



Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Thesis for the Degree of Bachelor of Science in Computer Science
15.0 credits

NON-CONTACT BASED PERSON'S SLEEPINESS DETECTION USING HEART RATE VARIABILITY

Fanny Danielsson
fdn16001@student.mdh.se

Examiner: Mobyen Uddin Ahmed
Mälardalen University, Västerås, Sweden

Supervisors: Hamidur Rahman
Mälardalen University, Västerås, Sweden

June 25, 2019

Abstract

Today many strategies of monitoring health status and well-being are done through measurement methods that are connected to the body, e.g. sensors or electrodes. These are often complicated and requires personal assistance in order to use, because of advanced hardware and attachment issues. This paper proposes a new method of making it possible for a user to self-monitoring their well-being and health status over time by using a non-contact camera system. The camera system extracts physiological parameters (e.g. Heart Rate (HR), Respiration Rate (RR), Inter-beat-Interval (IBI)) based on facial color variations, due to blood circulation in facial skin. By examining an individual's physiological parameters, one can extract measurements that can be used in order to monitor their well-being. The measurements used in this paper is features of heart rate variability (HRV) that are calculated from the physiological parameter IBI. The HRV features included and tested in this paper is SDNN, RMSSD, NN50 and pNN50 from Time Domain and VLF, LF and LF/HF from Frequency Domain. Machine Learning classification is done in order to classify an individual's sleepiness from the given features. The Machine Learning classification model which gave the best results, in forms of accuracy, were Support Vector Machines (SVM). The best mean accuracy achieved was 84,16% for the training set and 81,67% for the test set for sleepiness detection with SVM. This paper has great potential for personal health care monitoring and can be further extended to detect other factors that could help a user to monitor their well-being, such as measuring stress level.

Table of Contents

1. Introduction	1
2. Background	2
2.1. Physiological Parameters	2
2.1.1 Heart Rate Variability	3
2.2. Eye Tracking Features	5
2.3. Classification	5
2.3.1 ML Algorithms	6
3. Related Work	9
4. Problem Formulation and Research Questions	12
4.1. Limitations and Challenges	12
5. Materials and Method	13
5.1. Data Collection	13
5.2. Parameter Extraction	14
5.3. Feature extraction	16
5.4. Intelligent System Development	16
5.4.1 Software	16
5.4.2 Classification	17
6. Ethical and Societal Considerations	19
7. Results	20
7.1. Data Collection Report	20
7.2. Classification Models and Feature Results	20
7.2.1 KNN	20
7.2.2 LR	22
7.2.3 SVM	24
8. Discussion	28
9. Conclusions	31
9.1. Future work	31
References	34
Appendix A Appendices	35
1.1. Appendix 1	35
1.2. Appendix 2	36
1.3. Appendix 3	36

List of Tables

1	Table showing different HRV features in TD and FD.	3
2	ML Instances with known labels.	5
3	Table showing an overview of related work articles. Shows were the parameters of TD has or has not shown a relationship to sleepiness.	10
4	Table showing an example of the Excel file after extracting the IBI data for 10 participants. Top rows showing id and class and last rows showing the IBI data. .	15
5	Table displaying example of feature matrix.	16
6	The best results for KNN model with different features from the classification learner on MATLAB.	21
7	The best results for LR model with different features from the classification learner on MATLAB.	23
8	The best results for SVM model with different features from the classification learner on MATLAB.	25
9	Results for the extracted LR model.	27
10	Results for the extracted Cubic SVM model.	27
11	Results for the extracted Fine KNN model.	28

List of Figures

1	A visualization of a ECG signal showing IBI.	3
2	Visualization of the data split.	6
3	Illustration of SVM model, separates the non-linear data on a) into two classes b).	8
4	Two sample pictures from the video recordings.	13
5	Parameter extraction steps.	15
6	Example classification for sleepiness detection.	17
7	CM for Fine KNN with 10 features.	21
8	ROC curves for Fine KNN with 10 features.	21
9	CM for Cosine KNN with 7 features.	22
10	ROC curves for Cosine KNN with 7 features.	22
11	CM for LR with 10 features.	23
12	ROC curves for LR with 10 features.	23
13	CM for LR with 7 features.	24
14	ROC curves for LR with 7 features.	24
15	CM for Linear SVM with 10 features.	25
16	ROC curves for Linear SVM with 10 features.	26
17	CM for Linear SVM with 7 features.	26
18	ROC curves for Linear SVM with 7 features.	27
19	Google Form to be filled by the participants in the data collection.	35
20	Letter of Consent to be signed by the individuals participating in the data collection.	36
21	Technical explanation for the implementation of the intelligent system.	37

Abbreviations

AECS Average Eye Closure Speed

ANN Artificial Neural Network

CM Confusion Matrix

DFT discrete Fourier transform

ECG electrocardiogram

FD Frequency Domain

FFT Fast Fourier Transform

FNR False Negative Rates

FPR False Positive Rates

HB Heart Beat

HF High Frequency

HR Heart Rate

HRV Heart Rate Variability

IBI Inter-beat Interval

KNN K-Nearest Neighbors

KSS Karolinska Sleepiness Scale

LF Low Frequency

LR Logistic Regression

MeanNN Mean of selected IBIs

ML Machine Learning

NN50 Frequency of successive differences of IBIs that spanned more than 50 ms

PERCLOSE Percentage Eye Closure

pNN50 Percentage value of NN50

PSD Power Spectral Density

RMSSD Root mean square of the differences between consecutive IBIs

ROC Receiver Operating Characteristic

SDNN Standard deviation of IBIs

SDSD Standard deviation of differences between adjacent IBIs

SVM Support Vector Machines

TD Time Domain

TPR True Positive Rates

VLF Very Low Frequency

1. Introduction

In the field of computer vision, an active area of research is face feature detection [1]. This is because it has shown that it can be used for different kinds of real-world applications, such as monitoring and surveillance, drivers alertness system, flight systems, intelligent robots, etc. A system which can measure a drivers state, such as sleepiness, can help when using it to give warnings or provide safety precautions when the drivers alertness and attention levels are low (they are feeling sleepy).

Many driving accidents that happen are sleep related. In a study made 2009 by the Swedish insurance company Länsförsäkringar [2] it has shown that 3 out of 10 traffic accidents happen because of driver fatigue. But since it is known that driving tired is illegal in Sweden, there is probably more unrecorded accidents related to this. Further, the study showed that 77% of the participants said that they had driven while they were very tired. Negligence of sleep is a critical cause of excessive sleepiness which has shown to be a well-recognized cause of motor vehicle crashes [3]. The National Highway Traffic Safety Administration reports in 2013 [4], that 72,000 crashes, 44,000 injures, and 800 deaths were caused by drowsy driver's crashing their vehicles. Also here, these numbers are underestimated. Developing systems which may be used in order to monitor an individual's sleepiness can be critical in order to prevent accidents in the future.

Today, measurements to monitor sleepiness are extracted in many ways, for example via electrodes [5]. The electrodes are used to extract physiological parameters, such as inter-beat interval (IBI) to calculate heart rate variability (HRV) features. This paper describes the study on the development of a detection system which will make a user able to self-monitor their sleepiness and well-being over time. Introducing a new way of measuring an individual's sleepiness with a non-contact camera system. The camera system is used to withdraw physiological parameters from an individual. This is done by examining the color variation in an individual's face, caused by body blood circulation, using image processing [6]. The physiological parameter used to monitor sleepiness in this study is the HRV, which is extracted by measuring the changes in the time intervals between electrocardiogram (ECG) signals, thus the IBI [7]. The HRV features used are those that are in time domain (TD) and frequency domain (FD). From TD the SDNN, RMSSD, NN50, and pNN50 were tested. From FD the VLF- and LF bands are extracted and tested through obtaining the power density by applying the fast Fourier transform and power spectral density on the IBI time series. Further, the LF/HF ratio was also tested. In order to classify an individual's sleepiness into two classes (sleepy or not sleepy), I experimented with different Machine Learning classification models in order to decide which one would be most relevant for sleepiness detection. The models that I experimented with was K-nearest Neighbours, Logistic Regression (LR) and Support Vector Machines (SVM). To extract physiological parameters, to train and test the models, 10 individuals were chosen for data collection. Later only 6 of the individuals data were used to train the classification model. The classification model that showed best results for binary classification with HRV features were SVM. The results showed 84,16% for the training set and 81,67% for the test set. To obtain these results, the models were provided with 6 individuals data which were given by the SDNN, RMSSD, NN50, pNN50, VLF and LF/HF ratio features. The feature that showed really good results for sleepiness detection in all models were SDNN. LF feature on the other hand showed negative indication of sleepiness.

The goal of this work was to develop a system which could detect a person's sleepiness by using a non-contact camera system. Aiming to improve an individual's over all knowledge of their well-being. This could further aim to be tested in real world applications to help individual's identify when their attention and alertness levels are low (they are sleepy), which can be important in driving scenarios, air traffic monitoring and so on.

2. Background

In a human vision system, the human brain processes images that are derived from the eyes. The computer version of the human vision system is called computer vision. Computer vision is when a computer processes images from pictures or videos that are acquired from some kind of electronic camera source [8]. When processing, the computer extracts different information, for example, an individual's physiological parameters in order to verify health status or identify aspects of an individual's face which can be used in face recognition. With physiological parameters, one can as mentioned develop systems that can identify health status factors for an individual. One example is sleepiness identification. The definition of sleepiness is described to be the state of being sleepy ¹. In the US 20% of adults has reported that daytime sleepiness has interfered with their daily activities [9]. Subjective terms that people use related to excessive sleepiness is drowsiness, fatigue, inertness and so on. The cause of excessive sleepiness can be because of sleep deprivation, medication effects, illegal substance use along with other medical and psychiatric conditions.

To quantify an individual's sleepiness, thus define the sleepiness level of an individual, one can use different scales. A well-known subjective sleepiness scale is the karolinska sleepiness scale (KSS) [10]. It is used to quantify an individual's sleepiness level on a scale from 1 to 9.

The 9-points on the scale responds to²

- 1 = Extremely alert
- 2 = Very alert
- 3 = Alert
- 4 = Fairly alert
- 5 = Neither alert nor sleepy
- 6 = Some signs of sleepiness
- 7 = Sleepy, but no effort to keep alert
- 8 = Sleepy, some effort to keep alert
- 9 = Very sleepy, great effort to keep alert, fighting sleep

Furthermore, with exception from the sleepiness scale, there seem to be two significant ways of determining an individual's sleepiness. The two methods involve using either physiological parameters (as mentioned), such as IBI to calculate heart rate variability (HRV) features or using image processing in order to achieve eye measurements. These features/measurements can be used in order to detect sleepiness by examining their changes.

2.1. Physiological Parameters

Before explaining HRV one must know about a few different physiological parameters. First one is the heart beat (HB) which is a response to the pulse rate [11]. The pulse rate is the measurement of the arteries expansion and contraction during one minute. Further, heart rate (HR) is the speed of the HB or the number of pulses measured at a specific time interval. More in-depth about HR is that the rate is when the senatorial node depolarizes, this is because it is the source of cardiac depolarization. In Fig. 1 the number of waves in a minute is used to calculate the HR.

An interesting thing about HR is what Miriam R. Waldeck mentioned in her *Heart Rate During Sleep: Implications for Monitoring Training Status* [12]. She proposes a study where she measured an individual's distance from their average HR to the minimum HR. This study revealed that when a person is relaxed their HR was reduced by about 8 BPM.

¹<https://www.lexico.com/en/definition/sleepiness>

²<https://www.interdynamics.com/faqwd/karolinska-sleepiness-scale-kss/>

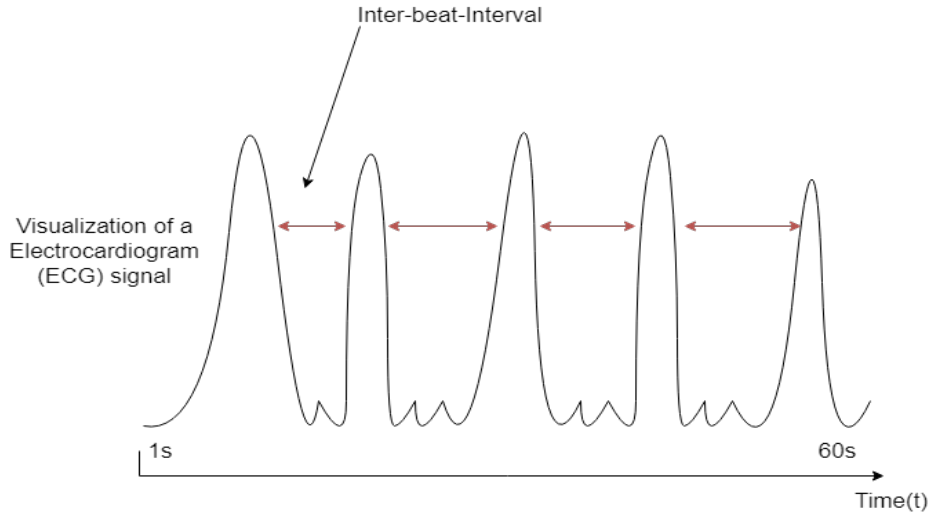


Figure 1: A visualization of a ECG signal showing IBI.

