Journal of Neural Engineering

PAPER

Unsupervised fNIRS feature extraction with CAE and ESN autoencoder for driver cognitive load classification

Ruixue Liu[1], **Bryan Reimer**[2], **Siyang Song**[3], **Bruce Mehler**[2] **and Erin Solovey**[1]

[1] Worcester Polytechnic Institute, P.O. Box 1212, Worcester, MA 016091, United States of America
[2] Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, United States of America
[3] University of Nottingham, Nottingham NG7 2RD, United Kingdom

E-mail: rliu2@wpi.edu

RECEIVED
5 August 2020

REVISED
22 November 2020

ACCEPTED FOR PUBLICATION
11 December 2020

PUBLISHED
8 March 2021

Keywords: driver cognitive load, functional near-infrared spectroscopy, fNIRS, convolutional autoencoder, echo state network

Abstract

Objective. Understanding the cognitive load of drivers is crucial for road safety. Brain sensing has the potential to provide an objective measure of driver cognitive load. We aim to develop an advanced machine learning framework for classifying driver cognitive load using functional near-infrared spectroscopy (fNIRS). *Approach*. We conducted a study using fNIRS in a driving simulator with the *N*-back task used as a secondary task to impart structured cognitive load on drivers. To classify different driver cognitive load levels, we examined the application of convolutional autoencoder (CAE) and Echo State Network (ESN) autoencoder for extracting features from fNIRS. *Main results*. By using CAE, the accuracies for classifying two and four levels of driver cognitive load with the 30 s window were 73.25% and 47.21%, respectively. The proposed ESN autoencoder achieved state-of-art classification results for group-level models without window selection, with accuracies of 80.61% and 52.45% for classifying two and four levels of driver cognitive load. *Significance*. This work builds a foundation for using fNIRS to measure driver cognitive load in real-world applications. Also, the results suggest that the proposed ESN autoencoder can effectively extract temporal information from fNIRS data and can be useful for other fNIRS data classification tasks.

1. Introduction

Road traffic accidents have claimed more than 1.35 million deaths each year around the world, with around 50 million people injured [1]. Meanwhile, according to a report from the National Highway Traffic Safety Administration (NHTSA), 36 560 lives were lost on United States roads in 2018, with around 400 000 people injured. This includes an estimated 2841 people killed by distracted drivers [2]. Distractions are often caused by a mix of auditory, vocal, visual, manual, and cognitive demands (e.g. [3]). As a complex and intensive activity, driving requires a driver to focus on not only the car, but also factors such as nearby vehicles, traffic signs, pedestrians, and lights. At the same time, the increased number of mobile devices and advanced in-car communication and infotainment systems are imposing different levels of cognitive load on the driver [4]. Research has shown both under-load and overload of driver's cognitive resources are related to road accidents [5]. When drivers are under-loaded, they can experience fatigue or drowsiness, and this may lead to reduced alertness and lowered attention. When drivers are overloaded, drivers are under stress and this may lead to insufficient attention and inadequate capacity and time for information processing [6, 7]. As a result, understanding the cognitive load of drivers has the potential to contribute to avoiding future accidents and hazards on the road [8].

Previous research has used several approaches to assess drivers' cognitive load, which can be divided into three main categories: subjective measures, performance measures, and physiological measures [8, 9]. Each of these approaches has both advantages and disadvantages [7]. Subjective measures can provide strong periodic indicators of load but require interrupting the task flow with probes or recalling events post hoc. Continuous objective measures, such as those that are physiological-based, can provide greater sensitivity to the time course changes in cognitive load during driving [10]. As such, various

types of physiological data have been collected for driver cognitive load studies, e.g. electroencephalogram (EEG) data [11, 12], heart rate [8, 10, 13], skin conductance [8, 10, 14] and eye movements [15].

Functional near-infrared spectroscopy (fNIRS) is a brain imaging technique, which has been shown to be useful for evaluating human cognitive load and working memory demand under various circumstances [16–20]. fNIRS emits near-infrared light into the brain. By measuring the light returned to the surface, the amount of oxygenated hemoglobin (HbO) and deoxygenated hemoglobin (HbR) can be calculated, which can indicate hemodynamic activity associated with brain activation in that area. As a portable and non-invasive technique, it has the potential to be used for driver cognitive load estimation [21, 22].

Most previous studies in this direction utilized traditional signal processing methods to analyze fNIRS signals without using state-of-the-art machine learning algorithms [21–23]. fNIRS data are high dimensional and high volume time series data. However, these studies either used a small segment or simple statistics to describe fNIRS data. The former approach requires the selection of small windows from the whole series and ignores global temporal dynamics, while statistics-based features lose both amplitude and temporal details. Motivated by this, we aim to explore advanced feature extraction methods for fNIRS data, to improve the classification accuracy for differentiating different levels of driver cognitive load using fNIRS.

Recent advances in deep learning allow task-specific features to be deep learned from various sources such as images, languages, and brain data [24–26], which are usually more powerful than handcrafted ones. The main idea of this paper is to learn high-level features using autoencoders, which are trained to reconstruct the original data in an unsupervised manner. In general, autoencoders can be divided into two categories, autoencoders with feed-forward neural networks and autoencoders with recurrent neural networks (RNNs). Feed-forward neural networks, such as the convolutional autoencoder (CAE), have shown powerful feature abstraction capability for extracting spatial and temporal dependencies from brain data [20, 27, 28]. Autoencoders building on RNNs, such as Echo State Networks (ESN), have shown to be very effective in extracting temporal patterns from multivariate time series data [29–33]. Research to date has not explored the application of RNN-based models for fNIRS feature extraction. In this work, we set out to employ both feed-forward neural networks and RNNs-based architectures for fNIRS feature extraction. Particularly, we employ the CAE and ESN autoencoder and compare their results on classifying fNIRS as an estimator of driver cognitive load.

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

The main contributions of this paper can be summarized as:

- We propose a machine learning framework for driver cognitive load classification using fNIRS data.
- We describe the application CAE and ESN autoencoder for unsupervised feature extraction from fNIRS data.
- We show that the proposed ESN autoencoder yields state-of-the-art classification accuracy for group-level models without window selection for fNIRS-based driver cognitive load classification.

2. Background

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

2.1. Driver cognitive load assessment

2.1.1. Secondary task paradigms during driving

As a driver's cognitive demand often includes competition between the driving task and non-driving related activities, driver cognitive load studies often utilize controlled and repeatable secondary task paradigms. Recent studies have adopted many types of secondary tasks and collected a variety of psychophysiological data for driver cognitive load analysis. Tsunashima *et al* used mental calculation tasks, which consisted of a low-demand task (one digit addition), a medium-demand task (one digit addition of three numbers) and high-demand task (subtraction and division with a decimal fraction), and evaluated the effectiveness of fNIRS for measuring differences in driver cognitive load [22]. In addition to steering and maintaining a set speed in a driving simulator, during secondary task periods designed to model increased

cognitive load, Wu *et al* asked participants to press one of the buttons on a panel when prompted by a command on the display screen [34]. Zhang *et al* employed a verbal task and a spatial-imagery task as secondary tasks. The verbal task required drivers to name words starting with a designated letter while the spatial-imagery task asked them to respond letters from A to Z under five rules that they predefined. During the task, eye tracker and head tracker were applied to obtain corresponding physiological data [35]. Putze *et al* asked participants to perform a visual search task and a mathematical cognitive task, while multiple biosignal streams (skin conductance, pulse, respiration, EEG) were collected [12].

Besides the aforementioned secondary tasks, recent studies have frequently adopted a type of secondary task called an *n*-back. A version of the *n*-back task was developed by the MIT AgeLab [9, 36] for the context of driving and later incorporated into ISO 14198 [37] as a standardized method to calibrate or otherwise characterize reference levels of demand placed upon a driver. In the standardized presentation of this form of the *n*-back, a series of single-digit numbers are presented via audio. Participants are asked to respond with the corresponding number *n* positions before the current number. As a result, the parameter *n* can easily adjust the level of working memory load. For example, using the *n*-back task as the secondary task, Solovey *et al* analyzed heart rate and skin conductance data from participants who were driving on the highway [8]. Li *et al* collected fNIRS and heart rate data while implementing an alternate *n*-back task in a simulated driving experiment [21]. The latter is an example of a study using a form of *n*-back task that presents a series of single letters. As each letter appears, the participant responds if the new letter matches a letter presented *n*-places back in the sequence (see Owen *et al* [38] for a review). This matching form is arguably more difficult for a given value of *n* [36].

