Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics

Designing An Embedded Feature Selection Algorithm For A Drowsiness Detector Model Based On EEG Data

Scientific Students' Association Report

Author:

Blanka Bencsik

Advisor:

István Reményi
Dr. Márton Szemenyei

2022

Contents

Abstract

Nowadays, the majority of road accidents are caused by errors due to driver fatigue. This reduces the safety of traditional driving, and also limits the widespread adoption of self-driving cars. Therefore, the monitoring and early detection of drivers' drowsiness plays a key role in the process of driving automation. The development of a robust and reliable drivers' drowsiness detector system is currently an open issue in this research field.

Several relevant indicators of fatigue exist, such as information derived from subjective self-assessment, expert assessment, reaction time and physiological signals (ECG, EEG, breathing, etc.), all of which at every timestamp can be jointly represented as large feature vectors in practice. Most likely these feature vectors contain redundancy, which, in addition to making the task of fitting a machine learning model to the problem challenging, decreases the problem's perspicuity and also the subsequent testability and development of the system. Thus, dimensionality reduction plays a vital role when talking about practical application.

The goal of my work is to design and implement a robust feature selection algorithm that can be later utilized as a building block in a system development of a drowsiness detector by highlighting the most contributing feature subsets. Based on the literature, EEG is one of the best indicators of fatigue, and, due to the characteristics of the sensor used to measure it, several features can be obtained from it describing human brain functions [27]. Therefore, I choose to work with this physiological signal. The selected public database enables the detection of two-state drowsiness, from which I obtain the EEG features used for drowsiness detection that appear in the literature. The expectation from the feature selection method is to determine the smallest feature subset with which the detector model can achieve comparably good performance as with using all features.

To solve this problem, I am designing an embedded feature selection algorithm inspired by a SOTA solution. It relies on a so-called Feature Prune Layer, that can be placed in front of the first layer of an arbitrary neural network. Its weights are point-to-point related to the input features, so each of them is meant to represent the importance of the corresponding feature. During model training, these weights change depending on the actual relevance of the input, according to the usual process of neural network updates. If certain conditions are met, the weights are deleted iteratively, until the desired number of features is reached. The method, therefore, ensures to reveal complex, non-linear relations between features during the training of the detector network.

The conducted work involves the feature extraction from the physiological data, the goal-directed modification of the initial SOTA method, and finally, the evaluation of the results. The achieved results are evaluated according to the performance of the models trained on the original and reduced feature sets and the credibility of the selected features based on the literature.

Chapter 1

Introduction

Several factors might be the cause of driver drowsiness, including sleep deprivation, physical exhaustion, medication side-effects and monotony. The last one is even more significant in the case of automated driving, where, due to the lack of active involvement, the driver is prone to become fatigued. At SAE level 2 and SAE level 3 of automated driving, the driver is out of the loop for prolonged periods, however they are expected to take over the control in certain scenarios [41]. It might lead to serious consequences if the driver is not alert and misses this action. Therefore, the ability of detecting drivers' drowsiness not only increases the safety of manual driving, but it facilitates the widespread adoption of automated driving. For such reason, the development of a robust and reliable machine learning-based driver assistant drowsiness detector system is a currently active, widely studied research topic.

Nevertheless, for the identification of drivers' drowsiness, various methods have been proposed using different indicators, such as subjective self-assessment, expert assessment, reaction time measurements, the percentage of eyelid closure over the pupils (PERCLOS) and different physiological signals, like electroencephalograms (EEG) describing brain function, electrooculograms (EOG) representing eye movements, electrocardiogram (ECG) representing heart waves, breathing, etc [27]. Unfortunately, in practice, all these indicators can be described as large feature vectors at every timestamp, which enlarges the input data's dimensionality significantly, also it is likely to contain redundancy. When dealing with machine learning problems, high dimensionality raises various issues, for example it increases the space and computational complexity, makes the clustering of similar features challenging, increases the risk of overfitting the machine learning model [39], moreover, it decreases the perspicuity and the testability of the given system. To overcome these issues, many dimensionality reduction methods exist, focusing on different main goals.

The goal of my work is to design and implement a robust feature selection algorithm that can be later utilized for the development of a drivers' drowsiness detector. For this purpose, an adequate data set has to be selected that can model the original problem well enough. EEG-based features have been proven to be one of the best indicators of drowsiness as a drowsiness detector model is able to provide accurate predictions when trained with EEG only without other sources of information [23]. In addition, due to the characteristics of the sensor used to measure it, large number of features can be obtained from it, therefore, I work with a data set that contains raw EEG data and is labeled with two-state drowsiness level values.

The proposed feature selection method was inspired by a SOTA embedded features selection algorithm which exploits the neural network updates' working principle for selecting the features with the highest predictive power, namely that its weights change depending on the actual relevance of the input. It relies on a so-called Feature Prune Layer, that can be placed in front of the first layer of an arbitrary neural network. Its weights are point-to-point related to the input features, so each of them is meant to represent the importance of the corresponding feature during the whole training process. The least important weights are deleted iteratively if certain conditions are met, until the desired number of features is reached.

The proposed method ensures to reveal complex, non-linear relations between the features during the training of the detector network and maximizes the amount of drowsiness-related information extracted from a set of EEG features that was extracted from the raw signal. As a result, I was able to reduce the feature number by 95 % with a minor deterioration in the model's accuracy and to produce a more accurate prediction when deleting 80 and 90 % of the initial features. Furthermore, the efficiency of the the proposed method is also proven by the fact that it outperforms the widely popular Principal Component Analysis feature selection algorithm.

In this paper the reader can first find a literature overview (Chapter 2) about the basics of working with EEG signals and how they can be utilized for drowsiness detection, the high dimensionality-related problems in the field of machine learning and basics of feature selection methods. This is followed by a brief introduction of the SOTA embedded feature selection method that primarily inspired my work. After that, I define the problem to be solved and briefly introduce the proposed method's architecture in Chapter 3. In Chapter 4, we can find detailed description about the goal-directed modification of the initial SOTA method in order to make it suitable for solving the defined problem. Finally, in Chapter 5, I present the achieved results and evaluate them in terms of the performance of the models trained on the original feature set and on reduced feature sets produced by traditional feature selection algorithms and by our method, and the credibility of the selected features based on the literature.

Chapter 2

Literature Overview

2.1 Driver Drowsiness Detection

The field of drivers' drowsiness detection have been actively studied in the past decades and several solutions have been proposed. Drowsiness detection methods are commonly grouped into the following categories based on the source of the data used for the detection:
1) behaviour-based
2) vehicle-based
3) physiological signal-based
4) hybrid methods.
The non-invasive behavior-based methods measure fatigue levels using parameters like eye closure ratio, eye blinking, head position, facial expressions and yawning. From these parameters behavioral features are extracted with the help of cameras and computer vision techniques. One of the most frequently used metric in this category is the Percentage of Eye Closures (PERCLOS) which is the ratio of eye closure over a period. Vehicle-based methods aim to detect fatigue from the different states of the vehicle, such as lane changing patterns, speed variability, steering wheel angle, etc. To collect these type of data, the employment of various sensors is required on the vehicle's different parts. Physiological signal-based approaches detect drowsiness based on the subjects' physiological condition, such as heart rate, brain changes, respiration, body temperature, etc. In order to measure these invasive biological parameters electrodes need to be places on the subjects' body [38].

The classification method determines the resolution of the detection: threshold-based and binary classification methods distinguish between drowsy and alert stages, while multi-class classification methods can predict several levels of fatigue. Multi-class classifiers are more suitable for estimating the severity of drowsiness, hence they can detect the drowsiness in its early stage and provide early warning. Unlike the aforementioned methods that predict discrete labels, regression methods can estimate continuous variables. The most widely used decision making models are radial basis function (RBF), support vector machine (SVM), artificial neural network (ANN), fuzzy interference (FI), linear discriminant analysis (LDA), receiver support vector regression (SVR), multiple linear regression (MLR), self-organizing neural fuzzy inference network (SONFIN), etc. [27].

In the case of supervised machine learning based decision making models, ground truth is used to label the training data; to determine the true drowsiness value of the events. Having a reliable ground truth is crucial as its reliability and precision directly implies the same characteristics of the decision making model. Ground truth can be obtained

by subjects' self-assessment, expert rating, reaction time and physiological signals [27]. In many studies, EEG has been reported to be the most reliable indicator of drowsiness as it directly describes the drivers' physical state [6][22]. However, its main drawback is that it requires sensors to be attached to the drivers' body, which may obstruct them. In addition, EEG signals might vary based on the subjects' age, gender, physical state, etc [35].

All things considered, several aspects has to be taken into account when developing a drowsiness detector system. Usually, the data acquisition is cumbersome and expensive as it requires either an environment simulator or a vehicle equipped with all the necessary pricey sensors. In addition, most of the measures used for producing ground truth data are highly subject-dependent. These factors altogether make the development of an effective, reliable drivers' drowsiness detector extremely challenging.

2.2 Electroencephalogram (EEG) Features

2.2.1 Measuring EEG Signals

Electroencephalograpy measures the electrical activities of different brain regions using surface electrodes placed on the scalp. Elecroencephalogram (EEG) is a graphic display of potential differences between two sites of the brain recorded over time [21]. EEG can be used to diagnose several medical conditions, such as epilepsy, Parkinson's Disease, autism, anxiety, sleep disorders and insomnia an many more. Moreover, different fields of research have also utilized it, namely brain-computer interfaces, biometrics, neuroscience and clinical applications, neuromarketing [45].

