

# Portable System to Detect driver drowsiness with Body Sensors (PoSDBoS) \*

Paul Pasler  
Reutlingen University  
Paul.Pasler@Student.Reutlingen-  
University.DE

## Abstract

Fahrerassistenzsysteme sind aus modernen Fahrzeugen nicht mehr wegzudenken. Müdigkeitserkennung hilft Sekundenschlaf oder müdigkeitsbedingte Unachtsamkeit zu vermeiden und verhindert somit schwere Unfälle. Systeme mit Körpersensoren zeigen in verschiedenen Studien sehr genau Ergebnisse und erkennen Müdigkeit frühzeitiger als andere Ansätze. In der vorgelegten Arbeit wurde ein solches System mit einem Elektroenzephalogramm (EEG) umgesetzt und getestet. Hierfür wurden Testdaten aufgenommen, verarbeitet und mit einem künstlichen Neuronalen Netz klassifiziert, sodass der aktuelle Status des Fahrers „Wach“ oder „Müde“ unterschieden werden kann.

## Schlüsselwörter

Fahrerassistenzsystem, Müdigkeitserkennung, Elektroenzephalogramm, Signalverarbeitung, Machine Learning, Neuronales Netz

## CR-Kategorien

A.0 [ACM]: Experimentation

\*

Betreuer Hochschule: Prof. Dr. Martinez  
Hochschule Reutlingen  
Natividad.Martinez@Reutlingen-  
University.de



Master Project IoT 2016

31. Juli 2016, Hochschule Reutlingen  
© 2016 Paul Pasler

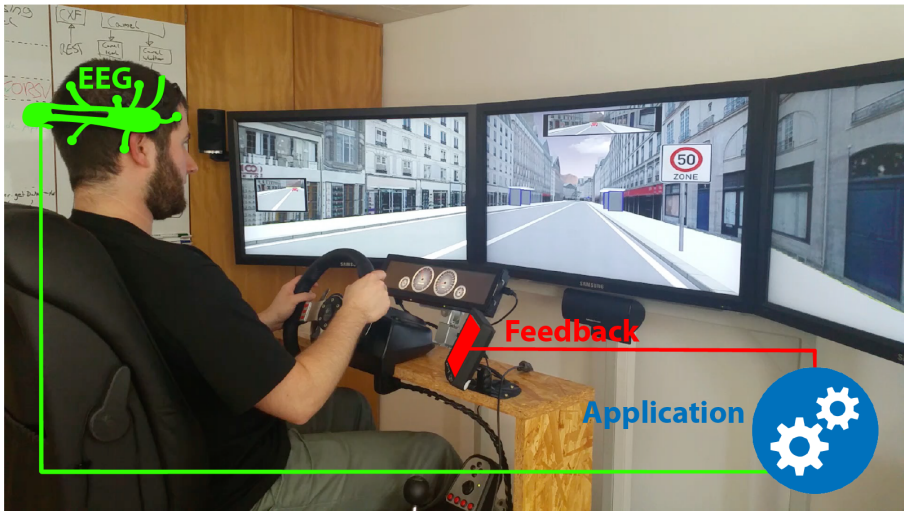
## 1 Einleitung

Fahrerassistenzsysteme sind innerhalb weniger Jahren von der Oberklasse, in die Mittel- und Kleinwagenklasse vorgedrungen. Die Unternehmensberatung Strategy Analytics schätzt, dass in den nächsten Jahren sechs mal so viele Fahrerassistenzsysteme verbaut werden als heute [1]. Sie bieten dem Fahrer erhöhten Komfort (Tempomat) oder die Sicherheit (Notbremsassistent). Laut der Boston Consulting Group, könnte der flächendeckende Einsatz von Fahrerassistenzsystemen, die Unfallrate um ein knappes Drittel zurückgehen lassen [2].

Zu Gruppe der Sicherheitsrelevanten Fahrerassistenzsysteme gehört auch die Müdigkeitserkennung. Beispielsweise rät die Müdigkeitserkennung „Attention Assist“ von Daimler dem Fahrer, zu gegebenen Anlass, eine Pause einzulegen und zeigt ein Kaffeesymbol im Cockpit an [3]. So wird müdigkeitsbedingte Unachtsamkeit oder Sekundenschlaf, die oftmals die Folge von schweren Unfälle sind, entgegengewirkt.