2.1.2. Driver cognitive load analysis

To analyze driver cognitive load using physiological data, researchers have proposed various data analysis methods. Tsunashima *et al* proposed a signal processing method based on multi-resolution analysis (MRA) using a discrete wavelet transform. The results on nine participants suggested that fNIRS data were effective for driver cognitive load evaluation [22]. However, they only conducted statistical analysis in this work, and did not apply machine learning. Wu *et al* proposed a queuing network based on the theory of human performance and neuroscience, and explored the cognitive characteristics of drivers' cognitive load caused by their actions with the vehicle information system [34]. Kim *et al* extracted EEG variation rates in five different driving situations, including left and right-turn, rapid-acceleration, rapid-deceleration, and lane-change [11].

In recent years, due to its success in classification tasks, machine learning has become a popular tool for driver cognitive load classification. Yang *et al* applied SVM and extreme learning machine (ELM) as the classifiers for eye gaze data, and the results show that the ELM-based method achieved better performance, with an accuracy of 76.4% for classifying high driver mental cognitive load from low driver mental cognitive load [39]. Solovey *et al* evaluated different machine learning classifiers for driver cognitive load by using heart rate data. They achieved a high accuracy of 89% for classifying consecutive 2-back elevated periods from normal driving, when using logistic regression with window selection [8]. Fridman *et al* [40] considered classification using 3D convolutional neural networks leveraging visual-only attributes alone to achieve 86% accuracy over a 3-class problem. Le *et al* trained and tested multiple classifiers for classifying driver cognitive load using fNIRS. They show that the decision trees achieved the best results with an accuracy of 82% for classifying different cognitive load elevated by the *n*-back task during driving. However, it is unclear which tasks and time window their classification was based on [23].

Results from previous work suggest that driving cognitive load is predictable by machine learning techniques using visual behavior and physiological data. However, researchers also pointed out that other factors rather than cognitive load, such as physical exertion and emotional state, can also influence physiological signals, which could result in conflicting or unreliable results [41]. fNIRS measures changes in cerebral hemodynamic activity and can be used to infer information on drivers' underlying cognitive activity directly. Moreover, it is safe, portable, easy to use, and quick to set up—characteristics that show promise for use in real-world settings. As such, fNIRS could provide an alternative for measuring driver cognitive load levels objectively. However, an fNIRS-based system using state-of-the-art using machine learning techniques for driver cognitive load classification is not fully explored.

2.2. fNIRS feature extraction

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

2.2.1. Hand-crafted features vs. deep learned features

Before CNN-based methods became the superior approach for feature extraction, the hand-crafted feature approach was used in most previous work. As

fNIRS data are time-series data, statistics obtained by specific time windows were often calculated as features. Aghajani *et al* classified different cognitive load levels elevated by the *n*-back task (*n* from 0 to 3), using the calculated slope, standard deviation, skewness, and kurtosis of each HbO and HbR signal, and the zero lagged correlation between HbO and HbR as features. These features were then selected based on their sensitivity to the changes in cognitive load. By using SVM and the moving window method, they achieved a mean accuracy of 74.8% for binary classification [42]. Similarly, Liu *et al* extracted the average HbO and HbR amplitude changes as features for classifying cognitive load elevated by the *n*-back task ($n = 0, 2, 3$). By using LDA, they achieved a mean accuracy of 53.9% for three-class classification [43].

Besides using statistical features, regression techniques were also employed to extract features from fNIRS data. In the work of Herff *et al*, features were extracted by fitting the slope of a straight line to the data in a specific window using linear regression during the *n*-back task. Their results show that classifying 3-back, 2-back, 1-back against a relaxed state achieved an accuracy of 81%, 80%, and 72%, respectively, while the accuracy for four-class classification is 45% [18].

With the advances in deep learning, more recent work has investigated using deep learning methods to automatically extract features from fNIRS data. For example, Trakoolwilaiwan *et al*, utilized four different CNNs to extract fNIRS features. The results show that CNNs achieved higher accuracy than the combination of SVM/ANN and hand-crafted features (mean, variance, kurtosis, skewness, peak, slope from HbO and HbR) [28]. Similarly, combined with the moving window method, Saadati *et al* showed that the CNN approach can improve the accuracy for cognitive load classification using fNIRS data, with an average accuracy of 82% [44].

These studies have demonstrated the advantages of advanced machine learning methods for automatic fNIRS feature extraction. However, challenges remain, including the fact that brain datasets are usually small due to the costly and time-consuming data collection process. At the same time, deep learning techniques require a large number of training data to achieve satisfactory results [45]. Also, since fNIRS data are time-series data, researchers need to take the spatial and temporal dynamics of fNIRS data into consideration when applying these models. In the next section, we outline these considerations and possible approaches.

2.2.2. Considerations

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

2.2.2.1. Sample window selection

In previous work using hand-crafted features as well as CNN-based methods for fNIRS-based cognitive load classification, window selection methods were utilized to carefully pick a small segment of a fixed size from the original data as the input. While this method might yield better classification results, it ignores the global temporal information and could result in overly optimistic classification results for real-world applications. Moreover, research has shown that due to the latency of the underlying physiological processes, fNIRS cognitive load classification may require a minimum window length of 10 s [18, 46]. Some previous work has not met this requirement which could lead to unreliable results. For example, Saadati *et al* used fNIRS data from a 3 s window to build CNN models [44]. Even though they achieved an accuracy of 89% for classifying cognitive load tasks, continuous time-windows from a single trial were used to form multiple samples in their work. This violates a key assumption behind machine leaning techniques that samples are independent and make their results unreliable.

In this work, we will regard each complete trial (30 s without window selection) as one sample for classification.

2.2.2.2. Individual vs. group models

Most previous work builds individual models for fNIRS-based cognitive load classification. However, research has shown that due to the small dataset of an individual participant and the high feature space of brain data, building individual models could lead to overfitting and achieving overly-optimistic results [47]. Therefore, researchers have shown the need for building group models (across participants) for fNIRS data classification [48, 49], which can enable researchers to get a larger dataset for model training and achieve more reliable results, as well as reduce the time for collecting brain data from a particular individual. However, due to inter-subject variability in hemodynamic responses, it is difficult to build robust models across participants based on fNIRS data [50–53].

There are only a few studies that have investigated building group models for fNIRS-based cognitive load classification. Putze *et al* implemented the *n*-back task in a virtual environment, and extracted the signal mean for all HbO and HbR channels, as well as the resulting slope and coefficient of each channel through linear regression as features. By pooling the data of all participants together and using shrinkage LDA as the classifier, they achieved a mean accuracy of 66% for classifying the 3-back period from the 1-back period, a mean accuracy of 64% for classifying the 2-back period from the 1-back period, and a mean accuracy of 42% for three-classes classification (1-, 2-, or 3-back) [46]. Liu *et al* also investigated fNIRS-based cognitive load classification

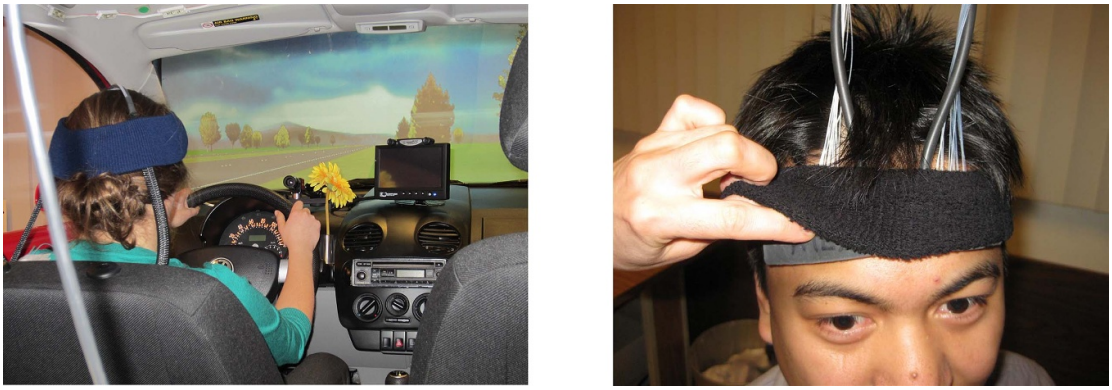

Figure 1. Driving simulation environment (left). The participants sit in the car and are instrumented with fNIRS (right). The screen in the front presents the simulated driving environment.

accuracy by learning from the data of other participants. They extracted the average HbO and HbR amplitude change between different windows from *n*-back tasks, and achieved a mean accuracy of 53.9% for three classes classification (0-, 2-, or 3-back) [43].