The International Federation of Clinical Neurophysiology standardized the electrode placement into the so-called 10-20 system. This system requires the use of at least 21 electrodes and enables the measurements to be proportional to the size and shape of the skull, provides an adequate coverage of the entire head and expresses the electrode designations in terms of brain areas. The designations consist of a letter which refers to the region of the brain (F: frontal, C: central, T: temporal,P: posterior, and O: occipital) and from a number which differentiates between left and right homologous regions - odd numbers indicate the left, even numbers indicate the right hemisphere, while "z" designation refers to the midline - in such way, that lower numbers reflect positions closer to the midline (Figure 2.2) [24].

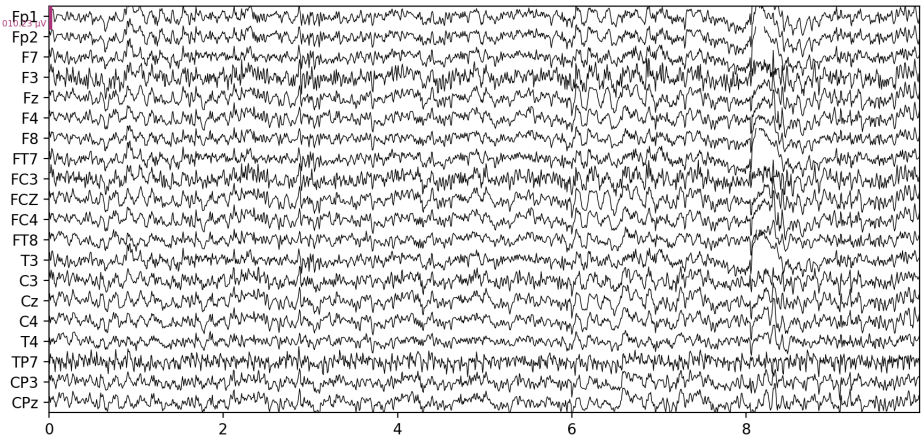

Figure 2.1: Raw EEG signals recorded on various electrodes.

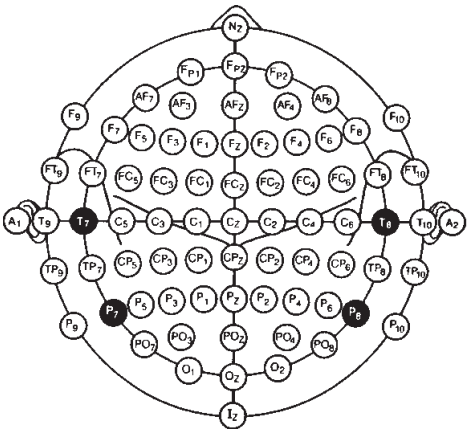

Figure 2.2: The ten-twenty electrode system of the International Federation [24].

2.2.2 EEG Feature Extraction

A wide range of features can be extracted from a raw EEG data (Figure 2.1) describing its characteristics which are used in different applications, however, in this paper, I only focus on features relevant to drowsiness detection. Based on [46] these can be categorized into the following groups: time-domain (mean, median, variance, skewness, number of zero-crossing, etc), frequency-domain, nonlinear, entropies, undirected spt., directed spt., complex networks, of which FFT-based features are the most commonly used in drivers' drowsiness detector systems (most popularly using 1 min time windows to extract the features) [27]. The Power Spectral Density (PSD) of the signal plays a vital role in calculating the frequency-domain features. It can be obtained with the Fast Fourier Transform algorithm (FFT) [34]), Welch's method [48] or Thompson multitaper method. Besiedes the widely favoured Fourier Transform, the signal can be transformed from time-domain to the frequency domain using wavelet decomposition [3] or matching pursuit decomposition [11] as well. While Fourier Transform decomposes the signal into sinusoids, in the case of wavelet decomposition, the decomposition is done by an underlying mother wavelet function. According to [46] the most frequently used frequency-domain features in all fields of EEG analysis are the relative powers of the most commonly used frequency bands, namely: delta (δ, 0.5–4 Hz), theta (θ, 4–8 Hz), alpha (α, 8–12 Hz), beta (β, 12–30 Hz), and gamma (γ, >30 Hz). However, different ratios between these bands also appear in EEG signal analysis: $\frac{\theta+\alpha}{\beta}$, $\frac{\alpha}{\beta}$, $\frac{\theta+\alpha}{\alpha+\beta}$, $\frac{\theta}{\beta}$, $\frac{\theta}{\theta+\alpha}$, $\frac{\alpha}{\theta+\alpha}$, $\frac{\theta+\alpha}{\theta+\beta}$ [4][32].

2.3 Feature Selection (FS)

2.3.1 Dimensionality Reduction in ML

In today's digital era tremendous amount of data is generated in every second with high dimensional features which are ubiquitous in various data science fields. When applying data mining and machine learning models on high dimensional data, the Curse of Dimensionality (COD) phenomenon is likely to occur: the volume of the space increases together with the dimensionality, causing the data to become sparse [15] - this usually means the features having zero values. A model trained with sparse data is prone to learn the noise, it cannot generalize, well which leads to overfitting and performance degradation on un-

seen data [28]. Besides, high dimensional data enhances the computational burden and decreases the perspicuity and the testability of the given problem.

To alleviate the aforementioned obstacles, many dimensionality reduction techniques have been introduced so far. During the dimensionality reduction process, the features that are redundant and not relevant to the task are omitted, yielding a more compact, more easily interpretable representation of the target concept with the most relevant features [2]. Dimensionality reduction is commonly categorized into two main groups: feature extraction (FE) and feature selection (FS). Feature extraction compresses the high dimensional feature set into a smaller one by constructing a new, lower dimensional feature space, usually by applying linear or nonlinear projection of the original set. It is preferred in applications where only the raw data is available which is not interpretable for a learning algorithm. However, in this case, the problem of further analysis arises, as we cannot retain the physical meaning of the new features. Feature selection, on the other hand, means the selection of a subset of relevant features from the initial set, keeping the physical meanings of the original features [28].

2.3.2 Feature Selection Categories Based on Selection Strategy

Feature selection is one of the most commonly used dimensionality reduction methods. Its general working principal consists of four main steps: generation of a feature subset, evaluation of the feature subset, checking the termination condition, result validation [29]. Feature selection methods can be categorized based on different perspectives. In terms of the availability of the labels in the training data set, feature selection methods can be divided into supervised (labels are available), unsupervised (labels are not available) and semi-supervised methods [28]. Aligned with the original problem statement, supervised solution was preferred in this study. Another categorization type relies on the selection strategy and distinguishes three main methods: filter, wrapper, embedded [30].

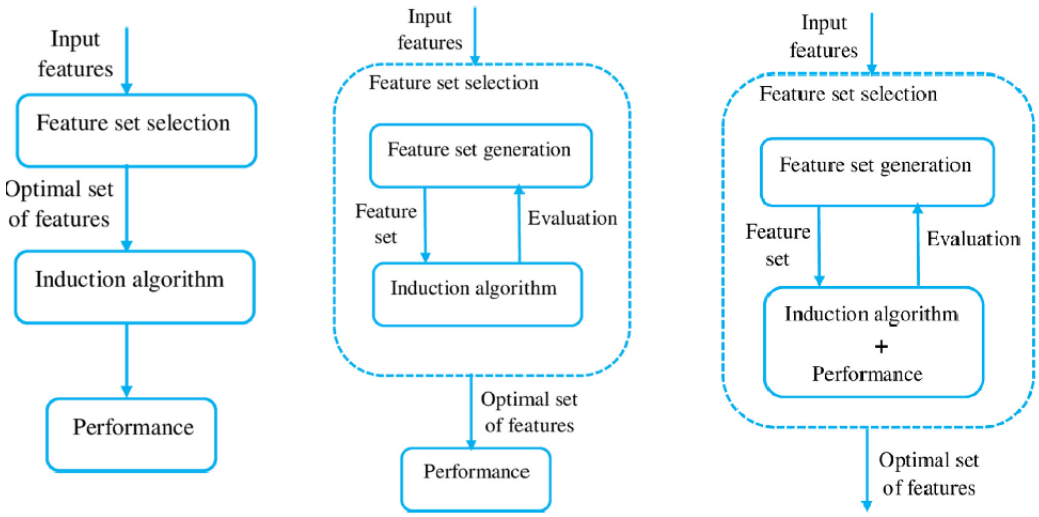

Figure 2.3: Flowchart of the feature selection process in different FS categories: Filter, Wrapper, Embedded [10].

2.3.2.1 Filter Methods

Filter methods utilize the data's intrinsic properties to assess feature importance. They calculate a score for each feature using different evaluation criteria which can be both univariate (examining each feature individually) and multivariate (multiple features are examined together in a batch). The features are then ranked according to these scores and a specified number of them with the lowest scores are filtered out, resulting the most predictive subset of features. In the case of filter methods, the selection is performed before the model training, therefore, the FS is considered as a pre-processing step. Among its most significant character traits, its independence from any learning algorithm should be mentioned, which makes filter methods usually faster than others, but raises the risk that the selected features may not be optimal for the given algorithm [28][30].

Several evaluation criteria exist for separating the features which approach the problem form different perspectives. The first option is to examine feature discriminative ability and select features such that within-class distance [17] is as small as possible while between-class distance [17] is as large as possible [31], meaning that features that strongly represent the given class and differ the most from features in other classes are selected into the subset. Some popular algorithms based on the aforementioned principle are the Fisher Score [12] and the Linear Discriminant Feature Selection [44] algorithms. Another idea is to exploit correlation measures, either to remove redundant features which can be applied in case of unsupervised learning as well, or to select the most similar - highly correlating - features to the target variable if labels are provided. For the first scenaio, Principal Component Analysis (PCA) is a widely used method which will be further discussed in this paper in Section 2.4.1. For the latter scenario, various statistical measures can be used, including Pearson's correlation coefficient (linear), ANOVA correlation coeff. (linear), Sperman's rank coefficient (nonlinear), Kendall's rank coeff. (nonlinear), Chi-Squared test and mutual information. Their applicability for a given problem depends on the data variable types.

