

# **Klassifikation von respiratorischen Ereignissen mit Earables und maschinellem Lernen**

Bachelor Thesis  
von

**David Laubenstein**

Chair of Pervasive Computing Systems/TECO  
Institute of Telematics  
Department of Informatics

First Reviewer:  
Second Reviewer:  
Supervisor:

Prof. Dr. Michael Beigl  
Franziska Mathis-Ullrich  
Tobias Röddiger

Project Period: 01/11/2019 – 01/04/2020



# Inhaltsverzeichnis

<b>1 Einführung</b>	<b>1</b>
<b>2 Grundlagen &amp; aktuelle Forschung</b>	<b>3</b>
2.1 Schlafmedizin . . . . .	3
2.2 Klassifizierung von Schlafstörungen . . . . .	4
2.2.1 Zentrales Schlafapnoe . . . . .	4
2.3 Maschinelle Lernverfahren . . . . .	5
2.3.1 Random Forest . . . . .	6
2.3.2 XGBoost . . . . .	7
2.3.3 Support Vektor Maschine (SVM) . . . . .	7
2.4 Forschung von Klassifikation anhand von IMU-Daten . . . . .	8
2.4.1 Accelerometer am Brustkorb . . . . .	8
2.4.2 Detektion direkt am Kopf mittels der Google Glass Brille . . . . .	9
2.4.2.1 Pulssignal . . . . .	10
2.4.2.2 Atemsignal . . . . .	11
2.4.3 Überwachung von Puls und Atmung mittels eSense-Earpods .	12
<b>3 Schlafanalyse</b>	<b>13</b>
3.1 Earable Plattform . . . . .	13
3.1.1 Was wird aufgezeichnet? . . . . .	13
3.1.2 Datenexport . . . . .	14
3.2 Polysomnographie-Systeme . . . . .	14
3.2.1 Datenexport . . . . .	16
3.3 Kamera . . . . .	16
3.4 Datensynchronisation . . . . .	16
3.5 Zusatzinformationen der Nutzer . . . . .	17
<b>4 Design</b>	<b>19</b>
4.1 Studienplanung . . . . .	19
4.2 Studienablauf . . . . .	20
<b>5 Implementierung</b>	<b>23</b>
5.1 App . . . . .	23
5.1.1 Plattform . . . . .	23
5.1.2 Messungsablauf . . . . .	24
5.1.3 Messung . . . . .	24
5.2 Anbindung an Auswertungspipeline . . . . .	25
5.2.1 Synchronisation der Daten . . . . .	25
5.3 Verarbeitungspipeline zur Klassifikation . . . . .	26

---

<b>6 Evaluation</b>	<b>29</b>
6.1 Gibt es passende Features? . . . . .	29
6.2 Ablauf der Evaluierung . . . . .	29
6.2.1 Data Labeling . . . . .	29
6.2.2 Kreuzvalidierungsverfahren . . . . .	30
6.2.2.1 Within Subject ( <i>k-fold cross validation</i> ) . . . . .	30
6.2.2.2 Leave One Subject Out (LOSO) . . . . .	31
6.2.3 Relevante Features . . . . .	31
<b>7 Ausblick</b>	<b>37</b>
<b>Literatur</b>	<b>39</b>





# 1. Einführung

Heutige Methoden, um respiratorische Ereignisse klassifizieren zu können, sind aufwendig und kostspielig. Zum Beispiel sind Schlafstörungen wie Schlafapnoe zu 85% undiagnostiziert.

