEG signals to classify mental fatigue in specific tasks such as driving [126], however, a task-generic model which can accurately classify either mental fatigue or a vigilance decrement through EEG signals has not yet been generated. The EEG markers of mental fatigue during vigilance tasks are consistent across both participants and different types of tasks, and mental fatigue is typically always associated with a vigilance decrement in vigilance tasks [16], thus, a model could be built which is capable of performing classification of a vigilance decrement in any vigilance task, through detection of mental fatigue in EEG signals, in any individual's EEG (i.e. a cross-participant model).

Recently, Yin et al. pursued the goal of building a task-generic cross-participant mental fatigue detector using extreme learning machines (ELMs) [127]. Two tasks were used which had the participants replicate the role of an aircraft's automated cabin air management system. Eight participants performed task 1, and six different participants performed task 2. Each task varied parameters within the task to create "low" and "high" mental fatigue conditions, with these conditions then corresponding to labelled trials of their respective condition. Models were then built from the EEG data for each task and each condition using entropy features, and spectral features (average power of the theta, alpha, beta, and gamma bands), as input features. The models were then tested upon the participant data of the opposite task, with classification accuracies ranging from 65-69%. An issue with relating this to vigilance decrement detection is that the tasks simply varied parameters within the task to create "low" and "high" mental fatigue conditions. These conditions were then used as the labels to train the classifier. This means the classifier was trained to identify EEG signals which correspond to these "low" and "high" mental fatigue conditions, and not actual vigilance decrements. For proper identification of a vigilance decrement, instead an objective measure of the participant's performance which is associated with the vigilance decrement (such as accuracy and/or response time) would need to be recorded and used to generate the labels of vigilance decrement vs no decrement for the machine learning classifier. Another issue is that it is unclear if the two stated tasks are analogous to two separate tasks in the real-world, such as the difference between driving and monitoring closed-circuit security cameras, as both tasks used in the experiment had participants performing the same role of the aircraft automated cabin air management system, with only certain parameters and conditions being varied between the two tasks. This suggests that their results are applicable to a varied version of the same type of task, but is not truly task-generic.

In this research, we build three different cross-participant models which use EEG signals to perform task-generic classification of the vigilance decrement on any individual. Two of the models are time-domain based, meaning they use the raw EEG time-series voltage values as their data, and the third model is frequency-domain based, using spectral features extracted from the average power of the five clinical EEG frequency bands. The data is comprised of two EEG datasets, with each dataset containing different participants, and each dataset containing different vigilance tasks (three different tasks in total). These datasets were collected by the 711th Human Performance Wing (HPW) in partnership with the University of Dayton through two different experiments for the purpose of studying event related potentials (ERPs) during a vigilance decrement across various vigilance tasks [15, 128]. Models are trained on data from two of the vigilance tasks and only a subset of the participants, and then tested using data from a separate vigilance task that the model has not seen, as well as participants that the model has not seen, which is crucial in order to avoid overestimated test accuracies in cross-participant EEG models [71].

This paper has the following structure. First, in Section 5.4, background is provided for the vigilance decrement and how it is linked to EEG. Next, in Section 5.5, we provide our methodology, first providing details on the datasets collected and the tasks used within those datasets, followed by details for the training and testing of all three models. Then, in Section 5.6, results are presented for all three models. Finally, in Section 5.7, results are compared and discussed, with conclusions and future work following in Section 5.8.

5.4 Related Work

Decision making and how it deteriorates in stressful work environments has been extensively studied since the late 1800's [119]. One of the main phenomena studied has been the concept of vigilance, which is the quality or state of being wakeful or alert [129]. Tasks which require vigilance fall under a taxonomy developed by Parasuraman and Davies [37], with the taxonomy classifying tasks into different categories based on specific information-processing transactions within the tasks themselves, such as signal discrimination (successive or simultaneous), task complexity, event rate, and sensory modality. For signal discrimination, simultaneous tasks are ones in which the critical stimulus and non-critical stimulus are both present at the same time for participants to use for comparison. Successive tasks, however, do not provide these stimulus to the participant at the same time, and therefore, it requires the participant to hold the non-critical stimulus in memory.

5.4.1 Vigilance Decrement.

Extensive research over the decades on vigilance and the vigilance decrement has found that the behavioral cause of the decrement is due to performing attention-demanding tasks over an extended period of time, ranging from tens of minutes to hours, depending on the task and its cognitive demand [130]. Performing these attention-demanding tasks for extended periods of time results in mental fatigue, and/or a decrease in sustained attention [131], with mental fatigue being defined as a gradual and cumulative phenomenon that is increased in magnitude by time spent on a tedious but mentally demanding task [132].

From a cognitive approach, the cause of a vigilance decrement has historically been studied using two contrasting classes of explanations, namely *overload* and *underload* [133]. A highly cited overload theory is the resource-depletion theory, which states that there are limited amounts of information processing resources that we cognitively possess, and that the level of effort/time-on-task required in vigilance tasks depletes these finite resources faster than they can be replenished, eventually leading to reduced detections of significant events, as these resources are needed for vigilance tasks [134]. Underload, in contrast, suggests that it is instead the lack of stimulation in vigilance tasks that results in the vigilance decrement, as the low levels of stimuli result in attention shifting away from the task, which then leads to distraction from off-topic thoughts, or task automaticity taking over, which results in reduced detections of significant events [135, 136]. Previous research has shown that the attention shift is primarily attentional lapses which come from perceptual decoupling from the trivial and unengaging nature of the task [137, 138]. However, this doesn't explain *where* attention is being shifted. A highly cited theory that attempts to explain where attention is shifted is the mind-wandering theory, which states that the attention is shifted to self-generated off-topic thoughts, leading to the vigilance decrement [138].

Both overload and underload approaches have highly cited research backing up their findings, and each approach claims to explain the cause of a vigilance decrement. As neither explains the full range of findings on their own, a recent approach which seeks to unify the two is the resource-control theory, which is a framework that combines both theories to explain the lapses in attention as a function of time-on-task [136]. The resource-control theory is consistent with overload theory in that it states that necessary cognitive resources are depleted by focusing on a task (executive control). However, it also states that executive control has to use cognitive resources to prevent mind-wandering. Thus, as cognitive resources are depleted, mind-wandering will occur, which results in attention shifting away from the task to thoughts not relevant to performing the task at hand. In this way, resource-control theory is also consistent with underload, as resource-control theory acknowledges mind-wandering and provides explanation for it occurring. Veksler and Gunzelmann developed a recent computational model to account for the vigilance decrement and had similar findings to the resource-control theory [139], albeit with *microlapses* as opposed to mind-wandering. In their model, microlapses are defined as brief gaps in cognitive processing that disrupt performance, resulting in a vigilance decrement in a similar manner to mind-wandering.

Numerous factors have also been found to affect the magnitude and timing of the vigilance decrement [140]. For magnitude, simultaneous stimulus, shorter signals [141, 142], task type/source complexity [143], and stimulus event rate [144, 145], all result in a greater vigilance decrement. For timing, the vigilance decrement varies depending on the task demands, with the vigilance decrement occurring earlier in more difficult tasks [130], and typically occurring within the first 20-35 minutes of a task, with half of the decrement occurring in the first 15 minutes [146].

5.4.1.1 Performance Measurement.

To identify in data whether a vigilance decrement has occurred, some measure of task performance through either accuracy (calculated as $\frac{\text{hits + correct rejections}}{\text{hits + false alarms + misses + correct rejections}}$), response time (RT), or both, is needed. Accuracy and RT are frequently correlated, such that slower responses are more accurate and vice versa, and this is referred to as the speed-accuracy trade-off [147, 148]. Due to this correlation, it is best to use both accuracy and RT to assess performance, and many different measures have been developed to combine both speed and accuracy into a single measure of performance. For example, there is the Inverse Efficiency Score (IES), which is the ratio of the mean RT and the proportion of correct responses (PC) [149], the Rate-Correct Score (RCS) which is the inverse of the correct RT-based IES [150], the Balanced Integration Score (BIS) which is a combined z-score of RT and accuracy [151], and many others. Recently, research by Mueller et al. examined 12 different measures of accuracy and RT on a vigilance task to determine their sensitivity to the vigilance decrement, and found that most single measures which combined accuracy and RT were slight improvements over just accuracy or RT alone [152]. While they found that the Linear Ballistic Accumulator model was the most sensitive and representative measure of the vigilance decrement, they also noted that it was difficult and cumbersome to use, and recommended the BIS measure overall.

The BIS measure is designed to give equal weights to both PCs and RTs, hence the name Balanced Integration Score, and is shown below in Equation 5.1. First, the PCs and RTs are standardized as shown in Equations 5.2 and 5.3, with participants j and standard deviations s, and then once standardized, the standardized RT is subtracted from the standardized PC. This gives the difference in standardized mean correct RTs and PCs. z_{pc} and z_{rt} can be calculated individually for each participant j, giving the BIS measurement for only that participant, or across all participants, giving the BIS measurement for the group.

$$BIS_j = z_{PC_j} - z_{RT_j}. \tag{5.1}$$

$$z_{pc_j} = \frac{PC_j - \overline{PC}}{s_{pc_j}}. \tag{5.2}$$

$$z_{rt_j} = \frac{RT_j - \overline{RT}}{s_{rt_j}}. \tag{5.3}$$

When calculating measures such as BIS from data collected during a vigilance task, trials must be binned in some manner for the standardized measures of z_{pc} and z_{rt} to be calculated. A common method is to divide the trials over the duration of the experiment into four time segments (bins) [128, 152, 153]. Once the number of bins is selected, BIS can then be calculated and compared for each bin to determine whether a vigilance decrement has occurred for the participant; a decreasing BIS indicates a decrement in vigilance. A typical method is to plot the bins on a graph to view the participant's performance over the course of the task, as well as to plot a line of best fit (least squares) to see how their performance trended over the course of the task, with a negative slope indicating a vigilance decrement over the course of the entire task.

5.4.2 EEG.

Physiological measurements such as EEG, electrocardiography (ECG), and electrooculography (EOG), have been progressively utilized to better understand the underlying mechanisms of mental fatigue and the vigilance decrement over the past two decades, with EEG receiving significant attention in research for its insight into the status of the brain [16]. EEG signals are a measure of the electrical activity in the brain using electrodes distributed over the scalp, and EEG is often referred to by its different clinical frequency bands, namely delta (2-4 Hz), theta (4-7 Hz), alpha (8-12 Hz), beta (13-29 Hz), and gamma (33-80 Hz). A physiological measurement such as EEG has the advantage of providing a more objective measurement of fatigue than a behavioral measure, as behavioral measures are subjective in nature and left to the experimenter's or participant's judgement. EEG studies investigating neural correlates of fatigue have found differing results based on the type of fatigue that the participant is experiencing, with the primary difference being fatigue from sleepiness (sleep fatigue) versus accumulating fatigue from cognitive processes and mental workload (mental fatigue). For example, neural correlates of sleep fatigue have been found to differ based on the task that is being performed. Driver fatigue research found that symptoms associated with sleepiness (e.g. prolonged eye closure) correlated to increases in spectral power for the alpha and beta bands [124], while in pilot fatigue studies, sleepiness was more associated with the opposite effect, with decreases in spectral power for the alpha band [154, 155]. Mental fatigue, however, has shown consistent neural correlates of increased spectral power for the alpha band across tasks [16]. This allows for the detection of mental fatigue across tasks and across participants. However, given that both types of fatigue can contribute to changes in performance, such as the vigilance decrement, yet have differing neural correlates, it is important to distinguish sleep fatigue from mental fatigue to reduce confounding variables.

Utilizing these neural correlates of EEG has been useful in detection of the vigilance decrement in previous research, with EEG spectral features being common features used to detect drowsiness, mental fatigue, and alertness [156–158]. Power spectral density (PSD) in combination with independent component analysis [158], the mean power of the frequency bands and their ratios [156, 159, 160], power spectral indices of wavelet transforms [161], and full spectrum log power are all spectral features that have been used [157, 162].

5.5 Methods

5.5.1 Datasets.

In this study, two EEG datasets are utilized, each collected through experiments by the United States Air Force Research Laboratory, 711th Human Performance Wing (HPW), in partnership with the University of Dayton. These experiments were each conducted for the purpose of studying ERPs during a vigilance decrement within various vigilance tasks [15, 128], however, the experiments were conducted separately and did not coincide. In one experiment, 32 participants completed three different tasks across a two hour session in the following order: the Hitchcock Air Traffic Controller (ATC) Task [163], the Psychomotor Vigilance Test (PVT) [164], and the 3-Stimulus Oddball Task [165]. The PVT was omitted from our research as the task length was short in duration ($<$10 min) along with a few amount of trials (< 100), making it difficult to segment into bins and quantify with the BIS measure. The Hitchcock ATC task and 3-Stimulus Oddball Task were performed as described in Sections 5.5.2.1 and 5.5.2.2, and trials for each task occurred as follows. The ATC Task included 200 practice trials with feedback provided every 50 trials, then a short break followed by 1600 trials without feedback or breaks. The 3-Stimulus Oddball Task included 20 practice trials, a short break, and 4 blocks of 90 trials each, with performance feedback after each block. Practice trials across both tasks are not utilized in analysis or model training. Some participants had incomplete data, and only the data from the 14 participants with complete data sets were analyzed.

The second experiment consisted of two sessions for each participant, conducted over two separate days, and utilized the line task described in Section 5.5.2.3. Each day, participants performed 200 practice trials and 4 blocks of 400 experimental trials each, with a short few minute break offered between each block. There were 29 participants, however only 26 of the participants returned the second day. The data from all 29 participants was utilized in the current study.

For both datasets, the tasks were presented on an LCD 60 Hz monitor using Psychophysics Toolbox [166] within MATLAB. EEG was recorded using a BioSemi Active II 64+2 electrode cap (10-20 system) with the 2 reference electrodes placed over the mastoids, with a sampling rate of 512 Hz. Vertical EOG (VEOG) and Horizontal EOG (HEOG) were also recorded [15, 128]. Baseline resting EEG was recorded before starting the experiment and checked for artifacts. Voltage offsets were reduced to less than 40mV to ensure low impedance, and any high impedance electrodes were re-gelled and re-applied.

