Feature selection of EEG-signal data for cognitive load

Mälardalens Högskola
Akademin för Innovation, Design och Teknik
Isac Persson
Thesis for the Degree of Bachelor in Computer Science
2017-05-17
Examiner: Mobyen Uddin Ahmed
Supervisor: Shaibal Barua
Abstract

Safely operating a vehicle requires the full attention of the driver. Should the driver lose focus as a result of performing other tasks simultaneously, there could be disastrous outcomes. To gain insight into a driver's mental state, the cognitive load experienced by the driver can be investigated. Measuring cognitive load can be done in numerous ways, one popular approach is the use of Electroencephalography (EEG). A lot of the data that can be extracted from EEG-signals, are redundant or irrelevant when trying to classify cognitive load. This thesis focuses on identifying EEG-features relevant to the classification of cognitive load experienced by drivers, through the use of feature selection algorithms. An experimental approach was utilized where three feature selection algorithms (ReliefF, BSS/WSS and BIRS) were applied to the available datasets. The feature subsets produced by the algorithms achieved higher classification accuracies compared to the use of all features. The best performing subset was generated by the ReliefF algorithm which achieved an accuracy of 66%. However, several other unique subsets achieved comparable results, therefore no single feature subset could be identified as most relevant for classification of cognitive load experienced by drivers. To conclude, the proposed approach could not identify features which could be used to confidently predict a driver's mental state.
Acknowledgements

This thesis was done under the Vehicle Driver Monitoring-project. Special thanks to the project partners Volvo Car and VTI for collecting and sharing the data that made this thesis work possible. Additionally, I would like to express my gratitude to my supervisor, Shaibal Barua, for his advice and guidance during the thesis process. I would also like to acknowledge my thesis examiner, Dr. Mobyen Uddin Ahmed, for his suggestions on several occasions during the thesis work.
Table of Contents

1. Introduction .......................................................................................................................... 5
  1.1 Problem formulation ........................................................................................................ 5
2. Background ............................................................................................................................ 6
  2.1 Electroencephalography (EEG) ...................................................................................... 6
  2.2 Cognitive Load ................................................................................................................ 6
  2.3 Classification .................................................................................................................. 6
    2.3.1 Support Vector Machines (SVM) ........................................................................... 6
  2.4 k-Fold Cross-Validation ............................................................................................... 7
  2.5 Feature Selection ........................................................................................................... 7
3. Related Work ....................................................................................................................... 8
4. Method ................................................................................................................................... 10
5. Data Analysis ....................................................................................................................... 11
  5.1 Datasets .......................................................................................................................... 11
  5.2 Removal of Outliers ...................................................................................................... 12
  5.3 Class Separability ......................................................................................................... 14
  5.4 New Formation of Datasets ......................................................................................... 15
6. Experiments ......................................................................................................................... 17
  6.1 Design ............................................................................................................................. 17
    6.1.1 Experiment 1 – Filter Algorithms ......................................................................... 17
    6.1.2 Experiment 2 – BIRS ........................................................................................ 18
  6.2 Implementation ............................................................................................................... 18
    6.2.1 Reproducibility ..................................................................................................... 18
    6.2.2 Data normalization ............................................................................................... 19
    6.2.3 Data Split .............................................................................................................. 19
    6.2.4 Feature ranking .................................................................................................... 19
    6.2.5 Support Vector Machine ....................................................................................... 20
    6.2.6 BIRS ...................................................................................................................... 20
7. Results .................................................................................................................................. 21
  7.1 Experiment 1 – Filter Algorithms ............................................................................... 21
  7.2 Experiment 2 - BIRS .................................................................................................. 23
8. Discussion ............................................................................................................................. 25
9. Conclusions ............................................................................................................................ 26
  9.1 Future Work ................................................................................................................... 26
References ................................................................................................................................... 27
List of Figures
Figure 1. Boxplots - Hidden Exit Dataset ................................................................. 12
Figure 2. Boxplots - Car From Right Dataset ............................................................... 13
Figure 3. Boxplots - Side Wind Dataset ................................................................. 13
Figure 4. Boxplots - FICA Dataset ........................................................................... 13
Figure 5. ecdf - Car From Right and Hidden Exit Datasets ........................................ 14
Figure 6. ecdf - Side Wind Dataset ........................................................................... 14
Figure 7. ecdf - FICA Dataset .................................................................................... 15
Figure 8. Boxplots - New Datasets ........................................................................... 16
Figure 9. ecdf - New Datasets .................................................................................... 16
Figure 10. Design of Experiment 1 ............................................................................ 17
Figure 11. Design of Experiment 2 ............................................................................ 18
Figure 12. Results Experiment 1 - HE & CFR Dataset ................................................... 22
Figure 13. Results Experiment 1 - Side Wind Dataset ................................................... 23
Figure 14. Results Experiment 2 - HE & CFR Dataset .................................................. 24
Figure 15. Results Experiment 2 - Side Wind Dataset .................................................. 24

List of Tables
Table 1. Results Table ............................................................................................... 21
1. Introduction
Safely operating a vehicle requires the full attention of the driver. There could potentially be disastrous outcomes if the driver loses focus as a result of performing other cognitively loading tasks whilst driving. Cognitive load describes the amount of working memory demand imposed by the performance of a particular task [1]. Since working memory is limited in capacity, performing tasks of various difficulties will reduce accessibility of working memory and increase cognitive load. High levels of cognitive load can result in decreased performance or failure to complete tasks [2]. Performing a cognitively loading task interferes with one’s ability to perform other tasks simultaneously. Hence, a driver’s ability to safely maneuver his/her vehicle can be greatly affected by performing other cognitively loading tasks whilst driving. It has been shown that drivers performing cognitively loading secondary tasks, experience more difficulty maintaining speeds lower than their preferred traveling speed [3]. Measuring cognitive load can be done in numerous ways, one popular approach is the use of EEG [4][5].