Another physiological parameter is the inter-beat interval (IBI) which is the time between two waves in Fig. 1. The unit of IBI can be in either milliseconds or seconds format [13].

2.1.1.1 Heart Rate Variability

HRV is the adjustment of the HR or the intervals between your HB [10]. HRV has been proven to have a correlation with an individual's sleepiness.

	Parameters	Units	Definitions
Time Domain	MeanNN	ms	Mean of selected IBIs
	SDNN	ms	Standard deviation of IBIs
	RMSSD	ms	Root mean square of the differences of IBIs
	SDSD	ms	Standard deviation of differences between adjacent IBIs
	NN50	count	Number of consecutive IBIs that differ more then 50ms
	pNN50	%	Percentage value of consecutive IBIs that differ more then 50ms
Frequency Domain	VLF	ms ²	Very low frequency
	LF	ms ²	Low frequency
	HF	ms ²	High frequency
	LFHFratio	ms ²	The ratio LF/HF

Table 1: Table showing different HRV features in TD and FD.

HRV has several features or metrics and these are time domain (TD), frequency domain (FD) and non-linear metrics [7]. The TD is focusing on measuring and monitoring the HRV within specific time ranges via statistic or geometric calculation [10]. It is said that TD is the simplest evaluation method of HRV.

The different statistic HRV features can be viewed in Tab. 1. The two geometric calculation for HRV can be viewed below.

- Integral of IBIs histogram divided by histogram height
- Baseline width of histogram of IBIs

FD is the number of times each event has occurred during a total period of observation, thus the amount of absolute or relative signal in component bands [7] [14]. The FD features can be used

to detect any differences in the data by observing peaks of the frequencies. The commonly used frequency bands can be viewed in Tab. 1.

The signal in a component band is measured by extracting its power density. To obtain the power density, a first step is to calculate the power spectrum. With this, one can go from a discrete-time signal into frequency. This is done with a discrete-time transformation [15].

The discrete-time transformation definition is as described below

Let $y(t)$ be a deterministic discrete-time signal, assume $\sum_{t=-\infty}^{\infty} |y(t)|^2 < \infty$, then the discrete-time Fourier transform of the signal is

$$y(\omega) = \sum_{t=-\infty}^{\infty} y(t)e^{-i\omega t}, \omega \in [-\pi, \pi] \quad (1)$$

which inverse is defined as

$$y(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} y(\omega)e^{i\omega t} d\omega \quad (2)$$

The discrete Fourier transform (DFT), which is its frequency Fourier transform (FFT), is obtained as

$$Y_k = \sum_{t=0}^{N-1} y(t)e^{i\frac{2\pi}{N}kt}, k = 0, \dots, N-1 \quad (3)$$

where $y(0) \dots y(N-1)$ is the time signal.

Further one must calculate and obtain the power spectral density (PSD) in order to see how the power of the signal is distributed with frequency.

Let $S(\omega) = |y(\omega)|^2$ be the energy spectral density, with this we got

(Parseval's theorem)

$$\sum_{t=-\infty}^{\infty} |y(t)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega \quad (4)$$

where $S(\omega)$ is the definition of energy as a frequency function.

The Weiner-khinchin theorem explains that the power spectrum of a zero-mean stationary stochastic process $y(t)$, can be calculated by obtaining the Fourier transform of its co-variance function $r(k)$. See definitions below.

$y(t)$'s auto co-variance sequence is defined as

$$r(k) = E\{y(t)y^*(t-k)\} \quad (5)$$

where $E\{\}$ is the expected value and $*$ is the complex conjugate.

The first definition of PSD is as illustrated below

$$\phi(\omega) = \sum_{k=-\infty}^{\infty} r(k)e^{-i\omega k} \quad (6)$$

Last the non-linear features are those that determine the quantity in unpredictable and complex series of IBIs [7]. However, these features are not considered in this thesis since I could not find any related studies involving these features. Hence literature did not show that non-linear features could be used to detect sleepiness.

HRV and its features can be used in order to identify and detect sleepiness by exterminating and comparing these different features. How to calculate the TD parameters of HRV can be further viewed in Tab. 1.

2.2. Eye Tracking Features

Another way of measuring sleepiness is by using image processing in order to retrieve eye measurements. Tang-Hsien Chang and Yi-Ru Chen mention in their article *Driver Fatigue Surveillance via Eye Detection* [16] that eye closure measurements are the most promising real-time measure of driver alertness. In order to retrieve different eye measurements for measuring sleepiness one could use image processing. Image processing is the process of finding information needed for the application domain [17]. This process can be divided into several steps. Further, image processing techniques and algorithms can be categorized into low level- and high-level processing. Low-level processing includes image compression, noise-filtering, and other more simple methods. But for high-level processing, it includes methods that can know and understand image content such as being able to identify and track movements of the eyes. Such a method is called eye tracker, and in general there are two significant ways of eye tracking [18]. The first one measures the position of the eye related to the head, and the second one measures the orientation of the eye in space.

Two measurements of the eyes that are used in correlation with sleepiness detection is percentage eye closure (PERCLOSE) and average eye closure speed (AECS). The most popular eye measurement is PERCLOSE [16]. PERCLOSE measures how much time the eye is 80% to 100% closed. It can be used for determining alertness because when an individual is tired the PERCLOSE measurements will increase.

Further AECS is not a measurement of how much the eye is closed more so how long an individual takes to go from open to closed eyes, thus the closure speed [16]. As with PERCLOSE, the AECS measurement will increase when an individual's alertness is low.

2.3. Classification

Machine Learning (ML) in general is the process in which a system learns based on a given set of data, rules or similar from instances [19]. One example is a system which can predict the weather at a given time based on earlier achieved data about the weather as a training set.

When a person is trying to identify the differences between a large amount of data there is a risk of mistakes occurring due to failures in analyzing or when trying to establish a relationship between multiple features [19]. A possible solution to this would be to apply ML techniques in order to achieve a more successful/improved system when it comes to efficiency and machine design.

ML		Instances		
instance	feature x	feature n	class
1	xxx		xx	1
2	xxx		xx	0
...
n	xxx		xx	1

Table 2: ML Instances with known labels.

Furthermore, in a ML instance, the data set is represented by a set of features [19]. These features can be binary, category, numerical or continuous.

The features, or the data, of a set can be split into different groups, See Fig. 2 for an example. These groups are training, validation and test set³. The training set is used to train a ML algorithm for it to give the accurate output, thus seeing and learning how this is done is dependant of the algorithm (weights, biases etc). The validation set is used to check how good the ML model is doing (evaluate a model), thus measure how much the accuracy has increased with iterations of

³<https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>

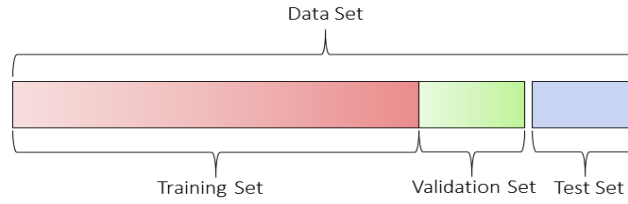


Figure 2: Visualization of the data split.

training. The validation set is often used in between the training set in order to see if the model has progressed so it can stop or continue the training. Last is the test set, it is used after the training set to evaluate the model.

The ML instances learning can be supervised, unsupervised or reinforced. Supervised learning is when the instances have known labels which represent the expected output [19]. Supervised learning can be used to train a given network. When the outputs are known an error can be calculated on how much those outputs differs from the expected ones. With this, the algorithm can be made so that it slowly learns to adjust in order to get closer to the expected output. Unsupervised learning on the other hand is when the labels are not known. With this technique one hopes to achieve unknown, but useful, classes of items. In reinforced learning the system is given an external trainer which measures how well the system is working. With this, the system does not know exactly what actions to take but it will with time know what actions to take in order to yield the best results. An example of this would be an algorithm that can solve a Super Mario course. In this case, the external trainer measures Marios position to the goal while it tries certain actions. If the position to the goal decreases without Mario dying the learner knows what action it took to get closer to the goal.

In ML there are two different categories of supervised learning. These are regression and classification. Regression is used when the expected outputs are continuous or numerical⁴. This means that the labels can be real values, thus integers or floats. In classification the labels are categorical or discrete and a mapping function is used to predict which category the input variables belongs to.

2.3.1 ML Algorithms

2.3.1.1 K-Nearest Neighbors

K-nearest Neighbors or KNN, is an algorithm which can be used for both classification and regression problems⁵. In a KNN the class membership is decided upon the K-nearest neighbors. This means that a value, K , is chosen and when trying to decide membership, K amount of neighbors are selected around the data point. Further the class label for the data point is set after the class labels on its selected neighbors.

2.3.1.2 Artificial Neural Network

A well known ML algorithm is the Artificial Neural Network (ANN). an ANN can return a specific output depending on a given specific input [16]. This is done by continuous training sets of data which makes the ANN able to learn and adjust to give accurate output to a specific input. The mathematical definition for a general ANN is

$$o(x) = f \sum_{i=0}^n w_i x_i \quad (7)$$

⁴<https://link.medium.com/1C5tYhnqTV>

⁵<https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>

x is a neuron, n is the number of input dendrites, o is one output axon, w is the weight which will determine how much the inputs will be weighted (analogy to synapses), f is an activation function which is a response to how powerful the output should be from the neuron. It is based on the summation of the inputs.

To be able to learn the ANN uses something called back-propagation to adjust its weights. This is done by calculating an error at the output layer and then transfers the errors adjusting the weights at each layer [16]. The error term e_x at the output layer on a single neuron k is

$$e_x = l_x - o_x \quad (8)$$

o_x is the calculated output, l_x is the desired output (label) of a neuron x .

The weights are adjusted by applying the above error term to calculate a delta value. The delta value is

$$\delta_x = e_x f'(o_x) \quad (9)$$

where f' is the derivative of the activation function. To determine how much the weight should be adjusted, delta y is calculated from the delta x of this layer. Leading to

$$\delta_y = n f'(o_y) \sum_{k=0}^K \delta_x w_{yx} \quad (10)$$

2.3..1.3 Logistic Regression

A similar algorithm to the ANN, is Logistic Regression (LR). Even though LR has regression in its name, it is used for classification. LR is used to assign observations to a discrete set of classes. As with an ANN, LR applies a functional form f and a delta parameter [20]. But the difference is that in LR the parameters can be interpreted, in an ANN this is not always the case, e.g. with the weights.

The class membership is decided by the general equation

$$P = \frac{1}{1 + e - (\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots \beta_n X_n)} \quad (11)$$

Or

$$P = \frac{1}{1 + e - (\beta_0 + \sum \beta_i X_i)} \quad (12)$$

A thing to note is that logistic regression model can be seen as an ANN without any hidden layers if the activation function is sigmoidal [20], thus

$$f(x) = \frac{1}{(1 + e^{-x})} \quad (13)$$

Which clearly shows a correlation to the two previous equations. The sigmoid function is used to return a given probability value which makes it possible to divide the input into two or more discrete classes⁶. One example where LR could be helpful is when trying to decide whether a student has passed or failed a test.

2.3..1.4 Support Vector Machines

A new supervised ML algorithm is Support Vector Machines or SVM [19]. An SVM focuses on separating classes between a hyper plane, thus defining a margin. By maximizing the margin, the distance between the separating hyper plane will increase and reach its largest possible distance. This has been proven to reduce the upper bound when it comes to predicting the outcome values for unseen data.

⁶<https://link.medium.com/1C5tYhnqTV>

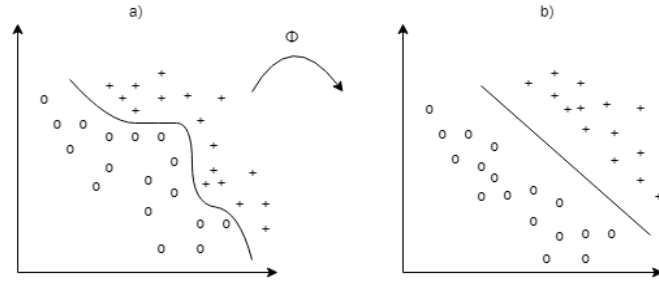


Figure 3: Illustration of SVM model, separates the non-linear data on a) into two classes b).

This technique is mostly used for non-linearly separable data because it can, with a help from a kernel function (ϕ), separate the data into two possible classes (categories) [21], See Fig. 3a and Fig. 3b.