2.3.2.2 Wrapper Methods

In contrast to the filter methods, in the case or wrapper FS methods, the learning algorithm has to be defined, in fact, wrappers exploit their black box nature to score subsets of features according to their importance and predictive power. Wrappers work iteratively, repeating the following steps until a stopping criteria is satisfied: they generate a subset from the initial features which are then evaluated with the help of the predefined learning algorithm [28]. The stopping criteria is usually defined as the combination of the desired number of selected features and the highest possible learning performance achieved when training the model with this subset.

Besides the learning algorithm and the stopping criteria, the space search strategy also has to be selected. Sequential search methods (also called hill-climbing or steepest ascent) are search strategies that use greedy techniques to examine features sequentially. They either start from the initial set of features and eliminate them one by one (sequential backward selection (SBS)) or start with an empty set and add features one by one (sequential forward selection (SFS)). One shortcoming of this method is that it can only guarantee local optimality. Genetic algorithms add some randomness to the search procedure, hence help to overcome the local optimum problem [1][30]. Other feature subset selection algorithms are the best-first search, branch-and-bound search, etc. [33][1].

Unfortunately, it is perceptible that in terms of speed, wrapper methods are not so efficient, due to the huge search space - 2^N where N is the number of features [30] -, which is even more problematic when dealing with very large sets of features. While some criticize this property of theirs and blame it for wrappers' rare application in practice [28], others claim that choosing an efficient search strategy can alleviate this obstacle [14].

2.3.2.3 Embedded

The embedded feature selection method is a trade-off between filter method's high speed but low accuracy and wrapper method's high accuracy, but expensive computational requirements. According to its descriptive name, in the case of embedded methods, the feature selection is integrated into the selected machine learning model's training procedure; the best feature subset is produced during the training of the chosen learning algorithm. Therefore, the performance of the model highly depends on the selected features. It has the merits of interacting with the model, but due to the lack of iterative feature subset evaluation, it is significantly more efficient than wrapper methods [28]. Similar to the wrappers, embedded methods are also not confined to supervised feature selection and can be applied for unsupervised feature selection [30].

The most widely used embedded methods are the regularization methods which aim to minimize fitting errors in order to fit the model to the feature set. To do so, they force feature coefficients to be as small as possible simultaneously [28]. Some popular example for the regularization approach are the LASSO, RIDGE and Elastic Nets. An embedded feature selection can also be done by any kind of tree-based algorithm, such as Decision Tree, RandomForest, ExtraTree, etc [40].

2.4 Feature Selection Methods Used In This Paper

In this section, next to the state-of-the-art feature selection method that inspired my work in the first place, I am going to introduce the widely popular PCA algorithm that was used as baseline in this paper in order to evaluate the performance of the proposed method.

2.4.1 Principal Component Analysis (PCA)

As mentioned previously, Principal Component Analysis (PCA) is a widely used - due to its easy application and non-parametric property - dimensionality reduction method that relies on linear algebra techniques. It projects the original, high-dimensional data into new dimensions in order to re-express it and explore hidden qualities. In a mathematical sense, the goal is to find the most meaningful basis by performing basis change transformations [42].

To better understand its working principle, imagine the data as an m $\times$ n matrix (X), where m denotes the number of variables (features) and n is the number of data points or samples. The first step of the PCA is to examine the correlation between the variables: to understand how the different variables vary from the mean with respect to each other. This is done by computing the the covariance matrix ($X_{cov} = X^T X$). The next step is to find the principal components. Principal components are uncorrelated, new variables constructed as linear combinations of the initial variables in such a way, that the information they contain is compressed into the first components. Geometrically, principal components express the directions of the data where the amount of variance is maximal.

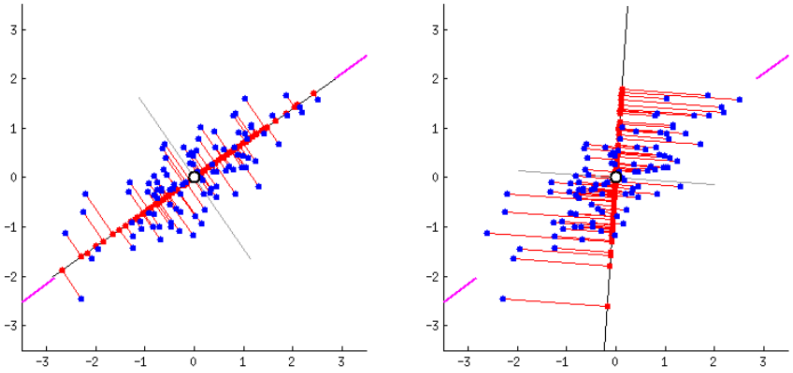

Figure 2.4: The first principal component is the direction along the pink line: the dispersion of the data points is the largest along that line (first graph). Along other directions (second graph), the dispersion of the data points is smaller [18].

As demonstrated in Figure 2.4, the first principal component is the direction along the pink line, as the dispersion of the data points is the largest along that line. In practice, finding these directions is achieved by determining the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors indicate the the directions of the axes where there is the most variance (principal components), while eigenvalues describe the amount of variance in each principal component. The ranking of the eigenvectors gives corresponding eigenvectors - hence the principal components - in order of significance. After that, we keep as many eigenvectors as the desired number of final features (P). The final step is to transform the original data set onto the axes represented by the principal components, which is done by a matrix multiplication: $X_{new} = PX^T$ [19][42].

2.4.2 Stepwise Weight Pruning Algorithm (SWPA)

SWPA is a novel embedded feature selection method proposed by [20]. Its main idea is to incorporate a so-called drop-in layer into a neural network architecture and prune its weights iteratively until the most important ones are left. Weight pruning refers to the process of removing parameters from an existing, accurate network. The method exploits the neural network updates' working principle, namely that its weights change depending on the actual relevance of the input. If the drop-in layer ($W \in \mathbf{R}^{1\times d}$ where d is the number of input elements) is the first layer in the network, and its weights are initialized to ones, the output of this layer $O = \{w_1x_1, ..., w_dx_d\}$ will be the multiplication of the corresponding input elements. Hence, if we set a weight w_i to 0 in the drop-in layer, that directly means that we removed input element x_i. Algorithm 1 summarizes the method's working principle.

The SWPA has been tested with 3 different data sets:
1) Smartphone Dataset for Human Activity Recognition (HAR) which contains different smartphone sensor data (e.g. accelerometer, gyroscope, etc.) recorded while subjects were performing basic activities like walking, lying, etc.
2) ISOLET which is a speech dataset containing recordings of subject pronouncing each letter of the alphabet
3) MNIST handwritten digits from 0-9 dataset which consist of 28× 28 gray-scale centered images.

Algorithm 1 Original Stepwise Weight Pruning Algorithm (SWPA) [20].

Input: training data $X \in \mathbf{R}^{n \times d}$, training labels Y, base network $f_\theta(.)$, Drop-in Layer W, Step Counter $n \in \mathbf{Z}_{\geqslant 1}$. Selection factor $f \in [0, 1]$
for *count* in $1, ..., n + 1$ **do**
  $O = \{w_1 x_1, ..., w_d x_d\}$
  if *count* > 1 **then**
    $k \leftarrow \frac{(1-f)*d}{n}$
    Sort the weights W of the Drop-in Layer based on their absolute value.
    Set the least k of them to 0.
  end if
  Train the base network on O
end for
Take the features corresponding to the top f fraction of the weights in W based on their absolute value and train them on the base network.

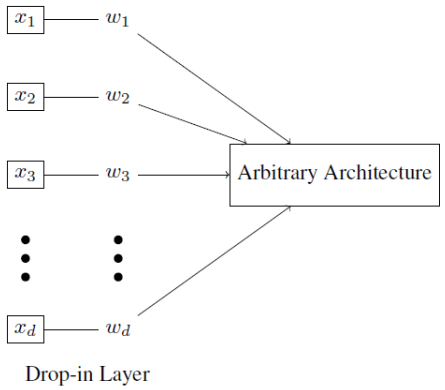

Figure 2.5: Modified network with the drop-in layer [20].

For the experiments they use a 3 layer feedforward neural network with a reduction factor of 2 which is trained for 20000 epochs if there is no degradation of the performance on any continuous set of 2000 epochs. The most important variables in the SWPA are:
1) the Step Counter (n): the features that have to be deleted in order to reach the desired number are removed in equal-sized groups in n steps.
2) the Selection Factor (f) which defines what percent of the initial feature number should be selected into the final subset. According to the paper, these variables are set to the following values: $n = 4$, $f = 0.1$.

To evaluate the achieved results, besides the random assignment, they use the Permutation Feature Importance (PFI) as a baseline with the number of random permutations of 10 which is an importance attribution technique commonly used for random forests. SWPA outperforms both the random assignment and the PFI on all datasets when the 10% of the original number of features are selected: for example on the MNIST dataset it yielded 0.941 accuracy, while using PFI the achieved accuracy was 0.893 and with random selection 0.714. With these results SWPA has proved itself to be a simple, yet efficient embedded feature selection method which is easy to apply in various tasks as the drop-in layer can be incorporated into any meural network architecture [20].

Chapter 3

Problem Statement

The goal of my work is to design and implement a robust feature selection algorithm that can be later utilized during the development of a drivers' drowsiness detector. According to the outstanding results achieved by SWPA introrduced in Subsection 2.4.2, its easy applicability and embedded property, I have found it to be a satisfactory choice for the basis of the designed FS method. However, the paper stays vague about the implementation of the drop-in layer. Although the description states clearly that the feature scoring depends on the weights in the drop-in layer which is the first layer of the used neural network architecture - therefore its weights change according to the importance of the input elements -, by observing Algorithm 1, the drop-in layer seems to be left out of the parameter update, as they always retrain the base network on its output [20].