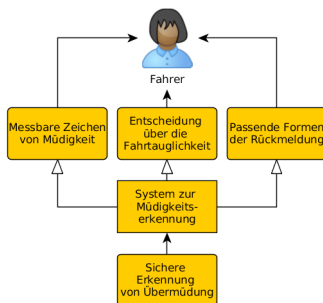
In Deutschland wurden 2015 rund 2,5 Mio. Unfälle polizeilich aufgenommen, die Zahl der Verkehrstoten liegt bei 3.450 [4]. Neben überhöhter Geschwindigkeit, zählt laut dem Deutschen Verkehrssicherheitsrat Müdigkeit zu den häufigsten Unfallursachen und ist damit für jeden fünften schweren Unfall verantwortlich [5]. Dies verdeutlicht das Potential einer frühzeitigen Erkennung von Müdigkeit und einer Meldung an den Fahrer.

Für eine sichere und korrekte Erkennung von Übermüdung ergeben sich mehrere Problem-



**Abbildung 1: Skizze des Systemaufbaus: Körpersensoren (Elektroenzephalografie / Elektrokardiogramm) liefert Daten an die Applikation und ein Feedback-Device warnt den müden Fahrer. Bild zeigt den Fahrersimulator der Reutlingen University.**

stellungen (Abb. 2). Anzeichen von Müdigkeit müssen vom System genau analysiert werden, um eine sichere Aussage über die Fahrtauglichkeit des Fahrers treffen zu können. Erkennt das System eine gefährliche Situation, muss der Fahrer in geeigneter Weise darauf hingewiesen werden.



**Abbildung 2: Das System zur Müdigkeitserkennung gliedert sich in mehrere Teilproblemstellungen auf.**

Daraus ergeben sich Anforderungen für ein multimodales System zur Müdigkeitserkennung (Abb. 1). Es existieren bereits diverse Systeme, denen es jedoch oftmals an Komfort und Portabilität mangelt. Ziel dieser Arbeit ist die Entwicklung eines Systems zur Müdigkeitserkennung mit einem EEG-Headset. Dessen Signale werden verarbeitet und mit einem KNN klassifiziert, mit dem Ziel den Fahrer rechtzeitig vor Übermüdung und deren Folgen zu warnen. Statt dem klassischen EEG wird ein Emotiv Epoc eingesetzt werden, dadurch soll die Beeinträchtigung des Fahrers möglichst gering gehalten werden. Auch wenn sich der Ansatz weniger für den Serienbetrieb eignet, soll das System für die Validierung von berührungslosen Systemen verwendet werden. Die notwendigen Testdaten werden im Rahmen des Projekts im Fahrersimulator der Reutlingen University aufgenommen. Dennoch soll das ganze System leicht portierbar sein, um es in einem echten Fahrzeug testen zu können.

zu können.

Die Ausarbeitung gliedert sich folgendermaßen. Im Kapitel 2 werden verschiedene Forschungsergebnisse zur Müdigkeitserkennung aufgezeigt und analysiert. Die Beschaffung der Daten und die durchgeführten Versuche sind Thema von Kapitel 4. Die Implementierung eines portablen Systems zur Müdigkeitserkennung mit Körpersensoren wird im Kapitel 5 vorgestellt. Kapitel 6 beschreibt die Ergebnisse und leitet in Kapitel 7 zu den weiteren Schritten und dem Fazit über. In den anschließenden Absätzen werden Grundlagen für die kommenden Kapitel erläutert.

## 2 Stand der Technik

Systeme zur Erkennung von Müdigkeit versuchen mit verschiedenen Parametern herauszufinden, ob sich die Person (der Fahrer) noch in einem aufmerksamen Zustand befindet. Die lässt sich in drei Bereiche unterteilen: Fahrverhalten, Computer-Vision (CV) und Körpersensoren. Die beiden ersten Bereich werden nur für die manuelle Markierung von Datensätzen genutzt. Die folgenden Ansätze beziehen sich ausschließlich auf die Erkennung mit Körpersensoren.

Für die Erkennung von Müdigkeit werden verschiedene Körpersensoren bzw. deren Kombination (multimodal) eingesetzt. Meist werden elektrische Spannung am und im Körper gemessen. Neben dem EEG, werden bspw. die Elektrokardiographie (EKG) oder Elektrookulografie (EOG) genutzt. Beim EKG wird die elektrische Aktivität des Herzmuskels erfasst, um bspw. die Herzfrequenz zu bestimmen. Das EOG misst die Bewegung der Augen, um bspw. Blinzeln zu erkennen.