3. Related Work

In literature of sleepiness detection, a huge field of research is sleepiness detection in vehicles in order to measure a drivers alertness levels. The motivation for these studies is that there seem to be many accidents occurring because of the negligence of sleep [11].

The method of measuring sleepiness starts with data extraction. Related work studies [11], [10], [22] and [14] use an individuals physiological parameters in order to measure sleepiness.

A study made by Prima Dewi Purnamasari and Aziz Zul Hazmi called *Heart Beat Based Drowsiness Detection System for Driver* [11] uses HB data in order to classify a drivers drowsiness level. In previous studies [12] it has shown that the HR can be reduced by 8 BPM when a person feels drowsy. Prima and Aziz system with HB as an indicator of drowsiness showed 96.52% as a success rate of their total respondents.

Further studies involving physiological parameters is the study made by Manik Mahachandra et al. called *Sensitivity of Heart Rate Variability as Indicator of Driver Sleepiness* [10]. In this study, they used HRV metrics as an indicator of sleepiness. These metrics were TD, FD, and fractal (Poincar plot method). RMSSD and SD1 of Poincar plot were the metrics that showed the best results because they could indicate sleepiness at an early stage which could be used to warn the driver.

Another related study that showed successful results for RMSSD were *Use of Subjective and Physiological Indicators of Sleepiness to Predict Performance during Vigilance Task* [22] by Kosuke Kaida et al. which involved vigilance tests. Their second finding was that SDNN could be useful in performance predictions. This was because it gave no negative predictions and was showed to be less influenced by individual differences.

In a related study by Udo Trutschel et al. called *Heart Rate Measures Reflect the Interaction of Low Mental Workload and Fatigue During Driving Simulation* [14], they could see that increase in sleepiness affects the TD measurement pNN50 in different ways. And in addition, Zcross which is the number of zero crossings in the given time window after z-transforming the IBIs. These parameters could be used in order to predict specific errors while driving, such as increment of lane variation. As for FD measurements, the VLF showed a very strong correlation to fatigue and performance data.

For FD measurements another study by Muhammad Awais et al. called *A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability* [5], revealed drowsiness-related changes in the LF component when an individual was going from awake to the drowsy state. Further an equation of LF/HF ratio, as shown below, also gave an indication of sleepiness since it would decrease apposed to the alert state.

$$R_{LF-HF} = \frac{LF}{HF} \quad (14)$$

Below one can see a table overview of the HRV features relation to sleepiness (Tab. 11).

Parameters	Shown relation to sleepiness	Did not show relation to sleepiness
MeanNN		
SDNN	[22]	
RMSSD	[10] [22]	
SDSD		
NN50		
pNN50	[14]	
VLF	[14]	[10]
LF	[14] [5]	[10] [22]
HF		[10] [22]
LF/HF ratio	[5]	[10]

Table 3: Table showing an overview of related work articles. Shows were the parameters of TD has or has not shown a relationship to sleepiness.

According to Tab. 11, the parameters SDNN, RMSSD, pNN50, VLF, LF and LF/HF Ratio seems to be common in sleepiness detection. In addition, we can see that the results are very different from study to study. For example LF has been shown to be related to sleepiness in at least two studies, but it has also been shown to not have a relation to sleepiness in two studies as well.

Another approach to measuring sleepiness which seems to be a future trend is done in studies [16] and [23]. Instead of using physiological parameters, image processing can be used in order to obtain eye measurements. These eye measurements are later used to classify an individual's sleepiness.

In a paper done by Tang-Hsien Chang and Yi-Ru Chen called *Driver Fatigue Surveillance via Eye Detection* [16] they propose a method of detecting driver alertness and fatigue via real-time extraction of eye measurements (PERCLOSE and AECS). To obtain face and eye features they used an infrared-only camera. This camera made it easier to obtain and detect the pupil since it will be displayed as a distinct white circle. Since this study involved image processing and eye features, they first needed to use a face detection technique to extract the face [16]. Their face detection technology involved ellipse-template matching algorithm. To obtain the eyes, they use a method where they subtract two frames which makes the image a black square with two white spots that represent the eyes. In order to calculate the eye size, they used something called a Sobel edge detector. The Sobel edge detector will obtain the upper and lower horizontal boundaries of the eyelids. Their system had an 97,8% accuracy rate of the eye detection module, but only 84,8% if a person was wearing glasses. Although their simplified image processing technique made the system low cost [16].

A related study that also used image processing techniques that involves the red parts of the image channel was *The Contactless Active Optical Sensor for Vehicle Driver Fatigue Detection* by Krzysztof Murawski et al. [23]. This system also uses PERCLOSE as measurements for eye activity. Furthermore, they use other measurements as PEROPEN, blink frequency, the activity of the eyes and pupil diameter.

What all the previous related studies, that did not use image processing, failed to do was to create a stable system that was not based upon sensor or electrodes placed on the body, resulting in inaccurate and bad readings if the sensor or electrodes slips off the body. Many, including themselves, discuss that this is a downside in their system. Further problems, as mentioned in [24], if such sensors are used to detect sleepiness of a driver, there is a possibility that it can hinder a drivers concentration. They also mention that when some sensors are attached, they have strict rules that glasses cannot be used and that movement needs to be restrained as much as possible. Because of this, further works that use other techniques for retrieving physiological parameters is necessary.

A downside when it comes to image processing techniques, in [16] and [23], was that when a participant is wearing glasses, there was a significant decrease in the accuracy because of problems

in the eye tracking method.

For quantifying an individual's sleepiness, related studies [10], [22] and [14] used KSS as their measurement of sleepiness on a numeric scale. Their results, as noted earlier, were successful for measuring sleepiness which shows that using some kind of scale, for example, KSS, can be useful in these kinds of systems. Also [14] notes that KSS correlates very well with their error measurements for driving and more.

Something to note is that for example, the related work by Tang Hsien and Yi [16] mentions a method of image processing which was simplified using only two important approaches. An infrared camera was used and together with that a tracking method for the eyes that could predict the placement of the eyes in the next frame, hence there are no need to do all the face detection and eye detecting steps again for each frame. This shows that there are related work studies that can help when it comes to choosing method(s) for the purpose of simplifying image processing. Image processing can help to improve the accuracy of drowsiness detection, for example, Prima and Aziz mention in their study [11] that for future work their system could be improved by adding other physiological sensors and combining it with methods of image processing.

For classifying sleepiness into multiple classes, two of the related studies used mML algorithms. In [16], the ML algorithm that was used for classification was an ANN. The ANN was used in order to give warnings when a threshold of the PERCLOSE and AECS measurements were breached [16]. Further, the warning module was also based upon user vigilance level and vehicle speed. Both ML algorithms in the related work were used in order to achieve multi-class outputs.

4. Problem Formulation and Research Questions

In daily life, sleep is one of the most important factors for an individual's well-being [25]. In addition, sleeping problems can increase the risk of hypertension, atherosclerosis, stroke, cardiac arrhythmias, and other cardiovascular conditions. The analysis and measurements of an individual's last night eye pupil or other areas of the face can be categorized by an intelligent algorithm in order to determine his/her sleepiness. The result can then make a user able to monitor their sleeping status and physiological parameters which can improve a users life-style via tailored user guidance.

We have consolidated that measuring sleepiness can be of importance in various areas which clarify the need for research and development in this area. Because of this, I decided on doing a thesis in this area. To achieve knowledge and identify what method(s) to use I came up with three questions:

- Q1: What features should be extracted from physiological parameters in order to model sleepiness?
- Q2: How can such a model be implemented using machine learning algorithms to classify sleeping status? Which machine learning algorithm(s) are best considering the correctness of sleepiness identification?
- Q3: What is the accuracy of such a system to identify/detect sleepiness?

4.1. Limitations and Challenges

This thesis was limited to only measure sleepiness by using HRV features as SDNN, RMSSD, NN50, and pNN50 from TD and VLF, LF and LF/HF ratio from FD. This was because literature revealed that these features could be relevant. I did not exclude the possibility of adding eye measurements for sleepiness detection. In the end, eye tracking methods were not included in the camera system because there was a time limitation and the literature had shown promising results for HRV features. The ML algorithms were restricted to use binary classification in order to classify sleepiness into two different categories (sleepy or not sleepy, 0 or 1).

The challenges of this thesis was to find literature which involved ML classification for sleepiness measurements. Further it was challenging to arrange meeting with the participants in the data collection since they all had a limited schedules and that one student were living outside of my home city. In the end I solved this by doing the data collection at different places in a period of 1-2 weeks. Mostly at the participant's residence (bringing a web camera and computer). The sleepiness detection subject and HRV features as a whole was interesting but also quite difficult to understand in some parts, for example the part on how to transfer time into frequency.

5. Materials and Method

By evaluating different results from the related work, I chose which methods that I thought would be most suitable for the study and development of a sleepiness detection system. In addition, this chapter describes what has been done besides this paper in order to solve the problem definition questions. The chapter covers data collection, parameter extraction, feature extraction, and intelligent system development with classification.

5.1. Data Collection

In order to achieve data in forms of physiological parameters I had to perform data collection on individuals. The first step was to find 10-15 participants to engage in the data collection. The participants were chosen from my friends and family members. All of them were students, and all except one studied at Mälardalens Högskola in Västerås. In the end, 10 participants were found and selected from available subjects in my close surrounding's. This was thought to be enough for its purpose after receiving some feedback about it.

The data collection was done through a 7 minute video recording with a web camera in an indoor environments with varying amount of ambient sunlight entering through windows, see Fig. 4 for sample pictures from two participants. The time of the day was late morning/early afternoon (midday). The web camera used was Logitech C920 (HD camera). All videos were recorded in color (24-bit RGB) at 30 frames per second (fps) with a pixel resolution of 1280 x 720 (720p) and saved in MP4 format on a laptop. The software used in order to record the individual with the web camera was called Logitech Capture. The participants were told about the study and what purpose their data will fill. During the video, they had been asked to sit in front of the web camera and to not move their face too much to the sides, to look fairly straight and to relax. After the video was taken, as a last step I subtracted the first and last minute of the 7 minute video. This was because occasionally the first and last minute of a video can be shaky due to clicking the record button, or that the participants might get nervous for a short while at the first and last minute of the recording.

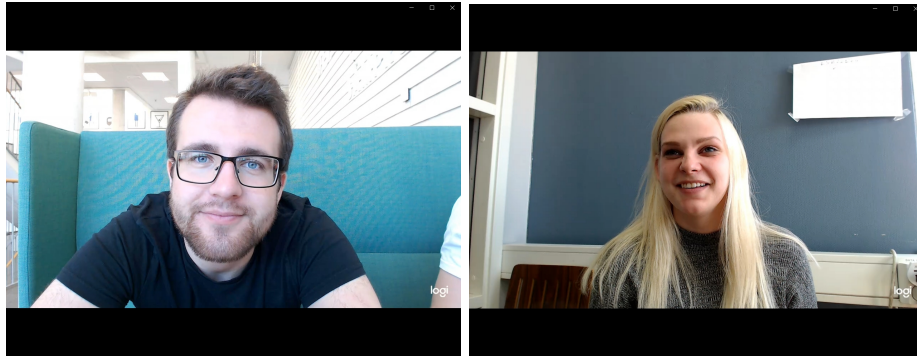


Figure 4: Two sample pictures from the video recordings.

Before the data collection, the participants were told to fill in a pre-study on how much they slept. This was in order to see when it would fit best to do the video recordings, since I needed participants that had slept both good and bad. The pre-study would show which days a participant had slept good or bad, and from that I could decide which days were best to get the most varying data. The pre-study was done each day for a week just before the actual video recording day. The pre-study had no exclusion/inclusion criteria, just to get a better understanding of a participant's sleeping patterns.

Just before the video recording, the participants were given a Google form to fill, see Appendix A 1 Fig. 19, which contained their respondent ID, gender, age, sleeping scale (see below) and if

they were wearing glasses during the video recording. The form was meant to give some additional information and define how the participants slept the night before the video recording.

In order to use supervised learning I needed to use some kind of sleepiness scale or sleep quantifier when collecting data on a participant. Some of the related work used KSS, or modified KSS, as their sleepiness representational scale [5] [14] [10] [22] but since I could not perform repetitive data collection for many weeks, I did not think that *"How sleepy are you feeling?"* would be a good question. This would probably have triggered adrenaline or nervousness in the participants leading to false or incorrect answers. If this question would have been asked for several occasions and for several weeks then I would have used it. Muhammad Awais et al. mentioned in their article *A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability* [5] that the KSS scale can be infeasible when using it in for example driving situations due to its subjective nature. Instead, I used my own scale. My own scale went from 0 - 2, and each number defined how good an individual had slept the night before the data collection. The number representation on the scale is as follows

- 0 - Neutral
- 1 - Bad
- 2 - Good

Since it can be difficult to determine what bad, neutral and good sleep is, I went for bad 0-5h, neutral 6-7h, and good 8h or more.