5.5.2 Vigilance Decrement Tasks.

5.5.2.1 Hitchcock Air Traffic Controller Task.

The Hitchcock ATC Task was designed to test theories surrounding sustained attention, workload, and performance, within a standardized controllable task that is relatively more representative of the real world [167]. Stimulus of a filled red circle and three concentric white circles are continually displayed to the participant. Two white line segments are then displayed over these stimuli in different configurations, as seen in Figure 30. The red circle represents a city, and the white line segments represent aircraft. Participants are instructed to respond (through press of a key on a keyboard) only if the two jets are on a collision course with one another, i.e. the white lines are colinear. If they are, this is a critical event, and a small minority of trials are critical events (3.3%), the rest being non-critical as seen in Figure 30. The stimulus appear every 2 seconds and only remain on screen for 300ms.

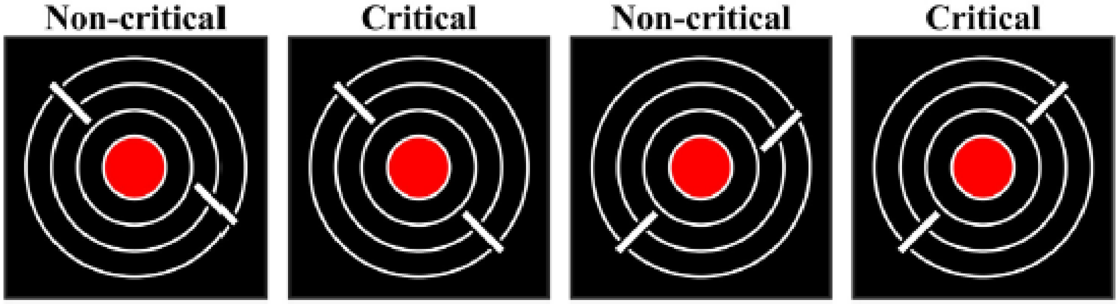


Figure 30. Examples of the different Air Traffic Controller Task stimuli.

5.5.2.2 3-Stimulus Oddball Task.

The 3-Stimulus Oddball Task was designed to assess how individuals discriminate targets, non-target distractors, and standard distractors, in various challenging scenarios [165]. In this task, three different visual stimuli can appear; targets, non-target distractors, and standard distractors. Targets and non-target distractors each appear separately in 10% of trials, and standard distractors appear in the remaining 80% of trials. As seen in Figure 31, the target is a large circle, the standard distractor a small circle, and the non-target distractor a large square. Stimuli are every 2 seconds with a 75ms duration. Participants are instructed to respond only to targets by pressing a response key on a keyboard, ignoring non-target distractors and standard distractors.

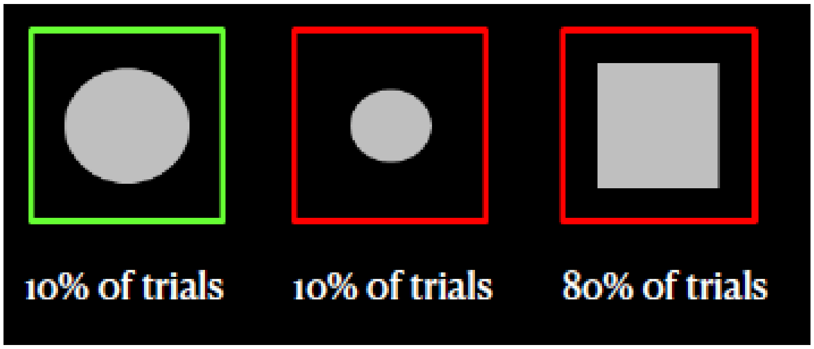


Figure 31. Shapes for the 3-Stimulus Task. The target is a large circle (left), the standard distractor a small circle (middle), and the non-target distractor a large square (right) [15].

5.5.2.3 Line Task.

In the Line Task, participants observe a series of pairs of parallel lines, and select whether or not each stimulus is critical. The critical stimuli vary among four conditions for the task, and with critical stimuli comprising 10% of the stimuli. The parallel lines are 0.75mm in width and variable in length based on trial condition [144]. The first and second conditions are successive-discrimination tasks, meaning the participant has to hold the critical stimulus in memory. In the first condition, the set of lines both being 1.46cm (short) is the critical stimulus, with both lines being 1.8cm (long) as the non-critical stimulus. In the second condition, these are reversed. The third and fourth conditions are simultaneous-discrimination tasks, meaning the participant is provided both the critical and non-critical stimulus at the same time for comparison. In the third condition, the critical condition occurs when the lines are different in length while in the fourth condition, these are reversed. Critical stimuli are sequenced such that there are at least four non-critical stimuli in between each pair of critical stimuli. Each participant completed both simultaneous and successive discrimination conditions (counterbalanced across sessions). Stimulus appeared on screen for 150ms and total trial duration was randomized to be between 1.3s and 1.7s. Figure 32 shows an example of the line stimulus.

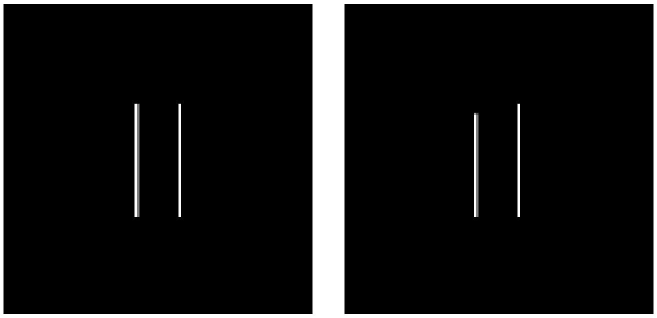


Figure 32. Examples of the different Line Task stimuli, with lines being the same length on the left, and different lengths on the right.

5.5.3 Preprocessing and Epoching of EEG signals.

Preprocessing of EEG data was performed through script batch processing using EEGLAB [108], and consisted of a combination of best practice steps from both Makoto's preprocessing pipeline [104] and the PREP pipeline [109]. Details for these steps can be found in Section 5.9, but worth noting is that the data is downsampled to 250 Hz. All tasks were relatively similar in trial duration, ranging from 1s to 1.7s, with inter-trial duration ranging from 1.2s to 2s. To avoid an epoching window which extends into the following trial for some tasks but not others, a 1s epoching window was selected based on both trial duration and inter-trial duration. Additionally, analysis performed by the 711 HPW demonstrated that a 1s window following stimulus-onset contained the majority of EEG activity for each task [15, 128]. This resulted in a sequence length of 250 for observations across all three tasks.

For labelling of the EEG signals, trials are divided into four bins for each task and the BIS measure (described in Section 5.4.1.1) is used to determine participant performance for each bin, with BIS values and the corresponding z-scores calculated separately for each individual. Performance for each task and each participant are plotted in Figure 33, including the best-fit line for each task and each participant. From the best fit lines in Figure 33, it can be seen that every participant, for every task, was at their highest performing state in the

1$^{\text{st}}$ bin, meaning every bin following the 1$^{\text{st}}$ bin was a vigilance decrement in comparison to the 1$^{\text{st}}$ bin. However, across the tasks, participants had varying performance following the 1$^{\text{st}}$ bin as can be seen in Figure 33, with some experiencing their largest decrement in the 2$^{\text{nd}}$, 3$^{\text{rd}}$, or 4$^{\text{th}}$ bins. This makes labelling across all four bins difficult while trying to also maintain a balanced dataset. Given this challenge, we opted to use the 1$^{\text{st}}$ and 4$^{\text{th}}$ bins for our model creation, labelling the 1$^{\text{st}}$ bin as attentive, and the 4$^{\text{th}}$ bin as a decrement, resulting in a perfectly balanced dataset.

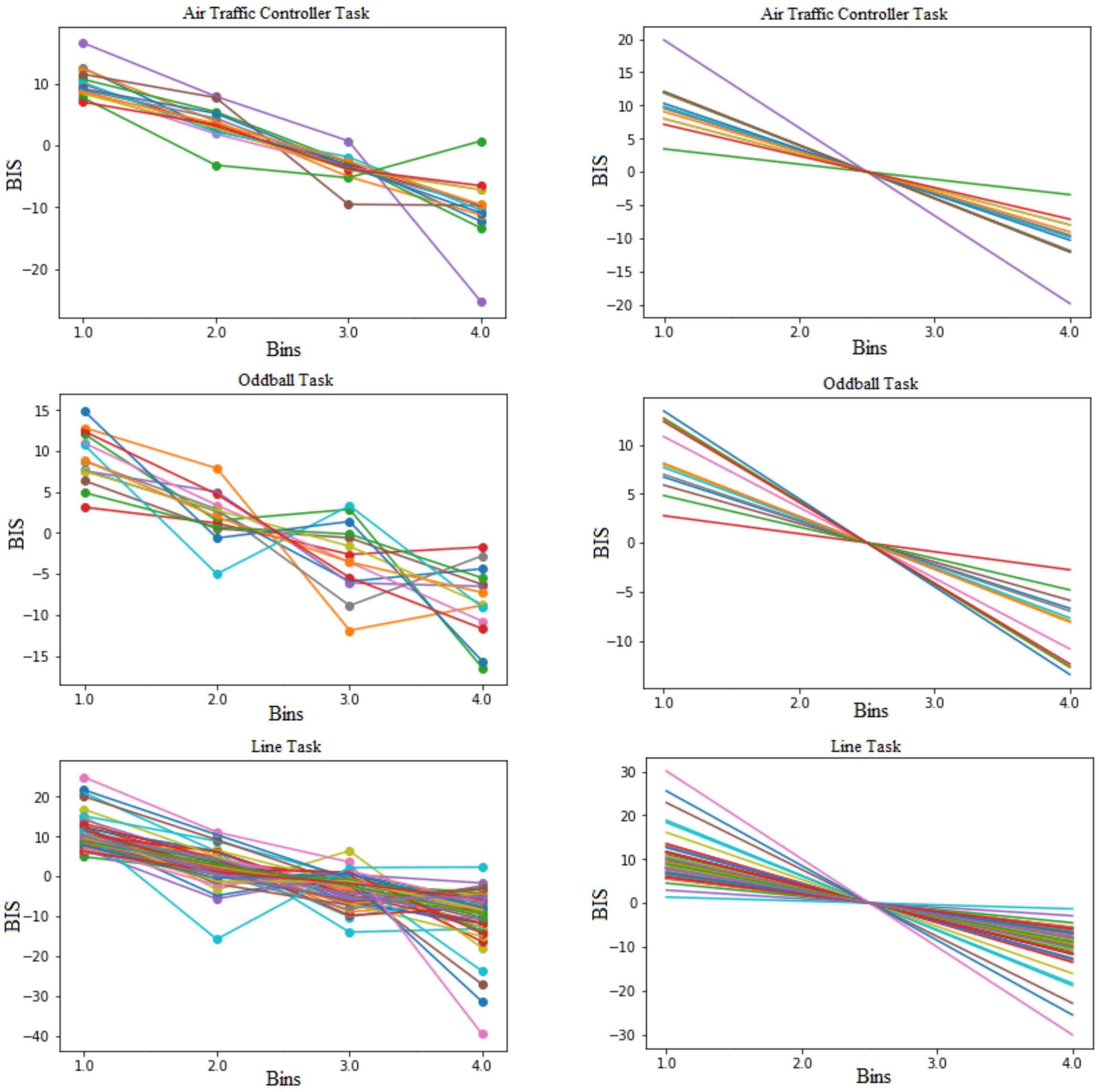


Figure 33. BIS measures and the corresponding best-fit lines for: (a) Air Traffic Controller task (top), (b) Oddball task (middle), and (c) Line task (bottom). BIS measures vary from bin to bin for each participant, with some participants decreasing steadily throughout the entire task, some decreasing initially and then recovering, or some alternating between decreasing and increasing BIS. Note that every participant's best-fit line has a negative slope, indicating that every participant's first bin is their most attentive bin with their largest BIS measure.

Proper labelling of the data is crucial for a machine learning model, and utilizing only the 1st bin as attentive maximizes tying the most attentive trials to their respective neural correlates. Additionally, the underlying mechanism that allows success in building a task-generic model is that mental fatigue is consistent in producing a vigilance decrement in these tasks, and that it is consistent in its neural correlates across different types of vigilance tasks [16]. As mental fatigue has been shown to accumulate over the duration of a vigilance task, the EEG data for the last bin is most likely to have the neural correlates of mental fatigue. As the last bin is a vigilance decrement for all participants across all tasks, using the 1st and 4th bins should maximize the likelihood that the data is labelled properly and will contain the underlying neural correlates to best ensure its success.

5.5.4 Model Creation.

To be effective in detection across participants, a model must be highly generalizable and resistant to the effects of non-stationarity and individual differences. For training and testing of a cross-participant model, this requires that data from participants used for model training must not be used for model validation or testing [71]. This is due to the individual differences and non-stationarity that are inherent within EEG data. If this rule is not followed, the model will likely have overestimated test accuracies, and additionally, the model will not train to be generalizable to a more general population, as the model will learn parameters which are likely only accurate for those participants. Additionally, as this is a task-generic model, the model should be tested with a vigilance task that is unseen by the model. To follow these guidelines, we adopted a leave-two-participants-out cross-validation (L2PO-CV) training method for all three models, resulting in 7-folds. The ATC and Line tasks were used to train the model, with the 3-Stimulus Oddball Task used for validation. This L2PO-CV method was used for training and validation of all three models. Both the ATC task and the Line tasks have the most amount of trials, with each participant having performed four times more trials in each of those tasks than the 3-Stimulus Oddball task, resulting in a more desirable ratio of training to validation data than if the ATC or Line tasks were used for validation. Additionally, this ensures there is training data from both experiments to allow additional generalization for the model, as the Line task was performed in a separate experiment, with an independently selected pool of participants. Ideally, CV would be performed across all three tasks, however this was infeasible due to the immense amount of training time it would require. All together this results in training folds with 41 participants and 53,600 observations total, and validation folds with 2 participants and 360 observations total.

As these cross-participant models are also task-generic, features must be invariant for not only the participants but also the task. For the frequency-domain model, the average power of the five traditional EEG frequency bands for all 64 scalp electrodes were selected as features, resulting in 320 spectral features for each observation, as literature demonstrated that the average power correlates with mental fatigue and is invariant across task, time, and participant [16]. However, an alternative to performing feature extraction by hand is to have the model extract salient features itself. Recently, autoencoders (AEs) have been shown to be more effective than handcrafted features in their ability to compose meaningful latent features from EEG across various classification tasks [53, 168, 169]. Another recent deep learning innovation is Temporal Convolutional Networks (TCNs), which are a new type of architecture for time-series data. TCNs have the advantage of processing a sequence of any length without having a lengthy memory history, leading to much faster training and convergence when compared to Long Short-Term Memory (LSTM) models [66]. For the time-domain models, a TCN-AE is used for one of the models, and a TCN for the other. In the next two sections, general information on TCNs and AEs is provided, followed by the proposed architectures, hyperparameters, and training and testing parameters for all three models.