Electroencephalography (EEG) is a noninvasive method that, with the help of electrodes placed around the scalp, measures voltage fluctuations from ionic current in the neurons of the brain. EEG-signals are used in a wide variety of research fields, for example in classification of sleep stages [6] or in measuring cognitive load of students learning new languages [7]. Substantial amounts of data can be extracted from EEG-signals, for example, assuming EEG-signal data is recorded from 32 electrodes and the signal power in 5 frequency bands is calculated individually per each 5 s, the number of extracted features amounts to 800 (32 electrodes * 5 frequency bands * 5 s) [8]. High dimensional feature vectors make classification problems more complex as they may contain redundant, irrelevant or noisy features which decrease the performance of classification algorithms [9].

To avoid the problems of high dimensional feature vectors and to increase classification accuracy, feature selection is needed. Feature selection, which is the focus of this thesis, can be regarded as the process of removing irrelevant, redundant or noisy features until only a set of features truly relevant to the problem remains.

1.1 Problem formulation
The focus of this thesis is identifying which subset of features are related to the levels of cognitive load experienced by individuals operating vehicles. This is done by the implementation of feature selection algorithms. The data used throughout the thesis is extracted from the EEG-signals of individuals performing a set of driving related tasks in a simulated environment.

The following specific research questions have been investigated during the thesis work:
• Which types of feature selection methods are suitable for the datasets?
• Which subset of features are most relevant for classification of a driver’s cognitive load?
2. Background

In this section, the necessary theory needed to understand the thesis is presented.

2.1 Electroencephalography (EEG)

Electroencephalography (EEG) is a noninvasive method for measuring voltage fluctuations from ionic current in the neurons of the brain. This is done with the help of electrodes placed around the test subject's scalp. Electrode placement usually follows the "International 10-20 System" in which electrodes are placed 10% or 20% of the total distance between specific locations of the skull [10]. Using a percentage based system accommodates differences in skull size amongst test subjects and ensures reproducibility.

There are several factors observed in analysis of EEG-signals, some of which include frequency distribution, signal amplitude and waveform morphology. The frequency range of EEG-signals are usually divided into several bands: less than 4 Hz (delta), 4 to 7 Hz (theta), 8 to 15 Hz (alpha), 16 to 31 Hz (beta) and greater than 32 Hz (gamma). The amplitude of EEG-signals is defined as the voltage (microvolts) measured from the peak to the through of a wave. The shape of an EEG signal is affected by a combination of frequency and amplitude, which makes it possible for certain features to be filtered out and thus detected. Waveform fluctuations occur in response to stimuli and depend on the state of the test subject, for example if the subject is alert or sleepy etc. [10].

2.2 Cognitive Load

Cognitive load can be defined as the amount of demand that performing a particular task places on an individual's cognitive system. Mental load, mental effort and performance are regarded as the main aspects of cognitive load [1]. Mental load refers to the interaction between task and subject on the basis of current knowledge about their characteristics. Therefore, it can be used for an "a priori" estimation of the expected demands on cognitive capacity. Mental effort is measured whilst a task is being performed and can be considered to reflect the actual levels of cognitive load, since it refers to the cognitive capacity allocated to meet the demand of the task. Performance describes the achievements of an individual performing a certain task, for example the amount of time spent on the task or the number of errors etc.

2.3 Classification

In the context of machine learning and statistical learning, classification refers to problems where the desired response is qualitative (also referred to as categorical), i.e. a variable representing one of K different classes [11]. In other words, the goal of classification is to identify which class a new observation should be assigned.

2.3.1 Support Vector Machines (SVM)

The Support Vector Machine (SVM) is a popular classifier that have been shown to perform well in various different settings and is commonly considered one of the best "out of the box" classifiers according to G. James et al [11]. SVMs are an extension of the support vector classifier and can handle linear as well as non-linear class boundaries with the help of kernel functions. For data that can be linearly classified, the SVM tries to identify the maximum-margin hyperplane that separates the different classes. However, if the data cannot be linearly separated, non-linear kernel functions are used to transform the feature space, allowing a maximum-margin hyperplane to be established.
Support vector Machines are a popular choice for classification in work related EEG-signal data. SVMs have been used for classification of memory workload levels [12], classification of sleep stages [6] and cross-task mental workload recognition [13].

2.4 k-Fold Cross-Validation
K-fold cross-validation is a re-sampling method in which the dataset is randomly divided into k folds. One fold is treated as a validation set and the remaining k-1 folds are used for training of the machine learning method. The classification accuracy of the model is determined on the validation fold. This process is repeated k times, such that all folds are used for validation once and k classification accuracies are acquired. The final accuracy estimate of the model is determined by averaging the individual k accuracies [11].

Cross-validation can be implemented with the goal of determining how well a model can be expected to perform on “unseen” data. However, in the context of this thesis, the goal of cross-validation is to evaluate various levels of flexibility by tuning parameters of selected classification methods.