When receiving the form and reaching the scale question, the participants were asked if they thought that they had slept neutral, bad or good the night before the video recording (with its representing hours of sleep). In the end, the participants were 5 females aged 27, 27, 26, 21 and 22 with their representing scale values good, bad, neutral, bad and good. Further there was 5 males aged 22, 19, 24, 20 and 23 with their representing scale values neutral, neutral, good, neutral and good. Furthermore, all participants had light skin color.

In addition to the Google form, the individuals were told to fill in a Letter of Consent, see Appendix A 2 Fig. 20. This was done in order to get consent to collect and use the video data along with assigning ownership of the video data to me and my academy. The participants were asked if they agreed for pictures to be taken from the video and used in seminars and papers. Everyone except one participant agreed to this.

5.2. Parameter Extraction

For parameter extraction, I used the human face to reveal an individual's physiological parameters. A non-contact based system [6] was used to retrieve the IBI from a participant in the previously mentioned video recordings. The system uses image processing techniques which examines the red, green, and blue (RGB) color values of the face to extract physiological parameters [6].

To begin with, the videos of the participants were offline sent to the system for analyzing and acquisition of the physiological parameter. After extracting all the frames of the video, the system calculates a region of interest (ROI) of the face using face detection. The face detection techniques that were used was the Viola and Jones method and a boosted cascade classifier. The Viola and Jones method [26] was used to get the coordinates of the faces location in the first frame. Further a boosted cascade classifier was used to obtain the x and y coordinates in the first frame, together with the height and width that defines a squared ROI around an individual's face.

After obtaining the ROI, it was used in order to extract RGB color values of each pixel in the face. This was done by dividing the color values into three RGB channels and then averaging each pixels RGB value. This yields an r, g, and b measurement point and forms a raw signal for each color. To de-trend the raw traces, the system uses a procedure which includes a smoothness priors approach by [27] alongside with normalizing the traces. As the last step, in order to quantify the IBI signal, the RGB traces are sent to ICA [28], PCA [29] and FFT [30] algorithms.

After subtracting the first and last minute of the 7-minute video it went through the camera system analyzing the video data and gave responding data in forms of physiological parameters (IBI) of a participant for 5 minutes of time (300 lines of data for each participant, representing each second in the 5 minute interval). With this data and the form containing the sleepiness scale, I had obtained the necessary parameters for the intelligent system.

The advantage of using a non-contact based camera system from [6] is that there are no sensors attached to an individual's body that can fall off or slide off from its accurate position. This was a problem that was mentioned in the related work article [11]. Furthermore, this makes the system unique from all previous studies. There is no known study using a non-contact based system for measuring sleepiness via physiological parameters.

In literature, as mentioned there seem to be two significant ways of determining the sleepiness of an individual's. The two methods involve using either physiological parameters [11] [10] [22] [14], such as IBI to calculate HRV features or using image processing in order to achieve eye measurements such as PERCLOSE and AECS [16] [23].

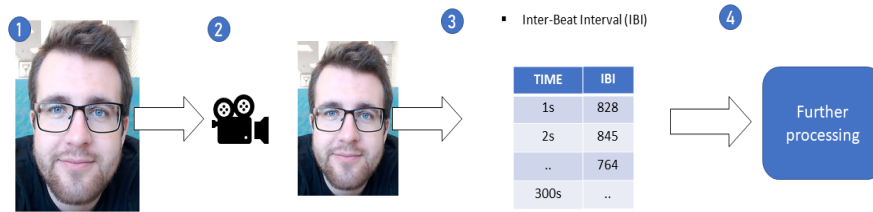


Figure 5: Parameter extraction steps.

I chose to use features of HRV related to [10], such as TD and FD in order to determine the sleepiness of an individual. To extract HRV features I had to do different calculations on an individual's IBIs. Therefore, I used the camera system to retrieve an individual's IBIs. In Fig. 5 we can see an example of the parameter extraction steps. The first step represents the recorded video, second step represents the non-contact camera system analyzing the video recordings to extract the IBI data, which output was an excel file with 300 rows representing the IBIs for each second for 5 minutes total, meaning that the frequency was 1 Hz (step 3).

id	1	2	3	4	..	10
class	0	1	1	0	...	2
1s	1066,40625	921,875	910,15625	953,125	...	1011,71875
2s	1101,5625	906,25	941,40625	992,1875	...	957,03125
..
..
300s	1046,875	929,6875	1027,34375	929,6875	...	1066,40625

Table 4: Table showing an example of the Excel file after extracting the IBI data for 10 participants. Top rows showing id and class and last rows showing the IBI data.

As mentioned, the IBI data was imported into an excel file, for an example See Tab. 4. At the top row of the table, the id indicates a participant's respondent ID and the class is the last night sleep scale value related to the participant. Under the ID and class, each IBI data row represents the IBI data for all seconds in the 5-minute video.

The reason for choosing these methods were, as they mention in one of the related work studies [10], that sleepiness was significantly correlated with HRV. Further they mention that there is a lack of reference to HRV sensitivity in sleepiness detection, hence the need for my study. By doing research about sleepiness detection I noticed that most systems use eye measurements. This must mean that it is a valuable source. But I chose to use HRV because there was no current eye tracking methods implemented in the system, and HRV has been proven to also provide good accuracy,

e.g. in [11]. Further eye tracking methods, as described in related work [16] and [23], encountered accuracy and detection problems when a participant was using glasses. This is also a factor in why I chose to focus on using physiological parameters extracted from facial features instead of just focusing on eye measurements since they can be disrupted by e.g. glasses.

5.3. Feature extraction

For feature extraction, I obtained features from TD and FD by doing calculations on an individual's IBIs. The TD features calculated from the IBIs were MeanNN, SDNN, RMSSD, SDSD, NN50, and pNN50, along with the FD features VLF, LF, HF and LF/HF ratio. Both TD and FD features were calculated with a 1 minute window size (60 seconds of IBI data). Further, the TD features were calculated as described in Tab.1 and the FD features were calculated as described in [15] and the 2. Background section, thus using the FFT to obtain the PSD. The MATLAB functions used for this was the `fft` function with 2048 as number of FFT points and the IBI, further computations was done with a re-sampling frequency of 256 Hz. The range of PSD to extract VLF, LF and HF was 0-0.04, 0.04-0.15 and 0.15-0.4.

	time(min)	TD features			FD features			class(label)
	1	f1	..	fn	F1	..	Fn	0 or 1
Person 1	..							
	5							
..								
Person 5	1							
	..							
	5							

Table 5: Table displaying example of feature matrix.

In order to store the calculated features for each individual I constructed a matrix. The matrix stores; the TD and FD features, rows representing which minute the TD and FD features were calculated for, and the sleep scale value (class). These values were stored for each participant. For an example of the feature matrix see Tab. 5.

5.4. Intelligent System Development

With the earlier achieved information from previous chapters in the report, I could start implementing the intelligent system. After calculating the TD and FD features of the IBIs and storing them into a feature matrix I had data which was important in order to train and test the accuracy of the system. Due to time limitations and other factors as mentioned earlier in the Parameter Extraction section, I did not implement new methods of image processing in order to include eye measurements for this system. This means that the system only includes features of HRV in order to determine/identify sleepiness of an individual.

For a more technical description of the implemented system see the attached document in Appendix A 3 Fig. 21.

5.4.1 Software

The software, which was used to implement the sleepiness detection system, was the desktop environment MATLAB, version R2019a. MATLAB is used by many engineers and scientist over

the world to analyze and design systems and products ⁷. MATLAB is a matrix-based language and is said to be the world's most natural way to express computational mathematics. Further, I used the MATLAB Classification Learner Tool in order to evaluate the accuracy of different ML algorithms along with extracting the best ML models (via the extract function button) to achieve the train and test accuracy myself.

MATLAB was used in at least one of the related work studies by Manik [10]. But I strongly suspect that it was used in many of them.

5.4..2 Classification





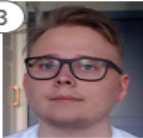

	#Outcome	Binary classifier
① 		1
② 		0
③ 		0

Figure 6: Example classification for sleepiness detection.

In order to classify sleepiness of an individual into two groups, sleepy or not sleepy (0 or 1) I decided to use ML classification. After doing research, I found that there seems to be many ML algorithms that could be used in order to counter this problem. KNN, ANN, LR, and SVM which are described in the Background chapter 2. were all valuable ML algorithms. Since I decided to use binary classification, LR was predicted to be the best option and give the best accuracy. This was because LR is intended for binary classification⁸, and so it seems to be the go-to strategy. SVM, for example, is often used for non-linearly separable data. One related study that I found that used LR was by Tomohiko Igasaki et al. called *Drowsiness Estimation Under Driving Environment by Heart Rate Variability and/or Breathing Rate Variability with Logistic Regression Analysis* [31]. To not overwhelm a reader with too much information I decided to focus on three algorithms, LR, KNN and SVM. My supervisor mentioned SVM and last KNN was chosen after seeing promising results. An ANN was considered since I had previously tried to implement it, but I did not end up completing it since I had troubles with the validation parts. In addition, the ANN was not included in the classification learner on MATLAB.

The motivation for using ML classification at all was as I mentioned in the Background chapter 2., it can help avoid mistakes when it comes to data analysis or when trying to establish a relationship between multiple features. See Fig. 6 for the intended binary classification.

Since I decided on binary classification I excluded the participants that had selected neutral on their sleepiness scale, thus I only used the participants that had slept good or bad. Originally there was 10 participants and 4 of them selected neutral, leading to an exclusion of 4 participants.

⁷https://www.mathworks.com/help/matlab/index.html?s_tid=CRUX_lftnav

⁸<https://www.quora.com/Which-is-better-for-binary-classification-SVM-or-logistic-regression>

The 6 remaining participants made the data set contain 4 participants that slept good and 2 slept bad, leading to a 33% 67% balanced set. 2 males and 2 females in the slept good section with their represented ages 23(m), 24(m), 22(f) and 27(f). Both males wore glasses. 2 females in the slept bad section with their represented ages 21 and 27.

I used MATLAB Classification Learner in order to evaluate what ML classification algorithms that gave the best validation accuracy for sleepiness detection. The algorithm settings for KNN and SVM are all adjusted according to the classification learner, thus I do not need to do this myself. This makes it possible for the program to evaluate what kind of kernel function, distance metric and so on that the model needs to provide the best accuracy. With this, there is no set kernel function, distance metric etc. since it changes by the classification learner.

First, all the features were selected and the system was trained together with 5, 7 and 10 folds cross validation. Cross-validation partitions the data into folds in order to prevent overfitting. After that, I needed to test the accuracy of sleepiness detection for the selected features. This was done by excluding different features and then training the system without them. The features that I focused on testing were SDNN, RMSSD, NN50, pNN50, VLF, LF and LF/HF ratio. By doing so, I could see what features gave the best accuracy and also what features that were not showing any correlations to an individual's sleepiness. The reason why I experimented with the selected features was that in related literature, they had shown relations to an individuals sleepiness. When it comes to leave-one-subject-out approach I noticed that this would be difficult to do with the limited amount of data after seeing that the validation accuracy was pretty low. But could definitely be done with an increased data set to see impacts from different participants.

Further after identifying which algorithms gave the best results, I extracted the functions (train classier) to obtain the training and test accuracy. This was because the classification learner only provided me with the training/validation accuracy. In order to do this, I divided the feature data from 6 participants into 80% training set and 20% test set. I extracted the test set as well since I wanted to try my trained model with unseen data and not just use the classification learner to get a validation accuracy. After that, I provided the algorithms with the training set, which output was an accuracy (train accuracy) and a trained model as output. After that, I could provide the test set to the trained model in order to calculate the test accuracy. I tried running the models with the training set and test set 10 times and then I calculated the mean (average), median, highest and lowest accuracy of these runs. I decided to calculate these values since that was what I knew something about (did not come across other methods), I calculated the matrix accuracy for the CMs but I realized that it was the same as the validation accuracy the classification learner would give me.

6. Ethical and Societal Considerations

Justin Zobel writes in his book *Writing for Computer Science* [32] that when doing research, a researcher is expected to be honest and their method(s) are expected to be done ethically. The scientific community has a set of rules, or so-called codes of conduct, that they work by. Some of them are; opinions should be clear thus not explained as facts, always stay truthful, protect the privacy of individual's that are involved and so on. Further, when it comes to ethics, they imply that one must also minimize the risk of harm to individuals, and to not plagiarize others work or present their own contribution in a wrong way. These codes of conduct have been in my mind when writing this paper and I have been taking them into consideration.