5.5.5 Temporal Convolutional Networks.

A TCN is a type of convolutional neural network (CNN) for 1D sequence data and was recently developed by Bai et al. [66]. A TCN utilizes dilated convolutions to process a sequence of any length, without having a lengthy memory history. TCNs are typically causal, meaning there is no information leakage from the future to the past, however they can be non-causal as well. The primary elements of a TCN consist of the dilation factor d, the number of filters n, and the kernel size k. The dilation factor controls how deep the network is, with dilations typically consisting of a list of multiples of two. Figure 34 provides a visual example of a causal TCN and aids in understanding the dilated convolutions on a sequence, with the dilation list in the figure being [1,2,4,8]. The kernel size controls the volume of the sequence to be considered within the convolutions, with Figure 34 showing a kernel size of 2. Finally, the filters are similar as they are in a standard CNN, and can be thought of as the number of features to extract from the sequence.

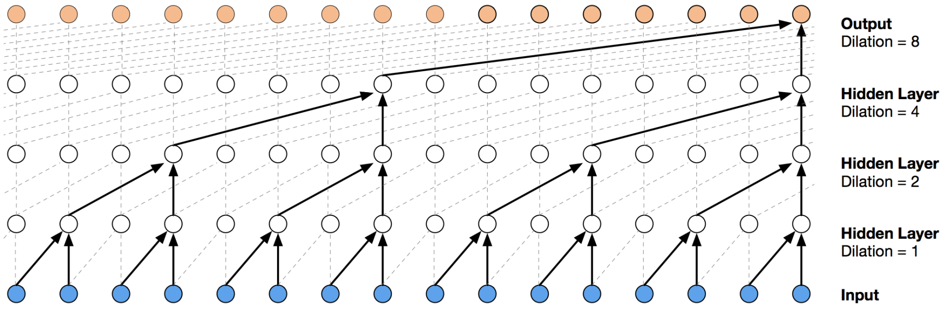


Figure 34. Visual illustration of a causal TCN. This TCN has a block size of 1, a dilation list [1,2,4,8] (i.e. dilation factor 8), and a kernel size of 2 [67]. This results in a receptive field of $2 \cdot 1 \cdot 8 = 16$.

These combined elements form a block as in Figure 34, and blocks can be stacked as they are in Figure 35. This increases the receptive field, which is the total length the TCN captures in processing, and is a function of the number of TCN blocks, the kernel size, and the final dilation, as shown in Equation 5.4. It is common to have a receptive field which matches the input sequence length, however the receptive field is flexible and can be designed to process any length, which is a primary advantage of TCNs. Other advantages include their ability to be trained faster than LSTMs/Gated Recurrent Unit (GRU) models of similar length, having a longer memory than LSTMs/GRUs when capacity of the networks is equivalent, and having similar or better performance than LSTMs/GRUs on a number of sequence related datasets [66, 68]

$$R_{field} = K_{size} \cdot N_{blocks} \cdot d_{final}. \tag{5.4}$$

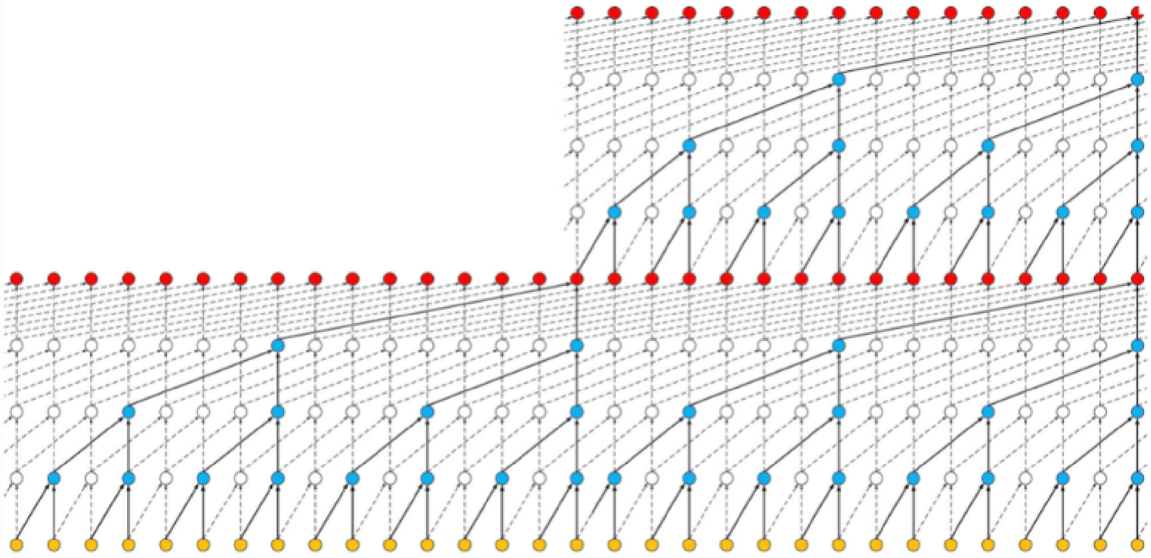


Figure 35. Visual illustration of a causal TCN with stacked blocks. This TCN has a block size of 2, a dilation list [1,2,4,8] (i.e. dilation factor 8), and a kernel size of 2 [68]. This results in a receptive field of $2 \cdot 2 \cdot 8 = 32$**.**

5.5.6 Autoencoders.

An autoencoder (AE) is a type of neural network architecture for unsupervised learning that is primarily used for reproduction of what is input into the network [58]. This is done through the use of two separate networks. One network named the *encoder* $f(\mathbf{x})$ compresses the input into a lower-dimensional representation called the *code* or the *latent-space* $\mathbf{h} = \mathbf{f}(\mathbf{x})$, and another network named the *decoder* reconstructs the input from the code $\mathbf{r} = \mathbf{g}(\mathbf{h})$. An example of a standard AE architecture can be seen in Figure 36. Because of the nature of the encoder, AEs are useful for dimensionality reduction, are powerful feature detectors, and can also be used for unsupervised pretraining of deep neural networks [69].

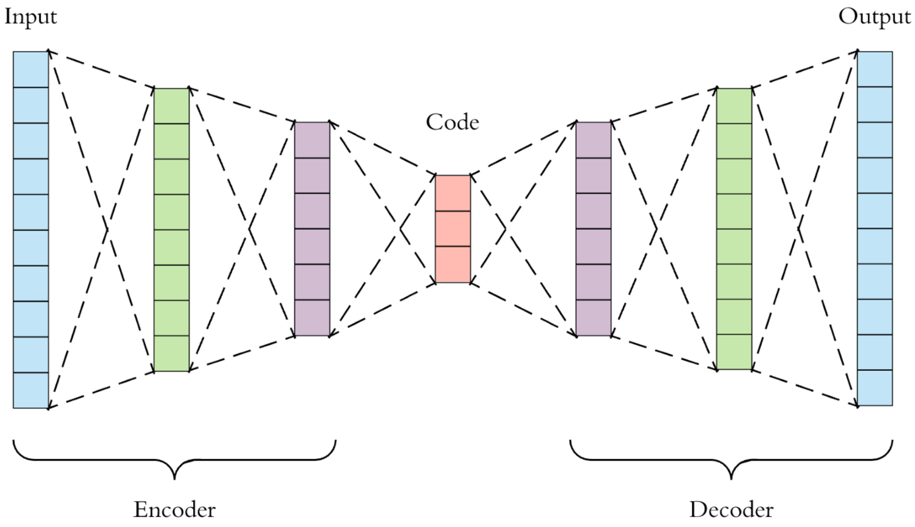


Figure 36. Visual representation of a standard AE architecture.

In Figure 36, the code **h** is constrained to have a smaller dimension than the input **x**. This is called being *undercomplete* and is typical of an AE, as it forces the AE to capture the most salient features of the training data, and thus, the AE doesn't overfit the training data and copy it perfectly [58].

5.5.7 Frequency-domain Model.

The frequency-domain model was a fully connected MLPNN as can be seen in Figure 37, and utilized spectral features extracted from the 1s epoched EEG signal using complex Morlet wavelet transforms in MATLAB to determine the mean power of the five traditional frequency bands: delta (2-4 Hz), theta (4-7 Hz), alpha (8-12 Hz), beta (13-29 Hz), and gamma (33-80 Hz) (details of this process are out of scope for this paper, and we refer the reader to chapters 12 and 13 in Mike Cohen's book, *Analyzing Neural Time Series Data* [18]). With 64 channels from the 64 electrode cap, this resulted in 320 spectral features for each observation ($5 \times 64 = 320$). To improve model training, the spectral features were also log transformed.

The model consisted of three hidden layers with hidden units hu, each followed by a dropout layer with dropout rate dr, with the ReLU activation function used for each hidden layer. As specified at the beginning of Section 5.5.4, L2PO-CV was used for training and validation of the MLPNN model. The Adam optimizer [115] was used to train the models for 300 epochs by minimizing the binary cross-entropy loss, and a hyperparameter sweep was performed over the hidden units hu, the dropout rate dr, and the learning rate lr.

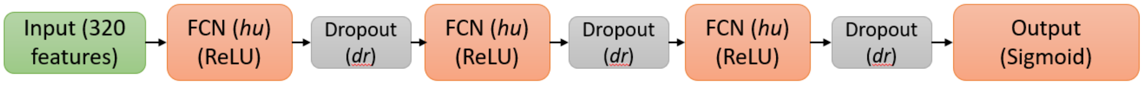


Figure 37. Visual representation of MLPNN classifier. The MLPNN architecture consists of three fully-connected hidden layers with hidden units hu and the ReLU activation, each followed by a dropout layer with a dropout rate dr.

5.5.8 Time-domain Models.

5.5.8.1 TCN-AE.

The TCN-AE architecture was modelled after work done by Thill et al., who recently developed one of the first published TCN-AE architectures for unsupervised anomaly detection in time series data for health monitoring of machines [170]. They credit the success of this model architecture to the architecture's ability to compose and encode salient latent features from the data, and doing so unsupervised. This architecture involves first training the AE to have the ability to reconstruct the EEG signal with minimal loss. Then the encoder of the trained AE encodes the EEG signal to its latent representation, and those latent features are used for training of a classification model. Their architecture was used as a basis for the TCN-AE model of this research, as the goal for this TCN-AE was to encode the most salient features of the EEG data, and then use those features as input to a fully connected neural network (FCN) classifier to perform classification.

The architecture of the TCN-AE is included below in Figure 38, with the encoder on the left, the decoder on the right, and the latent space in the bottom center. The encoder takes as input the EEG signal with dimensions of 250 × 64, with the 250 representing the sequence length of the 1s epoch downsampled to 250 Hz, and the 64 representing the different features from the 64 electrodes. The first layer is a TCN with hyperparameters as specified in Section 5.5.5, with d representing the dilation factor, k the kernel size, b the number of blocks, and n the number of filters. The TCN also used batch normalization, dropout, and recurrent dropout, with the dropout rate dr set as a hyperparameter. This is followed by a 1D convolution (Conv1D) with a kernel size of 1 for further dimensionality reduction and additional non-linearity [170], with L representing the number of filters for this convolution layer, which also represents the number of latent features, as there is no further dimensionality reduction after this layer. The ReLU activation function is used for both the TCN and Conv1D layers. Temporal average pooling is then performed with a size of 5 to reduce the sequence length by a factor of 5. This results in the latent space having a sequence length of $50 \times L$ number of features.

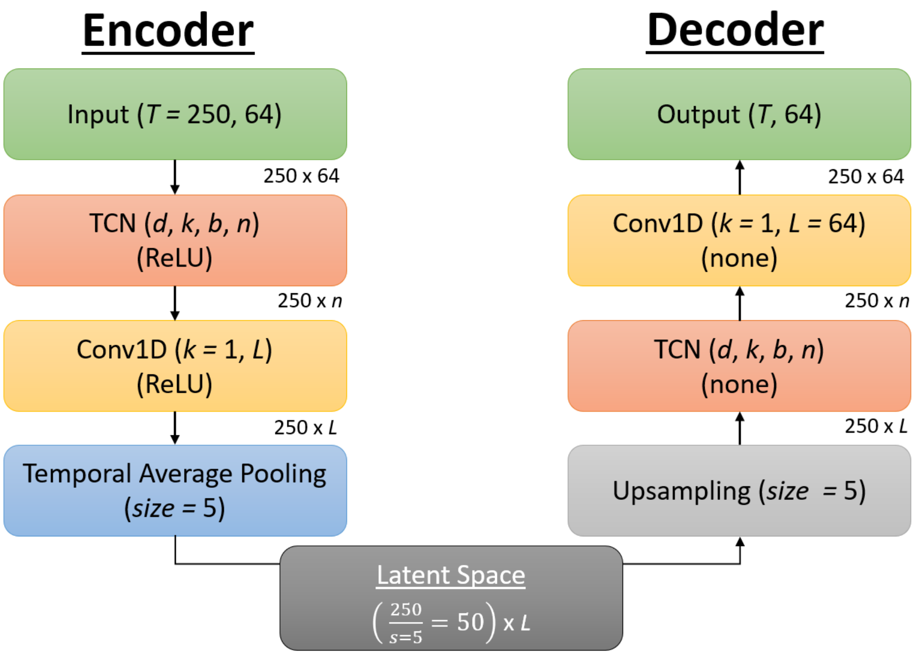


Figure 38. Visual representation of the TCN-AE architecture. Each block corresponds to a layer, with hyperparameters for that layer *italicized*. The activation function for the TCN and Conv1D layers is in parentheses, using ReLU for the encoder and no activation function for the decoder. The dimensions for the input are also provided in the upper-right of each layer as it passes throughout the architecture, with the dimensions starting at T = 250 for the sequence length, and 64 representing the features (corresponding to the 64 electrodes). The latent space dimensions are 50 × L, with L being a hyperparameter.

The decoder is similar to the encoder in its architecture, albeit in reverse. The sequence is first upsampled back to its original length of 250 using nearest neighbor interpolation. The sequence is then passed into a TCN which again has hyperparameters d, k, b, and n, followed by a Conv1D layer which increases the dimensionality of the sequence back to its original size of 64. There is no activation function for the TCN and Conv1D layers in the decoder, as this allows the values of the sequence length to take on any value to recreate the original signal.