Ansätze mit einem EKG allein, zeigten in verschiedenen Arbeiten kein eindeutiges Ergebnis [6], [7]. Beide versuchten Informationen aus der Herzfrequenzvariabilität zu erhalten und diese zu klassifizieren.

Khushaba et al. [8] versuchten EEG, EKG und EOG zu verbinden und verglichen verschiedenen Kombinationen. Mit einem „fuz-

zy wavlet“ basierten Algorithmus wurde die Signale aufbereitet und zeigten, dass ein EEG alleine bereits ausreicht. Die Kombination eines EEG mit EKG bzw. EOG verbesserte das Ergebnis nicht signifikant. Auch Johnson et. al [9] kamen zu der Erkenntnis, dass ein EEG ausreicht und das genutzte EOG nicht benötigt wird.

[10] konnten mit einem EEG die Zustände „Wach“, „Schläfrig“ und „Schlafend“ unterscheiden. Die Wavelet-Transformation und das genutzte künstliche Neuronale Netz (KNN) führten zu einer Erkennungsrate von 93%. Vuckovic et al. fanden den besten Algorithmus für die Initialisierung des KNNs: Der Learning Vector Quantization Algorithmus. Im Vergleich mit EEG-Experten erreichte das KNN eine Übereinstimmung von 90%. Huang et al. [11] nutzten ein Hidden Markov Modell zu Erkennung und erreichten eine gute Erfolgsrate. Lin et al. [12] nutzten die Unabhängige Komponenten Analyse (UKA) und Lineare Regression (LR) und konnten zeigen, dass hiermit bis zu 88% richtige Ergebnisse erzielt werden können.

Die betrachteten Arbeiten unterstreichen die Eignung des EEGs um Müdigkeit zu erkennen. Kombiniert mit anderen Sensoren können die Ergebnisse nur noch leicht verbessert werden.

## 3 EEG Headset

Für das Projekt wurde ein EMOTIV Epoc+ EEG Headset <sup>1</sup> (Abb 3) verwendet. Es besitzt 14 EEG Kanäle, sowie ein Gyroskop und sendet seine Daten via Bluetooth an den Rechner. Das Headset wird über den Kopf gestülpt und besitzen an den Sensorenden einen Filz der mit Kochsalzlösung befeuchtet wird.

Die Sensoren Anordnung ist an das internationale 10-20 System angelehnt (4). Die Rohdaten werden Abtastrate von 128 geliefert und enthalten neben dem Wert auch die Signalstärke (Qualität). Leider liefert die mitgelieferte SDK die Rohdaten nicht in

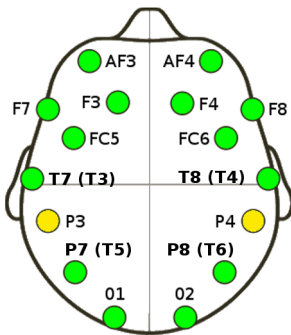
---

<sup>1</sup><http://emotiv.com/epoc>



**Abbildung 3: Das EMOTIV Epoc+ EEG Headset wird einfach über den Kopf gestülpt.**

Echtzeit, weshalb die Open-Source Lösung Emokit<sup>2</sup> eingebunden wurde.



**Abbildung 4: Die 14 Kanäle (Grün), sowie die beiden Qualitätssensoren (Gelb).**

## 4 Testdatenbeschaffung

Um Daten von übermüdeten Fahrern zu erhalten, kann natürlich kein Versuch im Straßenverkehr durchgeführt werden. Die Fremd- und Eigengefährdung wäre einfach zu groß. Darum fanden die Versuche in einer Simulationsumgebung statt. Im Fahrsimulator der IoT Labs<sup>3</sup> der Reutlingen University wird im Experiment eine Nacht-Autobahnfahrt simuliert. Verschiedene Studien [13], [14] legen nahe, dass sich Simula-

tionen zwar von der Realität unterscheiden, dass jedoch die Ergebnisse trotzdem valide und brauchbar sind.