2.5 Feature Selection
Feature selection is the process of removing irrelevant, redundant or noisy features from feature vectors, in order to increase performance of machine learning algorithms. Features which can be removed without affecting learning performance are referred to as irrelevant features. Whilst, redundant features on the other hand, are features which are correlated, i.e. “describes” the same thing as other features. They are relevant when observed individually, but the removal of one such feature does not affect the learning performance [14].

The three main objectives of feature selection are: improving accuracy of predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data [15]. By eliminating irrelevant, redundant and noisy features, the classification performance will be increased and the computational cost of classifiers will be decreased [6]. There are generally two ways of achieving feature selection: by ranking features based on some sort of criteria and selecting the top n ranking features or by combining features into smaller subsets and evaluating their performance [14].

Feature selection algorithms can be divided into three main groups [8],[9]:
- Filter algorithms: Selects features before passing them to the classification algorithm.
- Wrapper algorithms: Evaluates feature sets by using the classification algorithm as a subroutine for the feature selection task.
- Embedded algorithms: Selection is embedded within the induction algorithm.

The performance of a feature selection algorithm may differ depending on the type of problem that needs to be solved and the structure of the dataset that is used. Therefore, it can be advantageous to implement a few different algorithms to determine which algorithm performs the best on the specific problem at hand.
3. Related Work

A study of work and material related to classification of EEG-signal data and levels of cognitive load has been performed with the goal of creating a better understanding of which feature selection methods are commonly used in this field. The amount of studies related to classification of cognitive load from EEG data, that implemented feature selection, were rather few. Thus, more general work related to feature selection were also investigated. Below are some approaches to feature selection observed during the study, some of which are commonly used with EEG data:

Recursive Feature Elimination (RFE)

In working with the problem of cross-task mental workload recognition, Yufeng Ke et al [13], applied a modified implementation of a Recursive Feature Elimination algorithm for feature selection of EEG data.

The RFE-algorithm was constructed in such a way, that given an N-dimensional feature set, one feature was eliminated and to evaluate performance a SVM were trained and tested using the remaining N-1 features. This was repeated N times such that all features were removed once, resulting in N performance results. The feature, that when removed, resulted in the best performance was regarded as least contributing in the set and thus could be removed from the feature set. This entire process was repeated for the remaining features until only one feature remained.

As the RFE-algorithm recursively tests every feature subset, it can be considered impractical for high dimensional feature vectors, since larger feature sets results in a much higher computational demand.

Minimum Redundancy Maximum Relevance (mRMR)

The idea behind the mRMR-algorithm is to select the most relevant features with regards to class labels, whilst minimizing redundancy amongst the selected features. The algorithm uses mutual information to compute feature-to-feature and feature-to-class similarities [12].

The mRMR-algorithm has been used on EEG-signal data for classification of memory workload levels [12], as well as the classification of sleep stages [6]. Both studies used Support Vector Machines for classification.

ReliefF

The ReliefF-algorithm is an extension of the original Relief that is more robust, can handle incomplete or noisy data as well as classification problems with multiple classes. Depending on how well feature values distinguish between instances that are near each other, the algorithm tries to estimate the feature relevance. For a randomly selected instance, the algorithm searches for neighbors of the same class (referred to as “nearest hits”) and neighbors of a different class (referred to as “nearest misses”). If the distance from the randomly selected instance to a nearest hit is greater than to a nearest miss, the feature relevance will be slightly decreased and vice versa [8].

The ReliefF-algorithm has been used in studies related to classification of epileptic EEG events [16], as well as in the previously mentioned study of sleep stage classification [6].

Genetic Algorithms
Genetic algorithms are inspired by biological evolution and encode potential solutions as chromosome-like data structures [9]. Individuals are generated by a combination of features and their level of “fitness” are evaluated by the performance of a selected classifier. New generations of individuals, that will have an improved level of “fitness”, are created from combination of parents and mutation. This process is repeated until an optimization criteria has been met.

Genetic algorithms appear to be a popular approach to feature selection and have been used in work related to EEG-signal classification of schizophrenia [9] and motor imagery [17]. Both studies implement SVMs for classification.

**Forward Selection**
The forward selection method starts with an empty set of features. For every step in the process the feature set is extended by one. All remaining features are evaluated and the one that provides the highest increase in classification accuracy is added to the set. This entire process is repeated until none of the remaining features can increase the classification accuracy [8].

**Best Incremental Ranked Subset (BIRS)**
Best Incremental Ranked Subset is a two-phase algorithm. In the first phase, features are ranked per some evaluation criteria. Once the features have been ranked, the second phase starts and the features will be passed through from the highest to the lowest ranked. The classification accuracy of the highest ranked feature is obtained and the feature is marked as selected. Then, the classification accuracy for the selected and the “next” feature is obtained. If the next feature significantly improves the classification accuracy, it will be marked as selected as well. This process is repeated for the remaining features and on completion the algorithm returns the best subset found [14].

**BSS/WSS**
The BSS/WSS (Between-group Sum of Squares/Within-group Sum of Squares) algorithm is a filter which ranks features according to the ratio of the sum of squares differences between groups and within groups [18]. The algorithm is applied to all features in the dataset individually and a ratio-value is calculated on a feature to feature basis. The resulting ratio-values are sorted and the features are ranked accordingly.