L2PO-CV was used for training and validation of the reconstruction phase of the AE, with EEG signals standardized by channel for faster model convergence. The Adam optimizer [115] was used to train the autoencoder for 50 epochs for reconstruction of the EEG signal by minimizing the MSE loss, and hyperparameters were grid-searched using Ray

Tune version 1.3.0, with the hyperparameters consisting of the dilation factor d, the kernel size k, the number of blocks b, the number of filters n, the number of latent features L, the dropout rate dr, and the learning rate lr.

Once the autoencoder was trained for reconstruction, the weights of the encoder were locked and the encoder was then used to encode input sequences into latent features. The latent features were then flattened and used as input features into a FCN classifier. The TCN-AE architecture in its entirety can be seen in Figure 39. The FCN classifier had two hidden layers, each with the ReLU activation function, followed by a dropout layer, and a output layer using the sigmoid function. L2PO-CV was used for training and validation of the FCN for classification. The Adam optimizer [115] was used to train the models by minimizing the binary cross-entropy loss, and a hyperparameter sweep was performed over the number of hidden units for each layer, the dropout rate, and the learning rate.

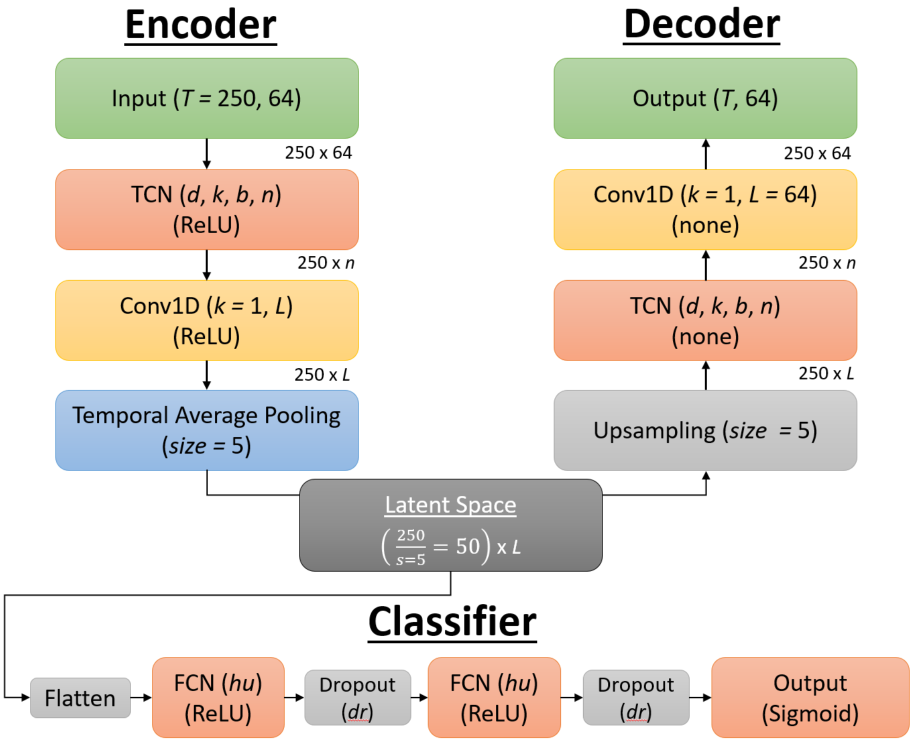


Figure 39. Visual representation of the TCN-AE classifier. The Encoder and Decoder comprise the AE architecture, with the latent space then used as input to the FCN classifier shown at the bottom. The FCN classifier architecture consists of two fully-connected hidden layers with hidden units *hu*, each followed by a dropout layer with a dropout rate *dr*.

5.5.8.2 TCN.

The TCN model can be seen in Figure 40, and was similar to the encoder portion of the TCN-AE architecture in that it consists of a TCN layer and a Conv1D layer, however this model differs in that prediction is performed after the Conv1D layer, using an output layer with a sigmoid activation function. The TCN layer has hyperparameters as specified in Section 5.5.5, with d representing the dilation factor, k the kernel size, b the number of blocks, and n the number of filters. The TCN also used batch normalization, dropout, and recurrent dropout, with the dropout rate dr set as a hyperparameter. The Conv1D has a kernel size of 1 and a filter size of 4, providing dimensionality reduction before the output layer. The ReLU activation function is used for both the TCN layer and the Conv1D layer. L2PO-CV was used for training and validation of the TCN for classification, with EEG signals standardized by channel for faster model convergence. The Adam optimizer [115] was used to train the models for 100 epochs by minimizing the binary cross-entropy loss, and a hyperparameter sweep was performed using Ray Tune and grid search over the dilation factor d, the kernel size k, the number of blocks b, the number of filters n, the dropout rate dr, and the learning rate lr.

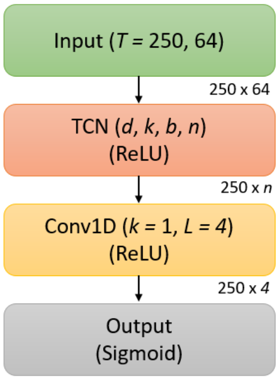


Figure 40. Visual representation of the TCN classifier. Each block corresponds to a layer, with hyper-parameters for that layer *italicized*, and the activation function in parentheses.

5.6 Results

Below are the results for both the frequency-domain model and the time-domain models. For each model, the best hyperparameter configuration is presented along with its CV balanced accuracy and confidence interval (CI). As accuracy is a binomial distribution, approximate binomial confidence intervals are used. Specifically we utilize Agresti Coull confidence intervals, as they typically maintain α while not being overly conservative [**?**]. Each model's CV balanced accuracy and its 95% Agresti Coull confidence interval is compared to random chance, i.e. a naïve classifier with accuracy of 50% (accuracy is 50% as this is a binary classification task). Validation accuracies are also provided for each participant by the participant's ID, along with their 95% confidence interval. At the end of this section, a table is provided with the participant validation accuracies for each model and the 7-fold CV accuracy for each model.

5.6.1 Frequency-domain Model.

Hyperparameter sweeps for the MLPNN model resulted in the best network achieving a 7-fold CV balanced accuracy of 64% (95% CI: 0.59, 0.69) with the following hyperparameters: hidden units of [250, 200, 150] (by layer), learning rate of 0.00001, and dropout rate of 0.5. This results in the model having CV accuracy statistically greater than random chance as evidenced by the confidence interval. Figure 41 depicts the validation accuracies for each participant for the MLPNN model, with 9 participants having validation accuracies statistically greater than random chance. Participants 2, 3, 7, 8, and 11, did not have validation accuracies greater than random chance.

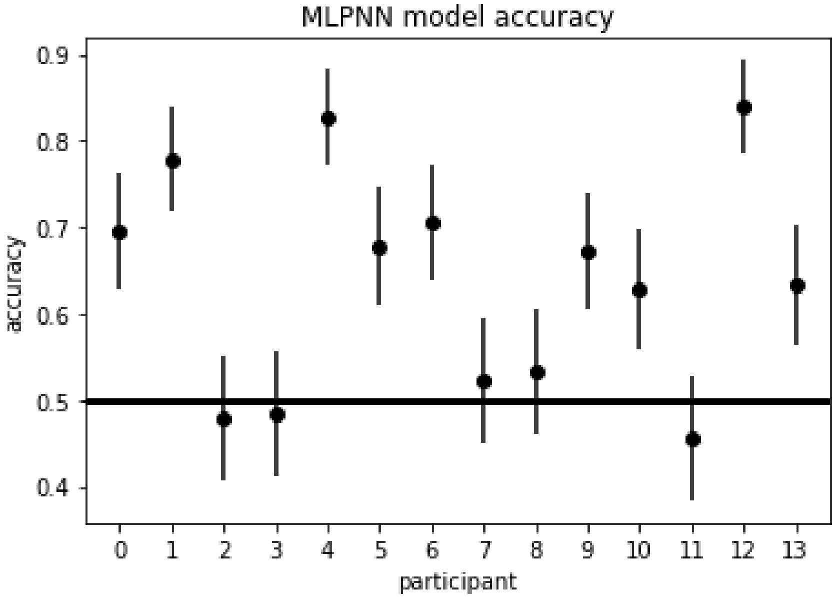


Figure 41. Participant validation accuracies for the MLPNN model, with 9 participants having validation accuracies statistically greater than random chance. Participants 2, 3, 7, 8, and 11, did not have validation accuracies greater than random chance. This model achieved a 7-fold CV accuracy of 64% (95% CI: 0.59, 0.69).

5.6.2 Time-domain Model - TCN-AE.

The best hyperparameters found for the TCN-AE signal reconstruction had the following configuration: dilations [1, 2, 4, 8, 16, 32], kernel size of 2, number of filters 36, number of blocks 2, learning rate of 0.0001, and dropout rate of 0.0; and resulted in a receptive field of $2 \cdot 2 \cdot 32 = 128$. For the classifier portion of the TCN-AE, all hyperparameter sweeps resulted in similar performance, with accuracies ranging between 48% and 52% for 7-fold CV balanced accuracy, with no set of hyperparameters resulting in a model which performed statistically better than chance. Individual participant accuracies were also investigated for each hyperparameter sweep, with two or less participants having significant performance for the hyperparameter sweeps. No participants had validation accuracies statistically greater than random chance.

5.6.3 Time-domain Model - TCN.

The best hyperparameter sweep for the TCN model yielded a 7-fold CV balanced accuracy of 56% (95% CI: 0.51, 0.61) with the following hyperparameters: dilations [1, 2, 4, 8, 16, 32], kernel size of 4, number of filters 10, number of blocks 2, learning rate of 0.0001, and dropout rate of 0.5; and resulted in a receptive field of $4 \cdot 2 \cdot 32 = 256$. This results in the model having CV accuracy statistically greater than random chance as evidenced by the confidence interval. Figure 42 depicts the validation accuracies for each participant for the TCN model, with 3 participants (1, 7, and 12) having validation accuracies statistically greater than random chance.

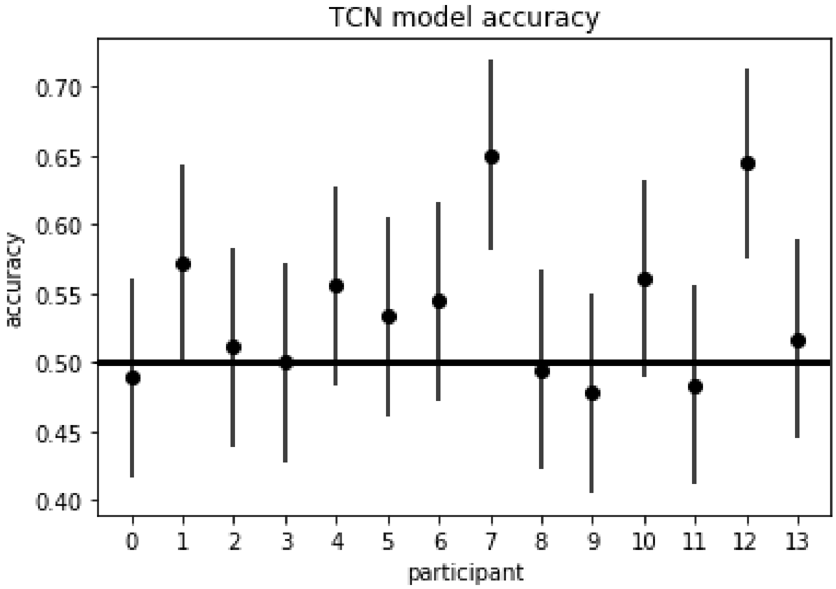


Figure 42. Participant validation accuracies for the TCN model, with 3 participants (1, 7, and 12) having validation accuracies statistically greater than random chance. This model achieved a 7-fold CV accuracy of 56% (95% CI: 0.51, 0.61).

Table 9 provides the participant validation accuracies and the 7-fold CV accuracy for all three models.

Table 9. Vigilance decrement classification model performance results for each model type. Participant validation accuracies and the 7-fold CV accuracy are provided for each model, with 95% confidence intervals provided in parentheses. For both participant accuracies and the 7-fold CV results across all participants, Bold signifies statistical significance of accuracy over random chance (defined as 50% for this binary classification task) as evidenced by the 95% confidence interval.

Participant #	MLPNN	TCN-AE	TCN
	Val Acc	Val Acc	Val Acc
0	**0.69 (0.62, 0.76)**	0.51 (0.44, 0.58)	0.49 (0.42, 0.56)
1	**0.78 (0.71, 0.83)**	0.49 (0.42, 0.56)	**0.57 (0.50, 0.64)**
2	0.48 (0.41, 0.55)	0.49 (0.42, 0.57)	0.51 (0.44, 0.58)
3	0.48 (0.41, 0.56)	0.52 (0.45, 0.59)	0.50 (0.43, 0.57)
4	**0.83 (0.77, 0.88)**	0.49 (0.42, 0.57)	0.56 (0.48, 0.63)
5	**0.68 (0.61, 0.74)**	0.55 (0.48, 0.62)	0.53 (0.46, 0.60)
6	**0.71 (0.63, 0.77)**	0.47 (0.40, 0.54)	0.54 (0.47, 0.62)
7	0.52 (0.45, 0.59)	0.56 (0.48, 0.63)	**0.65 (0.58, 0.72)**
8	0.53 (0.46, 0.60)	0.56 (0.49, 0.63)	0.49 (0.42, 0.57)
9	**0.67 (0.60, 0.74)**	0.54 (0.47, 0.62)	0.48 (0.41, 0.55)
10	**0.63 (0.56, 0.69)**	0.49 (0.42, 0.57)	0.56 (0.49, 0.63)
11	0.46 (0.38, 0.53)	0.46 (0.38, 0.53)	0.48 (0.41, 0.56)
12	**0.84 (0.78, 0.89)**	0.48 (0.41, 0.55)	**0.64 (0.57, 0.71)**
13	**0.63 (0.56, 0.70)**	0.53 (0.46, 0.60)	0.52 (0.44, 0.59)
7-fold CV	**0.64 (0.59, 0.69)**	0.52 (0.47, 0.57)	**0.56 (0.51, 0.61)**

5.7 Discussion

The frequency-domain model (MLPNN) had the highest level of performance of the three model types, with 7-fold CV accuracy statistically greater than random chance at 64% (95% CI: 0.59, 0.69), and nine of the fourteen participants having validation accuracies significantly greater than random chance, as evidenced by their respective 95% confidence intervals. The best time-series domain model (TCN) also had 7-fold CV accuracy statistically greater than random chance at 56% (95% CI: 0.51, 0.61), however, only three of the fourteen participants had validation accuracies significantly greater than random chance. Additionally, the MLPNN had significantly greater CV model accuracy than the TCN model, as evidenced by the 95% confidence interval for the difference between the two classifiers not containing 0, i.e. model accuracy difference of 8% (95% CI: 0.01, 0.15), and the MLPNN also had significantly more participants with validation accuracies greater than random chance than the TCN model, as evidenced by the McNemar's test statistic of $4.5 \geq 3.84$ ($p < .034$, $\alpha = 0.05$). Two of the participants in the MLPNN model, 4 and 12, had validation accuracies greater than 80%. Participant IDs of this model that did not have validation accuracies significantly greater than random chance were participants 2, 3, 7, 8, and 11, with participant 7 having the worst validation accuracy of 46%. Participant 7, however, was the participant with the highest validation accuracy for the TCN model, with participant 12 being the second highest. Participants having such differing levels of performance across all three model types suggests that low model performance for the TCN and the TCN-AE was not due to certain individual participants having poor quality of data.