From these studies, we can see that while it is beneficial to build group-level models using fNIRS data from a complete trial without window selection, it is difficult to achieve high accuracy for cognitive load classification. Thus, it would be valuable to research more advanced machine learning methods to extract temporal dynamics from fNIRS data without window selection and enable higher performance for group-level models. Considering the relatively small sample sizes of most fNIRS datasets, it could be difficult for the CNN-based method to fully extract temporal information from the data without overfitting [28]. Therefore, in this work, in addition to CNN-based methods (convolutional autoencoder), we also investigate the application of ESN autoencoder for extracting temporal patterns from fNIRS data.

3. Data collection

The goal of our study is to build a dataset of fNIRS data associated with different levels of working memory demands that come from secondary tasks during driving. While there is a wide range of tasks that a driver may perform, we use a variant of the *n*-back task as the secondary task, which has established capacity for eliciting scaled levels of working memory demand [8]. This task serves as a standardized [37] structured proxy for cognitively loading auditory-verbal working memory tasks that a driver may perform. The study was approved by the relevant institutional review board and informed consent was obtained for all participants.

3.1. Driving simulator

Our study was conducted in a driving simulator equipped with fNIRS. The driving simulator consisted of a fixed-base, full-cab Volkswagen New Beetle in front of an 8×8 ft projection screen (figure 1) with established validity for assessing changes in cognitive demand using the *n*-back [14] and visual manual based tasks [54]. Participants had an approximately 40-degree view of a virtual environment at a resolution of 1024×768 pixels. Graphical updates to the virtual world were computed by using Systems Technology Inc. STISIM Drive and STISIM Open Module based upon a driver's interaction with the wheel, brake, and accelerator. Additional feedback to the driver was provided through the wheel's force feedback system and auditory cues. The time-based triggering of visual and auditory stimuli was supported by custom data acquisition software and used to present prerecorded instructions for the cognitive task.

3.2. fNIRS recording

The fNIRS data were acquired using was a multichannel frequency domain Imagent from ISS Inc. Two probes were placed on the forehead to measure the two hemispheres of the anterior prefrontal cortex (figure 1). Each source emitted two near-infrared wavelengths (690 nm and 830 nm) to detect and differentiate between oxygenated and deoxygenated hemoglobin. Each source corresponds to four detectors, with the source-detector distances being 1.5, 2, 2.5, and 3 cm. The sampling rate was 11.8 Hz. The sensors were kept in place using headbands, which can also reduce light interference.

3.3. Driving task and secondary task

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

While driving, an auditory presentation—verbal response *n*-back task was employed to impose additional cognitive load while driving [36, 37]. In each 30-second task block, a series of single digits (0–9) were presented in random order (one at a time) at

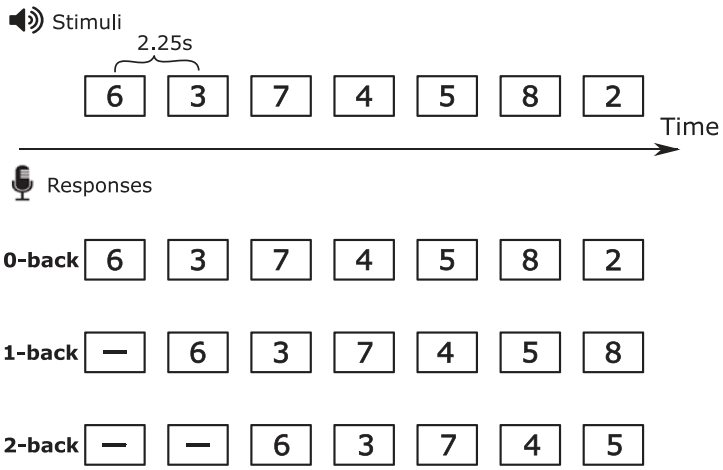

Figure 2. Example task block of auditory stimuli and the appropriate verbal responses for a 0-back task, a 1-back task, and a 2-back task.

2.25 s second intervals. As each new digit was presented, participants were to say out loud the digit *n* items back in the current sequence—the difficulty of the task increases as *n* increases. Three levels of difficulty were employed to present drivers with a low, moderate, and high level of secondary cognitive load. At the lowest cognitive load level (0-back), participants simply repeat each number as it is presented. At the moderate level (1-back), participants were required to respond with the number one item back in the sequence. In the most difficult level (2-back), participants responded with the number two item back in the sequence. Figure 2 describes an example set for the 0-back, 1-back and 2-back task.

3.4. Participants

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

3.5. Design and procedure

Participants were given instructions on how to complete the *n*-back task and practiced the task following training standards detailed in appendix A of [36] prior to entering the simulator. During the experiment, blocks were formed with a random ordering of each with three load levels (0-back, 1-back and 2-back), a 30-second period in which participants were asked to 'just drive,' (which we refer to as the *single-task driving* task) and a *blank-back* [55] where digits of the *n*-back were played with participants instructed to listen but not to respond. The blank-back condition is not considered in this analysis. Participants completed three blocks separated by a 90 s cool down.

4. Dataset curation

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

4.1. Behavioral data

We analyzed the participants' performance during the *n*-back conditions. Participants performed well on the secondary task, with an average accuracy of 100% on the 0-back task, 98.72% on the 1-back task, and 96.44% on the 2-back task. A one-way ANOVA shows significant differences between the three *n*-back levels in the number of errors ($F = 6.85; p < 0.001$). Furthermore, Tukey's post hoc tests showed that participants made significantly more errors during the 2-back task than the 0-back task ($p < 0.005$).

4.2. General dataset description

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

4.3. Dataset Preprocessing

Since signals measured by fNIRS may suffer from biological and technical artifacts, pre-processing is usually employed to enhance signal quality [56]. Following typical preprocessing techniques [57], we used a band-pass filter with a high pass value of 0.02 Hz and a low pass value of 0.5 Hz to remove the physiological noise (e.g. heart rate, respiration) and the instrumental noise. Raw light intensity data was then converted to HbO and HbR values using the Modified Beer–Lambert Law. Then, the correlation-based signal improvement (CBSI) is introduced to reduce motion artifacts. It has been shown that the CBSI method can effectively remove large spikes brought

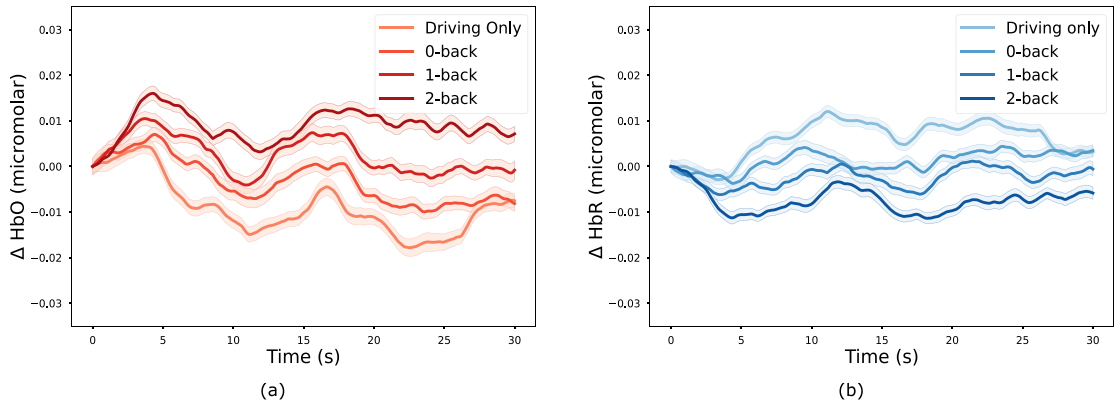

Figure 3. Variation of the changes in HbO and HbR concentration for different conditions. The figures show the mean (averaged across all channels and all individuals) and standard error over each condition. Shaded areas represent the standard error of the mean for each condition.

by head movements as well as enhance signal quality and spatial specificity [58]. All preprocessing was completed in MATLAB using HomER [59].