Along with the previous, Justin Zobel further mentions in his book that work within computer science, is expected to be honest and straightforward [32]. This is because if results are presented misleading, or completely wrong, it could lead to catastrophic consequences. Because of this, the results that are presented in this study will not purposely try to mislead you - the reader. When I apply my own opinions, that may not have scientific backing, it will be clear that it is the case.

Since the method states that data collection will be involved in this paper, one must think of the ethical and societal considerations related to this. This is because the participants in the data collection share their information willingly and because of this, the researcher must obtain a balanced relationship to the participant [32]. The relationship should include disclosure, trust, and awareness of potential ethical considerations.

The University of Glasgow has constructed a guide on several ethical considerations that should be taken into consideration when performing interviews [33], which can be valuable for video data collection as well. These are

- Procedures - These are basically the outlines on how the interview will be performed. The interviewees are expected to know them and be supplied with them in writing.
- Location - The location should be accustomed to the interviewees, and they should be offered an alternative to the main location if it is not suitable.
- Safety - If there are any safety concerns for the interviewees they should be notified of this and they should also let someone they trust know where and when they will be interviewed.
- Confidentiality - The interviewees should typically not be named unless it is very important to the research. They also have to give permission to this.
- Permission - All in all, the interviewees should give permission of all things that are done and recorded during the interview. Preferably the consent should be in writing.

The above considerations were thought of when preparing and executing the data collection. To gain permission and obtain a trusted relationship to the participants they were given a Letter of Consent (See Appendix A 2) to read and sign before the data collection. This letter gives them the opportunity to decline or accept that the video data will be collected and used. In addition, they prescribe the ownership of the data to me and the innovation, design and technology academy at Mälardalens Högskola. To let the participants know that they still have the freedom to leave the video recording room at any time, this is also stated in the letter. In addition, to prevent accident, injuries and other adverse consequences that may be caused by using a software product or service, all the video recordings will be saved locally. Therefore, there is a low risk of data leakage due to failure in the security of software products/services.

7. Results

In this section, the result accuracy will be presented for KNN, LR and SVM models. The features that I used for my experiment was SDNN, RMSSD, NN50, pNN50, VLF, LF and LF/HF ratio. Although I wanted to try with all features, so first I tried running the models with all 10 TD and FD features to see what accuracy they provided. The plots and matrices are displayed for the results of 10 and 7 features with KNN, LR and SVM. Further, there is a data collection report describing the significant results of the data collection.

7.1. Data Collection Report

When it comes to the data collection, all IBI data could successfully be extracted from the participants. All participants were positive and excited to be a part of this research and to provide data for the purpose of this paper and system development. The Letter of Consent was read and signed by everyone, and further all except one of the participants said that they would see no problem with me using snapshots from their videos in future papers, presentations and so on. One of my concerns after the data collection was that in some of the video recordings there were people in the background at short periods of time, I thought that this could give problems when collecting the IBI data with the non-contact camera system. Gladly, this did not end up being a noticeable problem for the system.

7.2. Classification Models and Feature Results

The results of the classification models and feature experiments are described with data from a confusion matrix (CM) and a receiver operating characteristic (ROC) curve. The CM shows where the classifier has performed poorly with true positive rates (TPR) and false negative rates (FNR). The ROC curve shows the false positive rates (FPR) on the x-axis which indicates the percentage of incorrectly assigned positive classes. And the TPR on the Y-axis which indicates the percentage of correctly assigned positive classes. Clarifying that, true positives shows the cases in which the model predicted 1 when the class label was 1 and true negatives shows when the model predicted 0 when the class label was 0 . The false positives shows when the model predicted 1 when the class label was 0 and false negatives shows the cases in which the model predicted 0 when the class label was 1 .

7.2..1 KNN

In this section, the validation accuracy for KNN with different features can be viewed. The accuracy is obtained from the classification learner on MATLAB. See Tab. 6 for feature experiment results. The table shows number of features, accuracy, model settings and number of k-folds (for validation).

KNN						
features	validation accuracy	model	number of neighbors	distance metric	distance weight	k-folds
10	86.7%	fine, weighted	1, 10	euclidean, euclidean	equal, squared inverse	7
7	83.3%	cosine	10	cosine	equal	5
6 (-SDNN)	80%	fine, weighted	1,10	euclidean, euclidean	equal, equal	5,7,10 (for all)
6 (-VLF)	83,3%	fine, weighted	1,10	euclidean, euclidean	equal, equal	5,7,10 (for all)
6 (-RMSSD)	86,7%	weighted	10	euclidean	equal	5, 7
6 (-pNN50)	86,7%	fine, weighted	1, 10	euclidean, euclidean	equal, squared inverse	5
6 (-LF/HF)	86,7%	cosine, weighted	10, 10	cosine, euclidean	equal, squared inverse	5
6 (-NN50)	86,7%	fine, weighted	1, 10	euclidean, euclidean	equal, squared inverse	5
6 (-LF)	90%	fine	1	euclidean	equal	5, 7

Table 6: The best results for KNN model with different features from the classification learner on MATLAB.

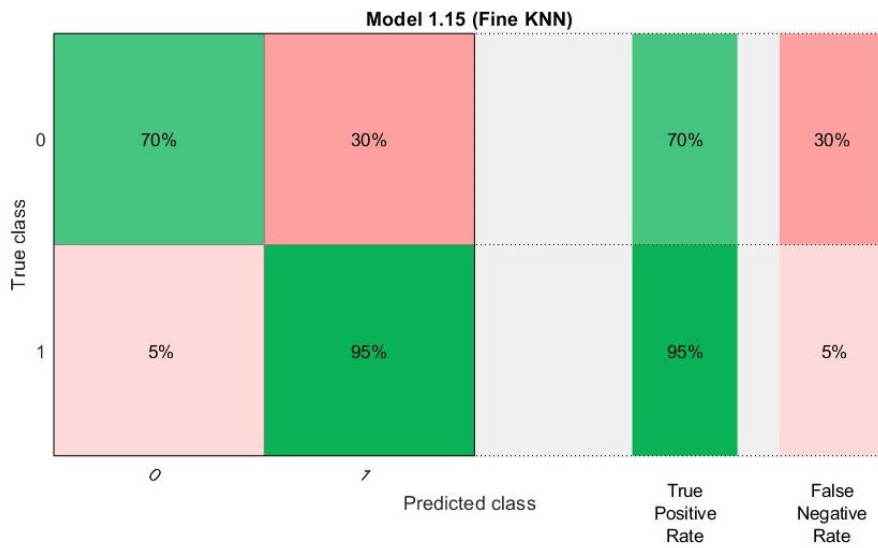


Figure 7: CM for Fine KNN with 10 features.

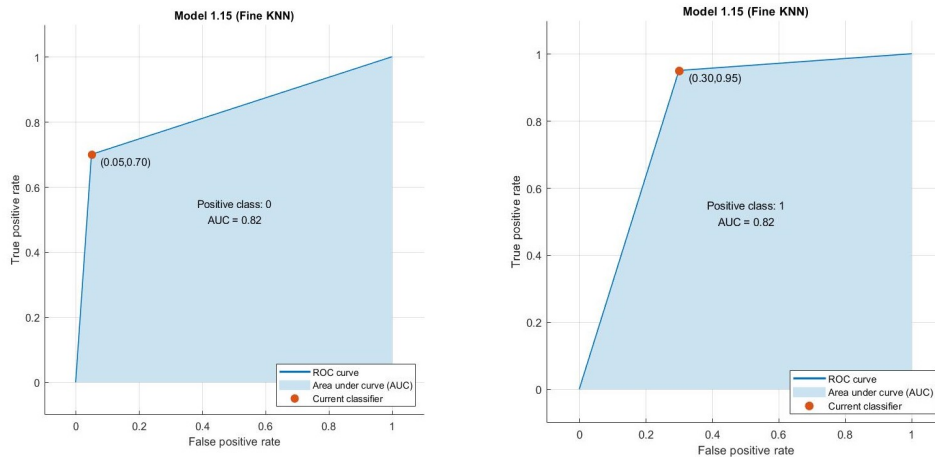


Figure 8: ROC curves for Fine KNN with 10 features.

When it comes to the results for 10 features one can see that it gave a validation accuracy of 86,7%

for Fine KNN. Its representing CM (See Fig. 7) shows 70% and 95% TPR for class 0 and 1 vs 30% and 5% FNR for class 0 and 1. The ROC curves in Fig. 8 shows 5% FPR and 70% TPR for class 0 and 30% FPR and 95% TPR for class 1.

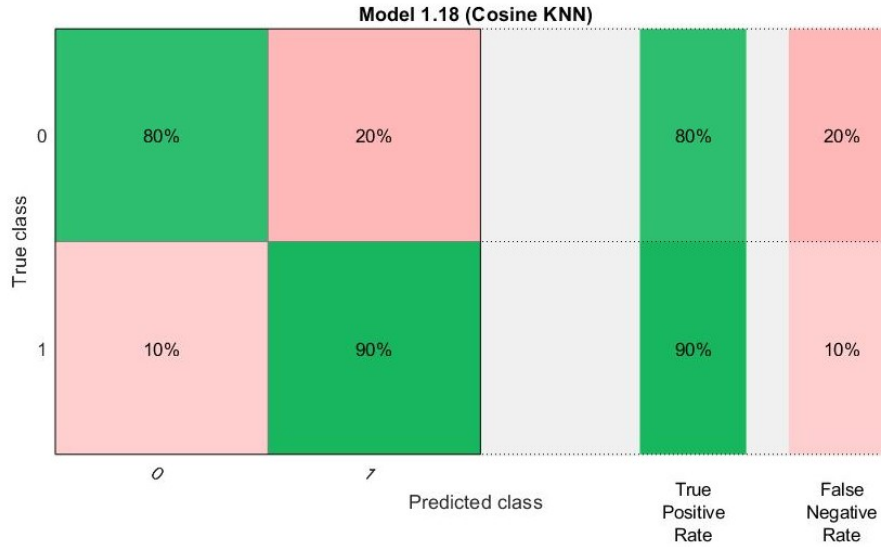


Figure 9: CM for Cosine KNN with 7 features.

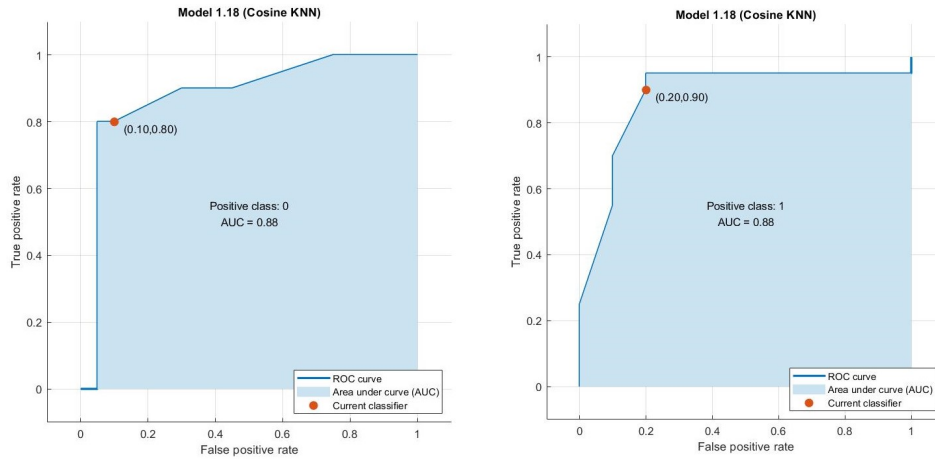


Figure 10: ROC curves for Cosine KNN with 7 features.

For 7 features one can see that it gave a validation accuracy of 83,3% for Cosine KNN. Its representing CM (See Fig. 9) shows 80% and 90% TPR for class 0 and 1 vs 20% and 10% FNR for class 0 and 1. The ROC curves in Fig. 11 shows 10% FPR and 80% TPR for class 0 and 20% FPR and 90% TPR for class 1.

7.2..2 LR

In this section the results for LR from the classification learner on MATLAB can be viewed. See Tab. 7 for feature experiment results. The table shows the features, accuracy and number of k-folds (for validation).

LR		
features	validation accuracy	k-folds
10	83,3%	7
7	83,3%	10
6 (-SDNN)	76,7%	5, 10
6 (-VLF)	83,3%	7, 10
6 (-RMSSD)	80%	7, 10
6 (-pNN50)	83,3%	7
6 (-LF/HF)	83,3%	10
6 (-NN50)	86,7%	7
6 (-LF)	86,7%	5

Table 7: The best results for LR model with different features from the classification learner on MATLAB.