One reason for the significant difference between the MLPNN and TCN models could lie in their difference of domains, i.e. frequency vs time. The literature suggests that changes in the average power of specific bands correlates to mental fatigue in sustained attention tasks [16], which also correlates to a vigilance decrement, and if these spectral features are the most salient information for mental fatigue, then there is no additional information gained by the network utilizing raw time-series signals versus spectral features. Furthermore, TCN performance is contingent on being able to learn that these spectral features are important given only the time-series signals, whereas these spectral features *are* the input for the MLPNN, so the MLPNN doesn't have to learn them. Thus, the MLPNN may have an advantage over the time-series domain models in that it could already have the most salient features to perform classification.

BIS measures of the 3-stimulus oddball task were investigated to determine if they correlated to model performance of the MLPNN model. If BIS measures are correlated to model performance, this would suggest that the magnitude of decline in a participant's task performance is correlated to how well the model can classify the EEG; i.e. the worse a decline in a participant's performance, the better the model can classify the EEG. Additionally, if the MLPNN model uses neural correlates of mental fatigue to perform classification, this would also suggest that as a participant becomes more mentally fatigued, they suffer a larger decline in task performance. To investigate if there was a correlation, BIS slopes of each participant, as well as the difference between the BIS measure of the first and last bins of each participant, were compared to the MLPNN model performance for that participant. These values are provided below in Table 10. The BIS slopes and MLPNN validation accuracies were not found to be correlated ($\rho = 0.07$, p = 0.82), nor were the BIS difference values and MLPNN validation accuracies ($\rho = -0.09$, p = 0.76).

Table 10. This table provides the BIS slope and difference between the BIS measures of first and last bin for the oddball task for each participant. Validation accuracy for the frequency-domain MLPNN model is also provided for each participant. Bold signifies statistical significance of accuracy over random chance (defined as 50% for this binary classification task) as evidenced by the 95% confidence interval.

Participant #	BIS Slope	BIS Difference (1st bin - 4th bin)	MLPNN Val Acc
0	-4.46	12.02	**0.69 (0.62, 0.76)**
1	-8.48	21.67	**0.78 (0.71, 0.83)**
2	-8.42	28.53	0.48 (0.41, 0.55)
3	-1.84	4.85	0.48 (0.41, 0.56)
4	-5.34	14.07	**0.83 (0.77, 0.88)**
5	-3.92	12.68	**0.68 (0.61, 0.74)**
6	-7.22	21.75	**0.71 (0.63, 0.77)**
7	-4.64	11.56	0.52 (0.45, 0.59)
8	-5.30	16.19	0.53 (0.46, 0.60)
9	-5.11	19.81	**0.67 (0.60, 0.74)**
10	-8.96	30.53	**0.63 (0.56, 0.69)**
11	-5.40	16.16	0.46 (0.38, 0.53)
12	-3.21	10.41	**0.84 (0.78, 0.89)**
13	-8.24	24.08	**0.63 (0.56, 0.70)**

For the MLPNN model, as this is an artificial neural network, there is no way to know for certain if the model is utilizing neural correlates of mental fatigue to determine if there is a vigilance decrement. However, if the model is utilizing neural correlates of mental fatigue, the lack of correlation between BIS measures and model performance suggests that the magnitude of the mental fatigue does not correlate to the magnitude of the vigilance decrement, or that the correlation is participant specific, i.e. some participants could be heavily fatigued and only suffer a slight decrease in performance, while some participants may have a significant decrease in performance when even moderately fatigued. Also, the vigilance decrement is a measure of task performance, and thus, in general, factors other than fatigue can affect a person's performance, such as outside distractions, lack of motivation to perform well, etc. It is possible that, even in a lab environment, factors such as this affected participant performance, resulting in a large BIS slope or BIS difference for certain participants, yet with only minimal mental fatigue accumulation.

Additionally, the literature notes that the neural correlates of mental fatigue and sleep fatigue manifest differently depending on the task, and that they can be opposites of one another, yet both types of fatigue affect task performance in a similar manner. Given this, it could be that some of the participants accumulated sleep fatigue, as opposed to mental fatigue, as the task continued on, resulting in a decrease in performance, but with neural correlates which differ from mental fatigue. As these neural correlates can be opposites of one another (e.g. an increase in spectral power for the alpha band as opposed to a decrease), it would be difficult for the model to generalize both of these types of fatigue.

5.8 Conclusions and Future Work

In conclusion, the model type that was most capable of classifying the vigilance decrement in an unseen task and unseen participant out of the models examined was the MLPNN frequency-domain model, utilizing spectral features extracted from the EEG, namely the average power of the five traditional EEG frequency bands. This finding is significant as thus far, a task-generic EEG cross-participant model of the vigilance decrement, i.e. a model capable of classifying the vigilance decrement in an unseen task and unseen participants, has not been built or validated. Previous work by Yin et al. in building a task-generic model did not utilize a different type of task in order to validate their model, and instead only varied parameters within a single task of operating an aircraft's cabin air management system in order to create two tasks. Our research in contrast utilized three different types of tasks (the air traffic controller task, the line task, and the 3-stimulus oddball task), and all of which are well established in the literature as vigilance type tasks.

To improve model performance, future work should incorporate more vigilance tasks for both training and testing as more EEG vigilance type datasets become available. Additionally, CV should be performed across all tasks to investigate if certain tasks provide more or less generalization and task invariance to the model. Selection of specific spectral features, such as certain frequency bands, should also be explored. By selecting only certain frequency bands, and/or certain regions of the head, model performance could be improved, as currently, the model utilizes a large number of features (320), but only certain features may be needed in order for the model to accurately classify the vigilance decrement, and removing these unnecessary features could reduce overfitting of the model. This feature importance could be determined through research of visualization techniques which allow for visual inspection of the features which result in maximum discrimination between the two classes (vigilance decrement vs not). Further investigation into mental fatigue vs sleep fatigue could also be useful. Experiments which note the sleepiness of participants throughout the experiment, either through objective measurements such as prolonged eye closure, or through subjective measurements such as observation and surveys, could result in separate data for neural correlates of mental fatigue vs sleep fatigue. These experiments could then be used for separate training and testing of the model, and this could reveal if incorporating both types of fatigue either aids the model or hinders it.

To further validate the model, future experiments should investigate devising tasks which result in an increase in vigilance. Currently, every participant experiences a vigilance decrement over the duration of each task, as the tasks are designed to do so. However, this presents a concern for model validation as the data is homogeneous across every participant. Ideally, for model training and validation, there would be data for both a vigilance decrement and a vigilance increase, to ensure the model could differentiate between the two, and to ensure the model isn't classifying based solely on task duration. These tasks could perhaps be achieved through planned breaks throughout the task, however, these experiments would require further validation themselves to ensure they reliably produce an increase in vigilance.

Separate but related work which should stem from this research would be to use EEG to determine when an individual is dropping below a standard level of performance. The vigilance decrement is useful as it informs when someone is experiencing a decrease in performance, however, this decrease in performance is relative to the person's own baseline level of performance. In certain tasks, it would be valuable to predict when an individual's expected performance would be too low for successful task completion. This research has demonstrated that EEG can be utilized to determine whether or not someone is experiencing a vigilance decrement, even in an unseen task, and thus it is possible that a model could utilize a participant's baseline measure of performance to determine if that participant has dropped below a performance threshold, however, more work is necessary for proper implementation. Additionally, a regression model could be investigated to predict the measure of performance itself.

Lastly, EEG research into the vigilance decrement should overall move towards more multi-task experiments, task agnostic models, and dataset sharing. This research demonstrated that the neural correlates of the vigilance decrement span across different task types and can be utilized to detect the vigilance decrement across these task types, however, to further pinpoint which specific features span across all of the different types of vigilance tasks, additional experiments which utilize multiple tasks are needed. Dataset sharing through repositories such as Kaggle [72] or the UCI machine learning data repository [73] would also further enable future research into task agnostic models, as experiments with different tasks could be combined for model building and neural correlate analysis. Future experiments require time and funding, however, dataset sharing could quickly enable this research by utilizing existing datasets across many types of vigilance tasks (piloting of aircrafts, driving, air traffic control, etc.).

5.9 Appendix A - EEG Preprocessing

Preprocessing of EEG data was performed through script batch processing using EEGLAB [108], and consisted of a combination of best practice steps from both Makoto's preprocessing pipeline [104] and the PREP pipeline [109].

1. Modifed EEGLAB to use double precision, as single precision can destroy natural commutativity of the linear operations.

2. Imported data into EEGLAB and included reference channels based on the equipment used (e.g. Biosemi's 64 scalp electrode cap uses channels 65 and 66 as reference channels, which are electrodes placed on the mastoids specifically for the purpose of referencing).

3. Down-sampled to 250 Hz for purpose of improving ICA decomposition by cutting off unnecessary high-frequency information, and also to reduce data size.

4. High-pass filtered the data at 1 Hz to reduce baseline drift, improve line-noise removal, and to improve ICA [110]. High-pass filter is done before line-noise removal

and 1 Hz is used as we were not performing event-related potential (ERP) analysis, which could be affected by using a 1 Hz high-pass filter, and would require an alternate strategy.

5. Imported channel info using International 10-20 system to allow for re-referencing.

6. Removed line noise using CleanLine plugin (default 60Hz notch filter) [111].

7. Removed bad channels using EEGLAB clean_rawdata plugin patented by Christian Kothe [112], which utilizes Artifact Subspace Reconstruction.

8. Interpolated all removed channels to minimize a potential bias in the average referencing step.

9. Re-referenced data to the average. Mastoid referencing isn't always sufficient [109], and re-referencing the data to the average helps suppress line noise that wasn't rejected by CleanLine [104].

10. Independent Component Analysis (ICA) 'runica' "'infomax': (extended)" algorithm variant was executed with the vertical EOG (VEOG) electrode used as input for the function.

11. ICA results from step 10 are used to remove artifact ICA components.

VI. Conclusions and Future Work

This dissertation has presented new findings in utilizing EEG in cross-participant models for classification in the domains of visual search and the vigilance decrement, and also more broadly in the general method of dataset partitioning for EEG cross-participant models. In the next section, these contributions are discussed in greater detail by chapter. Chapters I and II reviewed the domain of EEG classification and how this growing area of research could be applied to aiding future military operators as they continue to face operations which require visual search and sustained attention. This review included EEG's inherent non-stationarity and inter-participant variability, as careful dataset partitioning is required for EEG cross-participant models due to these aspects of EEG. Deep learning and the fundamental models utilized in this research were also reviewed. Finally, the background of the vigilance decrement and how it was elicited across various tasks was also presented.

6.1 Contributions and Findings

Chapter III demonstrated the effects of proper and improper methods of dataset partitioning for EEG cross-participant models, and work was performed both mathematically and empirically. Five publicly available datasets were utilized to train, validate, and test models using both proper and improper methods of dataset partitioning. To build these models, previous publications were replicated and the results between proper and improper are compared, demonstrating that model accuracy on unseen participants is significantly overestimated with improper dataset partitioning. The relationship between inter-participant variability and covariate shift was also demonstrated using an originally developed *shift to median* transformation method which was shown to reduce inter-participant variability and subsequently covariate shift through visualization using t-SNE. Model performance was also evaluated before and after the transformation and it was demonstrated that as inter-participant variability is reduced, and thus covariate shift is also reduced, that model performance is then increased. Shimodaira's equation for rescaling the loss of the training dataset to appropriately match the test loss in the presence of covariate shift was also utilized to demonstrate the presence of covariate shift when using the proper method of dataset partitioning versus the lack of covariate shift when using the improper method of dataset partitioning. This was done using a noval visualization method which utilized two-dimensional histogram estimators on a PCA projection of high-dimensional data to generate a heatmap for visualization of the loss rescaling weight ratio values of the Shimodaira equation. The results demonstrated that there was a significant increase in the amount of loss rescaling required for proper dataset partitioning versus improper dataset partitioning, which is indicative of the presence of covariate shift. In summary, the contributions and findings of Chapter III were:

A1: Utilized Shimodaira's loss rescaling on real data in order to demonstrate the presence of covariate shift between proper and improper methods of dataset partitioning using a novel heat map visualization of the loss rescaling weight ratio values, providing the first known method and use of Shimodaira's equation on real data in order to mathematically demonstrate the presence of a covariate shift.

A2: Developed a EEG data transformation technique to reduce inter-participant variability in the data and utilized this transformation to demonstrate how inter-participant variability affects covariate shift and resulting model performance.

A3: Empirical results show that not following the proper method of dataset partitioning can result in underestimation of error rates in models between 35% and 3900% in unseen participants, and also results in sub-optimal model creation for its intended purpose.

Chapter IV investigated how to detect and mitigate inefficient search by designing and conducting an experiment for natural inefficient search, and utilized various techniques in order to mitigate the inefficient search. The experiment was successful in creating a search environment where inefficient search was naturally the default behavior, with 80.86% of participant searches being inefficient prior to mitigation (with 7.18% of those being circular inefficient). Mitigation techniques that curbed the predominant behavior and encouraged efficient search were also found to be significant, with a 169% increase of efficient searches, resulting in a total of 51.41% of all searches being efficient, with the *nudge* and *hint* techniques being most effective, and the *explanation* and *instructions* techniques not found effective. Efficient search was also found to be faster than inefficient search, resulting in a 13% speed up, and also more accurate, with a 61% reduction in error rate. Physiological measures of EEG, ECG, EOG, GSR, and gaze tracking data, were also collected throughout the experiment. EEG was utilized to develop within and cross-participant models to evaluate the effectiveness of EEG for classification of inefficient and efficient search. Model types evaluated included Linear Discriminant Analysis (LDA), Random Forest Classifier (RFC), Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Temporal Convolutional Network (TCN). Due to the nature of the *nudge* mitigation altering the fundamental aspects of the search, such as gaze fixation duration and saccade intervals, four different methods of partitioning the data were utilized in order to isolate these effects for training and testing of the different model types. However, the models did not perform better than chance accuracy. Overall, the research in Chapter IV had the following findings:

B1: Without a mitigation technique or training of how to perform an efficient search, the majority of participant searches were inefficient.