4.4. Dataset overview

For an overview of the dataset, we calculated the folded average of HbO and HbR change across all participants for each condition. Specifically, we calculated the changes in HbO and HbR by subtracting the corresponding value of the starting point for each trial. Figure 3 shows the block averages of changes in HbO (red) and HbR (blue) for all participants across all channels and all *n*-back conditions. From figure 3(a), we can see that for all conditions, at the beginning of each trial, following neural activation, there is an increase in HbO, which is followed by a decrease in HbO due to the metabolic consumption of oxygen. Moreover, it is clear that the peak value of HbO increases as the difficulty of the task increase. The peak value of HbO is higher in the 1-back condition than 0-back condition and driving only, with the highest value during the 2-back condition. From figure 3(b), similarly, we can see that there is a decrease in HbR at the beginning of each trial, and followed by an increase. Also, The value of HbR is lower in the 1-back condition than 0-back condition and driving only, with the lowest value during the 2-back condition. Moreover, we tested the effect of *n*-back condition and channels using two-way repeated measures ANOVA and determined the main effects using Tukey's post hoc tests. we calculated the mean values for the driving only condition and three *n*-back conditions (0-back, 1-back, 2-back). The mean HbO and HbR values were then analyzed by a 4(condition)× 8(channel) repeated measures ANOVAs. Both the *n*-back condition and channels showed a significant effect on HbO and HbR ($p < 0.001$), while the interaction effect was not statistically significant. Furthermore, post hoc analyses showed that the 2-back task elicited higher HbO increases than the 0-back and the driving only condition ($p < 0.01$). our results are consistent with prior research and suggest heterogeneous activation at the prefrontal area as the difficulty of the task increase [18, 21, 42, 60]. Furthermore, this lays the foundation for our feature extraction and classification techniques.

5. Feature extraction methods

We investigate the application of convolutional autoencoders (CAE) and ESN autoencoder for learning useful representations from fNIRS data. The learned features can then be used as the input for classifiers.

5.1. Input

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

5.2. Convolutional autoencoders (CAE)

An autoencoder neural network is an unsupervised learning algorithm that aims to minimize reconstruction error between the input data and the output data, and is often used for pre-training neural networks [61]. Autoencoders consist of three main parts: the encoder, the bottleneck, and the decoder. The encoder learns how to compress the input data into a low-dimensional representation. The bottleneck is the layer containing the compressed representation of the data. The decoder part learns how to reconstruct the compressed data to be as close to the

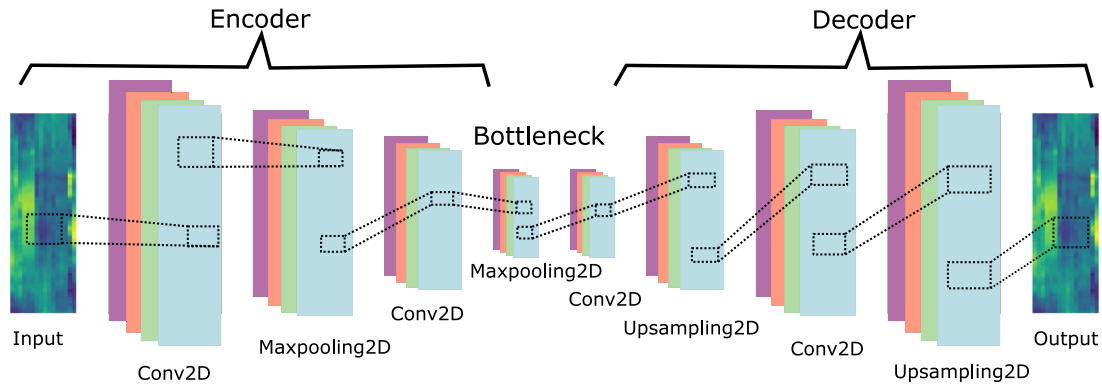

Figure 4. The architecture of convolutional autoencoders, which include the encoder, the bottleneck, and the decoder. After unspervised training, the bottleneck layer becomes the learned features for the input.

original input as possible. By minimizing the reconstruction loss through backpropagation, the compressed representation of the input becomes learned features that contain meaningful information of the input and are useful for future tasks. CAE uses convolutional layers in the encoder and decoder, which inherit the powerful feature abstraction ability of traditional CNNs and have been widely applied for extracting spatial and temporal dependencies from data. Particularly, it can preserve spatial locality by receptive field and parameter sharing. Additionally, convolutional layers can be followed by pooling layers for downsampling in the encoder part, while convolutional layers in the decoder are followed by unpooling layers for upsampling. Figure 4 shows the overview of applying CAE for feature extraction.

Specifically, in this work, to fully capture the spatial information contained by fNIRS signals collected by different channels and the time-series behavior of fNIRS data, fNIRS data was constructed as a set of 2D images, with the length of the image equal to the number of samples in the time window, and the width of image equal to the number of channels. For a given multi-channel fNIRS data input matrix X, and a set of n convolutional filters $\{F_1^{(1)}, \ldots, F_N^{(1)}\}$, the encoder computes:

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

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

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

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

$$L_\varepsilon(X, \tilde{X}) = \tfrac{1}{2}||X - \tilde{X}||^2. \quad (3)$$

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

Considering that the architecture of CAE can affect the resulting performance, we determine the best architecture of CAE for classifying driver cognitive load using fNIRS by investigating the effect of filter sizes, as well as depth and width on the classification accuracy.

5.3. Echo state network (ESN) autoencoder

The Echo State Network (ESN) is a family of recurrent neural network models with a strong architectural simplification. The connectivity and weights of hidden neurons in the recurrent neural network (called 'reservoir') are kept fixed and randomly assigned. Only output weights are learned during training so that the network can produce specific temporal patterns. As such, ESN has an unrivaled training speed compared to other recurrent neural networks. Previous work has shown that ESNs can achieve excellent performance in many fields, and are an efficient solution for multivariate time-series classification [62–65].

To improve classification accuracy by learning more powerful representations from the sequence of reservoir states, Chen *et al* proposed a 'model space' feature extraction approach by training a model for one-step-ahead prediction of the inputs, and then using the model parameters as features for classification. This approach has been successfully applied for multivariate time series classification and unsupervised EEG feature extraction [29, 31, 32]. Moreover, Bianchi *et al* proposed a 'reservoir model space' feature extraction approach, which consists of parameters from a model trained for one-step-ahead prediction of the future reservoir state, instead of the input. Their results show this approach can achieve superior classification accuracy on many multivariate time series datasets when comparing to state-of-the-art recurrent networks and time series kernels [66].

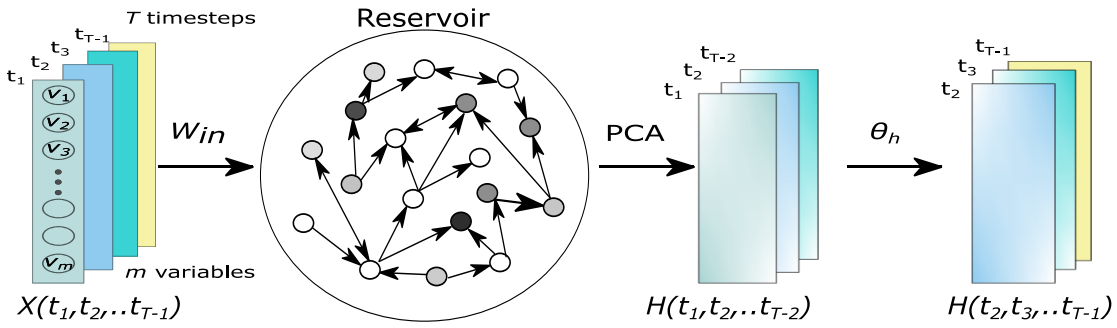

Figure 5. The overview of using Echo State Network autoencoder for fNIRS feature extraction in the reservoir model space. After unsupervised training, the learned parameters θ_h become the learned features for the input.

Therefore, in this work, we investigate the 'reservoir model space' approaches for fNIRS data feature extraction. Figure 5 shows the overview of using this approach for feature extraction. Specifically, we consider classification of fNIRS data consisting of M channels and observed for T time steps. The observation at time t is denoted as $x(t) \in \mathbb{R}^M$. We represent the multi-channel fNIRS data as a $T \times M$ matrix: $X = [x(1), \ldots, x(T)]^T$. For a echo state network with input weights W_{in} and recurrent connections W_r (randomly generated and left untrained), the state-update equation is:

$$h(t) = f(W_{in}x(t) + W_r h(t-1)), \tag{4}$$

where $h(t)$ is the reservoir state at time t. $f(.)$ is a non-linear activation function.