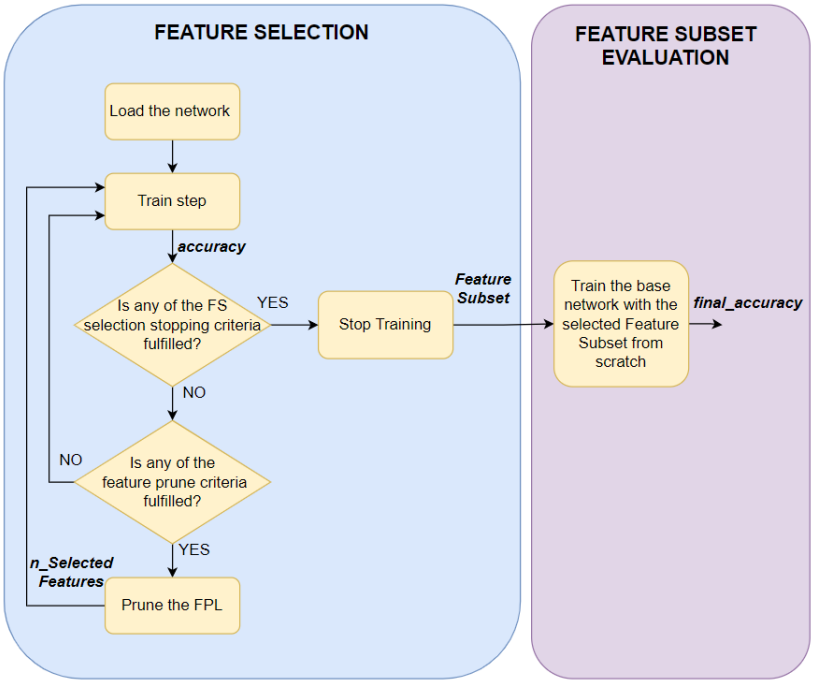

Figure 3.1: Flowchart of the proposed feature selection algorithm.

For this reason I rethink the idea proposed by [20] and complete it with additional properties in order to make it better suitable for the introduced problem. The flowchart of the final algorithm can be seen in Figure 3.1. This solution also strives to exploit the neural network updates' working principle, hence, I implement a layer similar to the drop-in

layer, called Feature Prune Layer (FPL) which is has the same size as the number of input features as is point-to-point related to them. The network is then trained until any of the FS selection stopping criteria is fulfilled. The FPL remains the part of the network for the whole training process and a pruning step is performed on it if any of the feature prune criteria is fulfilled. The pruning consist of removing the k weights from the FPL with the lowest magnitudes, where k is defined as in the case of SWPA. The feature selection part is followed by the feature subset evaluation, when the base network - without the FPL - is retrained from scratch with the selected feature subset. These two main parts are altogether considered as the proposed feature selection algorithm, and the performance is deduced from the accuracy achieved in the feature subset evaluation part.

Summarily, the proposed algorithm's task is to select the defined number of best predictive features from a set of EEG based features which are feasible for driver drowsiness detection according to the literature. To find the best setup of the algorithm, several tests have been performed examining the impact of the different hyperparameters. In the following chapter I am going to detail the aforementioned goal-directed modification of the SWPA, the feature pruning and feature selection stopping criteria, the hyperparameters in the model and the motive for their selection.

Chapter 4

Proposed Work

4.1 Data Preparation

4.1.1 Multi-Channel EEG Recordings Dataset

In this study the [8] public data set (processed) is used which is a processed version of [7] (original). The original data set contains multi-channel EEG recordings that were recorded during a sustained-attention driving task with the help of 27 subjects (aged between 22-28). During a 90-minute experiment, conducted in a VR driving environment with a dynamic driving simulator, the subjects were asked to keep the car in the center of the lane and respond quickly to the randomly introduced lane-departure events. These perturbations made the car drift to the left or to the right side of the lane (deviation onset). For obtaining the drowsiness level of the driver, in addition to the deviation onset, response onset (the subject steering the wheel in case of a departure event) and response offset (the car arriving back to its original position) occurring times have been recorded. These indicators of the drivers' promptness are instantaneous measures of the drowsiness level that can be calculated using the method described in [47]. The EEG signals were collected with the help of a wired EEG cap (Figure 4.1) with 32 Ag/AgCl electrodes (of which 2 were used as reference) based on a modified international 10-20 system [7].

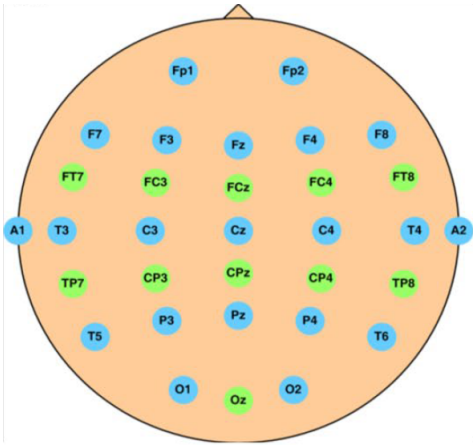

Figure 4.1: The layout of the electrodes in the EEG cap used for the experiments in [7].

The writers of paper [9] have produced the processed, balanced version of the original, pre-processed data set where the EEG data was digitalized at 500 Hz, an 1-Hz high-pass and 50-Hz low-pass filter was applied on it, followed by artefact rejection. They down-sample the EEG signals to 128 Hz, then extract equal-long, 3-second samples. Each sample was labeled with a 2-satate drowsiness level - drowsy or alert - using the aforementioned [47] method. The writers have devoted special effort to create a compact, balanced dataset, containing the most representative samples from different subjects, by carrying out the following steps:
1) they have discarded sessions where the number of samples from either class is less than 50
2) in case of multiple sessions belonging to the same subject, they have chosen the one with the most balanced class distribution
3) from each session they have selected alert samples with the shortest- and drowsy samples with the longest response time.

Step 1) and 3) ensures the balancedness of the classes, while step 2) results a balanced data from different subjects, hence, it is not likely that the classifier will be prone to favor the prediction of a specific subject. The final data set contains 2022 3 second long, pre-processed EEG samples collected from 11 different subjects [9].

4.1.2 EEG Feature Extraction

The chosen data set introduced Section 4.1.1 contains 3 second long time-domain EEG signals, referred as segments. In order to convert these signals into an interpretable format for any classification algorithm, we have to extract features that comprehensively describe the data set. Therefore, for every segment the commonly used frequency-domain EEG features introduced in Section 2.2.2 are determined:

$$\alpha\text{-PSD},\ \beta\text{-PSD},\ \theta\text{-PSD},\ \frac{\theta+\alpha}{\beta},\ \frac{\alpha}{\beta},\ \frac{\theta+\alpha}{\alpha+\beta},\ \frac{\theta}{\beta},\ \frac{\theta}{\theta+\alpha},\ \frac{\alpha}{\theta+\alpha},\ \frac{\theta+\alpha}{\theta+\beta}$$

The Power Spectral Density (PSD) is calculated with the help of the Welch's method [43], using a window size of 3 seconds. These aforementioned features are obtained from the signals measured individually on every electrode found on the EEG cap used to record the signals Figure 4.1. In addition, the calculated values are averaged over the frontal, the temporal and all the electrodes, as - according to the literature - some EEG frequency bands are more active on the frontal or on the temporal part of the brain Namely, these electrode positions are [7]:

Fp1, Fp2, F7, F3, Fz, F4, F8, FT7, FC3, FCZ, FC4, FT8, T3, C3, Cz, C4, T4, TP7, CP3, CPz, CP4, TP8, A1, T5, P3, PZ, P4, T6, A2, O1, Oz, O2, frontal, temporal, all

These calculations (Figure 4.2) have resulted a 330 element feature vector for each 3 s long EEG segment Figure 4.3. This feature set serves as the input for the designed feature selection algorithm, which aims to select the desired number of them with the highest predictive power. For the development phase, the data set has been split into train and test sets in 70 - 30 % ratio, while ensuring that the labels stay balanced by not letting the difference between the number of drowsy and alert labels to be greater than 20.

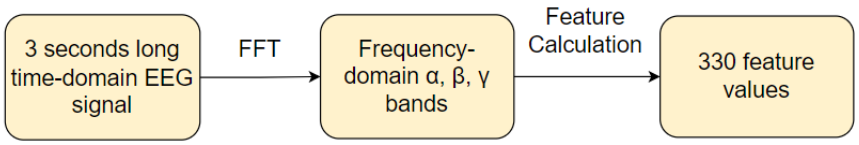

Figure 4.2: The process of feature extraction from time-domain EEG segments.

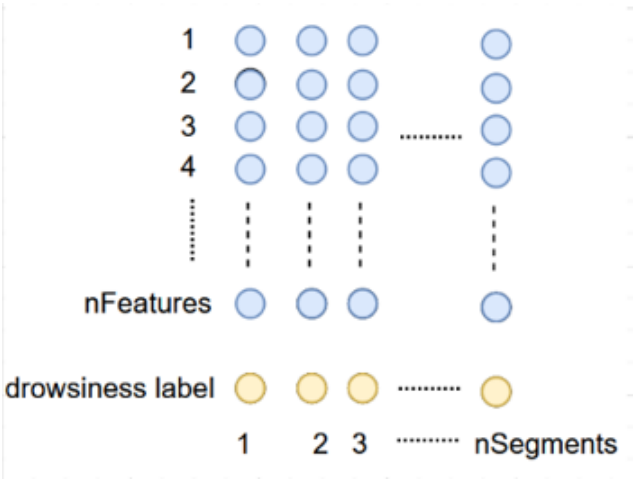

Figure 4.3: The structure of the generated EEG data set.