B2: Efficient search was found to be faster and more accurate than inefficient search.

B3: The *nudge* technique was the most effective mitigation technique, followed by the *hint* technique. The *explanation* and *instructions* techniques were not found to be effective.

B4: The effectiveness of EEG for classification of efficient search was also investigated, with multiple dataset partitioning methods and model classifiers being investigated for both within and cross-participant models. Overall, the models did not perform better than chance.

Chapter V investigated the ability to detect the vigilance decrement using EEG across participants (cross-participant) and tasks (cross-task). This is significant because the vigilance decrement is typically measured using sustained attention tasks which fall into different taxonomies based on their information-processing demands and thus, generalized behavior [5, 37]. Thus far, a generalized model which can classify the vigilance decrement across any sustained attention task using independent measures, such as physiological signals, has not been achieved. Additionally, models which can generalize and classify on any participant are desirable, as there is no additional time needed to either train a new model or update an existing model for every new user. Cross-task cross-participant models were evaluated for traditional machine learning model types in both the frequency-domain and time-domain, with the frequency-domain being an MLPNN using spectral features, and the time-domain being a TCN using the raw EEG voltage values. The MLPNN and TCN models both achieved cross-validation accuracies greater than random chance (50%), with the MLPNN performing best with 7-fold cross-validation (CV) balanced accuracy of 64% (95% CI: 0.59, 0.69), and 9 out of 14 participants with validation accuracies significantly greater than random chance. The TCN model resulted in 7-fold CV balanced accuracy of 56% (95% CI: 0.51, 0.61), with only 3 of the 14 participants having validation accuracies greater than random chance. When comparing these two models, the MLPNN was found to have significantly greater cross-validation accuracy than the TCN model, as evidenced by the 95% confidence interval of the model accuracy difference not containing 0 (8% (95% CI: 0.01, 0.15)). The MLPNN also had significantly more participants with validation accuracies greater than random chance, as evidenced by the McNemar's test statistic of $4.5 \geq 3.84$ ($p < .034$, $\alpha = 0.05$). The proposed novel TCN-AE model was explored due to its ability to distill the raw time-series voltage values to their most salient features, however, the TCN-AE did not achieve cross-validation accuracy greater than random chance, and none of the participants had validation accuracies greater than random chance. The poor performance of the TCN-AE could be due to the encoder learning features for the latent space which are most salient for signal reproduction, but which aren't the most optimal features for classification of the vigilance decrement. In fact, as the frequency-domain MLPNN significantly outperformed both of the time-domain TCN models, these findings suggest that spectral features may be most appropriate for vigilance decrement detection. Overall, this research has demonstrated that the vigilance decrement can be classified using EEG from unseen participants performing an unseen task, which is significant as thus far, models have only classified the vigilance decrement for seen tasks.

C1: The vigilance decrement can be classified in an unseen task using EEG from unseen participants, as evidenced by both the frequency-domain MLPNN and time-domain TCN achieving cross-validation accuracies greater than random chance.

C2: Frequency-domain models may outperform time-domain models for classification of the vigilance decrement due to the salient information contained within spectral features, as evidenced by the frequency-domain MLPNN having a significantly greater CV balanced accuracy of 64% (95% confidence interval (CI): 0.59, 0.69), and significantly more participants with validation accuracies greater than random chance (9 out of 14 participants).

6.2 Future Work

There are several lines of future work to be carried forward from this dissertation. For proper validation and testing of EEG cross-participant models, additional work is needed to standardize performance assessment and benchmarking of EEG cross-participant models, such as cementing the guidelines for proper dataset partitioning that are provided in this dissertation as an IEEE standard. These standards should then be widely disseminated across government, industry, and academia, with a focus on reaching all relevant conference and journal publications. In order to correct the trend of overestimated test accuracies in the EEG cross-participant model building body of research, peer-review must be bolstered, and thus all editorial teams and reviewers should be made aware of the standards, understand how they are to be properly followed, and the implications for model results if they are not followed. Special attention should also be focused on disseminating these standards to data contributors and owners and maintainers of dataset repositories. It is critical that datasets posted for public use are not prepared improperly, otherwise standards aren't adhered to from the start, and this sets up all users for failure. Ideally, in addition to ensuring the datasets are hosted properly, EEG data should always have (de-identified) participant labels available so that users may partition the data themselves. In this manner, users would be free to perform proper dataset partitioning with varying numbers of participants included in the training, validation, and testing sets, for their own research and investigation.

Further investigation into visual search could largely benefit the detection portion of the ESE research. Models tended to overfit due to the large amount of features used for the amount of data present (320 features and at most 480 trials per participant), despite utilizing regularization via dropout. Future work should explore reducing the number of features through feature selection, to select only the most salient features for training as suggested through literature review of visual search. Balancing of the datasets was also a challenge in this research. Four different datasets were generated due to the *nudge* technique altering fundamental aspects of the search. This considerably increases the amount of time to train models and explore hyperparameters, and is also sub-optimal in utilizing the full 480 trials provided by the participant. Future work should consider a modified ESE where the focus is solely on generating an EEG dataset for machine learning. With this approach, the experiment could explicitly instruct participants which type of search to use. This allows researchers to construct the experiment to result in a balanced dataset with much more precise labelling of the data, as the participants would be instructed block by block which visual search to perform. Epoching of the data was another challenge that could potentially be alleviated with this new experiment. Epoching was performed two seconds before the participant pushed the key for response, determined by an overall average of trial length to ensure the most amount of trials were epoched appropriately, however, data for trial length was imbalanced, with an average efficient search being 1.99 (95% CI: ± 0.37) seconds and an average inefficient search being 2.29 (95% CI: ± 0.50). By instructing the participants on which search to perform, this could potentially reduce the variable trial length and result in more accurate epoching of the data. Overall this results in an experiment and subsequent data that is less relevant to the real world, however, as models are currently unable to perform classification of efficient search, this could result in a dataset that is more apt for model creation and thus better determine the efficacy of using EEG for detection of inefficient and efficient search. Lastly, while this research focused on detection and mitigation of confirmation bias in visual search, the field should focus towards detection and mitigation of confirmation bias in any task. Experiments in the literature have demonstrated specific neural correlates for confirmation bias in general decision making [171], and thus it should be feasible to build a model which detects confirmation bias in decision making through EEG signals.

The vigilance decrement was classified in EEG from an unseen task and unseen participant, however further rigorous testing of this model is needed. Thus far, only one unseen task has been tested with the model. Future work should incorporate more vigilance tasks for both training and testing as more EEG vigilance type datasets become available. Additionally, cross-validation should be performed across all tasks to investigate if certain tasks provide more or less generalization and task invariance to the model. Utilizing all data within an experiment should also be explored for model creation, with a methodology created in order to appropriately label every bin as attentive or in a decrement, based on their performance measure. Alternatively, a regression model could be built to predict the trial's performance measure based on its corresponding bin's performance measure, which would allow use of all bins of trials. For this regression model, a separate methodology to classify a trial as attentive or in a decrement would then be needed after the performance measure is predicted by the model. Selection of specific spectral features, such as certain frequency bands, should also be explored. By selecting only certain frequency bands, and/or certain regions of the head, model performance could be improved, as currently, the model utilizes a large number of features (320), but only certain features or regions of the brain may be needed in order for the model to accurately classify the vigilance decrement, and removing these unnecessary features could reduce overfitting of the model. This feature importance of the neural network model could be determined through visualization techniques which allow for visual inspection of the model features which result in maximum discrimination between the two classes (vigilance decrement vs not). Further investigation into mental fatigue vs sleep fatigue could also be useful. Experiments which note the sleepiness of participants throughout the experiment, either through objective measurements such as prolonged eye closure, or through subjective measurements such as observation and surveys, could result in separate data for neural correlates of mental fatigue vs sleep fatigue. These experiments could then be used for separate training and testing of the model, and this could reveal if incorporating both types of fatigue either aids the model or hinders it. Lastly, EEG research into the vigilance decrement should overall move towards more multi-task experiments, task agnostic models, and dataset sharing. This research demonstrated that the neural correlates of the vigilance decrement span across different task types and can be utilized to detect the vigilance decrement across these task types, however, to further pinpoint which specific features span across all of the different types of vigilance tasks, additional experiments which utilize multiple tasks are needed. Dataset sharing through repositories such as Kaggle [72] or the UCI machine learning data repository [73] would also further enable future research into task agnostic models, as experiments with different tasks could be combined for model building and neural correlate analysis. Future experiments require time and funding, however, dataset sharing could quickly enable this research by utilizing existing datasets across many types of vigilance tasks (piloting of aircrafts, driving, air traffic control, etc.).

Bibliography

[1] A. M. Treisman and G. Gelade, “A feature-integration theory of attention,” *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[2] J. Rajsic, D. E. Wilson, and J. Pratt, “Confirmation bias in visual search.” *Journal of experimental psychology: human perception and performance*, vol. 41, no. 5, p. 1353, 2015.

[3] J. P. Gallaher, A. J. Kamrud, and B. J. Borghetti, “Detection and mitigation of inefficient visual searching,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 64, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2020, pp. 47–51.

[4] N. R. Council *et al.*, *Measuring human capabilities: An agenda for basic research on the assessment of individual and group performance potential for military accession*. National Academies Press, 2015.

[5] R. Parasuraman, J. S. Warm, and W. N. Dember, “Vigilance: Taxonomy and utility,” in *Ergonomics and human factors*. Springer, 1987, pp. 11–32.

[6] R. Parasuraman and M. Mouloua, *Automation and human performance: Theory and applications*. Routledge, 2018.

[7] J. S. Warm, R. Parasuraman, and G. Matthews, “Vigilance requires hard mental work and is stressful,” *Human factors*, vol. 50, no. 3, pp. 433–441, 2008.

[8] A. D. Fisk and W. Schneider, “Control and automatic processing during tasks requiring sustained attention: A new approach to vigilance,” *Human factors*, vol. 23, no. 6, pp. 737–750, 1981.

[9] A. Ariga and A. Lleras, “Brief and rare mental “breaks” keep you focused: Deactivation and reactivation of task goals preempt vigilance decrements,” *Cognition*, vol. 118, no. 3, pp. 439–443, 2011.

[10] T. Lan, A. Adami, D. Erdogmus, and M. Pavel, “Estimating cognitive state using EEG signals,” in *2005 13th European Signal Processing Conference*. IEEE, 2005, pp. 1–4.

[11] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, “Deep learning-based electroencephalography analysis: a systematic review,” *Journal of neural engineering*, vol. 16, no. 5, p. 051001, 2019.

[12] A. Craik, Y. He, and J. L. Contreras-Vidal, “Deep learning for electroencephalogram (EEG) classification tasks: a review,” *Journal of neural engineering*, vol. 16, no. 3, p. 031001, 2019.

[13] Z. Hu, Y. Sun, J. Lim, N. Thakor, and A. Bezerianos, “Investigating the correlation between the neural activity and task performance in a psychomotor vigilance test,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* IEEE, 2015, pp. 4725–4728.

[14] D. Wu, J.-T. King, C.-H. Chuang, C.-T. Lin, and T.-P. Jung, “Spatial filtering for EEG-based regression problems in brain–computer interface (BCI),” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 771–781, 2017.

[15] M. M. Walsh, G. Gunzelmann, and J. R. Anderson, “Relationship of p3b single-trial latencies and response times in one, two, and three-stimulus oddball tasks,” *Biological psychology*, vol. 123, pp. 47–61, 2017.

[16] X. Hu and G. Lodewijks, “Detecting fatigue in car drivers and aircraft pilots by using non-invasive measures: The value of differentiation of sleepiness and mental fatigue,” *Journal of safety research*, vol. 72, pp. 173–187, 2020.

[17] A. Y. Kaplan, A. A. Fingelkurts, A. A. Fingelkurts, S. V. Borisov, and B. S. Darkhovsky, “Nonstationary nature of the brain activity as revealed by EEG/MEG: methodological, practical and conceptual challenges,” *Signal processing*, vol. 85, no. 11, pp. 2190–2212, 2005.

[18] M. X. Cohen, *Analyzing neural time series data: theory and practice.* MIT press, 2014.

[19] V. Rasoulzadeh, E. Erkus, T. Yogurt, I. Ulusoy, and S. A. Zergeroğlu, “A comparative stationarity analysis of EEG signals,” *Annals of Operations Research*, vol. 258, no. 1, pp. 133–157, 2017.

[20] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama, “Application of covariate shift adaptation techniques in brain–computer interfaces,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 6, pp. 1318–1324, 2010.

[21] G. Matthews and M. Amelang, “Extraversion, arousal theory and performance: A study of individual differences in the EEG,” *Personality and individual differences*, vol. 14, no. 2, pp. 347–363, 1993.

[22] P. Medrano, E. Nyhus, A. Smolen, T. Curran, and R. S. Ross, “Individual differences in EEG correlates of recognition memory due to DAT polymorphisms,” *Brain and behavior*, vol. 7, no. 12, p. e00870, 2017.

[23] H.-P. Landolt, “Genetic determination of sleep EEG profiles in healthy humans,” in *Progress in brain research.* Elsevier, 2011, vol. 193, pp. 51–61.

[24] D. J. Smit, D. I. Boomsma, H. G. Schnack, H. E. H. Pol, and E. J. de Geus, “Individual differences in EEG spectral power reflect genetic variance in gray and white matter volumes,” *Twin research and human genetics*, vol. 15, no. 3, pp. 384–392, 2012.

[25] S. D. Muthukumaraswamy, R. A. Edden, D. K. Jones, J. B. Swettenham, and K. D. Singh, “Resting GABA concentration predicts peak gamma frequency and fMRI amplitude in response to visual stimulation in humans,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 20, pp. 8356–8361, 2009.

[26] S. D. Muthukumaraswamy and K. D. Singh, “Visual gamma oscillations: the effects of stimulus type, visual field coverage and stimulus motion on MEG and EEG recordings,” *Neuroimage*, vol. 69, pp. 223–230, 2013.