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

$$h(t+1) = V_h h(t) + v_h. \tag{5}$$

The parameters $\theta_h = \{V_h, v_h\}$ are learned by minimizing a ridge regression loss function. These parameters then becomes the representations for the input and used for classification. Also, since dimensionality reduction applied on top of $h(t)$ can enhance the representations' generalization capability, we applied principle component analysis (PCA) on $h(t)$ [66].

The performance of ESN can be influenced by the number of hidden neurons and the internal connectivity of the reservoir [32]. Therefore, in this work, we determine the optimal parameters for ESN for classifying driver cognitive load using fNIRS by investigating the effect of the number of hidden neurons and the internal connectivity of the reservoir on the classification accuracy.

6. Classification methods

After extracting features from fNIRS data, a classifier is needed to map the features to classes.

6.1. Convolutional neural networks (CNNs)

For features extracted using the CAE, research has shown that learned weights of the encoder can be used to initialize CNNs' convolution layers, which can yield a better classification performance [67]. Therefore, in this work, we chose to use a CNN with unsupervised pre-training as the classifier for features extracted using CAE. CNNs can be constructed by removing the decoder part and adding fully connected layers. Specifically, we add two fully connected layers and output neurons with the rectified linear unit (ReLU) activation function. Each layer has 200 units, and 100 units, respectively. We implemented an optimizer using RMSprop with a learning rate of 0.001. The parameters of the CNNs including the pre-trained weights are then fine-tuned through optimizing.

6.2. Multilayer perceptron (MLP)

For features extracted using the ESN autoencoder and the Conv-ESN autoencoder, we choose Multilayer Perceptron (MLP) as the classifier. MLPs have been widely used in previous work and have shown high performance for fNIRS data classification. MLP is a feed-forward neural network with multiple fully-connected layers. Similarly, we use an MLP consisting of two hidden layers with the ReLU activation function. Each hidden layer has 200 units, and 100 units, respectively. We also implemented an optimizer using RMSprop with a learning rate of 0.01.

7. Classification results

We report the classification results achieved using features extracted with CAE and the ESN autoencoder. Moreover, to evaluate the effectiveness of these approaches, we also extract commonly-used hand-crafted features from fNIRS data and compare their results. The average values of HbR and HbO and the slope over the whole window of all channels are used as hand-crafted features. Specifically, the classification results of using features extracted with CAE was achieved by fine-tuning the CNN. The classification results of using features extracted with ESN autoencoder and hand-crafted features was achieved by training the MLP. In addition, we compare the classification results achieved when using only HbO,

Table 1. Parameter optimization table for CAE.

Depth	Filter sizes	Width	Single-task driving vs. 2-back	Single-task driving vs. 1-back	Single-task driving vs. 0-back	Four-classes classification
2	$7 \times 2, 5 \times 2$	32, 16	71.61 ± 1.22	67.23 ± 2.03	65.80 ± 1.43	44.64 ± 1.82
2	$7 \times 3, 5 \times 3$	32, 16	70.25 ± 2.23	67.42 ± 1.45	63.23 ± 1.23	43.75 ± 2.06
3	**$7 \times 2, 5 \times 2, 3 \times 2$**	**16, 16, 8**	**73.25 ± 1.59**	**68.75 ± 1.04**	**65.71 ± 1.87**	**47.21 ± 3.52**
3	$7 \times 3, 5 \times 3, 3 \times 3$	16, 16, 8	71.92 ± 1.76	67.73 ± 1.66	64.67 ± 1.73	45.33 ± 2.47
4	$7 \times 2, 5 \times 2, 5 \times 2, 3 \times 2$	16, 16, 8, 8	70.20 ± 1.34	67.19 ± 1.59	63.08 ± 1.32	43.46 ± 2.28
4	$7 \times 3, 5 \times 3, 5 \times 3, 3 \times 3$	16, 16, 8, 8	68.35 ± 1.46	66.13 ± 1.86	62.33 ± 1.67	42.78 ± 2.05

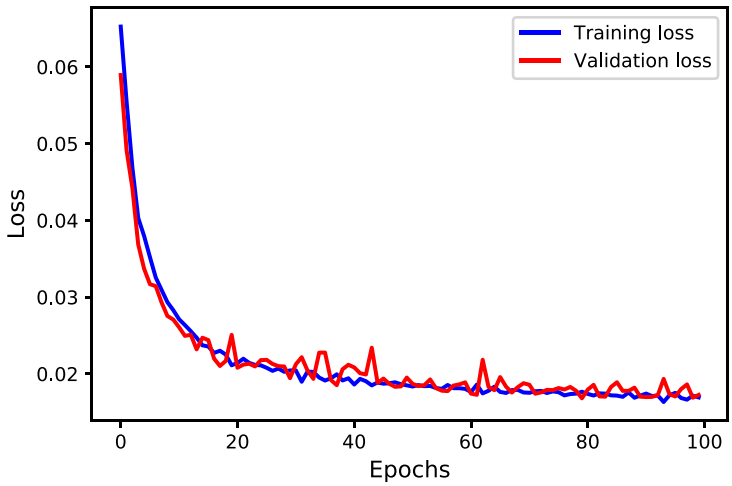

Figure 6. The mean squared error loss for training and validation sets of the CAE network with the optimal architecture, when classifying 2-back against *single-task driving*.

using only HbR, and using the combination of HbO and HbR as input with different feature extraction methods.

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

7.1. Convolutional autoencoder results

Table 1 shows the classification accuracy for differentiating different cognitive load levels from fNIRS data with the fine-tuned CNN with unsupervised pre-training using the CAE. To determine the optimal architecture for the CAE, table 1 compares the classification accuracy achieved with CAEs consisting of different filter sizes and widths (all convolutional layers are followed by a max-pooling layer with filters of size 2×2). The accuracies are the mean accuracies of 10×10 cross-validation. We can see that the architecture of the CAEs can slightly affect the classification accuracy. Specifically, when the depth is 3, and the filter sizes are 7×2, 5×2, 3×2 with a width of 16, 16, 8, we achieved the highest classification accuracy for differentiating different cognitive load with fNIRS data. As expected, classifying 2-back against *single-task driving* achieved the best results of 73.25% accuracy (precision = 74.16%, recall = 68.53%, F1-score = 71.14%), while classifying 1-back and 0-back against *single-task driving* achieved an accuracy of 68.75% (precision = 70.75%, recall = 62.90%, F1-score = 66.56%) and 65.71% (precision = 69.39%, recall = 59.26%, F1-score = 63.92%), respectively. For the four-class classification task (single-task driving vs. zero-back vs. one-back vs. two-back), we achieved an accuracy of 47.21% (chance accuracy 25%).

Furthermore, figure 6 shows the training loss and validation loss for the CAE with the optimal architecture across 100 epochs for the task of classifying 2-back against *single-task driving*. It is clear that the validation loss and training loss were converged at around the 80th epoch. More importantly, they almost dropped simultaneously, indicating that the proposed training approach allows the model to learn good generalization capability without overfitting.

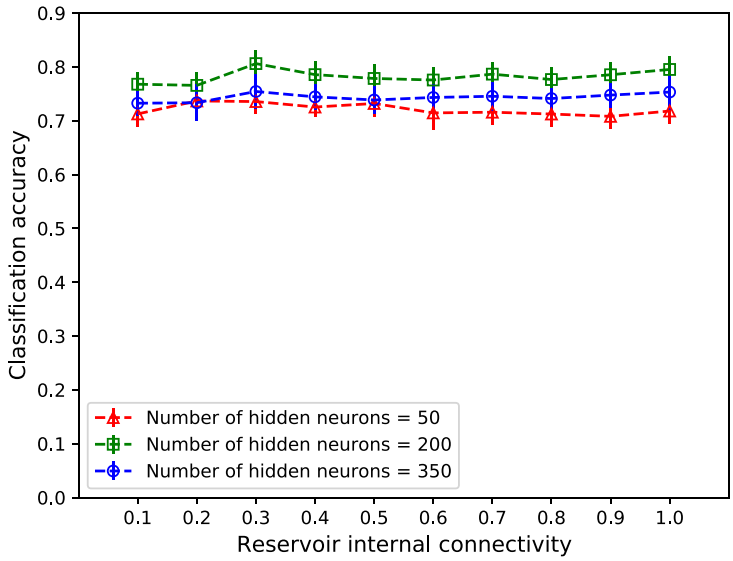

Figure 7. fNIRS data classification accuracy for *2-back* vs. *single-task driving* when using ESN autoendoers for feature extraction, with different reservoir internal connectivity. The accuracy reported represents the mean accuracy of the 10-fold cross-validation with 10 repetitions.