4.2 Used Metrics

In order to introduce the design and planning process of the proposed solution, it is crucial to keep the final goal in mind which includes the awareness of the desired outcome measured with the chosen metrics. With the intention of making the following sections easily readable, in this subsection, I introduce the the two main metrics used in this paper.

Mean Average Precision (mAP) [%]: As far as classifiers are concerned, mAP is one of the most important indicators of their performance. It is defined as the average precision calculated separately for each individual class, averaged over all the classes. Ideally, this value is determined separately for the training set during the training of the classifier and for a different test set during the validation which is carried out after each specified number of iterations (epochs).

Overfitting [%]: The difference between the train and test mAP refer to the generalization ability of the classifier. If the test mAP is lower than the train mAP, it implies the so-called overfitting phenomenon: the classifier is unable to perform well on unseen data. Here, I simply define the overfitting metric as the signed difference between the train and test mAP values. Therefore, the aim is to achieve as small overfitting value as possible.

4.3 Development Of The Feature Selection Method

4.3.1 Iterative Feature Pruning

Similarly to the SWPA (2.4.2), the feature scoring method of the proposed embedded FS method algorithm also relies on the neural network updates' working principle. A so-called Feature Prune Layer (FPL) is attached to the front of the classifier network (base network), which has the same size as the number of input features and is point-to-point related to them. Consequently, during the training, its weights change according to the importance of the input features, hence, deleting a weight from the FPL means the removal of the corresponding feature from the original feature set.

If a predefined feature pruning criterion (see Section 4.3.2) gets fulfilled, a subset of the remaining weights in the FPL will be deleted. The amount of the deleted weights in a pruning step is defined as follows:

$$N_{deleted\_weights} = \frac{f * d}{n} \tag{4.1}$$

where d corresponds to the remaining number of features in the FPL, n is a counter of the pruning steps and $f \in [0.1, 0.4]$ is a constant that directly contributes to the amount of removed weights in a given step. Choosing a higher f results a coarser pruning strategy. Nevertheless, the use of n prompts that the further we move with the training the more gentle the weight pruning gets. The feature scoring and the iterative feature pruning process is demonstrated by Figure 4.4.

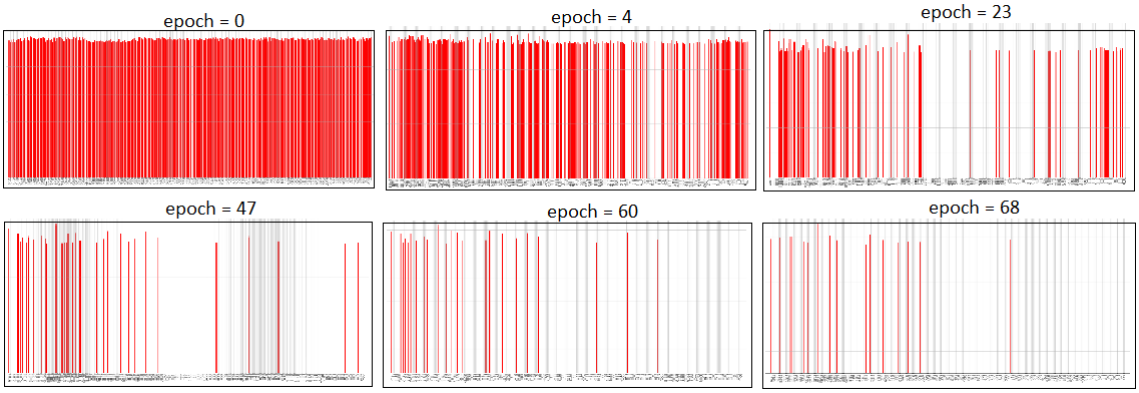

Figure 4.4: A few steps of the iterative feature pruning process. The vertical axis shows the corresponding scores for each feature. As we move forward with the training, the scores change according to the feature importance. At a given pruning step, the k number of features with the smallest magnitude is deleted, until only the desired number of features remain.

4.3.2 Feature Pruning Criteria

A pruning step is performed on the FPL if any of the following feature pruning criteria is fulfilled:
1) The test mAP (mAP) reaches a predefined value ($final\_mAP$)
2) The predefined number of epochs is reached (max_epochs)
3) The overfitting reaches a pedefined level ($max\_overfitting$)

Commonly, when a neural network model is pruned, its performance slightly drops and it needs a few iterations of training to regain its earlier accuracy. Depending on the coarseness of the weight pruning defined by Equation 4.1 and the given training phase, the degradation of the accuracy varies in the different scenarios. For example, if $N_{deleted\_weights}$ is a large number, a significant amount of the weights is going to be deleted from the FPL even in the first pruning step, which is likely to cause a heavier accuracy degradation than if it was pruned with a smaller $N_{deleted\_weights}$. In addition, the longer we train the network, the more confident it gets, meanwhile, the pruned amounts will decrease due to their inverse relationship with the step counter. Because of this, it is not ideal to train the network for the same number of epochs between each feature pruning step, as it may need dissimilar amount of iterations to regain its accuracy. As discussed previously, the test mAP is one of the best indicators of a network's performance, hence, I use it to determine the appropriate moment of the next pruning step. According to the first feature pruning criteriom, the next pruning step can be performed if the network's accuracy reaches the predefined $final\_mAP$ after the last reduction.

It is possible that the network will never be able to reach the desired $final\_mAP$ after a certain point. In order to prevent the training from getting stuck in an infinite loop, according to the second criterion, a feature pruning step may also be carried out if the network has been trained for a predefined maximum number of epochs (max_epochs) since the previous one. Lastly, a pruning step also takes place if none of the aforementioned criteria is fulfilled, but the overfitting reaches the $max\_overfitting$ threshold. This is likely to happen if $N_{deleted\_weights}$ is too small, and the pruning is performed in a slower pace than the network's regeneration ability. A summary of the selected values for the previously discussed thresholds can be found in Table 4.1.

Table 4.1: Thresholds for the feature pruning criteria.

final_mAP	**max_epochs**	**max_overfitting**
[0.65, 0.95]	20	0.05

4.3.3 Algorithm Structure

The embedded nature of the proposed FS algorithm is due to the fact that the feature selection process is carried out during the training of the classifier network, while continuously performing the feature pruning introduced in the previous subsections. The training may be terminated if any of the following feature selection stopping criteria is fulfilled:
1) The desired number (des_feat_num) of features are left in the feature set which is the ideal case
2) The network was trained for a maximum number of epochs (max_epochs_final). This

ensures that the training will not get stuck in an infinite loop if the desired number of features cannot be reached with the selected hyperparameters.

After the termination, the final subset of selected features is evaluated on the base classifier network - the same architecture but without the Feature Prune Layer. This is considered as the end of the feature selection process and the final results are the ones achieved with this step: the performance of the base classifier with the selected feature subset ($newset\_mAP$, $newset\_overfitting$). The whole process of the feature selection is demonstrated by Algorithm 2.

Algorithm 2 Proposed Feature Selection Algorithm.

Input: original EEG feature set $[1 \times k]$
$network \leftarrow$ initialize ▷ FPL to ones, rest randomly
$n \leftarrow 0$ ▷ pruning step counter
$epochs \leftarrow 0$ ▷ epochs between two pruning steps
$epochs\_final \leftarrow 0$ ▷ all epochs during training
$mAP \leftarrow 0, overfitting \leftarrow 0$
while $(k > des\_feat\_num)$ OR $(epochs\_final < max\_epochs\_final)$ **do**
  if $(mAP \geq final\_mA)$ OR $(epochs \geq max\_epochs)$ OR $(overfitting \geq max\_overfitting)$ **then**
    $k, n =\leftarrow$ perform weight pruning on the FPL
    epochs $\leftarrow 0$
  end if
  $mAP, epochs, epochs\_final \leftarrow$ train()
end while
$newset\_mAP$,newset_overfitting$\leftarrow$ take the new k-sized feature subset and train the base network on it from scratch

4.3.4 Hyperparameter Selection

While reading the previous subsections, it was perceptible that during the FS process some of the defined thresholds were handled as variables. The changing of these variables strongly influences the outcome of the FS algorithm: the composition of the final feature subset. While we might have an assumption about how the changing of these variables individually affect the outcome, the problem gets more complex if we combine them. Moreover, due to the neural networks' black box nature, they act as hyperparameters and it is impossible to define their value consequently. The proposed FS method has two hyperparameters:
1) $\mathbf{f} \in [0.1, 0.5]$ parameter which directly contributes to the amount of removed weights in a given step. Choosing a higher f results a coarser pruning strategy. (Equation 4.1)
2) **final_mAP** $\in [0.65, 0.95]$ which is on of the feature pruning criteria (see Section 4.3.2).

In order to determine the best selection of these hyperparameters for the FS method, I have conducted several experiments testing their different values introduced by Algorithm 3.