[27] M. X. Cohen, “Hippocampal-prefrontal connectivity predicts midfrontal oscillations and long-term memory performance,” *Current Biology*, vol. 21, no. 22, pp. 1900–1905, 2011.

[28] M. Sugiyama, M. Krauledat, and K.-R. MÃžller, “Covariate shift adaptation by importance weighted cross validation,” *Journal of Machine Learning Research*, vol. 8, no. May, pp. 985–1005, 2007.

[29] H. Raza, “Adaptive learning for modelling non-stationarity in EEG-based brain-computer interfacing,” Ph.D. dissertation, Ulster University, 2016.

[30] M. Sugiyama and M. Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

[31] H. Raza, G. Prasad, and Y. Li, “Dataset shift detection in non-stationary environments using EWMA charts,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2013, pp. 3151–3156.

[32] ——, “EWMA based two-stage dataset shift-detection in non-stationary environments,” in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2013, pp. 625–635.

[33] ——, “Adaptive learning with covariate shift-detection for non-stationary environments,” in *2014 14th UK Workshop on Computational Intelligence (UKCI)*. IEEE, 2014, pp. 1–8.

[34] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[35] J. Rajsic and J. Pratt, “More than a memory: Confirmatory visual search is not caused by remembering a visual feature,” *Acta psychologica*, vol. 180, pp. 169–174, 2017.

[36] J. Rajsic, J. E. T. Taylor, and J. Pratt, “Out of sight, out of mind: Matching bias underlies confirmatory visual search,” *Attention, Perception, & Psychophysics*, vol. 79, no. 2, pp. 498–507, 2017.

[37] R. Parasuraman and D. Davies, "A taxonomic analysis of vigilance performance," in *vigilance*. Springer, 1977, pp. 559–574.

[38] R. G. Hefron, B. J. Borghetti, J. C. Christensen, and C. M. S. Kabban, "Deep long short-term memory structures model temporal dependencies improving cognitive workload estimation," *Pattern Recognition Letters*, vol. 94, pp. 96–104, 2017.

[39] R. Hefron, B. Borghetti, C. Schubert Kabban, J. Christensen, and J. Estepp, "Cross-participant eeg-based assessment of cognitive workload using multi-path convolutional recurrent neural networks," *Sensors*, vol. 18, no. 5, p. 1339, 2018.

[40] T. I. Laine, K. Bauer, J. W. Lanning, C. A. Russell, and G. F. Wilson, "Selection of input features across subjects for classifying crewmember workload using artificial neural networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 32, no. 6, pp. 691–704, 2002.

[41] A. Berger, F. Horst, S. Müller, F. Steinberg, and M. Doppelmayr, "Current state and future prospects of EEG and fNIRS in robot-assisted gait rehabilitation: A brief review," *Frontiers in human neuroscience*, vol. 13, p. 172, 2019.

[42] A. C. Merzagora, M. Izzetoglu, R. Polikar, V. Weisser, B. Onaral, and M. T. Schultheis, "Functional near-infrared spectroscopy and electroencephalography: a multimodal imaging approach," in *International Conference on Foundations of Augmented Cognition*. Springer, 2009, pp. 417–426.

[43] M. Muthalib, A. R. Anwar, S. Perrey, M. Dat, A. Galka, S. Wolff, U. Heute, G. Deuschl, J. Raethjen, and M. Muthuraman, "Multimodal integration of fNIRS, fMRI and EEG neuroimaging." *Clinical Neurophysiology*, vol. 124, no. 10, pp. 2060–2062, 2013.

[44] A. M. Chiarelli, P. Croce, A. Merla, and F. Zappasodi, "Deep learning for hybrid EEG-fNIRS brain–computer interface: application to motor imagery classification," *Journal of neural engineering*, vol. 15, no. 3, p. 036028, 2018.

[45] H. Raza, D. Rathee, S.-M. Zhou, H. Cecotti, and G. Prasad, "Covariate shift estimation based adaptive ensemble learning for handling non-stationarity in motor imagery related EEG-based brain-computer interface," *Neurocomputing*, vol. 343, pp. 154–166, 2019.

[46] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015.

[47] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *International conference on machine learning*, 2016, pp. 2839–2848.

[48] S. Jirayucharoensak, S. Pan-Ngum, and P. Israsena, “EEG-based emotion recognition using deep learning network with principal component based covariate shift adaptation,” *The Scientific World Journal*, vol. 2014, 2014.

[49] S. Sun, C. Zhang, and D. Zhang, “An experimental evaluation of ensemble methods for EEG signal classification,” *Pattern Recognition Letters*, vol. 28, no. 15, pp. 2157–2163, 2007.

[50] H. Raza, H. Cecotti, Y. Li, and G. Prasad, “Learning with covariate shift-detection and adaptation in non-stationary environments: Application to brain-computer interface,” in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[51] H. Raza and S. Samothrakis, “Bagging adversarial neural networks for domain adaptation in non-stationary EEG,” in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–7.

[52] S. R. Liyanage, C. Guan, H. Zhang, K. K. Ang, J. Xu, and T. H. Lee, “Dynamically weighted ensemble classification for non-stationary EEG processing,” *Journal of neural engineering*, vol. 10, no. 3, p. 036007, 2013.

[53] X. Li, Z. Zhao, D. Song, Y. Zhang, J. Pan, L. Wu, J. Huo, C. Niu, and D. Wang, “Latent factor decoding of multi-channel EEG for emotion recognition through autoencoder-like neural networks,” *Frontiers in Neuroscience*, vol. 14, p. 87, 2020.

[54] T. G. Dietterich, “Ensemble methods in machine learning,” in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[55] C. Alippi and M. Roveri, “Just-in-time adaptive classifiers—part ii: Designing the classifier,” *IEEE Transactions on Neural Networks*, vol. 19, no. 12, pp. 2053–2064, 2008.

[56] X. J. Zhu, “Semi-supervised learning literature survey,” University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2005.

[57] K. J. Miller and G. Schalk, “Prediction of finger flexion: 4th brain-computer interface data competition,” *BCI Competition IV*, vol. 1, pp. 1–2, 2008.

[58] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[59] E. R. Kandel, J. H. Schwartz, T. M. Jessell, D. of Biochemistry, M. B. T. Jessell, S. Siegelbaum, and A. Hudspeth, *Principles of neural science*. McGraw-hill New York, 2000, vol. 4.

[60] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning*, 2013, pp. 1310–1318.

[61] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[62] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Gated feedback recurrent neural networks,” in *International conference on machine learning*, 2015, pp. 2067–2075.

[63] ——, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[64] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” in *International conference on machine learning*, 2015, pp. 2342–2350.

[65] C. Olah, “Github,” Aug 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[66] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.

[67] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[68] P. Remy, “Temporal convolutional networks for keras,” https://github.com/philipperemy/keras-tcn, 2020.

[69] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2019.

[70] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot, “Higher order contractive auto-encoder,” in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2011, pp. 645–660.

[71] A. Kamrud, B. Borghetti, and C. Schubert Kabban, “The effects of individual differences, non-stationarity, and the importance of data partitioning decisions for training and testing of EEG cross-participant models,” *Sensors*, vol. 21, no. 9, p. 3225, 2021.

[72] H. Begleiter, “Neurodynamics laboratory,” *State University of New York Health Center at Brooklyn*, 1999.

[73] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[74] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[75] M. Wattenberg, F. Viégas, and I. Johnson, “How to use t-SNE effectively,” *Distill*, 2016. [Online]. Available: http://distill.pub/2016/misread-tsne

[76] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[77] J. Birjandtalab, M. B. Pouyan, and M. Nourani, “An unsupervised subject identification technique using EEG signals,” in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 816–819.

[78] S. Bickel, “Learning under differing training and test distributions,” 2009.

[79] J. Min, P. Wang, and J. Hu, “The original EEG data for driver fatigue detection,” Jul 2017. [Online]. Available: https://figshare.com/articles/dataset/The_original_EEG_data_for_driver_fatigue_detection/5202739/1

[80] B. Rahmani, C. K. Wong, P. Norouzzadeh, J. Bodurka, and B. McKinney, “Dynamical hurst analysis identifies EEG channel differences between PTSD and healthy controls,” *PloS one*, vol. 13, no. 7, p. e0199144, 2018.

[81] B. Roach, “EEG data from basic sensory task in schizophrenia,” 2020. [Online]. Available: https://www.kaggle.com/broach/button-tone-sz

[82] M. Arevalillo-Herráez, M. Cobos, S. Roger, and M. García-Pineda, “Combining inter-subject modeling with a subject-based data transformation to improve affect recognition from EEG signals,” *Sensors*, vol. 19, no. 13, p. 2999, 2019.

[83] R. S. Ellis, *Entropy, large deviations, and statistical mechanics*. Taylor & Francis, 2006, vol. 1431, no. 821.

[84] J. Min, P. Wang, and J. Hu, “Driver fatigue detection through multiple entropy fusion analysis in an EEG-based system,” *PLoS one*, vol. 12, no. 12, p. e0188756, 2017.

[85] K. A. Lee, G. Hicks, and G. Nino-Murcia, “Validity and reliability of a scale to assess fatigue,” *Psychiatry research*, vol. 36, no. 3, pp. 291–298, 1991.

[86] T. Chalder, G. Berelowitz, T. Pawlikowska, L. Watts, S. Wessely, D. Wright, and E. Wallace, “Development of a fatigue scale,” *Journal of psychosomatic research*, vol. 37, no. 2, pp. 147–153, 1993.

[87] H. Yu and B. M. Wilamowski, “Levenberg-marquardt training,” *Industrial electronics handbook*, vol. 5, no. 12, p. 1, 2011.

[88] H. Wang, Y. Li, X. Hu, Y. Yang, Z. Meng, and K.-m. Chang, “Using EEG to improve massive open online courses feedback interaction.” in *AIED Workshops*, 2013.

[89] Z. Ni, A. C. Yuksel, X. Ni, M. I. Mandel, and L. Xie, “Confused or not confused? disentangling brain activity from EEG data using bidirectional LSTM recurrent neural networks,” in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2017, pp. 241–246.

[90] H. Wang, Z. Wu, and E. P. Xing, “Removing confounding factors associated weights in deep neural networks improves the prediction accuracy for healthcare applications,” in *BIOCOMPUTING 2019: Proceedings of the Pacific Symposium*. World Scientific, 2018, pp. 54–65.

[91] L. Ingber, “Statistical mechanics of neocortical interactions: Canonical momenta indicators of electroencephalography,” *Physical Review E*, vol. 55, no. 4, p. 4578, 1997.

[92] X. L. Zhang, H. Begleiter, B. Porjesz, and A. Litke, “Electrophysiological evidence of memory impairment in alcoholic patients,” *Biological Psychiatry*, vol. 42, no. 12, pp. 1157–1171, 1997.

[93] L. Farsi, S. Siuly, E. Kabir, and H. Wang, “Classification of alcoholic EEG signals using a deep learning method,” *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3552–3560, 2020.

[94] J. M. Ford, V. A. Palzes, B. J. Roach, and D. H. Mathalon, “Did i do that? abnormal predictive processes in schizophrenia when button pressing to deliver a tone,” *Schizophrenia bulletin*, vol. 40, no. 4, pp. 804–812, 2014.

[95] R. Buettner, M. Hirschmiller, K. Schlosser, M. Rössle, M. Fernandes, and I. J. Timm, “High-performance exclusion of schizophrenia using a novel machine learning method on EEG data.” in *HealthCom*, 2019, pp. 1–6.

[96] G. Casella, S. Fienberg, and I. Olkin, “An introduction to statistical learning,” 2013.

[97] R. Li, J. S. Johansen, H. Ahmed, T. V. Ilyevsky, R. B. Wilbur, H. M. Bharadwaj, and J. M. Siskind, “The perils and pitfalls of block design for EEG classification experiments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 316–333, 2020.

[98] IEEE Standards Association, “Standards roadmap: Neurotechnologies for brain-machine interfacing,” 2021. [Online]. Available: https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/presentations/ieee-neurotech-for-bmi-standards-roadmap.pdf

[99] R. Chavarriaga, C. Carey, J. L. Contreras-Vidal, Z. Mckinney, and L. Bianchi, “Standardization of neurotechnology for brain-machine interfacing: state of the art and recommendations,” *IEEE Open Journal of Engineering in Medicine and Biology*, 2021.

[100] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," *science*, vol. 185, no. 4157, pp. 1124–1131, 1974.

[101] D. Kahneman and G. Klein, "Conditions for intuitive expertise: a failure to disagree." *American psychologist*, vol. 64, no. 6, p. 515, 2009.

[102] R. S. Nickerson, "Confirmation bias: A ubiquitous phenomenon in many guises," *Review of general psychology*, vol. 2, no. 2, pp. 175–220, 1998.

[103] M. Ashcraft and G. Radvansky, "Cognition 6th edition," 2013.

[104] M. Miyakoshi, "Makoto's preprocessing pipeline," 2020. [Online]. Available: https://sccn.ucsd.edu/wiki/Makoto's_preprocessing_pipeline

[105] O. Ledoit and M. Wolf, "Honey, i shrunk the sample covariance matrix," *The Journal of Portfolio Management*, vol. 30, no. 4, pp. 110–119, 2004.

[106] S. Kumar, A. Sharma, and T. Tsunoda, "Brain wave classification using long short-term memory network based OPTICAL predictor," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[107] A. Smart Eye and S. Gothenburg, "Smart eye pro," 2019.

[108] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of neuroscience methods*, vol. 134, no. 1, pp. 9–21, 2004.

[109] N. Bigdely-Shamlo, T. Mullen, C. Kothe, K.-M. Su, and K. A. Robbins, "The PREP pipeline: standardized preprocessing for large-scale EEG analysis," *Frontiers in neuroinformatics*, vol. 9, p. 16, 2015.

[110] I. Winkler, S. Debener, K.-R. Müller, and M. Tangermann, "On the influence of high-pass filtering on ICA-based artifact reduction in EEG-ERP," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 4101–4105.

[111] T. Mullen, "Cleanline EEGLAB plugin," *San Diego, CA: Neuroimaging Informatics Toolsand Resources Clearinghouse (NITRC)*, 2012.

[112] C. A. E. Kothe and T.-P. Jung, "Artifact removal techniques with signal reconstruction," Apr. 28 2016, uS Patent App. 14/895,440.

[113] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[114] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*, 2018.