7.2. Echo state network autoencoder results

Figure 7 shows the comparison results of fNIRS data classification accuracy when using ESN autoencoders for feature extraction, with different reservoir internal connectivity. For simplicity, we only show the classification accuracy for differentiating *2-back* vs. *single-task driving* here. The accuracy reported is the mean accuracy of 10-fold cross-validation with 10 repetitions, and the standard deviation of each point reflects the variation of the accuracy caused by the reservoir's randomness. We can see that the reservoir's internal connectivity only slightly changes the classification results, with the best classification accuracy achieved when the connectivity is around 0.3. Moreover, we can see that the variance of accuracy due to the randomness of echo state network randomness is small (around 3.0%), which is consistent with prior work [32]. As such, we can conclude that fNIRS data classification results based on ESN autoencoder are robust against the reservoir's randomness.

Figure 8 shows the impact of the number of hidden neurons in ESN autoencoders on fNIRS data classification accuracies, when the internal connectivity is set to 0.3. We can see that the classification accuracy first increases as the number of hidden neurons in the ESN autoencoder increase, and then decreases. The best classification results are achieved when the number of hidden neurons is 200. Specifically, classifying *2-back* against *single-task driving* achieved a mean accuracy of 80.61% (precision = 79.08%, recall = 81.80%, *F*1-score = 80.38%), while classifying *1-back* and *0-back* against *single-task driving* achieved a mean accuracy of 73.86% (precision = 74.16%, recall = 72.70%, *F*1-score = 73.26%) and 71.28% (precision = 72.54%, recall = 67.26%, *F*1-score = 69.60%), respectively. For the four-class classification task, we achieved an accuracy of 52.45%.

7.3. Comparison results with different inputs

Table 2 shows the classification accuracy, precision, recall, and *F*1-score for classifying different levels of driver cognitive load when using hand-crafted features, CAE, the proposed ESN autoencoder, while using only HbO, using only HbR, and using the combination of HbO and HbR as the input. The classification results of CAE and ESN autoencoder are the best results achieved by these approaches through parameter optimization (see sections 7.1 and 7.2, respectively). From table 2, we can see that, in general, when using the CAE and the proposed ESN autoencoder, using the combination of HbO and HbR as the input achieved slightly better classification results than using only HbO or only HbR. However, when using hand-crafted features, for classifying *2-back* against *single-task driving* and *0-back* against *single-task driving*, using only HbO as the input achieved slightly better classification results than using only HbR or using the combination of HbO and HbR; while for classifying *1-back* against *single-task driving* and four-classes classification, using the combination of HbO and HbR as the input achieved slightly better classification results than using only HbO or only HbR. These results suggest that both CAE and ESN autoencoder can effectively extract useful information from the combination of HbO and HbR, while the hand-crafted features from HbO and HbR could contain redundant information and reduce the model performance.

Moreover, we can see the proposed ESN autoencoder achieved superior classification results for fNIRS-based driver cognitive load classification.

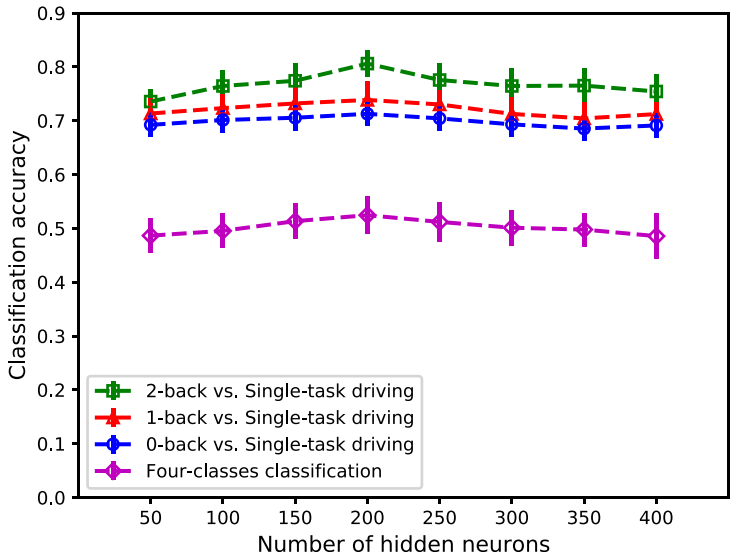

Figure 8. The impact of number of hidden neurons in ESN autoencoders on fNIRS data classification accuracies, when the internal connectivity is set to 0.3. The accuracies represent the mean accuracy of 10-fold cross-validation with 10 times repetition.

Table 2. Comparison of classification accuracy, precision, recall, and F1 score achieved by using different feature extraction methods, while using only HbO, using only HbR, and using the combination of HbO and HbR. *SD* refers to the *single-task driving* condition.

		Hand-crafted features			CAE			ESN autoencoder		
		HbO	HbR	HbO + HbR	HbO	HbR	HbO + HbR	HbO	HbR	HbO + HbR
2-back v.s *SD*	Accuracy	64.85	63.89	62.94	71.30	69.48	73.25	78.70	77.80	80.61
	Precision	66.66	66.04	65.45	74.17	73.08	74.16	77.72	76.36	79.08
	Recall	56.72	57.45	58.18	67.26	63.63	68.53	81.81	81.58	81.67
	*F*1-score	61.26	61.36	61.45	70.40	67.96	71.14	79.68	78.97	80.38
1-back v.s *SD*	Accuracy	58.31	57.40	60.21	66.57	65.75	68.75	72.21	71.30	73.86
	Precision	58.99	58.24	60.45	68.40	66.80	70.75	74.70	74.16	74.82
	Recall	59.93	58.18	63.63	62.18	58.54	62.90	69.07	67.26	72.70
	*F*1-score	59.38	58.11	61.97	65.16	62.52	66.56	71.62	70.40	73.26
0-back v.s *SD*	Accuracy	59.26	56.49	55.58	65.08	64.84	65.71	69.48	68.52	71.28
	Precision	59.22	57.48	56.72	68.71	66.36	69.39	73.08	70.74	72.54
	Recall	59.99	56.36	54.54	57.44	56.72	59.26	63.63	62.90	67.26
	*F*1-score	59.43	56.85	55.58	62.60	61.20	63.92	67.96	66.59	69.60
Four-classes	Accuracy	37.32	36.67	37.94	44.57	45.67	47.21	50.12	49.78	52.45

Specifically, compared to the highest classification accuracy achieved using hand-crafted features, ESN autoencoder improved the classification accuracy by 15.76%, 12.85% and 11.17% for classifying *2-back* against *single-task driving*, *1-back* against *single-task driving*, and *0-back* against *single-task driving*, respectively; while the classification accuracy for four-classes classification was improved by 14.51%. When compare to using CAE for feature extraction, the ESN autoencoder improved the classification accuracy by 7.36%, 5.11% and 5.55% for classifying *2-back* against *single-task driving*, *1-back* against *single-task driving*, and *0-back* against *single-task driving*, respectively; while the classification accuracy for four-classes classification was improved by 5.24%. Furthermore, statistical tests results on the best classification accuracy achieved by different methods show that the ESN autoencoder outperformed CAE for classifying *2-back* against *single-task driving* and *1-back* against *single-task driving* ($p < 0.05$, 10×10 cross-validation with a corrected paired Student *t*-test [68]), while there are no significant differences between the classification accuracy for classifying *0-back* against *single-task driving* and four-classes classification. When compared to using hand-crafted features, both CAE and ESN autoencoder achieved significantly higher accuracy for all classification tasks ($p < 0.01$, 10×10 cross-validation with a corrected paired Student *t*-test [68]). These results suggest that the proposed ESN autoencoder can effectively extract useful temporal information for fNIRS data classification.

8. Discussion

Physiological data has shown to be useful for measuring driver cognitive load non-intrusively and

continuously. However, physiological data are not always entirely reliable [7, 41].

To improve robustness, brain sensing can provide an additional objective measure of driver cognitive load level. In this work, we describe an advanced machine learning framework for driver cognitive load classification using fNIRS data. To collect an fNIRS data set with different driver cognitive load levels, we conducted a study in a driving simulator where participants were asked to perform an auditory-vocal working memory secondary-task (*n*-back). We then investigate advanced machine learning methods to extract useful features from fNIRS data for classification.