**Conclusions**
As is evident by the study of related work, the most common method of classification seems to be SVMs. However, for feature selection, numerous different methods appear in work related to EEG-signal data. The selection of a certain method is highly dependent on the structure of the data at hand. The datasets used in this thesis contained 270 features, which in combination with the available computational resources made wrapper algorithms such as RFE and Forward Selection unsuitable for use. In this thesis work, two filter algorithms (ReliefF and BSS/WSS) are used since they have low computational demand. Additionally, one hybrid algorithm (BIRS), is used since it utilizes feature rankings and incrementally creates feature subsets, thus evaluating more feature combinations than the filters, without needing as much computational resources as for example RFE.
4. Method

To answer the research questions outlined in the thesis, several methodologies have been utilized. Initially, a literature study of material related to feature selection was performed to acquire knowledge of the current "State of the Art". Once information about commonly used algorithms, such as their areas of application and suitability for certain types of data, had been collected, it was easier to select suitable algorithms for implementation and testing in the later stages of the thesis work.

The datasets used throughout the thesis, which were provided by the supervisor, contained data extracted from EEG-signals of individuals performing driving related tasks in a simulated environment. Although artifacts had already been removed from the data during extraction, an analysis of the datasets was performed. By analyzing the structure of the data and its features, decisions about suitable algorithms could be made. Additionally, potential outliers could be identified and removed before any feature selection was performed.

The combined knowledge acquired by the literature study and data analysis aided in the selection of algorithms. Once a set of 3 suitable algorithms had been selected, they were implemented in the high-level language R. Two experiments were constructed and each experiment was performed twice for each of the datasets. The classifier utilized in the two experiments was a Support Vector Machine (see section 2.3.1), which was used to evaluate the performance of the features subsets. The results of the experiments were used as a basis for conclusions about the features most relevant to cognitive load.
5. Data Analysis

All data used throughout the thesis work was provided by the supervisor. Although this thesis did not focus on the extraction or collection of data, an understanding of the process that generated the dataset assisted in analyzing it. The data were extracted from the EEG-signals of individuals using a driving simulator to operate a vehicle in an urban environment. Data was collected for 4 different tasks performed in a single driving session by each of the test subjects:

- Car From Right: The test subject drove along a road when a car appeared from the right and traveled in the direction of test subject’s vehicle.
- Hidden Exit: As the test subject drove, a hidden exit appeared on the right-hand side of the road after a curve, which the driver passed.
- Side Wind: The test subject drove along a road and a force (wind) was applied to the vehicle pushing it in a specific direction.
- FICA: As the test subject drove along a road, the car in front stopped suddenly.

Each task was performed 4 times, with the exception of FICA which was only performed once. Data for 3 different events/classes (Pre-task, Task, Post-task) were collected for each of the tasks. For Car From Right, Hidden Exit and Side Wind, an additional n-Back task was performed simultaneously during 2 of the 4 Task-events. The extraction of data from the EEG-signals of the test subjects, resulted in an initial dataset consisting of 1254 observations (rows) and 275 features (columns). Only 270 of the 275 features were used for further analysis, as 5 of the features contained metadata such as task names and subject IDs etc.

5.1 Datasets

The initial dataset was loaded into R and filtered by task name to create a subset of data for each task. The four subsets were exported to .csv files:

- HiddenExitDataset.csv (396 observations and 275 features)
- CarFromRightDataset.csv (396 observations and 275 features)
- SideWindDataset.csv (396 observations and 275 features)
- FICAData.csv (66 observations and 275 features)

As each dataset only contained data related to one specific task, individual analysis of the different tasks could more easily be conducted.

The features that the datasets consist of, represents the Power Spectral Density (PSD) which have been estimated from each EEG-channel. PSD was extracted for the frequency bands: δ(<4 Hz), θ(4-7 Hz), α(8-12 Hz), β(12-30 Hz), and γ (31-50 Hz). Additionally, the following four features were also extracted: (θ+α)/β, α/β, (θ+α)/(α+β), and θ/β. Nine features were extracted from each of the 30 EEG-channels, resulting in a total of 270 features. The features were denoted as “EEG_channel”_“feature”. For example, feature “f8_theta”, represented the PSD of the theta band (4 to 7 Hz) for EEG-channel “f8”.

Ethical Considerations

The data was collected under the Vehicle Driver Monitoring project, where VTI and Volvo Car were responsible for ethical considerations. Participation in the study was voluntary and each participant had the right to cancel the study at any time. The participants received written and oral information regarding the study and each
participant signed a consent form. Personal privacy was covered by the Personal Data Act (1998: 204).

The datasets used during this thesis work, contained no information about the test subjects (names, ages etc.). Since the datasets posed no threat to the personal integrity of the test subjects, no further ethical considerations regarding the handling of the data had to be made.

5.2 Removal of Outliers

Initially, the datasets were reviewed manually with the help of boxplots and histograms to create an understanding of the data distributions across the features. Values that appeared to significantly differ from the majority of observations could be potential outliers. For example, in the “Car From Right”-dataset most values were in the range of 0-50, whilst some values exceeded 4000 which were a clear indication of an outlier. Such extreme outliers could indicate that an error had occurred either during the EEG recording phase (for example, faulty electrodes etc.) or during the extraction of data from the raw signal data.

To identify and remove outliers in the datasets, the `outlier()`-function from R's `outliers`-package [19] was utilized. The function identified the values with the biggest difference from the sample mean for each feature in the datasets. Once a set of potential outlier-values had been identified they were compared to upper and lower threshold values generated by calculating the sample median $+$ the median absolute deviation $\times$ 2.5 as proposed by Christophe Leys et al [20].

$$M-2.5 \times MAD < x < M + 2.5 \times MAD$$

If the value was greater or smaller than the proposed thresholds, it was considered an outlier and removed from the dataset.