Algorithm 3 Experiments For Hyperparameter Selection

$results \leftarrow []$
for $final\_mAP$ in 0.65, ..., 0.95 **do**
  for f in 0.1, ..., 0.5 **do**
    $newset\_mAP \leftarrow$ select the $des_feat_num$of number of features from the original set with $final\_mAP$ and f
    results.append($newset\_mAP$, $newset\_overfitting$)
  end for
end for
find the best performance in $results$ and take the corresponding hyperparameters

4.3.5 Feature Prune Layer Realization

In terms of the implementation, the FPL is realized the same way as a linear layer, but instead of matrix multiplication it computes the Hadamard product between its weights and the input. When talking about neural network pruning, we can distinguish two main types based on the structure of the weight removal: structured and unstructured pruning [5]. In the case of structured pruning entire groups of weights are removed (like channels, filters or layers), while unstructured pruning corresponds to deleting weights individually by setting their value to zero. For removing weights from the FPL unstructured pruning is used. When implementing unstructured pruning in practice, such difficulty arises that during the parameter update, the zeroed weights in the networks' layers also get updated regardless of their current value. To overcome this issue, instead of actually setting the weights in the layer to zero, I use a mask with which the weights of the FPL are multiplied in every forward pass. This way, in every pruning step, only the mask gets modified, therefore there is no need to detach the FPL from the computational graph, which avoids the decrease of the algorithm's speed. The illustration of the FPL's implementation can be seen in Figure 4.5.

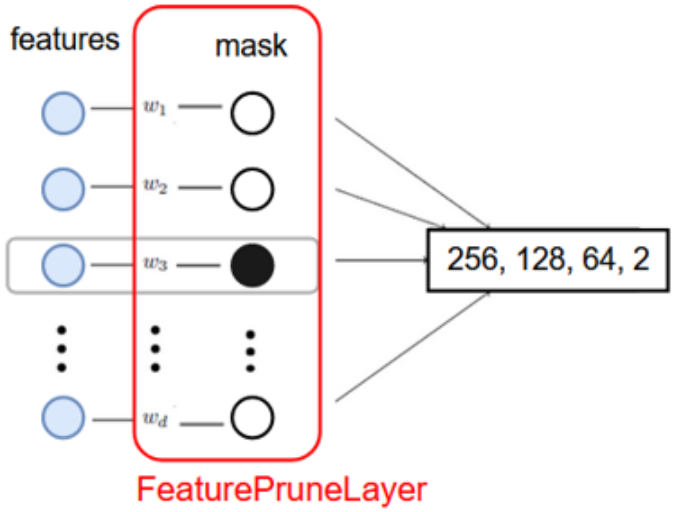

Figure 4.5: Architecture of the network used for feature selection.

4.3.6 Classifier Network

Similarly to the SWPA, I also use a small network with 4 hidden layers for the classification. In order to deduce the most beneficial size for these layers, I have trained multiple networks with different layer sizes on the whole feature set. Each training session ran for 70 epochs, using Adam optimizer, a batch size of 64 and an initial learning rate 0.001. Every architecture has been tested 4 times and the average of the achieved results is demonstrated in Table 4.2. The **do** sign between the layers in the network architectures stands for the dropout layer which serves as a regularization technique by randomly zeroing out some elements of the input with probability p using samples from a Bernoulli distribution [16]. Here, I used probability p = 0.5. The **mAP** and **overfitting** are defined the same way as described in Section 4.2. The $_{\mathbf{avg}}$ and $_{\mathbf{std}}$ extensions refer to the average and the standard deviation of the performed tests' results respectively.

Table 4.2: Results of predicting the drowsiness labels using different neural network architectures.

Name	Network Architecture	$\mathbf{mAP_{avg}}$	$\mathbf{overfitting_{avg}}$	$\mathbf{mAP_{std}}$	$\mathbf{overfitting_{std}}$
arch1	64; 32; 16; 2	0.893	0.029	0.004	0.0015
arch2	64; do; 32; do; 16; do; 2	0.881	0.016	0.0043	0.009
arch3	128; 64; 32; 2	0.907	0.032	0.0042	0.0042
arch4	128; do; 64; do; 32; do; 2	0.899	0.016	**0.0015**	**0.0013**
arch5	**256; 128; 64; 2**	**0.926**	0.037	0.042	0.004
arch6	256; do; 128; do; 64; do; 2	0.902	**0.014**	0.004	0.009

Arch4 seems to provide the most stable training as the standard deviation of both metrics is the smallest in this case. Nevertheless, **arch5** achieves the highest **mAP**, and given that the rest of the metrics do not vary significantly between the different architectures, the 256; 128; 64; 2 network architecture is chosen to be used for the further experiments.

Chapter 5

Results

5.1 Reducing The Feature Subset

The aim of the experiments introduced in this section is to find out to what extent it is possible to reduce the feature set without a major degradation in the classification performance. The top 5, 10, and 20 % of the original feature set are examined. For finding the desired number of features, the proposed FS method is tested with all the possible hyperparameter settings described in Section 4.3.4. As f varies in the range of [0.1, 0.4] with a step size of 0.1 and final_mAP varies in [0.6, 0.95] with a step size of 0.05, all their variations resulted 32 test cases for each feature subset size. Each test case resulted a feature subset, with which the base network was trained 4 times, and the achieved results have been averaged. Table 5.1 summarizes these results: the 3 bests performing test cases are shown for each feature subset, compared to the case when the classifier is trained on the original feature subset.

Table 5.1: Results of different sized feature subsets generated with the proposed FS method.

All features (330)	
mAP	**overfitting**
0.926	0.037

TOP 20 %				**TOP 10 %**				**TOP 5 %**			
Results		**Hyperparameters**		**Results**		**Hyperparameters**		**Results**		**Hyperparameters**	
mAP	**overfit.**	**f**	**final _mAP**	**mAP**	**overfit.**	**f**	**final _mAP**	**mAP**	**overfit.**	**f**	**final _mAP**
0.953	**0.033**	**0.2**	**0.95**	**0.941**	**0.028**	**0.2**	**0.7**	**0.916**	**0.01**	**0.2**	**0.75**
0.948	0.039	0.3	0.95	0.935	0.027	0.2	0.65	0.906	0.022	0.2	0.7
0.946	0.031	0.3	0.9	0.931	0.039	0.2	0.9	0.901	0.016	0.3	0.75
0.916	0.036	0.3	0.75	0.887	0.017	0.4	0.9	0.795	0.002	0.3	0.95

The results indicate that it is possible to reduce the original feature number by 95 % without significant performance degradation: in the best performing test case, the classifier's accuracy is 91.6 % accompanied by 1 % overfitting, which is just slightly worse from the results achieved when training with the entire feature set: 92,6 % mAP and 3.7 % overfitting. This proves that the proposed FS algorithm is able to select the most important features in terms of their contribution to the prediction. Moreover, when reducing the feature number by 90 % and 80 %, the classifier's accuracy increases by 1.5 % and 2.7 % respectively. At first thought, this phenomenon might be unexpected, as one would

think that the loss of information due to the reduction of the input will definitely lead to a performance degradation. However, in a recent study, it has been proven that - even linear, correlation-based - feature selection indeed can improve the performance of a classifier neural network model [37]. This might be possible as feature selection also reduces the noise in the input data, which helps the model to better generalize.

A general observation is that f has a stronger impact on the outcome than final_mAP. When training with a very small f, the pruning is performed in such slow steps, that the desired number of features cannot be reached within the maximum iteration limit. On the other hand, if its too large, the weights are removed sooner than their value would stabilize, which leads to an improper selection strategy. Changing the final_mAP does not evoke significant differences: with the fixed value of $f = 0.2$, the results of the test cases remain close to each other even when choosing vastly different values for final_mAP - see the test cases for choosing the top 10 % of the original features. Nevertheless, $f = 0.2$ and final_mAP = 0.75 seems to be an advantageous hyperparameter combination for selecting the top 10 % of the features, and as we have seen, the changing of the final_mAP does not have a significant effect on the outcome, further experiments are carried out using this setting.

5.2 Reproductibility

The EEG cap that provides the data has a quite dense electrode distribution, meaning that the signals measured on the adjacent electrodes may be similar. Therefore, the original feature set with PSD values for each electrode is likely to contain redundant information and many correlating features. Due to the random initialization of the weights in the classifier network, it can happen that in different runs, among the correlating features, different ones will be selected into the final subset. This will result slightly different performances in different runs when the feature selection is performed with the exact same settings. However, the features themselves might be different, the difference between the comprehensive descriptive power of the generated feature subsets is negligible. This is proven by the results in Table 5.2 which shows the performance of the feature selection algorithm from 4 different sessions, using the same settings in each of them: $f = 0.2$, final_mAP = 0.75, des_feat_num = 10 % of the original set. Similarly as before, with each obtained feature subset, the base network was trained 4 times, and the averaged results are presented in the table. Even though the feature subsets are not completely the same (Figure 5.1), the performances achieved in the different sessions are close to one another.