Previous research has shown the superiority of CNNs-based approach for automatically extracting features from fNIRS data comparing to hand-crafted features. However, a moving window method was often used in previous work to carefully pick a small segment from the original data as the input. While using the moving window method could result in better classification accuracy, this approach ignores the global temporal information and makes the results over-optimistic for deploying in real-world applications. Particularly, a small segment of the fNIRS data has limited capability to represent the cognitive process for measuring driver cognitive load. Therefore, we set out to investigate feature extraction methods from a long period of fNIRS data without window selection. Nevertheless, due to overfitting, the small sample sizes of fNIRS datasets make it challenging for the CNN-based method to fully extract temporal information from a long time series data [28].

As such, in this work, we investigate the application of both CNN-based autoencoder and RNN-based autoencoder for extracting patterns from fNIRS data. Specifically, we compare the classification results achieved using CAE and ESN autoencoder. CAE learns a compressed representation of the input by reconstructing the original input and has been widely used in many machine learning problems. After unsupervised training, CAE can then be used for fine-tuning CNN in classification tasks. On the other hand, ESN has been proven an efficient solution for many multivariate time series data classification problems, but it has not been explored for applying on fNIRS data. To the best of our knowledge, this is the first work to explore the application of ESN autoencoder for extracting temporal patterns from fNIRS data. Specifically, the ESN autoencoder aims to perform one step-ahead prediction for each reservoir state, and learned output weights become the features. Our results show that both CAE and ESN autoencoder are suitable for fNIRS feature extraction, while ESN autoencoder achieved higher classification accuracy than CAE for fNIRS-based driver cognitive load classification. Furthermore, since ESN autoencoder is an unsupervised feature extraction method, it can be used in various fNIRS-based machine learning problems. Apart from the higher performance, compared to other RNNs, ESN is computationally efficient and has a fast training speed, which makes it useful for real-time fNIRS data classification. For future work, we will explore the application of ESN autoencoder in other fNIRS data classification tasks.

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

9. Conclusion

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

Acknowledgments

We would like to thank Reza Moradinezhad, Calan Farley, Robert Jacob, Daniel Afergan and Missy Cummings. This material is based upon work supported by the National Science Foundation under Grant Nos. 1835307 and 1136996 awarded to Computing Research Association for the CI Fellows project. Additional support for this work was provided by the US DOT's Region I New England University Transportation Center at MIT and the Toyota Class Action Settlement Safety Research and Education Program. The views and conclusions being expressed are those of the authors, and have not been sponsored, approved, or endorsed by Toyota or plaintiffs' class counsel.

ORCID iDs

Ruixue Liu https://orcid.org/0000-0003-3139-2804
Erin Solovey https://orcid.org/0000-0003-2423-4963

References

[1] World Health Organization and others 2018 Global Status Report on Road Safety 2018: Summary *Technical Report* (World Health Organization)
[2] National Highway Traffic Safety Administration 2018 Distracted driving (available at: https://www.nhtsa.gov/risky-driving/distracted-driving) (accessed 8 April 2020)
[3] Reimer B, Mehler B, Dobres J, McAnulty H, Mehler A, Munger D and Rumpold A 2014 Effects of an 'Expert Mode' voice command system on task performance, glance behavior & driver physiology *Proc. 6th Int. Conf. Automotive User Interfaces and Interactive Vehicular Applications* pp 1–9
[4] Tchankue P, Wesson J and Vogts D 2011 The impact of an adaptive user interface on reducing driver distraction *Proc. 3rd Int. Conf. Automotive User Interfaces and Interactive Vehicular Applications* pp 87–94
[5] Engström J, Johansson E and Östlund J 2005 Effects of visual and cognitive load in real and simulated motorway driving *Transp. Res.* F **8** 97–120
[6] Coughlin J F, Reimer B and Mehler B 2011 Monitoring, managing and motivating driver safety and well-being *IEEE Pervasive Comput.* **10** 14–21
[7] Paxion J, Galy E and Berthelon C 2014 Mental workload and driving *Front. Psychol.* **5** 1344
[8] Solovey E T, Zec M, Perez E A G, Reimer B and Mehler B 2014 Classifying driver workload using physiological and driving performance data: two field studies *Proc. Conf. Human Factors in Computing Systems* (New York: ACM) pp 4057–66
[9] Mehler B, Reimer B, Coughlin J F and Dusek J A 2009 Impact of incremental increases in cognitive workload on physiological arousal and performance in young adult drivers *Transp. Res. Rec.* **2138** 6–12
[10] Mehler B, Reimer B and Coughlin J F 2012 Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task: an on-road study across three age groups *Hum. Factors* **54** 396–412
[11] Kim H S, Hwang Y, Yoon D, Choi W and Park C H 2014 Driver workload characteristics analysis using EEG data from an urban road *IEEE Trans. Intell. Transp. Syst.* **15** 1844–49
[12] Putze F, Jarvis J-P and Schultz T 2010 Multimodal recognition of cognitive workload for multitasking in the car *2010 20th Int. Conf. Pattern Recognition* (Piscataway, NJ: IEEE) pp 3748–51
[13] van Gent P, Melman T, Farah H, van Nes N and Bart van A 2018 Multi-level driver workload prediction using machine learning and off-the-shelf sensors *Transp. Res. Rec.* **2672** 141–52
[14] Reimer B and Mehler B 2011 The impact of cognitive workload on physiological arousal in young adult drivers: a field study and simulation validation *Ergonomics* **54** 932–42
[15] Reimer B, Mehler B, Wang Y and Coughlin J F 2012 A field study on the impact of variations in short-term memory demands on drivers' visual attention and driving performance across three age groups *Hum. Factors* **54** 454–68
[16] Gateau T, Durantin G, Lancelot F, Scannella S and Dehais F 2015 Real-time state estimation in a flight simulator using fNIRS *PLoS One* **10** e0121279
[17] Yue G, Miao S, Han J, Liang Z, Ouyang G, Yang J and Xiaoli Li 2018 Identifying ADHD children using hemodynamic responses during a working memory task measured by functional near-infrared spectroscopy *J. Neural Eng.* **15** 035005
[18] Herff C, Heger D, Fortmann O, Hennrich J, Putze F and Schultz T 2014 Mental workload during n-back task—quantified in the prefrontal cortex using fNIRS *Front. Hum. Neurosci.* **7** 935
[19] Peck E M, Afergan D, Yuksel B F, Lalooses F and Jacob R J K 2014 Using fNIRS to measure mental workload in the real world *Advances in Physiological Computing* (Berlin: Springer) pp 117–139
[20] Ho T K K, Gwak J, Park C M and Song J I 2019 Discrimination of mental workload levels from multi-channel fNIRS using deep leaning-based approaches *IEEE Access* **7** 24392–403
[21] Li L-peng, Liu Z-gang, Zhu H-yan, Zhu L and Huang Y-chun 2019 Functional near-infrared spectroscopy in the evaluation of urban rail transit drivers' mental workload under simulated driving conditions *Ergonomics* **62** 406–19
[22] Tsunashima H and Yanagisawa K 2009 Measurement of brain function of car driver using functional near-infrared spectroscopy (fNIRS) *Comput. Intell. Neurosci.* **2009** 164958
[23] Le A S, Aoki H, Murase F and Ishida K 2018 A novel method for classifying driver mental workload under naturalistic conditions with information from near-infrared spectroscopy *Front. Hum. Neurosci.* **12** 431
[24] Gondara L 2016 Medical image denoising using convolutional denoising autoencoders *2016 IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)* (Piscataway, NJ: IEEE) pp 241–6
[25] Huang H, Hu X, Zhao Y, Makkie M, Dong Q, Zhao S, Guo L and Liu T 2017 Modeling task fMRI data via deep convolutional autoencoder *IEEE Trans. Med. Imaging* **37** 1551–61
[26] Liou C-Y, Cheng W-C, Liou J-W and Liou D-R 2014 Autoencoder for words *Neurocomputing* **139** 84–96
[27] Hennrich J, Herff C, Heger D and Schultz T 2015 Investigating deep learning for fNIRS based BCI *2015 37th Annual International Conf. IEEE Engineering in Medicine and Biology Society (EMBC)* (Piscataway, NJ: IEEE) pp 2844–47
[28] Trakoolwilaiwan T, Behboodi B, Lee J, Kim K and Choi J-W 2017 Convolutional neural network for high-accuracy functional near-infrared spectroscopy in a brain–computer interface: three-class classification of rest, right- and left-hand motor execution *Neurophotonics* **5** 011008
[29] Aswolinskiy W, Reinhart Re F and Steil J 2018 Time series classification in reservoir-and model-space *Neural Process. Lett.* **48** 789–809
[30] Bao W, Yue J and Rao Y 2017 A deep learning framework for financial time series using stacked autoencoders and long-short term memory *PLoS One* **12** e0180944
[31] Chen H, Tiňo P, Rodan A and Yao X 2013 Learning in the model space for cognitive fault diagnosis *IEEE Trans. Neural Netw. Learn. Syst.* **25** 124–36
[32] Sun L, Jin B, Yang H, Tong J, Liu C and Xiong H 2019 Unsupervised EEG feature extraction based on echo state network *Inf. Sci.* **475** 1–17
[33] Zhang J-S and Xiao X-Ci 2000 Predicting chaotic time series using recurrent neural network *Chin. Phys. Lett.* **17** 88
[34] Wu C and Liu Y 2007 Queuing network modeling of driver workload and performance *IEEE Trans. Intell. Transp. Syst.* **8** 528–37
[35] Zhang Y, Owechko Y and Zhang J 2004 Driver cognitive workload estimation: a data-driven perspective *Proc. 7th Int. Conf. Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)* (Piscataway, NJ: IEEE) pp 642–47
[36] Mehler B, Reimer B and Dusek J A 2011 MIT AgeLab delayed digit recall task (n-back) (Cambridge, MA: Massachusetts Institute of Technology) vol 17
[37] ISO/TS 14198 (11.2012) 2019 Road vehicles-Ergonomic aspects of transport information and control systems—calibration tasks for methods which assess driver