**Hidden Exit**

After removal of outliers the “Hidden Exit”-dataset consisted of 352 observations, with the following distribution amongst the classes:

- Pre-task: 119 observations
- Task: 117 observations
- Post-task: 116 observations

![Figure 1. Boxplots - Hidden Exit Dataset](image-url)
Car From Right
13 outliers were removed from the “Car From Right”-dataset, resulting in a total of 384 remaining observations, distributed between the classes as follows:
- Pre-task: 127 observations
- Task: 129 observations
- Post-task: 128 observations

Side Wind
17 outliers were removed from the “Side Wind”-dataset, resulting in 379 remaining observations. The distribution of observations between the classes looked as follows:
- Pre-task: 122 observations
- Task: 128 observations
- Post-task: 129 observations

FICA
After removal of outliers, the “FICA”-dataset contained 62 observations with the following distribution between the two classes:
- Pre-task: 30 observations
- Task: 32 observations
5.3 Class Separability
To create an understanding of how effective the features were at separating the different classes (Pre-task, Task and Post-task), a two-sample t-test was applied on all features and the resulting p-values were compared. The lower the p-value, the more effective the feature was at separating the classes. To create a visual representation of the class separability for the datasets, the empirical cumulative distribution function (ecdf) of the p-values were plotted.

By examining the shape of the resulting graphs, conclusions could be made regarding the number of features that could effectively separate the various classes. For example, the “CFR – Task & Post”-graph in figure 5 showed that around 10% of all features in the “Car From Right”-dataset had a p-value of less than 0.05, which indicated that roughly 27 out of the 270 features had strong discriminative power when trying to separate Task and Post-task events.

Figure 5. ecdf - Car From Right and Hidden Exit Datasets

Figure 5 show that the distribution of features with discriminative power were very similar across the different classes for the “Car From Right” and “Hidden Exit”-datasets. Likewise, the graphs for the “Side Wind”-dataset (figure 6) show comparable results for Pre-task & Post-task separability. However, Pre-task & Task as well as Task & Post-task separability appeared to be slightly less effective than that of the other datasets.

Figure 6. ecdf - Side Wind Dataset
As the “FICA”-dataset only contained observations of the Pre-task and Task classes, only one ecdf-graph was plotted (figure 7). Similarly to the results of the other datasets there were no features with strong discriminative power for separating Pre-task and Task events.

![FICA - Pre & Task](image)

**Figure 7. ecdf - FICA Dataset**

### 5.4 New Formation of Datasets

Since the focus was to detect cognitive load, which was affected by the test subjects performing n-back tasks whilst driving, the datasets needed to be re-structured. The newly structured datasets only contained the Task-event data. Pre- and Post-task data were removed because during these time segments the test subjects were still driving the same route which could affect the experienced cognitive load. To which extent this affected the data cannot be known without a control setting, and as that was not available, the time segments were discarded. Additionally, the first 10 seconds of data were also discarded to allow for calibration of the EEG-signals with the scenario for each event.

As the new datasets only contained Task-event data, focus was shifted from classification of Pre-task, Task and Post-task events, to classification of n-back and no n-Back events. The events where an n-Back task was performed simultaneously to the main task were expected to result in higher levels of cognitive load, as such, n-Back events were given the class label “High” whilst no n-Back events were attributed the class label “Low”.

Data collected from the Car From Right and Hidden Exit tasks were combined into one single dataset, since both scenarios consisted of visual cues. The Side Wind data was kept separate since the scenario was influenced by the wind factor and deemed more demanding than the other two scenarios. The FICA scenario was removed since FICA was a critical situation scenario and no n-Back task was performed during any of the Task-events.

The new datasets were processed in the same way as the original dataset. Although the new datasets appeared to contain significantly fewer extreme outliers, a total of 26 observations were identified as outliers and removed from the datasets. After removal of outliers the “Hidden Exit and Car From Right”-dataset contained 248 observations (126 High and 122 Low) and the “Side Wind”-dataset contained 122 observations (61 High and 61 Low).
The class separability of the new datasets appeared to be quite weak as no features showed strong discriminative power. For the "Hidden Exit and Car From Right"-dataset only around 2% of the features had a p-value of less than 0.1. The "Side Wind"-dataset appeared to have even weaker separability of the classes, as only around 1% of the features had p-values of less than 0.2.
6. Experiments

To identify which subset of features that are most relevant to the classification of a driver’s cognitive load, a set of experiments were constructed. Using the knowledge gained from the literature study and the data analysis as a basis, three suitable feature selection algorithms were chosen. With regards to the structure of the datasets it was decided that two filter algorithms (ReliefF and BSS/WSS) as well as one hybrid algorithm (BIRS) would be used to evaluate the feature relevance for cognitive load. As a result, two experiments were designed and implemented in the R programming language. The experiments were performed four times each, twice for each dataset alternating between the two filter algorithms.

6.1 Design

The first experiment was designed to utilize the filter algorithms, whilst the second experiment utilized the hybrid feature selection algorithm. However, the general designs of the experiments were quite similar and they differed mainly in the way that feature subsets were constructed and evaluated.

Both experiments were structured to perform the same initial operations i.e. loading, normalizing and splitting the desired datasets. Once the data had been correctly partitioned into training and test sets, feature ranking was performed. The filter algorithms ReliefF and BSS/WSS were utilized in both experiments and applied to the training sets in order to rank the features. The operations following the feature ranking differed between the experiments as described below.