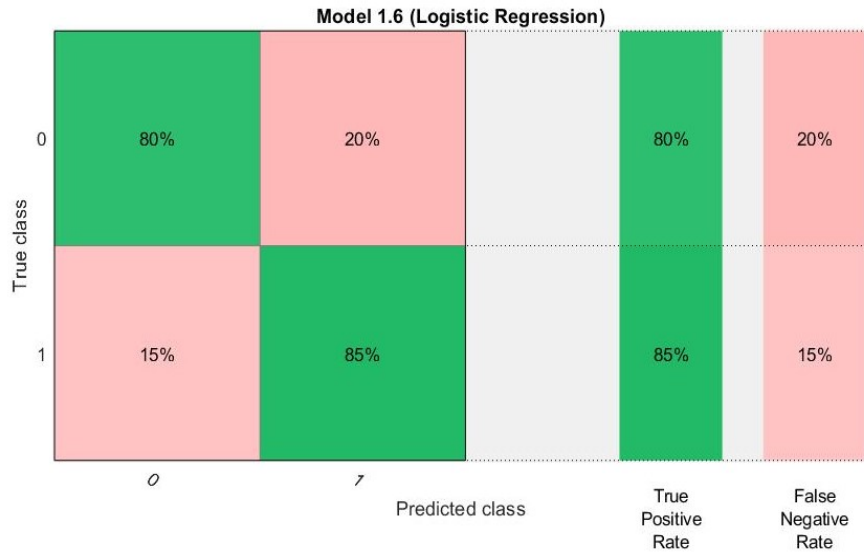


Figure 11: CM for LR with 10 features.

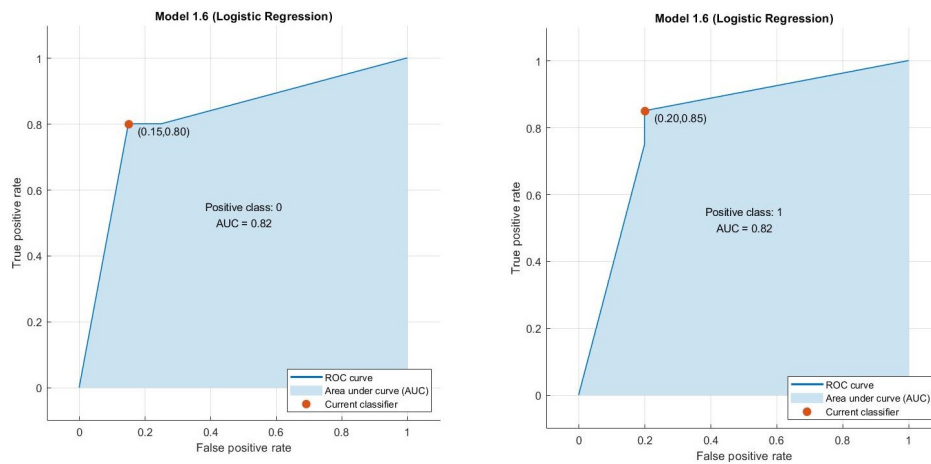


Figure 12: ROC curves for LR with 10 features.

When it comes to the results for 10 features one can see that it gave a validation accuracy of 83,3%. Its representing CM (See Fig. 11) shows 80% and 85% TPR for class 0 and 1 vs 20% and 15% FNR for class 0 and 1. The ROC curves in Fig. 12 shows 10% FPR and 70% TPR for class 0 and 30% FPR and 90% TPR for class 1.

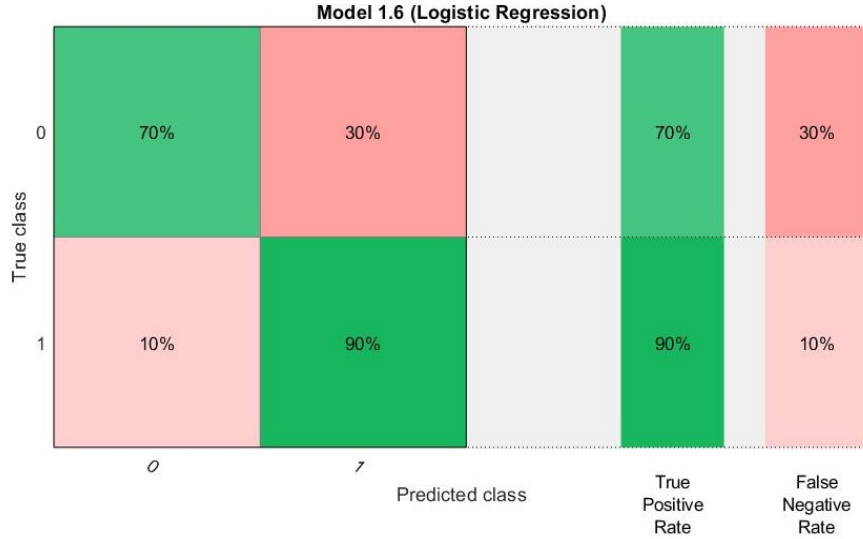


Figure 13: CM for LR with 7 features.

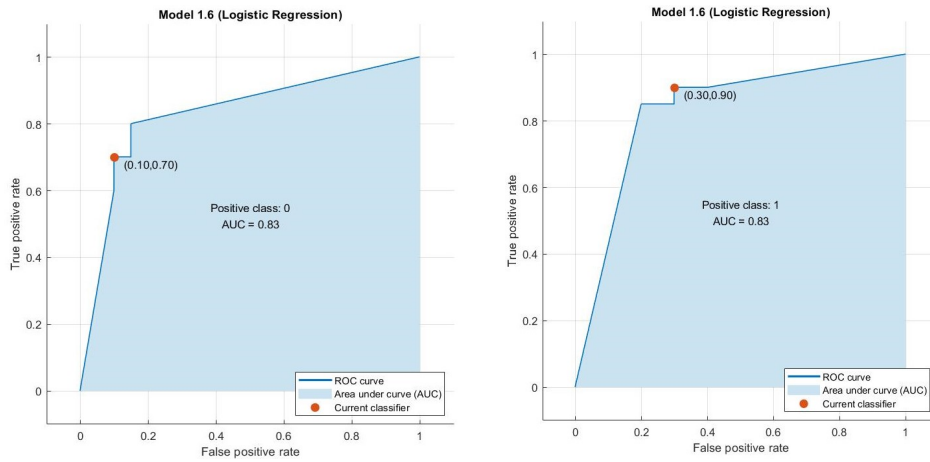


Figure 14: ROC curves for LR with 7 features.

For 7 features one can see that it gave the same accuracy as with 10 features, thus 83,3%. Its representing CM (See Fig. 13) shows 70% and 90% TPR for class 0 and 1 vs 30% and 10% FNR for class 0 and 1. The ROC curves in Fig. 14 shows 10% FPR and 70% TPR for class 0 and 30% FPR and 90% TPR for class 1.

7.2.3 SVM

In this section the results for SVM from the classification learner on MATLAB can be viewed. See Tab. 8 for feature experiment results. When it comes to the results for 10 features one can see that it gave a validation accuracy of 90% for linear SVM. The table shows number of features,

accuracy, model settings and number of k-folds (validation). The kernel scale for all is automatic except for the Medium Gaussian SVM which kernel scale is 2,4.

SVM			
features	validation accuracy	model and kernel function	k-folds
10	90%	linear	10
7	90%	linear	10
6 (-SDNN)	86,7%	linear, cubic	10
6 (-VLF)	90%	linear	7, 10
6 (-RMSSD)	90%	linear	10
6 (-pNN50)	86,7%	linear	7, 10
6 (-LF/HF)	90%	linear	5, 7
6 (-NN50)	86,7%	quadratic	5
6 (-LF)	90%	cubic	5, 7

Table 8: The best results for SVM model with different features from the classification learner on MATLAB.

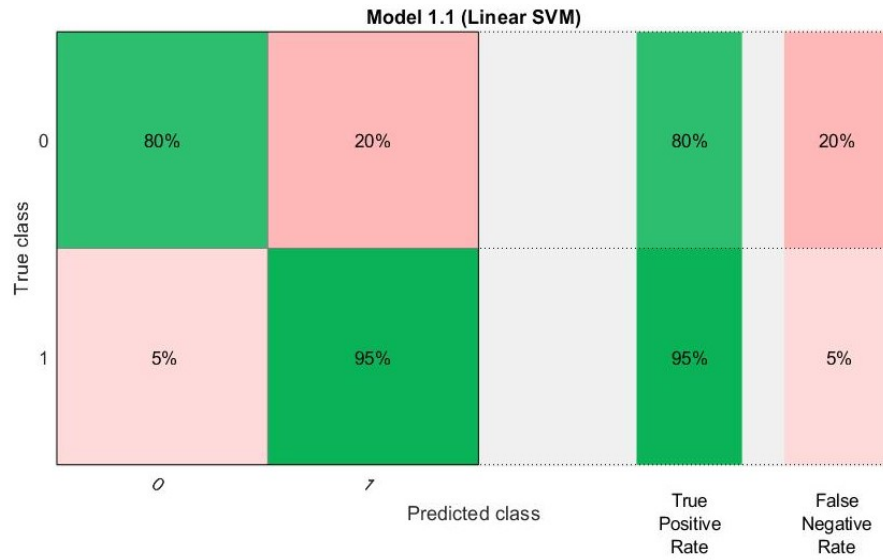


Figure 15: CM for Linear SVM with 10 features.

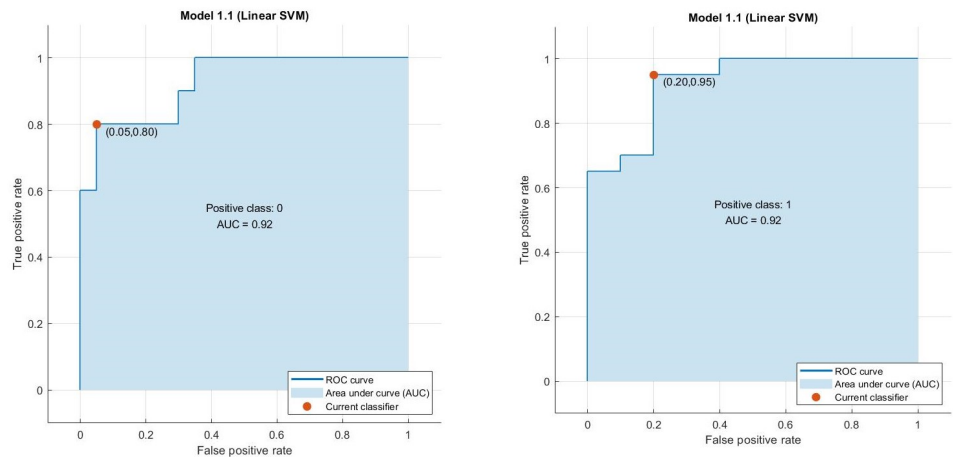


Figure 16: ROC curves for Linear SVM with 10 features.

SVM’s 10 feature CM can be viewed in Fig. 15. It shows 80% and 95% TPR for class 0 and 1 vs 20% and 5% FNR for class 0 and 1. The ROC curves in Fig. 16 shows 5% FPR and 80% TPR for class 0 and 20% FPR and 95% TPR for class 1.

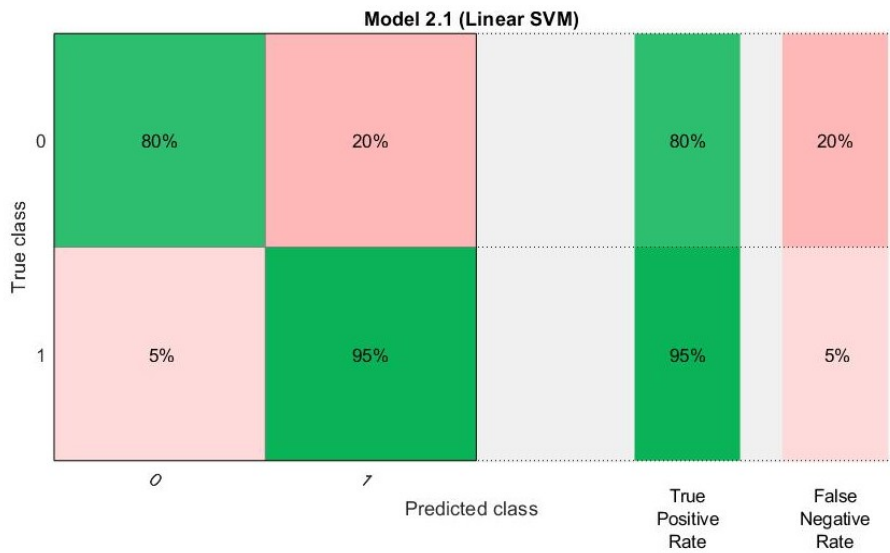


Figure 17: CM for Linear SVM with 7 features.