demand due to the use of in-vehicle systems (ISO International Organization for Standardization)
[38] Owen A M, McMillan K M, Laird A R and Bullmore E 2005 N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies *Human Brain Mapp.* **25** 46–59
[39] Yang Y, Sun H, Liu T, Huang G-B and Sourina O 2015 Driver workload detection in on-road driving environment using machine learning *Proc. ELM-2014 Volume 2* (Berlin: Springer) pp 389–98
[40] Fridman L, Reimer B, Mehler B and Freeman W T 2018 Cognitive load estimation in the wild *Proc. 2018 Chi Conf. Human Factors in Computing Systems* pp 1–9
[41] Lohani M, Payne B R and Strayer D L 2019 A review of psychophysiological measures to assess cognitive states in real-world driving *Front. Hum. Neurosci.* **13**
[42] Aghajani H, Garbey M and Omurtag A 2017 Measuring mental workload with EEG+ fNIRS *Front. Hum. Neurosci.* **11** 359
[43] Liu Y, Ayaz H and Shewokis P A 2017 Multisubject "learning" for mental workload classification using concurrent EEG, fNIRS and physiological measures *Front. Hum. Neurosci.* **11** 389
[44] Saadati M, Nelson J and Ayaz H 2019 Convolutional neural network for hybrid fNIRS-EEG mental workload classification *Int. Conf. Applied Human Factors and Ergonomics* (Berlin: Springer) pp 221–32
[45] Nagasawa T, Sato T, Nambu I and Wada Y 2020 fNIRS-GANs: data augmentation using generative adversarial networks for classifying motor tasks from functional near-infrared spectroscopy *J. Neural Eng.* **17** 016068
[46] Putze F, Herff C, Tremmel C, Schultz T and Krusienski D J 2019 Decoding mental workload in virtual environments: a fNIRS study using an immersive n-back task *2019 41st Annual Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)* (Piscataway, NJ: IEEE) pp 3103–106
[47] Combrisson E and Jerbi K 2015 Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy *J. Neurosci. Methods* **250** 126–36
[48] Bandara D, Velipasalar S, Bratt S and Hirshfield L 2018 Building predictive models of emotion with functional near-infrared spectroscopy *Int. J. Hum. Comput. Stud.* **110** 75–85
[49] Erdoğan S B, Özsarfati E, Dilek B, Kadak K S, Hanoğlu Lutfü and Akın A 2019 Classification of motor imagery and execution signals with population-level feature sets: implications for probe design in fNIRS based BCI *J. Neural Eng.* **16** 026029
[50] Gemignani J, Middell E, Barbour R L, Graber H L and Blankertz B 2018 Improving the analysis of near-infrared spectroscopy data with multivariate classification of hemodynamic patterns: a theoretical formulation and validation *J. Neural Eng.* **15** 045001
[51] Hu X-S, Hong K-S and Ge S S 2012 fNIRS-based online deception decoding *J. Neural Eng.* **9** 026012
[52] Liu R, Walker E, Friedman L, Arrington C M and Solovey E T 2020 fNIRS-based classification of mind-wandering with personalized window selection for multimodal learning interfaces *J. Multimodal User Interfaces* (https://doi.org/10.1007/s12193-020-00325-z)
[53] Sereshkeh A R, Yousefi R, Wong A T and Chau T 2018 Online classification of imagined speech using functional near-infrared spectroscopy signals *J. Neural Eng.* **16** 016005
[54] Wang Y, Mehler B, Reimer B, Lammers V, D'Ambrosio L A and Coughlin J F 2010 The validity of driving simulation for assessing differences between in-vehicle informational interfaces: a comparison with field testing *Ergonomics* **53** 404–20
[55] Mehler B and Reimer B 2013 An initial assessment of the significance of task pacing on self-report and physiological measures of workload while driving *Proc. 7th Int. Symp. Human Factors in Driver Assessment, Training and Vehicle Design* pp 170–76
[56] Naseer N and Hong K-S 2015 fNIRS-based brain-computer interfaces: a review *Front. Hum. Neurosci.* **9** 3
[57] Pinti P, Scholkmann F, Hamilton A, Burgess P and Tachtsidis I 2018 Current status and issues regarding pre-processing of fNIRS neuroimaging data: An investigation of diverse signal filtering methods within a general linear model framework *Front. Hum. Neurosci.* **12** 505
[58] Cui X, Bray S and Reiss A L 2010 Functional near infrared spectroscopy (NIRS) signal improvement based on negative correlation between oxygenated and deoxygenated hemoglobin dynamics *Neuroimage* **49** 3039–46
[59] Huppert T J, Diamond S G, Franceschini M A and Boas D A 2009 HomER: a review of time-series analysis methods for near-infrared spectroscopy of the brain *Appl. Opt.* **48** 0–33
[60] Li T, Luo Q and Gong H 2010 Gender-specific hemodynamics in prefrontal cortex during a verbal working memory task by near-infrared spectroscopy *Behav. Brain Res.* **209** 148–53
[61] Gehring J, Miao Y, Metze F and Waibel A 2013 Extracting deep bottleneck features using stacked auto-encoders *2013 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (Piscataway, NJ: IEEE) pp 3377–81
[62] Bianchi F M, Scardapane S, Uncini A, Rizzi A and Sadeghian A 2015 Prediction of telephone calls load using echo state network with exogenous variables *Neural Netw.* **71** 204–13
[63] Li D, Han M and Wang J 2012 Chaotic time series prediction based on a novel robust echo state network *IEEE Trans. Neural Networks and Learning Systems* **23** 787–99
[64] Ma Q, Shen L, Chen W, Wang J, Wei J and Yu. Z 2016 Functional echo state network for time series classification *Inf. Sci.* **373** 1–20
[65] Tanisaro P and Heidemann G 2016 Time series classification using time warping invariant echo state networks *2016 15th IEEE Int. Conf. Machine Learning and Applications (ICMLA)* (Piscataway, NJ: IEEE) pp 831–36
[66] Bianchi F M, Scardapane S, Løkse S and Jenssen R 2020 Reservoir computing approaches for representation and classification of multivariate time series *IEEE Trans. Neural Netw. Learn. Syst.* (https://doi.org/10.1109/TNNLS.2020.3001377)
[67] Masci J, Meier U, Cireşan D and Schmidhuber Jurgen 2011 Stacked convolutional auto-encoders for hierarchical feature extraction *Int. Conf. Artificial Neural Networks* (Berlin: Springer) pp 52–59
[68] Bouckaert R R and Frank E 2004 Evaluating the replicability of significance tests for comparing learning algorithms *Pacific Conf. Knowledge Discovery and Data Mining* (Berlin: Springer) pp 3–12