6.1.1 Experiment 1 – Filter Algorithms

The first experiment was structure to only utilize the feature rankings produced by ReliefF and BSS/WSS. Once the features had been ranked, subsets were constructed by simply taking the top $k$ ranking features. Initially the $k$ value was set to the number of features in the datasets, resulting in all features being used for training the classifier. The $k$ value was then reduced in predefined arbitrary steps and new increasingly smaller subsets were created. Each new subset was used to train the classifier which were then tested on the test set. Once the $k$ value reached one, the performance of all subsets was presented. Conclusions could then be made regarding which subset of features resulted in the highest classification accuracy.

![Figure 10. Design of Experiment 1](image-url)
6.1.2 Experiment 2 – BIRS
The second experiment was also structured to rely on ReliefF and BSS/WSS to rank features. However, once the feature ranking had been established, a feature subset was created by using a wrapper algorithm. Starting with only the highest ranking feature, the classifier was trained and the performance evaluated on the test set. The second highest ranking feature was then added to the selected subset and the classifier was once again trained and the performance evaluated. If the new feature improved the classification accuracy it was kept in the selected subset, however, if the performance decreased or remained the same as for the previous subset, the feature was discarded. This process was repeated until all top $k$ ranking features had been tested, after which the final selected subset and its performance was presented.

![Design of Experiment 2](image)

6.2 Implementation
The experiments were implemented using the R programming language and software environment. The motivation for using R was its popularity in fields such as statistics, data mining and data analysis as well as the wide variety of high quality libraries/packages available for use. By being able to use pre-existing packages instead of re-implementing certain components, the time required to implement and execute the experiments was drastically reduced. Below is a description of the various components and their implementations.

6.2.1 Reproducibility
To ensure reproducibility across all iterations of the experiments, the `set.seed(888)` function in R was used with an arbitrary seed value of “888”. This was done to ensure that all “randomized” actions were performed the same way every time the experiments were executed. For example, when $k$-fold cross validation was performed, the `set.seed()` function ensured that the dataset was “randomly” partitioned the same way every iteration of the experiments.

By using the `set.seed()`-function, the experiments could accurately be reproduced. Additionally, it could be concluded that any differences in accuracy between the experiments were a result of the performance of the selected algorithms and not dependent on any “randomized” actions within the experiments.
6.2.2 Data normalization

The datasets were normalized to the range of [0-1] by applying the following formula to all features:

\[ X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \]

Where \( X \) represents the feature values, \( X_{\text{max}} \) is the sample maximum and \( X_{\text{min}} \) is the sample minimum.

6.2.3 Data Split

Each dataset was split into a training set containing 70% of the observations and a test set containing the remaining 30%. The training set was used for training and tuning of the classifier, whilst the test set was used to evaluate the performance of the final classification model.

6.2.4 Feature ranking

Feature ranking was utilized in both experiments and performed by using two different filter algorithms, ReliefF and BSS/WSS.

ReliefF

The ReliefF algorithm randomly selects an instance/observation and then searches for \( k \) of its nearest neighbors from the same class (referred to as “nearest hits”) as well as \( k \) nearest neighbors from a different class (referred to as “nearest misses”). If the value of a feature for the randomly selected instance differs greatly from the average of the “nearest hits”, the feature separates instances of the same class which is not desirable, thus the relevance estimate of the feature is decreased. However, if the feature separates instances of different classes, which is desirable, the relevance estimate of the feature is increased.

The algorithm was implemented using R’s “CORElearn”-package [21]. The selected implementation was “ReliefFexpRank” with a \( k \) value of 70, as recommended by Robnik-Sikonja and Kononenko [22]. The algorithm functions such that the \( k \) nearest instances had their weight exponentially decreased with increasing rank. Rank of the nearest instances was determined by the distance from the randomly selected instance.

BSS/WSS

The BSS/WSS filter algorithm ranks features based on the ratio of the sums of squares differences between and within groups. The algorithm was implemented in R such that the ratio values for the individual features were calculated based on the following formula [23]:

\[
\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k (\bar{x}_{kj} - \bar{x}_j)^2}{\sum_i \sum_k (x_{ij} - \bar{x}_{kj})^2}
\]

Where \( \bar{x}_{kj} \) represents the mean value of all observations that belong to class \( k \) and \( \bar{x}_j \) represents the mean value of all observations. \( x_{ij} \) represents the value of the \( i \)-th observation. Once the ratios had been calculated, the features were ranked by their ratio values from highest to lowest. The final feature ranking was then used in later stages of the two experiments.
6.2.5 Support Vector Machine

Both experiments utilized a Support Vector Machine (SVM) with a Radial Basis Function kernel, for classification. The SVM was implemented using the “caret”-package [24] available in R. The classifier was trained on the training sets using repeated 10-fold cross validation. Suitable values for the tuning parameters of the SVM (sigma and cost) were found using grid-search with exponentially growing sequences of sigma and cost (for example, \(\text{cost} = 2^{-5}, ..., 2^{15}\) etc.) as proposed by Chih-Wei Hsu et al [25].

Once the SVM had been trained on the training set it was applied to the test set and its performance recorded. The performance of the SVM were then used to evaluate the quality of the selected feature subsets.

6.2.6 BIRS

The Best Incremental Ranked Subset (BIRS) algorithm was implemented in R based on the pseudocode presented below. It utilized the previously described filter algorithms (ReliefF and BSS/WSS) for ranking the features. Once a feature ranking had been established, a subset was incrementally generated by testing one feature at a time, starting with the highest ranking one. If the “new” feature improved the performance of the classifier it was kept in the set, else it was discarded.