Der Fahrsimulator (Abb. 5) besteht aus einem Simulationsrechner mit drei 20" Monitoren, einem echten Fahrersitz, Lenkrad, Schaltung und Pedalen. Für die Simulation wird die Open-Source Software OpenDS<sup>4</sup> genutzt. Per TCP/IP werden alle notwendigen Fahrzeugdaten vom Simulationsrechner auf den Datensammler und dort in ein virtuelles Steuergerät-Software (Vector CANoe<sup>5</sup>) geschickt. Über einen CAN-Bus können die Daten dann wieder, mittels einer Schnittstelle (CAN-Interface), ausgelesen werden. Die eigentliche Applikation wird auf dem Embeddedrechner ausgeführt.

Für den Versuch werden neben den rohen EEG Signalen, auch die Fahrzeugdaten, sowie die Simulation und der Fahrer aufgenommen. Weiterhin kann der Versuchsleiter auch Besonderheiten protokollieren (Abb. 6).

Im Versuch sollte der Fahrer eindeutige Anzeichen von Müdigkeit zeigen. Dies kann durch verschiedene Versuchsparameter begünstigt werden. So zeigt eine Studie, dass Unfälle meist zwischen 2:00 - 6:00, sowie 14:00 - 16:00 Uhr passieren [14]. Auch die Schlafmenge von weniger als 6 Stunden in der Nacht vor dem Experiment erhöht die Chance auf Anzeichen [13]. Das Geschlecht oder Alter der Probanden ist nicht relevant. Vor dem Experiment sollten jedoch keine Drogen, Alkohol oder Kaffee eingenommen werden. Ein Führerschein ist von Vorteil, aber nicht zwingend notwendig.

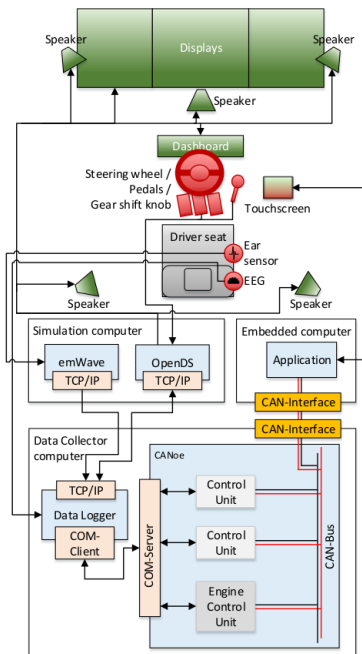
Auch die Teststrecke (Abb. 7) trägt zur Erhöhung der Müdigkeit bei. Monotone Autobahnfahrten die größtenteils geradeaus verlaufen, ohne andere Verkehrsteilnehmer und konstanter Geschwindigkeit führen eher zu einer Übermüdung. Nach diesen Kriterien wurde eine endlose zweispurige Autobahnkarte mit einer Geschwindigkeit von konstant 130Kmh erstellt. Sie spielt zudem

<sup>2</sup><https://github.com/openyou/emokit>

<sup>3</sup><http://iotlab.reutlingen-university.de/>

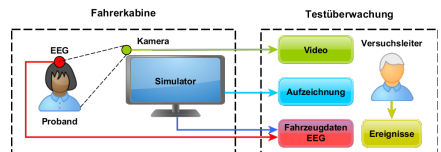
<sup>4</sup><https://www.opens.eu>

<sup>5</sup>[https://vector.com/vi\\_canoe\\_de.html](https://vector.com/vi_canoe_de.html)



**Abbildung 5: Der Aufbau des Simulators der Reutlingen University mit den drei Rechnern für die Simulation, Datensammlung und Applikation.**

Nachts und ist eher Dunkel gehalten, was besonders anstrengend für die Augen ist. Für einen Versuch werden 40 Minuten angesetzt. Fünf Minuten werden für eine kurze Einführung, 30 Minuten für die Testfahrt und wieder fünf Minuten für eine kurze Einschätzung mit Fragebogen. Anhand der aufgenommenen Daten werden nun Stellen gesucht, an denen die Testperson eindeutige Anzeichen von Müdigkeit zeigt. Eindeutig sind Verhaltensweisen wie häufiges Gähnen und Einnicken (Kopf fällt nach vorn) - Diese Merkmale werden häufig in CV-Ansätzen genutzt. Auch Verhaltensmerkmale wie, von der Spur abkommen und heftig Gegenlenken oder deutliche Veränderungen der Geschwindigkeit können Anzei-