Table 5.2: Results of 4 different sessions with the same hyperparameter settings: $f = 0.2$, final_mAP = 0.75, des_feat_num = 10 %.

experiment	mAP	overfitting
repr1	0.924	0.036
repr2	0.932	0.025
repr3	0.938	0.025
repr4	0.927	0.027

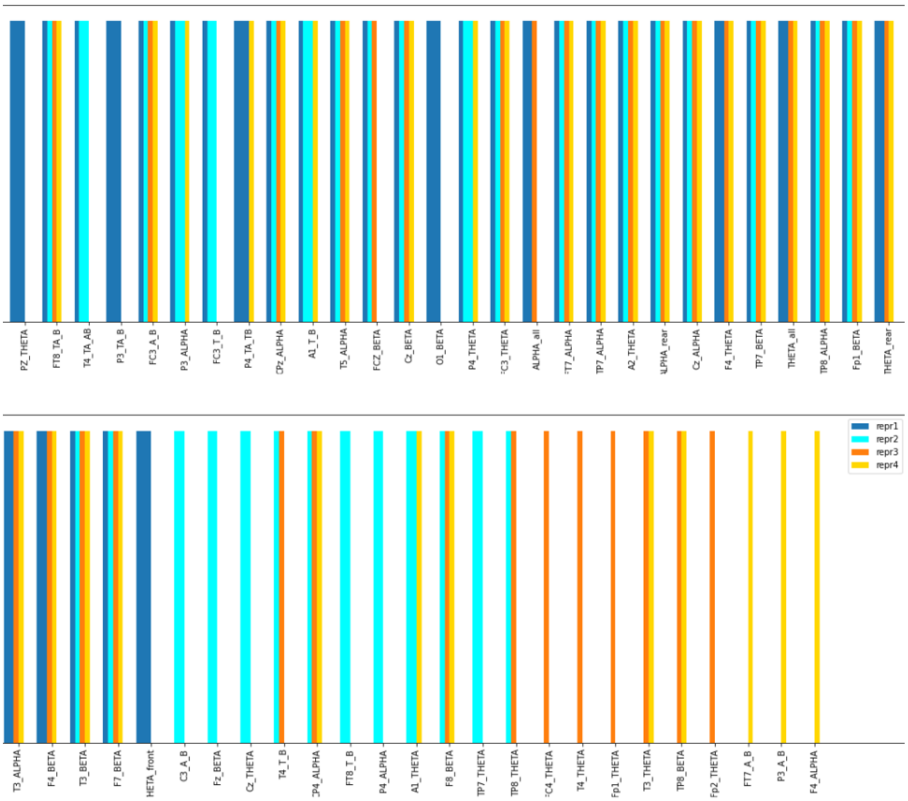

Figure 5.1: The selected features in 4 different sessions with the same hyperparameter settings: $f = 0.2$, final_mAP = 0.75, des_feat_num = 10 %. Each color corresponds to a session, thus, features that are represented with more than one color, were selected into the final subset multiple times. This implies that these features have the highest predictive power.

5.3 Comparison To Other FS Methods

To get a thorough view of the proposed method's efficiency, it is compared to the widely used PCA feature selection method and also to the random selection. The top 5, 10, and 20 % of the original feature set have been selected using the proposed method, the PCA and a random selection. In the case of the PCA, the projection was not completed. Instead, the des_feat_num number of features that mostly contributed to the first principal component have been selected, and the base network is trained with them. The averaged results form 4 runs are summarized in table Table 5.3. In all three scenarios, the proposed method outperforms both the random selection and the PCA.

Figure 5.2 shows the chosen features by the proposed method and PCA with their final scores. The selected features are highly dissimilar in the case of the the FS methods: while the proposed method mostly selects the sole α, θ, β PSDs, PCA assigns higher scores to the $\frac{\alpha}{\alpha+\theta}$ feature, measured on different electrodes.

The selected features by the proposed FS method also appear in other studies as best indicators of driver fatigue among other EEG features. In [13] they examined the relationship between reaction ability, physiological signals and driving fatigue, and concluded that among the frequency domain features β-PSD has the greatest correlation with the

Table 5.3: Classification performances when training with the subsets of the TOP 20, 10 and 5 % of the original features generated using different feature selection methods.

	TOP 20 %		**TOP 10 %**		**TOP 5 %**	
FS method	**mAP**	**overfit.**	**mAP**	**overfit.**	**mAP**	**overfit.**
random	0.648	0.23	0.539	0.313	0.504	0.327
PCA	0.886	0.034	0.75	0.056	0.673	0.049
proposed	**0.94**	**0.034**	**0.927**	**0.027**	**0.916**	**0.01**

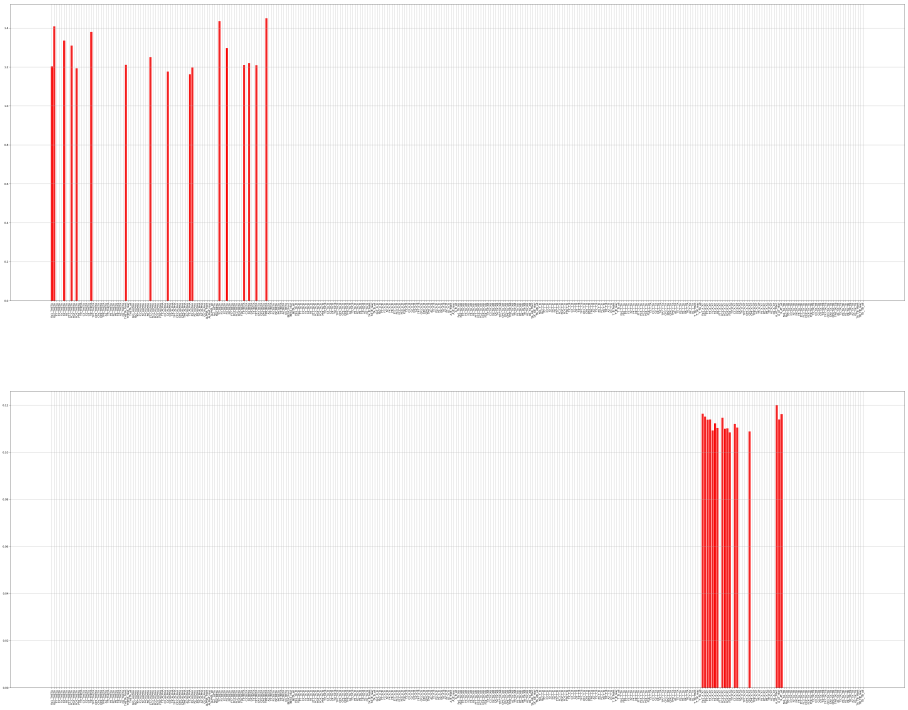

Figure 5.2: Selected TOP 5 % features by the proposed algorithm (up) and by PCA (bottom).

reaction time based on Grey correlation analysis. This is due to the fact, that β waves appear in case of excitement or alertness. The experiments carried out in study [26] in order to detect driving sleep-onset state showed that the effect of the mutual addition of α, β and θ waves is more satisfactory compared to when these waves are used alone. Similarly to this conclusion, it can be seen that the proposed method always selects these α-PSD, β-PSD and θ-PSD based features together into the final subset, regardless of the different settings or runs Figure 5.1. Lastly, the credibility of the θ-PSD's presence in the final subsets is proven by the fact that hight θ activity refers to the microsleep state [36] which indicates high level drowsiness and sleep-onset state [25].

Chapter 6

Summary

The stated goal - the development of a feature selection (FS) algorithm that can be later utilized for the development of a robust and reliable drivers' drowsiness detector - has been successfully achieved. Inspired by an idea introduced in a SOTA paper, I have designed an embedded FS method which exploits the neural networks' working principle for feature scoring: the classifier network is supplemented with a so-called Feature Prune Layer (FPL) that has the same size as the number of input features, is point-to-point related to them and the magnitude of its weights represent the importance of the corresponding features. In order to find the desired number of features with the best predictive power, the FPL was pruned iteratively during the classifiers' training. As electroencephalogram (EEG) measures the electrical activities in the brain and has been proven to be one of the best indicators of drowsiness, I have used this data source in this study. The initial feature set has been constructed from frequency-based features extracted from a public data set that contains raw EEG data recorded during sustained-attention driving tasks.

The proposed feature selection algorithm proves its efficiency by the fact that is able to reduce the original feature set even by 95 % without major degradation in the accuracy: using the best performing hyperparameter setting, the classifier's accuracy drops only by 1 % while the overfitting decreases by 2.7 %. When moderately reducing the initial feature set, the proposed FS algorithm is able to reduce the noise and extract the vital information. This is revealed by the results when selecting the top 10 % and top 20 % of the initial features the classifier's accuracy increases by 1.5 % and 2.7 % respectively. The proposed method outperforms the random selection and the widely popular Principal Component Analysis when reducing the original feature set by 90%: it achieves 38.8 % and 17.7 % higher accuracy respectively. Moreover, the selected features by the proposed FS method also appear in other studies as best indicators of driver fatigue among other EEG features, which confirms the solution's credibility. One drawback of the introduced solution is caused by the fact that the feature scoring relies on the weights in the classifier network, hence in its current state, the feature selection is highly sensitive to the network's random initialization; the results may slightly vary in different runs. For this reason, the further plans include discovering the stabilization opportunities for the method. In addition, I also plan to compare the results to a widely used nonlinear FS method and finally, to examine its generalization ability over different drowsiness indicator features.

Bibliography

[1] Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997. ISSN 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(97)00043-X. URL https://www.sciencedirect.com/science/article/pii/S000437029700043X. Relevance.

[2] Preface. In Ian H. Witten, Eibe Frank, and Mark A. Hall, editors, *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages xxi–xxvii. Morgan Kaufmann, Boston, third edition edition, 2011. ISBN 978-0-12-374856-0. DOI: https://doi.org/10.1016/B978-0-12-374856-0.00021-3. URL https://www.sciencedirect.com/science/article/pii/B9780123748560000213.

[3] Hojjat Adeli, Ziqin Zhou, and Nahid Dadmehr. Analysis of eeg records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, 123(1):69–87, 2003. ISSN 0165-0270. DOI: https://doi.org/10.1016/S0165-0270(02)00340-0. URL https://www.sciencedirect.com/science/article/pii/S0165027002003400.

[4] Shaibal Barua, Mobyen Uddin Ahmed, Christer Ahlström, and Shahina Begum. Automatic driver sleepiness detection using eeg, eog and contextual information. *Expert Systems with Applications*, 115:121–135, 2019. ISSN 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.07.054. URL https://www.sciencedirect.com/science/article/pii/S0957417418304792.

[5] Davis W. Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John V. Guttag. What is the state of neural network pruning? *ArXiv*, abs/2003.03033, 2020.

[6] Timothy Brown, John Lee, Chris Schwarz, Dary Fiorentino, and Anthony Mcdonald. *Assessing the feasibility of vehicle-based sensors to detect drowsy driving*. 01 2014.