The original algorithm tests all features of the dataset, however, due to the computational time required to train the classifier once using the available hardware, it was unreasonable to test all features. Therefore, this implementation only tested the top \(k\) ranking features. By observing the results of the first experiment, conclusions regarding suitable \(k\) values could be made. It was decided that the top 30 ranking features for the “Side Wind”-dataset and the top 60 ranking features for the “Hidden Exit & Car From Right”-dataset would be tested. This was because improvement in accuracy was observed in the first experiment once the subsets contained fewer features than the selected \(k\) values (see section 7.1).

Pseudocode for the BIRS algorithm:

```plaintext
Rankings = ReliefF or BSS/WSS(trainingSet)
BestPerformance = 0
SelectedSubset = []
k = 60 //For the HE & CFR dataset
k = 30 //For the SW dataset
for i = 1 to k{
    TempSubset = SelectedSubset + Rankings[i]
    SVM = trainSVM(TempSubset)
    TempPerformance = testSVM(SVM, testSet)
    if TempPerformance > BestPerformance then{
        BestPerformance = TempPerformance
        SelectedSubset = TempSubset
    }
}
```
7. Results

In this section, the results of the experiments are presented. Table 1 shows the highest classification accuracy achieved by using feature subsets generated by the feature selection algorithms. Classification accuracy was improved across the board using feature selection, with the exception of one case, where a reduced feature set achieved the same accuracy as all features. For the “Hidden Exit and Car From Right”-dataset the ReliefF algorithm performed the best in terms of classification accuracy as the generated subset achieved 66%. However, for the “Side Wind”-dataset, the BIRS algorithm utilizing the BSS/WSS feature rankings performed the best with an accuracy of 61%.

Table 1. Results Table

<table>
<thead>
<tr>
<th></th>
<th>HE &amp; CFR</th>
<th>SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Features</td>
<td>62%</td>
<td>38%</td>
</tr>
<tr>
<td>ReliefF</td>
<td>66%</td>
<td>47%</td>
</tr>
<tr>
<td>BSS/WSS</td>
<td>62%</td>
<td>53%</td>
</tr>
<tr>
<td>BIRS &amp; ReliefF</td>
<td>64%</td>
<td>55%</td>
</tr>
<tr>
<td>BIRS &amp; BSS/WSS</td>
<td>64%</td>
<td>61%</td>
</tr>
</tbody>
</table>

7.1 Experiment 1 – Filter Algorithms

Here follow the results from experiment 1, where the filter algorithms ReliefF and BSS/WSS was utilized. The results are presented separately for each of the two datasets used in this thesis.

Hidden Exit and Car From Right

Using all features in the “Hidden Exit and Car From Right”-dataset resulted in a classification accuracy of 62% on the test set. By using the top 15 features produced by the ReliefF algorithm, a classification accuracy of 66% was achieved. No increase in accuracy occurs by using any top $k$ ranking features for BSS/WSS. However, the feature set can be reduced to only the top 80 features whilst still achieving equal accuracy to that of all features (see figure 12).

Features selected using the ReliefF algorithm:

A classification accuracy of 38% was achieved by using all features in the “Side Wind” dataset. Reduction of the feature set based on rankings produced by the filter algorithms resulted in a significant increase of accuracy. Using the top 2 ranking features produced by BSS/WSS, resulted in an accuracy of 53%. Both the top 5 and top 20 features produced by the ReliefF algorithm resulted in a classification accuracy of 47% (see figure 13).

**Features selected using the ReliefF algorithm:**

1. "fc6_th_by_bt"    "cz_th_al_by_bt"    "cp1_theta"
4. "cp6_al_by_bt"    "p7_gamma"

**Features selected using the BSS/WSS algorithm:**

1. "p4_gamma"    "o2_delta"
Here follow the results from experiment 2, where the hybrid feature selection algorithm BIRS was utilized. The results are presented separately for each of the two datasets used in this thesis.

Hidden Exit and Car From Right

When applied to the “Hidden Exit and Car From Right”-dataset using the ReliefF feature rankings, the BIRS algorithm selected 5 features and achieved a classification accuracy of 64%. Which is slightly lower than the 66% accuracy achieved by simply using the top 15 ranking features. Using the BSS/WSS feature rankings, 10 features were selected that resulted in an accuracy of 64%, which is a slight increase from the 62% achieved by using the top 80 ranking features.

Features selected using the BIRS algorithm with ReliefF feature rankings:

[1] "f8_theta"     "fpz_th_by_bt"     "fc2_alpha"
[4] "f8_th_al_by_bt"   "f8_gamma"

Features selected using the BIRS algorithm with BSS/WSS feature rankings:

[1] "cp6_beta"  "fc1_delta"  "cp2_gamma"
[4] "pz_gamma"   "c4_delta"   "fc2_th_al_by_al_bt"
[7] "p8_gamma"  "fz_th_al_by_al_bt"  "fc2_gamma"
[10] "cp6_delta"
For the “Side Wind” dataset, the BIRS algorithm produced feature subsets with higher accuracies than those achieved by only using ReliefF and BSS/WSS. Using the ReliefF rankings, 5 features were selected and a classification accuracy of 55% was achieved, which is an increase from the 47% achieved by using the top 5 features. Using the BSS/WSS rankings, 4 features were selected and an accuracy of 61% was achieved. Once more, an increase from the 53% achieved by using just the top 2 ranking features.