**Abbildung 6: Der Versuchsaufbau für die Erkennung von Müdigkeit mit den Datenströmen aus der Fahrerkabine zur Testüberwachung.**



**Abbildung 7: Die Autobahnkarte für den Versuch verläuft endlos geradeaus, Nachts und bei konstanter Geschwindigkeit.**

chen für eine Unachtsamkeit wegen Müdigkeit sein. Diese Stellen werden dann in den EEG Daten mit dem Label „Müde“ markiert, alle anderen mit „Wach“. Die EEG Sequenzen können dann auf eindeutige Varianzen untersucht werden. Im nächsten Kapitel ist dies Thema der Datenaufbereitung.

## 5 Müdigkeitserkennung mit einem EEG-Headset

Abbildung 8 zeigt die Gesamtübersicht des entwickelten Systems dar. Alle blauen Klassen der Anwendung wurden in Python 2.7 implementiert. Der Fahr Simulator und seine Schnittstellen (rosa) in Java und C#. Abschnitt 5.1 befasst sich mit den Rohdaten und deren Weiterreichung im System. Die Verarbeitung der Rohdaten ist Thema von Abschnitt 5.2. Im folgenden Abschnitt werden daraus die passenden Merkmale extrahiert. In Abschnitt 5.4 wird die Arbeitsweise des Klassifikators näher beleuchtet.

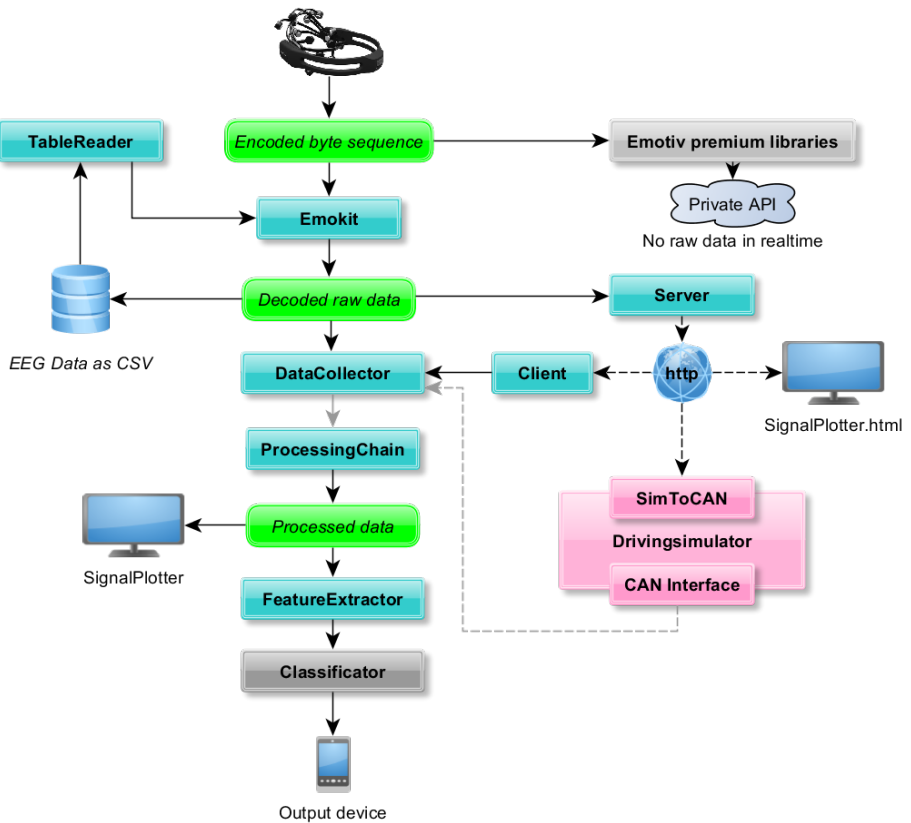


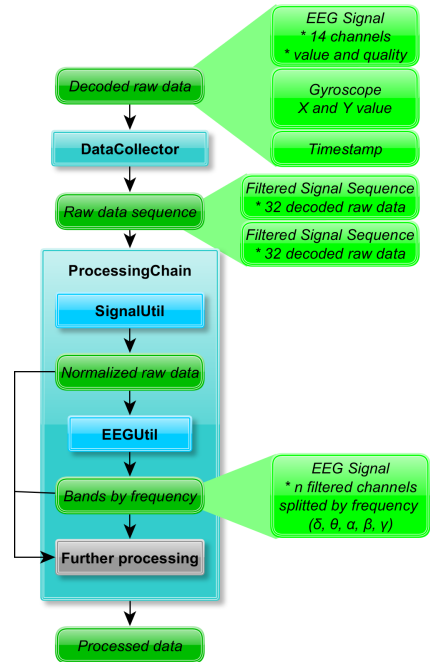
Abbildung 8: Das entwickelte System zur Müdigkeitserkennung.