[7] Zehong Cao, Chun-Hsiang Chuang, Jung-Kai King, and Chin-Teng Lin. Multi-channel EEG recordings during a sustained-attention driving task. *Scientific Data*, 6(1), apr 2019. DOI: 10.1038/s41597-019-0027-4. URL https://doi.org/10.1038%2Fs41597-019-0027-4.

[8] Jian Cui. Eeg driver drowsiness dataset. https://doi.org/10.6084/m9.figshare.14273687.v3.

[9] Jian Cui, Zirui Lan, Tianhu Zheng, Yisi Liu, Olga Sourina, Lipo Wang, and Wolfgang Müller-Wittig. Subject-independent drowsiness recognition from single-channel eeg with an interpretable cnn-lstm model. In *2021 International Conference on Cyberworlds (CW)*, pages 201–208, 2021. DOI: 10.1109/CW52790.2021.00041.

[10] Pradip Dhal and Chandrashekhar Azad. A comprehensive survey on feature selection in the various fields of machine learning. *Applied Intelligence*, 52, 03 2022. DOI: 10.1007/s10489-021-02550-9.

[11] Piotr J Franaszczuk, Gregory K Bergey, Piotr J Durka, and Howard M Eisenberg. Time–frequency analysis using the matching pursuit algorithm applied to seizures originating from the mesial temporal lobe. *Electroencephalography and Clinical Neurophysiology*, 106(6):513–521, 1998. ISSN 0013-4694. DOI: https://doi.org/10.1016/S0013-4694(98)00024-8. URL https://www.sciencedirect.com/science/article/pii/S0013469498000248.

[12] Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, 02 2012.

[13] Mengzhu Guo, Shiwu Li, Linhong Wang, Meng Chai, Facheng Chen, and Yunong Wei. Research on the relationship between reaction ability and mental state for online assessment of driving fatigue. *International Journal of Environmental Research and Public Health*, 13:1174, 11 2016. DOI: 10.3390/ijerph13121174.

[14] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3(null):1157–1182, mar 2003. ISSN 1532-4435.

[15] Trevor J. Hastie, Robert Tibshirani, and Jerome H. Friedman. The elements of statistical learning: Data mining, inference, and prediction, 2nd edition. In *Springer Series in Statistics*, 2001.

[16] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, 2012. URL https://arxiv.org/abs/1207.0580.

[17] Dr. MD Rashedul Islam, Md Rahim, Md. Rajibul Islam, and Jungpil Shin. *Genetic Algorithm Based Optimal Feature Selection Extracted by Time-Frequency Analysis for Enhanced Sleep Disorder Diagnosis Using EEG Signal*, pages 881–894. 01 2020. ISBN 978-3-030-29512-7. DOI: 10.1007/978-3-030-29513-4_65.

[18] Zakaria Jaadi. A step-by-step explanation of principal component analysis (pca). URL https://builtin.com/data-science/step-step-explanation-principal-component-analysis. Accessed: 2022-10-23.

[19] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. DOI: 10.1098/rsta.2015.0202. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2015.0202.

[20] Vinay Varma K. Embedded methods for feature selection in neural networks, 2020. URL https://arxiv.org/abs/2010.05834.

[21] Peter W. Kaplan, Selim R Benbadis, Aatif M. Husain, and William O. Tatum. Handbook of eeg interpretation. 2007.

[22] Rami N. Khushaba, Sarath Kodagoda, Sara Lal, and Gamini Dissanayake. Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm. *IEEE Transactions on Biomedical Engineering*, 58(1):121–131, 2011. DOI: 10.1109/TBME.2010.2077291.

[23] Rami N. Khushaba, Sarath Kodagoda, Sara Lal, and Gamini Dissanayake. Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm. *IEEE Transactions on Biomedical Engineering*, 58:121–131, 2011.

[24] George H. Klem, Hans Lüders, Herbert H. Jasper, and Christian Erich Elger. The ten-twenty electrode system of the international federation. the international federation of clinical neurophysiology. *Electroencephalography and clinical neurophysiology. Supplement*, 52:3–6, 1999.

[25] Saroj K.L. Lal and Ashley Craig. A critical review of the psychophysiology of driver fatigue. *Biological Psychology*, 55(3):173–194, 2001. ISSN 0301-0511. DOI: https://doi.org/10.1016/S0301-0511(00)00085-5. URL https://www.sciencedirect.com/science/article/pii/S0301051100000855.

[26] Boon Giin Lee, Boon-Leng Lee, and Wan-Young Chung. Mobile healthcare for automatic driving sleep-onset detection using wavelet-based eeg and respiration signals. *Sensors (Basel, Switzerland)*, 14:17915–17936, 10 2014. DOI: 10.3390/s141017915.

[27] Gang Li and Wan-Young Chung. Electroencephalogram-based approaches for driver drowsiness detection and management: A review. *Sensors*, 22(3), 2022. ISSN 1424-8220. DOI: 10.3390/s22031100. URL https://www.mdpi.com/1424-8220/22/3/1100.

[28] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Comput. Surv.*, 50(6), dec 2017. ISSN 0360-0300. DOI: 10.1145/3136625. URL https://doi.org/10.1145/3136625.

[29] Huan Liu and Hiroshi Motoda. Feature selection for knowledge discovery and data mining. In *The Springer International Series in Engineering and Computer Science*, 1998.

[30] Huan Liu and Hiroshi Motoda. *Computational Methods of Feature Selection*. 01 2008. ISBN 9781584888789. DOI: 10.1201/9781584888796.

[31] Mahdokht Masaeli, Glenn Fung, and Jennifer G. Dy. From transformation-based dimensionality reduction to feature selection. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 751–758, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

[32] Henrik Mårtensson, Oliver Keelan, and Christer Ahlström. Driver sleepiness classification based on physiological data and driving performance from real road driving. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):421–430, 2019. DOI: 10.1109/TITS.2018.2814207.

[33] Patrenahalli M. Narendra and Keinosuke Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26:917–922, 1977.

[34] Gustavo H.B.S. Oliveira, Luciano R. Coutinho, Josenildo C. da Silva, Ivan J.P. Pinto, Júlia M.S. Ferreira, Francisco J.S. Silva, Davi V. Santos, and Ariel S.

Teles. Multitaper-based method for automatic k-complex detection in human sleep eeg. *Expert Systems with Applications*, 151:113331, 2020. ISSN 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113331. URL https://www.sciencedirect.com/science/article/pii/S0957417420301561.

[35] Pallavi Pandey and K.R. Seeja. Subject independent emotion recognition from eeg using vmd and deep learning. *Journal of King Saud University - Computer and Information Sciences*, 34(5):1730–1738, 2022. ISSN 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2019.11.003. URL https://www.sciencedirect.com/science/article/pii/S1319157819309991.

[36] Malik Peiris, Paul Davidson, P.J. Bones, and Richard Jones. Detection of lapses in responsiveness from the eeg. *Journal of neural engineering*, 8:016003, 02 2011. DOI: 10.1088/1741-2560/8/1/016003.

[37] Martijn J. Post, Peter van der Putten, and Jan N. van Rijn. Does feature selection improve classification? a large scale experiment in openml. In *IDA*, 2016.

[38] Muhammad Ramzan, Hikmat Ullah Khan, Shahid Mahmood Awan, Amina Ismail, Mahwish Ilyas, and Ahsan Mahmood. A survey on state-of-the-art drowsiness detection techniques. *IEEE Access*, 7:61904–61919, 2019. DOI: 10.1109/ACCESS.2019.2914373.

[39] Mustafa Abdul Salam, Ahmad Taher Azar, Mustafa Samy Elgendy, and Khaled Mohamed Fouad. The effect of different dimensionality reduction techniques on machine learning overfitting problem. *International Journal of Advanced Computer Science and Applications*, 12(4), 2021. DOI: 10.14569/IJACSA.2021.0120480. URL http://dx.doi.org/10.14569/IJACSA.2021.0120480.

[40] Marco Sandri and Paola Zuccolotto. *Variable Selection Using Random Forests*, pages 263–270. 01 2006. ISBN 978-3-540-35977-7. DOI: 10.1007/3-540-35978-8_30.

[41] Nadja Schömig, Volker Hargutt, Alexandra Neukum, Ina Petermann-Stock, and Ina Othersen. The interaction between highly automated driving and the development of drowsiness. *Procedia Manufacturing*, 3:6652–6659, 2015. ISSN 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2015.11.005. URL https://www.sciencedirect.com/science/article/pii/S235197891501121X. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.

[42] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014. URL http://arxiv.org/abs/1404.1100.

[43] O M Solomon, Jr. Psd computations using welch's method. [power spectral density (psd)]. 12 1991. DOI: 10.2172/5688766. URL https://www.osti.gov/biblio/5688766.

[44] Fengxi Song, Dayong Mei, and Hongfeng Li. Feature selection based on linear discriminant analysis. In *2010 International Conference on Intelligent System Design and Engineering Application*, volume 1, pages 746–749, 2010. DOI: 10.1109/ISDEA.2010.311.

[45] Mahsa Soufineyestani, Dale Dowling, and Arshia Khan. Electroencephalography (eeg) technology applications and available devices. *Applied Sciences*, 10(21), 2020. ISSN 2076-3417. DOI: 10.3390/app10217453. URL https://www.mdpi.com/2076-3417/10/21/7453.

[46] Igor Stancin, Mario Cifrek, and Alan Jović. A review of eeg signal features and their application in driver drowsiness detection systems. *Sensors (Basel, Switzerland)*, 21, 2021.

[47] Chun-Shu Wei, Yu-Te Wang, Chin-Teng Lin, and Tzyy-Ping Jung. Toward drowsiness detection using non-hair-bearing eeg-based brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26:400–406, 2018.

[48] Peter D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15:70–73, 1967.