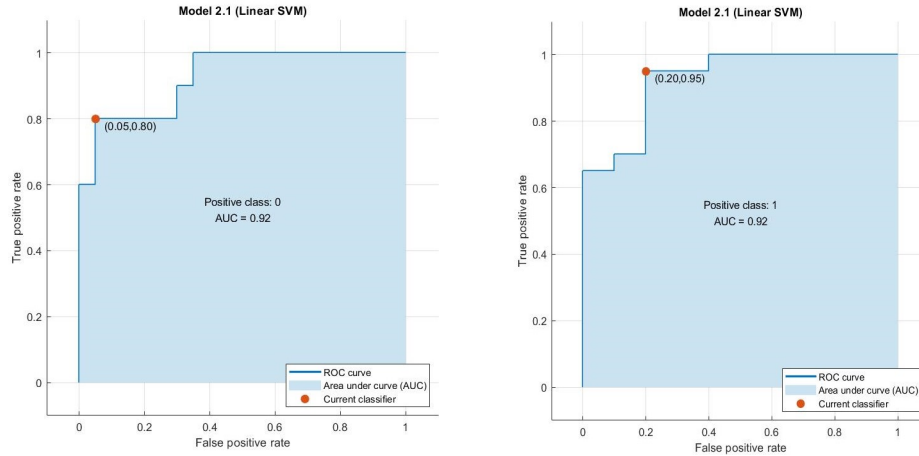


Figure 18: ROC curves for Linear SVM with 7 features.

For 7 features one can see that it gave the same accuracy as with 10 features, thus 90%. The CM can be viewed in Fig. 17) shows 80% and 95% TPR for class 0 and 1 vs 20% and 5% FNR for class 0 and 1. Which is the same matrix as can be seen for 10 features. The ROC curves in Fig. 18 shows 5% FPR and 80% TPR for class 0 and 20% FPR and 95% TPR for class 1.

7.2..3.1 Evaluation

The absolute best accuracy in the MATLAB Classification Learner for SVM and KNN was 90%. The best accuracy for LR was 86,7%. When experimenting with the features I could see that when excluding the LF feature, the accuracy increased for all the models. Because of this, I decided that I would extract the three models with only 6 features included (thus exclude the LF feature). The SVM model extracted was Cubic SVM and the KNN model was Fine KNN. For the extracted models I used 5-fold validation since it would give me the most promising results.

In the below tables 9, 10 and 11 one can see the results for the extracted LR, SVM and KNN models. The tables displays the mean, median, maximum and minimum accuracy's of 10 runs.

LR		
	training accuracy	test accuracy
Mean (Average)	79.7%	78,33%
Median	81.25%	83,33%
Maximum	91.67%	100%
Minimum	66.67%	50%

Table 9: Results for the extracted LR model.

SVM		
	training accuracy	test accuracy
Mean (Average)	84.16%	81.67%
Median	83.33%	83.33%
Maximum	91.67%	100%
Minimum	70.83%	50%

Table 10: Results for the extracted Cubic SVM model.

KNN		
	training accuracy	test accuracy
Mean (Average)	82.50%	76.67%
Median	83.33%	83.33%
Maximum	87.50%	100%
Minimum	75%	50%

Table 11: Results for the extracted Fine KNN model.

8. Discussion

In this part, I will be discussing about the results obtained and shown in the previous section (See 7. Results section). I will be reviewing and comparing the results with the previously given problem formulation questions from section 4., along with comparisons of the state-of-the-art and related works.

My first question was

- Q1: What features should be extracted from the physiological parameters in order to model sleepiness?

As a result of analysing different related work literature, I could determine what physiological parameter that was relevant for sleepiness detection, hence propose an answer to the above question.

Literature showed that the physiological parameter IBI could be relevant. Further research revealed what HRV features could be extracted from the IBI in order to model sleepiness. These features were SDNN, RMSSD, NN50, pNN50 from TD and VLF and LF from FD [11], [10], [22] and [14]. Since I also extracted meanNN, SDSD and HF, I also did try the models with 10 features. When removing these features I could see that the accuracy did not decrease nor increase in SVM and LR models. Although for KNN, the accuracy would decrease by 3,4% after removing these features. This shows that the features can be relevant for some models.

The mean accuracy received for the extracted SVM model was 84,16% for the training set and 81,67% for the test set. For LR it was 79,7% and 78,33%. Last for KNN the mean accuracy was 82,50% and 76,67%. This means that the best results of using HRV features for sleepiness detection gave a mean accuracy of 84,16% for the training set and 81,67% for the test set with the SVM model. I believe that this might compete with some of the state-of-the-art architectures, at least [16] since their system would get an accuracy of 84,8% if a participant was wearing glasses. Otherwise further related work had over 95% accuracy, which is hard to compete with. However, I think that it is not a complete catastrophe either.

In LR, the feature that seemed to be most important was SDNN, since when removing it the accuracy would decrease by 6,6%. For KNN, the accuracy would decrease by 3% when the SDNN feature was removed. For SVM, SDNN was also important, this was because when removing it, the accuracy would decrease by 3,3%. This means that for all models, the SDNN parameter was important and showing correlations to an individual's sleepiness. In the related study [22] the SDNN parameter was shown to give no negative predictions, hence it was seen to have a strong correlation to an individual's sleepiness. Further parameter that showed indication of sleepiness in their study was RMSSD. In my results I could see that when removing this parameter in the LR model, it would decrease the accuracy meaning that it also showed correlations to sleepiness in my study. Related study [10] also showed that RMSSD can be used to detect sleepiness.

The parameters that would increase the accuracy for all models after being removed was LF, meaning that it was the absolute worst parameter. LF/HF ratio did not show any increase nor decrease in accuracy for the models after being removed. These results were different from what was seen in related work [5], where LF/HF ratio was showing correlations to an individual's sleepiness.

The pNN50 feature would give an increase in accuracy for SVM but not for LR. This might be because LR is dependent of features that can be linearly separable and SVM is not. So I think that this means that the pNN50 features were hard to linearly separate for LR and therefore SVM could separate this feature better. Just because the accuracy would increase for LR when this feature was removed, it does not mean that it is not relevant for sleepiness detection it just was not fit for that certain model. In related study [14], pNN50 feature was also shown to be related to an individual's sleepiness.

Additional parameters that were shown to not give any increase nor decrease in accuracy was; VLF and RMSSD for SVM, VLF for KNN.

For KNN the additional features that gave an accuracy increase after being removed was RMSSD, pNN50 and NN50. This makes me think that KNN might be one of the worst models with the settings that was given in the classification learner. I believe so because related work studies had shown relation to an individual's sleepiness with these parameters and also this could be seen in the LR and SVM models. Maybe this has do to with how the classification learner chooses the amount of neighbors, because for all it would choose either 1 or 10. If choosing 1 neighbour and the data points has a nonlinear relationship, it can lead to overfitting if a data point in an area that is generally negative, is predicted positive. In Tab. 11 we can see that the KNN model would give the lowest mean test accuracy of all models. This might be because of overfitting.

So, as a result of the feature experiment, I would say that SDNN showed really good results for sleepiness detection in all algorithms (KNN, LR and SVM) vs LF which showed poor results for sleepiness detection.

Furthermore, my second question was

- Q2: How can such a model be implemented using machine learning algorithms to classify sleeping status? Which machine learning algorithm(s) are best considering the correctness of sleepiness identification?

It was challenging for me to find related work which had been using ML algorithms in order to classify sleepiness. What I found was the related study by Tang-Hsien and Yi-Ru [16]. As mentioned in the related work method they used an ANN, but in their work, they used more than two groups as their output (multi-class classification). The outputs that they used in their alert system was based upon how dangerous the situation was in correlation to sleepiness, vehicle speed, and vigilance level [16]. The outputs were categorized into four different groups, which were Safety, Caution, Danger and Very Dangerous.

When it comes to my system, I decided to use binary classification in order to determine if an individual belongs to one out of two groups. In the 5. Method section I mentioned that in literature, LR was explained to be intended for binary classification. With this, I predicted that LR would give the best accuracy. As a result and seen in the 7. Results section, I found that LR can be used in order to classify sleepiness and that it gave fairly good accuracy but not as good as SVM. As discussed above about question one, the KNN model that I extracted might not be valid because of the possible overfitting problem. Although I could see promising results for the LR and SVM models which means that they could be relevant for binary classification and sleepiness detection. But since SVM had the best mean accuracy of 84,16% for the training set and 81,67% for the test set, I would say that it appears to be the strongest option for sleepiness detection. SVM had an overall good accuracy for the features since it gave a validation accuracy of 90% in the Classification Learner for 6 out of 9 experiments, see Tab. 8. In addition, I think that since SVM gave the best mean accuracy, the feature data might be non-linearly separable. Which would be a reason in why SVM gave the best results.

My last question was

- Q3: What is the accuracy of such a system to identify/detect sleepiness?

As mentioned before, the best mean accuracy was 84,16% (training set) and 81,67% (test set) for binary classification with SVM to identify/detect sleepiness. The data provided to train and test the system was restricted since I wanted to extract the LF feature since it showed poor results

for all models. In addition to this, I had to exclude the neutral values for the participants in the data collection. Further the training and test set were divided into 80% and 20% of the feature data each which I think was a factor in both the training and test mean accuracy. I believe that the system accuracy for the limited amount of training data is positive. Although we could see that with this amount of data, the KNN models results might have been because of overfitting. I am not sure if this has to do with how the classification learner picks its neighbours settings or that it needs more data to pick suitable settings. For SVM and LR I believe that the data was enough to not lead to overfitting, but we can see in CM results (See 7. Results section) that the models would provide a better percentage value of TPR and FNR for class 1. This might have been because the data set was not completely balanced (more people had slept good then bad), I believe that this can be fixed with a larger and more balanced data set.

Another thing that might have affected the results is that I did the data collection during midday. Even though it is known that lack of sleep can lead to excessive sleepiness during the day [9], it might not be the optimal choice to wait so long before the video recording. Maybe a participant has had time to drink coffee or eat food which can lead to a more alert state. This is something that I did not think of during my data collection. I did not come across anything in my readings, surroundings or previous knowledge that made me think of this when planning to collect data. Also, I had not done any data collection before which means that I did not think of the importance of outside factors (time of day, energy boosters, skin color, light variation, etc).

When it comes what I explained about the possible problem of people disrupting the videos in the background, I thought that this could have been a problem for the camera system and decrease the accuracy. Although, as seen in the accuracy, I believe that with the limited amount of data, the accuracy for the different algorithms were really good. I think that the people in the background did not become a problem otherwise the accuracy would have been worse. This might have been because the person in the background was so far back, maybe at least 2 meters away, and the camera system did not detect or try to detect that individual. A limitation to the camera system is that it can only detect and extract one individual's physiological parameters at a time, and this could have been a factor in why it did not try and detect a second person entering the frame.

As mentioned, the classification algorithm that gave the best accuracy was SVM since it had the best mean accuracy of 10 runs. I believe that this was because of its way of handling non-linearly separable data. Since LR uses the sigmoid function (to give probabilities) and a tipping point value to classify data, the data must be linearly separable. SVM can handle complex problems but still use a "linear" fashion ⁹. If I could have experimented with less features, I could have tried to identify which feature(s) that had linearly separate data which could have increased the accuracy for LR. But for that I need to collect more data. And for KNN, as explained before, it had problems with the model (neighbor) settings.

⁹<https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f>

9. Conclusions

In this section the overall report will be summarized and a conclusion of the end results and end analysis will be presented. Furthermore, the potential future work will be mentioned when it comes to improvements and extensions of the system.

As we can see in the 7. Results section HRV features can, in fact, be used for sleepiness detection since valid results in the form of accuracy were given for several features with different classification algorithms. This means that physiological parameters, such as IBI and HRV features, are relevant and could be used in order to model sleepiness. This answers my first question which was *"What features should be extracted from the physiological parameters in order to model sleepiness?"*. The SDNN, pNN50 and RMSSD features from TD did show to be useful in sleepiness detection. The feature data experiments strongly suggested that SDNN was by far the best measurement for identifying if an individual was sleepy or not sleepy. Other measurement such as LF did not at all show as great potential. Related study [14] showed that LR can be used to detect sleepiness. In the present study, however, I found that it had negative correlations.

10 participants were included in a data collection where they quantified their sleepiness on a scale from 0-2. The IBI data, to calculate the HRV features, was extracted using a non-contact camera system by [6]. In the end, 6 participants data were used for the classification models. The best validation accuracy from MATLABs Classification Learner was 90% which were given by the classification algorithms SVM (Linear and Cubic) and KNN (Fine). Further, with the extracted models, SVM gave the best mean accuracy for the training and test set which was 84,16% and 81,67%. This means that the SVM model, which gave the best correctness (mean accuracy) can, in fact, be a relevant model to identify/detect sleepiness. In addition, the classification learner can be used in order to evaluate and extract implemented models which would fit best to a given classification problem. This answers my second question *"How can such a model be implemented using machine learning algorithms to classify sleeping status? Which machine learning algorithm(s) are best considering the correctness of sleepiness identification?"*. But as mentioned in the 8. Discussion section, it might be a problem with how the classification learner chooses settings for KNN (or that I had a too small data set) because my results for the KNN model might be because of overfitting. When it comes to the last question, *"What is the accuracy of such a system to identify/detect sleepiness?"*, the mean accuracy, as mentioned, of such a system reached 84,16% at best for the training set and 81,67% for the test set to identify/detect sleepiness with SVM.