## 5.1 Datensammlung

Das EEG Hedset schickt enkodierte Byte-Sequenzen via Bluetooth, an die proprietären Emotiv Premium Libraries oder die Open-Source Lösung Emokit (vgl. 3). Die Emokit Klasse wurde für das Projekt leicht modifiziert, sodass sie sich ins System einfügt. So wurde Unterstützung für das neueste EPOC+ Modelle implementiert, sowie die Möglichkeit, Testdaten aus dem TableReader zu versenden. Die dekodierten Rohdaten enthalten 14 EEG Kanäle mit Wert und Qualität, die Gyroskopwerte in X- und Y-Richtung und einen Zeitstempel (Abb. 9). Diese Daten können als CSV gespeichert werden, sowie an einen HTTP-Server oder direkt an den DataCollector übergeben werden. Über den Server integriert die SimToCAN Anwendung via HTTP die EEG Daten in den Fahrsimulator und das virtuelle Steuergerät (4).

Der DataCollector kann Daten direkt oder aus einem HTTP-Client empfangen. Die Schnittstelle zum Fahrsimulator bzw. zum CAR-Interface ist vorbereitet. So ist es möglich die Datensammlung und die Datenverarbeitung auf verschiedenen Rechnern durchzuführen.

Die Aufgabe der DataCollector-Klasse ist es, die einzelnen Signale in Sequenzfenstern von 128 Signalwerten zu aggregieren. Das ist die Abstrakte des EEG-Headsets und entspricht somit in etwa den Signalen einer Sekunde. Es sind zwei dieser Fenster implementiert und sie überschneiden sich in der Hälfte. So ist gewährleistet, dass signifikante Stellen nicht verloren gehen. Die Fensterfunktion ist ein simples Rechteck, sodass alle Werte gleich gewichtet werden. Eine andere Fensterfunktion bspw. mit Glockenkurvenverlauf (Hamming- oder Hann-Fenster) wäre einfach einzubauen. Zudem fügt der DataCollector nur konfigurierte Kanäle hinzu, sodass sich die Datenmenge reduziert. Im folgenden Abschnitt werden die gefilterten Sequenzfenster nun verarbeitet und aufbereitet.



**Abbildung 9: Die Daten werden immer weiter reduziert und verarbeitet.**

## 5.2 Datenverarbeitung

Die gefilterten EEG-Sequenzen durchlaufen nun eine Verarbeitungskette (Abb. 9).

Im ersten Schritt werden die Signale auf das Intervall 1 bis -1 normalisiert, dazu werden die Signale durch den jeweiligen Maximalwert bzw. Betrag des Minimalwertes geteilt. Dieses Vorgehen stellt sicher, dass die absolute Amplitude keinen Einfluss auf die Gewichtung im Klassifikator hat. Zudem wird die Datenmenge wiederum reduziert.

Im zweiten Schritt werden die Signale in Frequenzbänder (EEG-Bänder) unterteilt. Hierzu werden bestimmte Frequenzbereiche aus dem Signal entfernt, sodass nur die gewünschten Frequenzen erhalten bleiben. Diese EEG-Bänder gliedern sich in folgende Frequenzbereiche und werden nach griechischen Buchstaben benannt:

- $\delta$  : 0,1 bis < 4Hz
- $\theta$  : 4 bis < 8Hz
- $\alpha$  : 8 bis < 13Hz
- $\beta$  : 13 bis < 30Hz
- $\gamma$  : > 30Hz