Features selected using the BIRS algorithm with ReliefF feature rankings:
1. "p7_gamma"
2. "cp6_al_by_bt"
3. "fc6_th_by_bt"
4. "fz_theta"
5. "fc5_th_al_by_bt"

Features selected using the BIRS algorithm with BSS/WSS feature rankings:
1. "p4_gamma"
2. "o2_delta"
3. "c4_th_al_by_bt"
4. "cp6_gamma"
8. Discussion
The results show that the utilized feature selection algorithms improved classification accuracy in all cases except one, where the accuracy remained the same as for all features. The most significant improvement was observed in the “Side Wind”-dataset where classification accuracy was increased by twenty-three percentage points using the BIRS algorithm with BSS/WSS feature rankings. The ReliefF algorithm performed the best for the “Hidden Exit and Car From Right”-dataset with a modest accuracy improvement of four percentage points. Out of the three feature selection algorithms used, the hybrid method BIRS performed the best overall since it generated the two highest performing subsets for the “Side Wind”-dataset as well as the second and third highest performing subsets for the “Hidden Exit and Car From Right”-dataset.

Regarding the research question of “Which types of feature selection methods are suitable for the datasets?”. The filter algorithms were a suitable choice because of their low computational demand. Whilst the filters improved classification accuracy, the hybrid algorithm BIRS performed even better. This can be expected since it evaluates different combinations of features using the feature rankings generated by the filter algorithms as a basis. Although the BIRS algorithm have higher computational demand, it is considered the most suitable feature selection algorithm for the datasets used throughout the thesis work. However, it is plausible that a slightly better performing feature subset could be found by a wrapper algorithm that evaluates more feature combinations than the BIRS algorithm. Due to the computational demand of such algorithms as well as the limitation of hardware and time, no such algorithms could be evaluated.

Regarding the second research question of “Which subset of features are most relevant for classification of a driver’s cognitive load?”. No single subset of features could be identified as most relevant for classification of cognitive load experience by drivers. This is based on the fact that the highest classification accuracies achieved across the experiments only reached 66% for the “Hidden Exit and Car From Right”-dataset and 61% for the “Side Wind”-dataset. Classification using all features i.e. when no feature selection was performed, resulted in accuracies of 62% for the “Hidden Exit and Car From Right”-dataset and 38% for the “Side Wind”-dataset. Even though the feature selection algorithms improved classification accuracy for both datasets, the highest achieved accuracies were not enough to confidently predict drivers cognitive load. Furthermore, the difference in accuracy between the various subsets were small even though they shared no features. The two highest performing subsets for the “Hidden Exit and Car From Right”-dataset, produced by ReliefF and BIRS with BSS/WSS rankings, are completely unique i.e. they have no shared features, even though they only differ in accuracy by two percentage points (66% and 64%). The same is true for the “Side Wind”-dataset, where the BIRS algorithm generated two completely unique subsets that only differed in accuracy by six percentage points (61% and 55%). Additionally, by comparing the top performing feature subsets for each of the two datasets, no shared features could be found. This could indicate that none of the identified features are relevant for cross-task classification of cognitive load.

Although the implemented feature selection algorithms improved classification accuracy for both datasets, the subsets only achieved accuracies in the range of 53-66%, which is too low to confidently identify features as truly relevant to cognitive load. The overall
low classification accuracies achieved for the two datasets can be explained by the results of the data analysis (see section 5.4), which suggests that none of the individual features had strong discriminative power when trying to separate observations of high and low cognitive load. It is possible that the poor classification results could be improved by adding more observations of the two classes to the datasets, this would allow the classifier to have more data for training. Another thing that could be used to improve classification, is the introduction of other indicators of cognitive load. By combining the EEG-features, which perform poorly on their own, with other indicators, it is possible that higher classification accuracies could be achieved.

9. Conclusions
In this thesis, feature selection was utilized to investigate the relevance of EEG-signal features to the classification of cognitive load experienced by drivers. A data analysis was conducted to create an understanding of the datasets as well as to make decisions about suitable feature selection algorithms. Considering the limited computational power of the available hardware and the results of the data analysis, two filter algorithms (ReliefF and BSS/WSS) and one filter/wrapper-hybrid (BIRS) were deemed suitable for the datasets.

Two experiments were designed and implemented to identify which features were most relevant to classification of cognitive load. All chosen feature selection algorithms produced subsets that increased the classification accuracy in comparison to using all features. The highest classification accuracy for the “Hidden Exit and Car From Right”-dataset was 66%, which was achieved using the ReliefF algorithm. For the “Side Wind”-dataset an accuracy of 61% was achieved using BIRS with BSS/WSS feature rankings.

Several subsets generated by the feature selection algorithms achieved similar classification accuracies even though they were completely unique i.e. they shared no features amongst each other. Additionally, the overall low classification accuracy (66% and 61%) suggests that the features have weak discriminative power when trying to separate observations of high and low cognitive load. As a result, no single feature subset could be identified as most relevant to classification of cognitive load experienced by drivers.

9.1 Future Work
As mentioned in the discussion section, the hardware used to perform the experiments had quite limited computational power which made it unreasonable to use wrapper algorithms. As such, it would be interesting to see, given sufficient computational resources, if better performing feature subsets could be found by a wrapper algorithm that evaluates more feature combinations than the BIRS algorithm.

Since most individual features appear to have weak discriminative power for separating the classes of the datasets (see section 5.4) it could also be interesting to investigate if the EEG-features could be combined with other indicators of cognitive load to improve classification accuracy.
References