[115] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[116] F. Chollet *et al.*, *Deep learning with Python*. Manning New York, 2018, vol. 361.

[117] S. K. Lal and A. Craig, “A critical review of the psychophysiology of driver fatigue,” *Biological psychology*, vol. 55, no. 3, pp. 173–194, 2001.

[118] P. L. Ackerman and R. Kanfer, “Test length and cognitive fatigue: an empirical examination of effects on performance and test-taker reactions.” *Journal of Experimental Psychology: Applied*, vol. 15, no. 2, p. 163, 2009.

[119] N. H. Mackworth, “The breakdown of vigilance during prolonged visual search,” *Quarterly Journal of Experimental Psychology*, vol. 1, no. 1, pp. 6–21, 1948.

[120] I. Sasahara, N. Fujimura, Y. Nozawa, Y. Furuhata, and H. Sato, “The effect of histidine on mental fatigue and cognitive performance in subjects with high fatigue and sleep disruption scores,” *Physiology & behavior*, vol. 147, pp. 238–244, 2015.

[121] K. C. Smolders and Y. A. de Kort, “Bright light and mental fatigue: Effects on alertness, vitality, performance and physiological arousal,” *Journal of environmental psychology*, vol. 39, pp. 77–91, 2014.

[122] Y. Shigihara, M. Tanaka, A. Ishii, S. Tajima, E. Kanai, M. Funakura, and Y. Watanabe, “Two different types of mental fatigue produce different styles of task performance,” *Neurology, Psychiatry and Brain Research*, vol. 19, no. 1, pp. 5–11, 2013.

[123] L. C. Hogan, M. Bell, and R. Olson, “A preliminary investigation of the reinforcement function of signal detections in simulated baggage screening: further support for the vigilance reinforcement hypothesis,” *Journal of Organizational Behavior Management*, vol. 29, no. 1, pp. 6–18, 2009.

[124] A. Craig, Y. Tran, N. Wijesuriya, and H. Nguyen, “Regional brain wave activity changes associated with fatigue,” *Psychophysiology*, vol. 49, no. 4, pp. 574–582, 2012.

[125] C. Zhao, M. Zhao, J. Liu, and C. Zheng, “Electroencephalogram and electrocardiograph assessment of mental fatigue in a driving simulator,” *Accident Analysis & Prevention*, vol. 45, pp. 83–90, 2012.

[126] Z. Gao, X. Wang, Y. Yang, C. Mu, Q. Cai, W. Dang, and S. Zuo, “EEG-based spatio–temporal convolutional neural network for driver fatigue evaluation,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2755–2763, 2019.

[127] Z. Yin and J. Zhang, “Task-generic mental fatigue recognition based on neurophysiological signals and dynamical deep extreme learning machine,” *Neurocomputing*, vol. 283, pp. 266–281, 2018.

[128] A. Haubert, M. Walsh, R. Boyd, M. Morris, M. Wiedbusch, M. Krusmark, and G. Gunzelmann, "Relationship of event-related potentials to the vigilance decrement," *Frontiers in psychology*, vol. 9, p. 237, 2018.

[129] H. Head, "Vigilance," *Brit. Journ.. Psychol*, 1926.

[130] J. E. See, S. R. Howe, J. S. Warm, and W. N. Dember, "Meta-analysis of the sensitivity decrement in vigilance." *Psychological Bulletin*, vol. 117, no. 2, p. 230, 1995.

[131] B. S. Oken, M. C. Salinsky, and S. Elsas, "Vigilance, alertness, or sustained attention: physiological basis and measurement," *Clinical neurophysiology*, vol. 117, no. 9, pp. 1885–1901, 2006.

[132] S. Charbonnier, R. N. Roy, S. Bonnet, and A. Campagne, "EEG index for control operators' mental fatigue monitoring using interactions between brain regions," *Expert Systems with Applications*, vol. 52, pp. 91–98, 2016.

[133] F. Al-Shargie, U. Tariq, H. Mir, H. Alawar, F. Babiloni, and H. Al-Nashash, "Vigilance decrement and enhancement techniques: a review," *Brain sciences*, vol. 9, no. 8, p. 178, 2019.

[134] R. A. Grier, J. S. Warm, W. N. Dember, G. Matthews, T. L. Galinsky, J. L. Szalma, and R. Parasuraman, "The vigilance decrement reflects limitations in effortful attention, not mindlessness," *Human factors*, vol. 45, no. 3, pp. 349–359, 2003.

[135] W. S. Helton and J. S. Warm, "Signal salience and the mindlessness theory of vigilance," *Acta psychologica*, vol. 129, no. 1, pp. 18–25, 2008.

[136] D. R. Thomson, D. Besner, and D. Smilek, "A resource-control account of sustained attention: Evidence from mind-wandering and vigilance paradigms," *Perspectives on psychological science*, vol. 10, no. 1, pp. 82–96, 2015.

[137] I. H. Robertson, T. Manly, J. Andrade, B. T. Baddeley, and J. Yiend, "'Oops!': performance correlates of everyday attentional failures in traumatic brain injured and normal subjects," *Neuropsychologia*, vol. 35, no. 6, pp. 747–758, 1997.

[138] T. Manly, I. H. Robertson, M. Galloway, and K. Hawkins, "The absent mind:: further investigations of sustained attention to response," *Neuropsychologia*, vol. 37, no. 6, pp. 661–670, 1999.

[139] B. Z. Veksler and G. Gunzelmann, "Functional equivalence of sleep loss and time on task effects in sustained attention," *Cognitive science*, vol. 42, no. 2, pp. 600–632, 2018.

[140] D. Gartenberg, B. Z. Veksler, G. Gunzelmann, and J. G. Trafton, "An ACT-R process model of the signal duration phenomenon of vigilance," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2014, pp. 909–913.

[141] P. Desmond, G. Matthews, and J. Bush, "Sustained visual attention during simultaneous and successive vigilance tasks," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 45, no. 18. SAGE Publications Sage CA: Los Angeles, CA, 2001, pp. 1386–1389.

[142] C. Baker, "Signal duration as a factor in vigilance tasks," *Science*, vol. 141, no. 3586, pp. 1196–1197, 1963.

[143] G. Teo and J. L. Szalma, "The effects of task type and source complexity on vigilance performance, workload, and stress," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 55, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2011, pp. 1180–1184.

[144] R. Parasuraman and M. Mouloua, "Interaction of signal discriminability and task type in vigilance decrement," *Perception & Psychophysics*, vol. 41, no. 1, pp. 17–22, 1987.

[145] T. M. Lanzetta, W. N. Dember, J. S. Warm, and D. B. Berch, "Effects of task type and stimulus heterogeneity on the event rate function in sustained attention," *Human Factors*, vol. 29, no. 6, pp. 625–633, 1987.

[146] W. H. Teichner, "The detection of a simple visual signal as a function of time of watch," *Human factors*, vol. 16, no. 4, pp. 339–352, 1974.

[147] R. G. Pachella, "The interpretation of reaction time in information processing research," MICHIGAN UNIV ANN ARBOR HUMAN PERFORMANCE CENTER, Tech. Rep., 1973.

[148] W. A. Wickelgren, "Speed-accuracy tradeoff and information processing dynamics," *Acta psychologica*, vol. 41, no. 1, pp. 67–85, 1977.

[149] J. T. Townsend and F. G. Ashby, "Methods of modeling capacity in simple processing systems," in *Cognitive theory*. Psychology Press, 2014, pp. 211–252.

[150] D. J. Woltz and C. A. Was, "Availability of related long-term memory during and after attention focus in working memory," *Memory & Cognition*, vol. 34, no. 3, pp. 668–684, 2006.

[151] H. R. Liesefeld and M. Janczyk, "Combining speed and accuracy to control for speed-accuracy trade-offs (?)," *Behavior Research Methods*, vol. 51, no. 1, pp. 40–60, 2019.

[152] S. T. Mueller, L. Alam, G. J. Funke, A. Linja, T. Ibne Mamun, and S. L. Smith, "Examining methods for combining speed and accuracy in a go/no-go vigilance task," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 64, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2020, pp. 1202–1206.

[153] G. Gunzelmann, J. B. Gross, K. A. Gluck, and D. F. Dinges, "Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling," *Cognitive science*, vol. 33, no. 5, pp. 880–910, 2009.

[154] J. A. Caldwell, K. K. Hall, and B. S. Erickson, "EEG data collected from helicopter pilots in flight are sufficiently sensitive to detect increased fatigue from sleep deprivation," *The International Journal of Aviation Psychology*, vol. 12, no. 1, pp. 19–32, 2002.

[155] L. L. Di Stasi, C. Diaz-Piedra, J. Suárez, M. B. McCamy, S. Martinez-Conde, J. Roca-Dorda, and A. Catena, "Task complexity modulates pilot electroencephalographic activity during real flights," *Psychophysiology*, vol. 52, no. 7, pp. 951–956, 2015.

[156] L. Cao, J. Li, Y. Sun, H. Zhu, and C. Yan, "EEG-based vigilance analysis by using fisher score and PCA algorithm," in *2010 IEEE International Conference on Progress in Informatics and Computing*, vol. 1. IEEE, 2010, pp. 175–179.

[157] T.-P. Jung, S. Makeig, M. Stensmo, and T. J. Sejnowski, "Estimating alertness from the EEG power spectrum," *IEEE transactions on biomedical engineering*, vol. 44, no. 1, pp. 60–69, 1997.

[158] C.-T. Lin, R.-C. Wu, S.-F. Liang, W.-H. Chao, Y.-J. Chen, and T.-P. Jung, "EEG-based drowsiness estimation for safety driving using independent component analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 12, pp. 2726–2738, 2005.

[159] H. Ji, J. Li, L. Cao, and D. Wang, "A EEG-based brain computer interface system towards applicable vigilance monitoring," in *Foundations of Intelligent Systems*. Springer, 2011, pp. 743–749.

[160] G. Borghini, G. Vecchiato, J. Toppi, L. Astolfi, A. Maglione, R. Isabella, C. Caltagirone, W. Kong, D. Wei, Z. Zhou *et al.*, "Assessment of mental fatigue during car driving by using high resolution eeg activity and neurophysiologic indices," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012, pp. 6442–6445.

[161] C. Zhang, C.-X. Zheng, and X.-L. Yu, "Automatic recognition of cognitive fatigue from physiological indices by using wavelet packet transform and kernel learning algorithms," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4664–4671, 2009.

[162] R. Rosipal, B. Peters, G. Kecklund, T. Åkerstedt, G. Gruber, M. Woertz, P. Anderer, and G. Dorffner, "EEG-based drivers' drowsiness monitoring using a hierarchical gaussian mixture model," in *International Conference on Foundations of Augmented Cognition*. Springer, 2007, pp. 294–303.

[163] E. M. Hitchcock, J. S. Warm, G. Matthews, W. N. Dember, P. K. Shear, L. D. Tripp, D. W. Mayleben, and R. Parasuraman, "Automation cueing modulates cerebral blood flow and vigilance in a simulated air traffic control task," *Theoretical Issues in Ergonomics Science*, vol. 4, no. 1-2, pp. 89–112, 2003.

[164] D. F. Dinges and J. W. Powell, "Microcomputer analyses of performance on a portable, simple visual RT task during sustained operations," *Behavior research methods, instruments, & computers*, vol. 17, no. 6, pp. 652–655, 1985.

[165] M. D. Comerchero and J. Polich, "P3a and p3b from typical auditory and visual stimuli," *Clinical neurophysiology*, vol. 110, no. 1, pp. 24–30, 1999.

[166] D. H. Brainard, "The psychophysics toolbox," *Spatial vision*, vol. 10, no. 4, pp. 433–436, 1997.

[167] E. M. Hitchcock, W. N. Dember, J. S. Warm, B. W. Moroney, and J. E. See, "Effects of cueing and knowledge of results on workload and boredom in sustained attention," *Human factors*, vol. 41, no. 3, pp. 365–372, 1999.

[168] K. S. Prabhudesai, L. M. Collins, and B. O. Mainsah, "Automated feature learning using deep convolutional auto-encoder neural network for clustering electroencephalograms into sleep stages," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2019, pp. 937–940.

[169] D. Ayata, Y. Yaslan, and M. Kamasak, "Multi channel brain eeg signals based emotional arousal classification with unsupervised feature learning using autoencoders," in *2017 25th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2017, pp. 1–4.

[170] M. Thill, W. Konen, and T. Bäck, "Time series encodings with temporal convolutional networks," in *International Conference on Bioinspired Methods and Their Applications*. Springer, 2020, pp. 161–173.

[171] R. K. Minas, R. F. Potter, A. R. Dennis, V. Bartelt, and S. Bae, "Putting on the thinking cap: using neurois to understand information processing biases in virtual teams," *Journal of Management Information Systems*, vol. 30, no. 4, pp. 49–82, 2014.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*
16-09-2021

2. REPORT TYPE
Doctoral Dissertation

3. DATES COVERED *(From - To)*
Sept 2018 - Sept 2021

4. TITLE AND SUBTITLE
Advancing Proper Dataset Partitioning and
Classification of Visual Search and the Vigilance
Decrement Using EEG Deep Learning Algorithms

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F4FGA08305J006

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Kamrud, Alexander J., Maj

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT-ENG-DS-21-S-011

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Dr. James Lawton
875 N. Randolph, Ste. 325
Arlington, VA 22203
james.lawton.1@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR/RTA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A:
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

EEG classification of visual search and vigilance tasks has vast potential in its benefits. This dissertation investigates EEG models built to utilize any individual's EEG signals, i.e. cross-participant models, in the areas of: dataset partitioning for proper training and validation; classification of the efficiency of an operator's search; and classification of whether an operator is in a vigilance decrement. First, the necessity of proper dataset partitioning for EEG cross-participant models is demonstrated with empirical results showing that improper partitioning of datasets can lead to error rates underestimated between 35% and 3900%. Next, the results of a EEG visual search experiment are presented, with techniques tested to mitigate inefficient search, with efficient search found to be faster and more accurate, and the nudge and hint techniques found to be effective in mitigation of inefficient search. Lastly, EEG cross-participant models are presented which classified whether or not a participant was in a vigilance decrement during an unseen vigilance type task, with the best model having accuracy significantly greater than random chance with 64% validation accuracy.

15. SUBJECT TERMS
Deep Learning; EEG; cross-participant; inter-participant variability; vigilance decrement; visual search

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Brett J. Borghetti, AFIT/ENG
U	U	U	UU	193	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 x4612 brett.borghetti@afit.edu