Den Frequenzbändern werden verschiedene Eigenschaften zugesprochen

## TODO Frequenzbänder

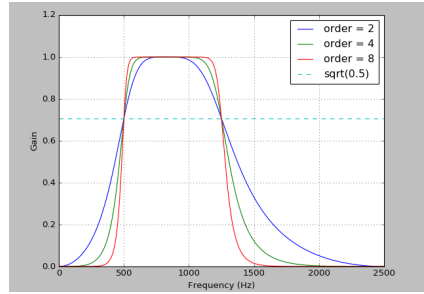
Um die einzelnen Frequenzbänder zu erhalten ist eine Filterfunktion notwendig. Für die Anwendung wurde hierfür ein Butterworth-Filter[15] eingesetzt. Gleichung 1 zeigt die Übertragungsfunktion mit  $A_0$ : Gleichspannungsverstärkung,  $\Omega = \frac{f}{f_s}$ : auf Grundfrequenz normierte Frequenz und  $n$ : Ordnung des Filters. Abbildung 10 zeigt exemplarisch Filterfunktionen verschiedener Ordnung mit den Grenzen von 500 bis 1250Hz. Alle Frequenzen darunter und darüber werden deutlich abgeschwächt bzw. gehen gegen null. Der Butterworth-Filter verläuft nahe Eins im gewünschten Bereich, fällt an den Grenzen ab und stellt sicher, dass das Signal an den Grenzen um  $\frac{1}{\sqrt{2}} \approx 0.7071$  gemindert wird. Je höher die Ordnung, desto steiler geht die Funktion durch die angegebenen Grenzen. Der Filter lässt sich gut in Hardware realisieren.

$$|A|^2 = \frac{A_0^2}{1 + k_{2n}\Omega^{2n}} \quad (1)$$

## 5.3 Mermalsextraktion

## 5.4 Klassifikator

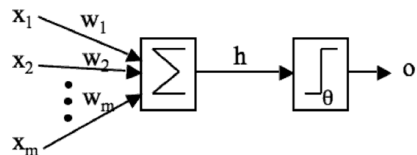
Nachdem in vorherigen Abschnitt eine passende Merkmalsmenge erstellt wurde, geht es nun um die Entscheidung, ob der Fahrer Müde oder Wach ist bzw. ob das System eine Müdigkeitsmeldung erscheinen lässt. Für diese Klassifizierung werden im allgemeinen Machine-Learning-Algorithmen verwendet. Anhand von markierten Datensätzen



**Abbildung 10: Die Filterfunktion eines Butterworthfilters 2., 4., und 8. Ordnung im Frequenzbereich von 500 bis 1250Hz.**

wird versucht den Algorithmus zu Trainieren (Überwachtes Lernen). Dies dient dem Ziel, dass er auch unbekannte Daten klassifizieren kann. Dieser Vorgang wird Generalisierung bezeichnet und ist auch im menschlichen Lernen ein wichtiger Schritt.

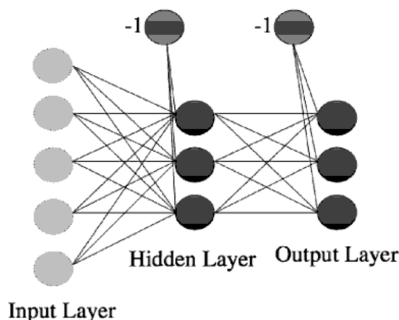
Für die Anwendung wurde zur Klassifizierung ein künstliches Neuronales Netz ausgewählt. Es basiert auf einem erweiterten McCulloch-Pitts-Neuron [16] und ist der Funktionsweise des menschlichen Gehirns bzw. seinen Neuronen nachempfunden[17]. Ein KNN lässt sich im einfachsten Fall durch eine Merkmalsmenge  $X = x_1, x_2, \dots, x_n$ , dazugehörige Gewichte  $W = w_1, w_2, \dots, w_n$ , eine Übertragungsfunktion  $\Sigma$  und eine Schwellwertfunktion  $\theta$  beschreiben (Abb. 11).



**Abbildung 11: Darstellung eines McCulloch-Pitts-Neurons. Die Merkmale  $X$  werden mit den Gewichten  $W$  multipliziert und in  $\Sigma$  summiert. Wenn  $h > \theta$  „feuert“ das Neuron ( $o = 1$ ) [17].**



Dieser vereinfachte Aufbau kann schon einfache Aufgaben, wie bspw. ein logisches „UND“, erfüllen. Schon ein logisches „XOR“ lässt sich nicht mehr abbilden. Dafür muss eine oder mehrere Schichten von Neuronen (Hidden Layers) hintereinander geschaltet werden (Abb. 12).