9.1. Future work

Since the ML algorithms where restricted to using binary classification further work could be to use multi-class classification instead. With this, one could do experiments and evaluate what ML models that are most relevant for multi-class sleepiness detection. The base of this thesis was, as mentioned, to use a non-contact camera system to measure sleepiness. In order to do this, I had to perform data collection on participants in order to obtain videos for the camera system. Now there was only 10 individuals participating in the data collection which did not provide that much data. I believe that if there was more data to train the system with the overall accuracy of the classification algorithms would improve. Therefore, a possible future extension would be to add more data by doing an additional data collection. Also, for data collection something to consider would be to include real road driving scenarios for the participants to see how well the system is doing in a real-world application. In addition, as described in the 5. Method section, I would want to do repetitive data collection on the participants for many weeks, and to use an existing sleepiness scale (such as KSS) in the future.

In the end, as mentioned in the 8. Discussion section and seen throughout the report there was no eye measurements parameters added. This could be a possible future work to extend the system with. It could make the system more robust and accurate since there can be more features added to detect sleepiness. Related study [16] and [23] showed accurate results for sleepiness detection with eye measurements as PERCLOSE and AECS, this could be two valid measurements to add

as a start of future work. Furthermore, the non-linear features of HRV was never used as well. This could be something to consider in the future.

In the area of health status monitoring the system could be used to detect more factors that make it possible to monitor an individual's well-being. The non-contact camera system can be used to extract more physiological parameters that could be used in order to extend the system. Stress or Cognitive Load can be something interesting to research about and see if they can be measured via physiological parameters.

References

- [1] D. Reney and N. Tripathi, “An Efficient Method to Face and Emotion Detection,” in *presented at 5th International Conference on Communication Systems and Network Technologies, Gwalior, India*, 2015, p. 493.
- [2] “Trötthet orsak till många trafikolyckor,” *ViBilägare*, Jun 2009. [Online]. Available: <http://www.vibilagare.se/nyheter/trotthet-orsak-till-manga-trafikolyckor-2575>
- [3] D. J. Gottlieb, J. M. Ellenbogen, M. T. Bianchi, and C. A. Czeisler, “Sleep deficiency and motor vehicle crash risk in the general population: A prospective cohort study,” *BMC Medicine*, vol. 16, no. 1, pp. 1–10, 2018.
- [4] “Drowsy driving: Asleep at the wheel,” *Centers for Disease and Control and Prevention (CDC)*, Nov 2018. [Online]. Available: https://www.cdc.gov/features/dsdrowsydriving/index.html?fbclid=IwAR2-yOS7_Qv_V8WvmeIvEhmlq4mIcScP51reLZfhuS0jIPfJ_uGVIJKyC-g
- [5] M. Awais, N. Badruddin, and M. Driberg, “A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and Wearability,” *Sensors (Switzerland)*, vol. 17, no. 9, pp. 1–16, 2017.
- [6] H. Rahman, M. U. Ahmed, and S. Begum, “Non-contact Physiological Parameters Extraction using Camera,” in *the 1st Workshop on Embedded Sensor Systems for Health through Internet of Things (ESS-H IoT)*, October 2015.
- [7] F. Shaffer and J. P. Ginsberg, “An Overview of Heart Rate Variability Metrics and Norms,” in *Frontiers in Public Health*, vol. 5, no. September, 2017, pp. 1–17.
- [8] M. S. Nixon and A. S. Aguado, *Feature Extraction Image Processing for Computer Vision Third Edition*. Academic Press, 2012.
- [9] P. Lavie, J. Zomer, I. Eliaschar, Z. Joachim, E. Halpern, A. H. Rubin, and G. Alroy, “Excessive daytime sleepiness and insomnia,” *Archives of otolaryngology (Chicago, Ill. : 1960)*, vol. 108, no. 6, pp. 373–7, 1982. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/7092673>
- [10] M. Mahachandra, Yassierli, I. Z. Sitalaksana, and K. Suryadi, “Sensitivity of heart rate variability as indicator of driver sleepiness,” in *2012 Southeast Asian Network of Ergonomics Societies Conference: Ergonomics Innovations Leveraging User Experience and Sustainability, SEANES 2012*. IEEE, 2012, pp. 1–6.
- [11] P. Dewi Purnamasari and A. Zul Hazmi, “Heart Beat Based Drowsiness Detection System for Driver,” in *Proceedings - 2018 International Seminar on Application for Technology of Information and Communication: Creative Technology for Human Life, iSemantic 2018*, no. 22. IEEE, 2018, pp. 585–590.
- [12] M. R. Waldeck and M. I. Lambert, “Heart rate during sleep: Implications for monitoring training status,” *Journal of Sports Science and Medicine*, vol. 2, no. 4, pp. 133–138, 2003.
- [13] M. U. Ahmed, S. Begum, and M. S. Islam, “Heart Rate and Inter-beat Interval Computation to Diagnose Stress Using ECG Sensor Signal,” in *MRTC Report*, 2010, p. 4.
- [14] U. Trutschel, M. Golz, C. Heinze, D. Sommer, B. Sirois, and D. Edwards, “Heart rate measures reflect the interaction of low mental workload and fatigue during driving simulation,” p. 261, 2012.
- [15] T. Cui, “Spectrum Analysis of Heart Rate Variability,” 2013. [Online]. Available: <http://lup.lub.lu.se/luur/download?func=downloadFile&recordId=4092312&fileId=4092333>
- [16] T. H. Chang and Y. R. Chen, “Driver fatigue surveillance via eye detection,” in *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*. IEEE, 2014, pp. 366–371.

- [17] V. H. Milan Sonka and R. Boyle, *Image Processing, Analysis, and Machine Vision Fourth Edition*. Cengage Learning, 2014.
- [18] A. T. Duchowski, *Eye Tracking Methodology Theory and Practice Third Edition*. Springer, 2017.
- [19] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Supervised Machine Learning: A Review of Classification Techniques,” *Artificial Intelligence Review*, vol. 26, pp. 159–190, 2006.
- [20] S. Dreiseitl and L. Ohno-Machado, “Logistic regression and artificial neural network classification models: A methodology review,” *Journal of Biomedical Informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.
- [21] D. A. Salazar, J. I. Vélez, and J. C. Salazar, “Comparison between SVM and Logistic Regression: Which One is Better to Discriminate?” *Revista Colombiana de Estadística*, vol. 35, no. 2 SPEC. ISSUE, pp. 223–237, 2012.
- [22] K. Kaida, T. Åkerstedt, G. Kecklund, J. P. Nilsson, and J. Axelsson, “Use of Subjective and Physiological Indicators of Sleepiness to Predict Performance during a Vigilance Task,” *Industrial Health*, vol. 45, no. 4, pp. 520–526, 2007.
- [23] K. Murawski, T. Sondej, K. Rozanowski, O. Truszczyński, M. Macander, and L. Macander, “The contactless active optical sensor for vehicle driver fatigue detection,” in *Proceedings of IEEE Sensors*, 2013, pp. 3–6.
- [24] A. Chowdhury, R. Shankaran, M. Kavakli, and M. M. Haque, “Sensor Applications and Physiological Features in Drivers’ Drowsiness Detection: A Review,” *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3055–3067, 2018.
- [25] G. E. Plante, “Metabolism,” *Departments of Medicine (Nephrology), Physiology, and Pharmacology, and Institute of Geriatrics*, vol. 55, no. 2, October 2006.
- [26] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” no. February, 2005, pp. I–511–I–518.
- [27] M. P. Tarvainen, P. O. Ranta-Aho, and P. A. Karjalainen, “An advanced detrending method with application to HRV analysis,” *IEEE transactions on bio-medical engineering*, vol. 49, no. 2, pp. 172–5, 2002. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/12066885>
- [28] P. Comon, “Independent component analysis, a new concept,” p. 287314, 1994.
- [29] M. Krammer, “Nonlinear principal component analysis using autoassociative neural networks,” p. 233243, 1991.
- [30] R. L. Veenkant, “A Serial Minded FFT,” no. 3, pp. 180–185, 1972.
- [31] T. Igasaki, K. Nagasawa, N. Murayama, and Z. Hu, “Drowsiness estimation under driving environment by heart rate variability and/or breathing rate variability with logistic regression analysis,” in *Proceedings - 2015 8th International Conference on BioMedical Engineering and Informatics, BMEI 2015*, no. Bmei. IEEE, 2016, pp. 189–193.
- [32] J. Zobel, *Writing for Computer Science, Third Edition*, 2014.
- [33] Uni of Glasgow, “Ethical Issues in Interviews,” *Uni of Glasgow*, no. July 2002, p. 1, 2002. [Online]. Available: https://www.gla.ac.uk/media/media_237671_en.pdf

A Appendices

1.1. Appendix 1

Data collection

This form is meant to collect some additional data for each respondent in the data collection.

**Obligatorisk*

Respondent ID *

Ditt svar

Gender

☐ Female

☐ Male

☐ Prefer not to say

Age

Ditt svar

Sleepiness scale *

☐ Bad

☐ Neutral

☐ Good

☐ Övrigt:

Will you be having glasses during the video? *

☐ Yes

☐ No

☐ Övrigt:

Figure 19: Google Form to be filled by the participants in the data collection.

1.2. Appendix 2

Consent Form

Terms of Agreement

Below are sets of statements that give you the depositor, a series of options in terms of how you wish your video data, or a portion of your video data, to be archived. Please note that 'video data' always refers to 'anonymized video data' (i.e. you are not named). Further, during the data collection, the respondent will remain the right to leave the room at any time.

By signing below:

I agree to deposit the whole/an agreed portion of my video data to Fanny Danielsson and members of the innovation, design, and technology academy at Mälardalens Högskola.

I agree that my video data/an agreed portion of my video data will be available from 13/4-2019 and 10 years further to researchers and the public for scholarly and educational.

By giving my permission:

I do/do not agree that Fanny Danielsson and members of the innovation, design, and technology academy at Mälardalens Högskola, may use this video data/an agreed portion of the video data, without any further approval on my part. It is/is not necessary to contact me again for permission.

I hereby assign ownership of the video data/portion of the video data to Fanny Danielsson and members of the innovation, design, and technology academy at Mälardalens Högskola. I understand that I nevertheless retain copyright, subject to the rights I have granted Fanny Danielsson and members of the innovation, design, and technology academy at Mälardalens Högskola to make copies and publish and to grant permissions to others to do so.

Signature

Date

Name

Respondent ID number (to be entered by research team)

Figure 20: Letter of Consent to be signed by the individuals participating in the data collection.

1.3. Appendix 3

Technical Report

By Fanny Danielsson

The below definitions and functions are a description of what has been implemented and used in the MATLAB code of the intelligent system.

1. Definitions

computationMatrix = the 10x7 computation matrix for storing TD and FD measurements for 5 minutes of time, along with the sleepiness scale for each participants.

numberOfIndividuals = number of participants in the data collection, in this case 10.

numberOfMinutes = number of minutes of IBI data. In this case 5 minutes.

trainSet = 80% of computation matrix, used to train a classification model. Including label.

testSet = 20% of computation matrix, used to test a trained classification model. Not including labels.

testSetLabels = Storing the actual labels for the test set.

testAccuracy = The test accuracy for a trained classification model.

validationAccuracy = The training accuracy returned after training a classification model.

2. Functions

2.1 Heart Rate Variability metric calculation

VDM_hrvTimeDomainAnalysisStatistical(IBIinterval) = function for calculating time domain measurements.

VDM_hrvFrequencyDomainAnalysisFFT_NEW(IBI,256, 'hamming', 2048, 0, 0.04, 0.04, 0.15, 0.15, 0.4) = function for calculating frequency domain measurements.

2.2 Classification

trainClassifierLR = Returns a trained LR classifier and its accuracy. Obtained from MATLAB Classification Learner.

Input:

trainingData: a matrix with the same number of columns and data type as imported into the app.

Output:

trainedClassifier: a struct containing the trained classifier. The struct contains various fields with information about the trained classifier.

trainedClassifier.predictFcn: a function to make predictions on new data.

validationAccuracy: a double containing the accuracy in percent.

trainClassifierKNN = Same as above but classifier is KNN.

trainClassifierSVM = Same as above but classifier is SVM.

trainedModel.predictFcn(testSet) = Exported classifier to make predictions for new data.

Figure 21: Technical explanation for the implementation of the intelligent system.