**Abbildung 12: Darstellung eines Neuronalen Netzes mit mehreren Schichten (Multi Layer Perceptron, MLP)[17].**

## 6 Ergebnis

## 7 Fazit und Ausblick

### Literatur

- [1] Strategy Analytics. Advanced driver assistance systems forecast - aug 2015. <https://www.strategyanalytics.com/access-services/automotive/powertrain-body-chassis-and-safety/market-data/report-detail/advanced-driver-assistance-systems-forecast---aug-2015>, 2015. Zugriff: 2015-10-28.
- [2] Xavier Mosquet, Michelle Andersen, and Aakash Arora. A roadmap to safer driving through advanced driver assistance systems. <https://www.bcgperspectives.com/Images/MEMA-BCG-A-Roadmap-to-Safer-Driving-Sep-2015.pdf>, 2015. Zugriff: 2015-10-28.
- [3] Daimler AG. Attention assist, 2008. Available at <http://media.daimler.com/dcmedia/0-921-658892-49-1147698-1-0-0-0-0-1-11702-854934-0-1-0-0-0-0-0.html>, Zugriff: 2015-08-13.
- [4] Statistisches Bundesamt. Zahl der verkehrstoten steigt im jahr 2015 voraussichtlich auf etwas 3450. [https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2015/12/PD15\\_463\\_46241.html](https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2015/12/PD15_463_46241.html), 2015. Zugriff: 2015-12-31.
- [5] Claudia Evers. Unterschätzte Risikofaktoren Übermüdung und ablenkung als ursachen für schwere lkw-unfälle. [http://www.dvr.de/presse/seminare/904\\_20.htm](http://www.dvr.de/presse/seminare/904_20.htm), 2008. Zugriff: 2015-10-20.
- [6] Jose Vicente, Pablo Laguna, Ariadna Bartra, and Raquel Bailon. Detection of driver's drowsiness by means of hrv analysis. In *Computing in Cardiology, 2011*, pages 89–92, Sept 2011.
- [7] E. Rogado, J.L. Garcia, Rafael Barea, Luis M. Bergasa, and Elena Lopez. Driver fatigue detection system. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1105–1110, Feb 2009.
- [8] Rami N. Khushaba, Sarath Kodagoda, Sara Lal, and Gamini Dissanayake. Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm. *Biomedical Engineering, IEEE Transactions on*, 58(1):121–131, Jan 2011.
- [9] Robin R. Johnson, Djordje P. Popovic, Richard E. Olmstead, Maja Stikic, Daniel J. Levendowski, and Chris Berka. Drowsiness/alertness algorithm development and validation using synchronized EEG and cognitive performance

to individualize a generalized model. *Biological psychology*, 87(2):241–250, May 2011.

- [10] Abdulhamit Subasi. Automatic recognition of alertness level from eeg by using neural network and wavelet coefficients. *Expert Syst. Appl.*, 28(4):701–711, May 2005.
- [11] Ruey S. Huang, Chung J. Kuo, Ling-Ling Tsai, and Oscar T.C. Chen. Eeg pattern recognition-arousal states detection and classification. In *Neural Networks, 1996., IEEE International Conference on*, volume 2, pages 641–646 vol.2, Jun 1996.
- [12] Chin teng Lin, Ruei cheng Wu, Sheng fu Liang, Wen hung Chao, Yu jie Chen, and Tzyy ping Jung. Eeg-based drowsiness estimation for safety driving using independent component analysis. *IEEE Trans. Circuits Syst. I, Reg. Papers*, pages 2726–2738, 2005.
- [13] Johan Engstrom, Emma Johansson, and Joakim Ostlund. Effects of visual and cognitive load in real and simulated motorway driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8(2):97–120, March 2005.
- [14] Jim Horne and Louise Reyner. Vehicle accidents related to sleep: a review. *Occupational and Environmental Medicine*, pages 289–294, May 1999.
- [15] Stephen Butterworth. On the Theory of Filter Amplifiers. *Wireless Engineer*, 7, 1930.
- [16] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [17] Stephen Marsland. *Machine learning : an algorithmic perspective*. Chapman & Hall/CRC machine learning & pattern recognition series. CRC Press, Boca Raton, 2009. A Chapman & Hall book.