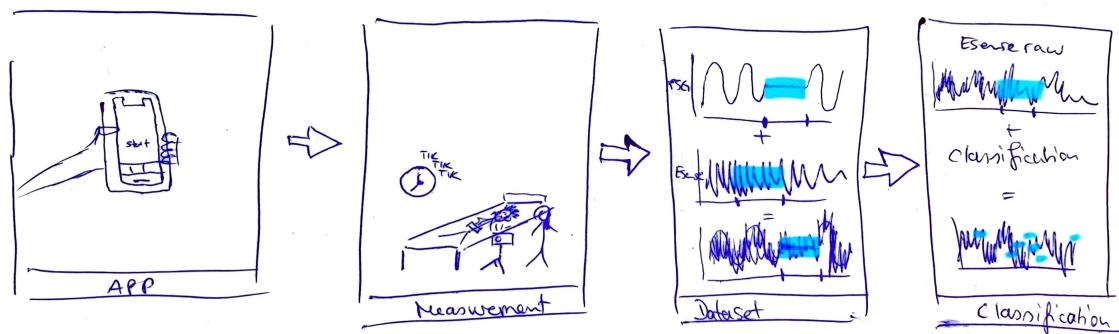


Abbildung 1.1: Ablauf der Bachelorarbeit



## 2. Grundlagen & aktuelle Forschung

### 2.1 Schlafmedizin

Unter dem Begriff *Schlafmedizin* versteht man die Lehre von Diagnostik, Klassifikation und Behandlung von Störungen während des Schlafs [3]. Trotz Erwähnungen in der Antike findet Schlafmedizin erst seit des letzten Jahrhunderts Bedeutung. Mithilfe der Polysomnographie konnten unterschiedliche Schlafphasen zyklischen Ablaufs erkannt werden. Zudem konnten den Schlafphasen physiologische Eigenschaften nachgewiesen werden. Heutzutage sind ca. 80 Schlafstörungen in dieversen Bereichen bekannt, welche neben psychologischen Testverfahren überwiegend elektrophysiologisch untersucht und behandelt werden. Patienten werden mit ambulanten Hilfsmitteln oder stationär in einem Schlaflabor untersucht und anschließend von einem technisch ausgebildeten Personal analysiert.

Hierbei wird eine Polysomnographie durchgeführt, womit genauere Messergebnisse im Vergleich zu einem ambulanten Hilfsmittel (z.B eine Langzeitbewegungsmessung), erreicht werden können. Während einer Polysomnographie werden Gehirnströme, Augenbewegungen und Muskelspannungen erkannt, wodurch die einzelnen Schlafphasen unterschieden werden können. Die verschiedene Sensorwerte werden dann in Form eines Hypnogramms mittels der Kriterien des AASM (engl. *American Association for Sleep Medicine*) ausgewertet (siehe 2.1). Hierbei werden die Schlafstadien in *Wachphase (w)*, *Einschlafen (N1)*, *leichten Schlaf (N2)*, *Tiefschlaf (N3)* und *Rapid-Eye-Movement-Schlaf (REM)* unterteilt.

Im Schlaflabor werden zudem auch Untersuchungen der Müdigkeit, der Tagesschläfrigkeit und der Aufmerksamkeit vorgenommen [3].

Atmungsstörungen können oft die Ursache von Schlaganfällen, Herzinfarkten oder den eben genannten Symptomen sein, welche zu erheblichen psychischen Störungen führen können. Aus diesem Grund ist es wichtig, so genau wie möglich Schlafstörungen bestimmen zu können, um Folgeerkrankungen zu verhindern.

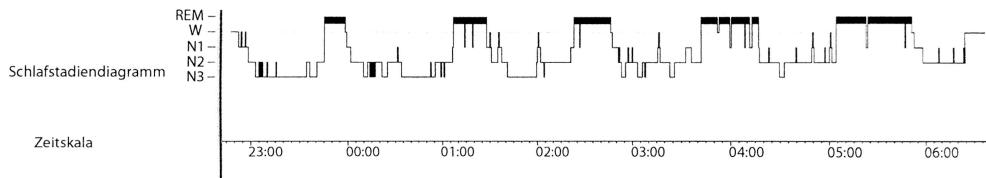


Abbildung 2.1: Beispielaufzeichnung, dargestellt in einem Hypnogramm [13]

## 2.2 Klassifizierung von Schlafstörungen

Zur Charakterisierung von Schlafstörungen werden viele Biosignale während des Schlafs registriert, welche entscheidene Merkmale liefern [13]. Aufgrund dieser Annahme wurden Klassifikatoren für Schlafstörungen entwickelt.

- Ein- und Durchschlafstörungen (Insomnien)
- Schlafbezogene Atmungsstörungen
- Hypersomnien zentralnervösen Ursprungs
- Zirkadiane Rhythmus schlafwachstörungen
- Störungen in Verbindung mit Schlaf, Schlafstadien oder partiell Erwachen (Parasomnien)
- Schlafbezogene Bewegungsstörungen
- Andere Schlafstörungen

Diese Gliederung orientiert sich an der *ICSD-3* [13].

Diese Bachelorarbeit fokussiert sich darauf, ein zentrales Schlafapnoe zu klassifizieren, was Bestandteil der „schlafbezogenen Atmungsstörungen“ ist. Schlafapnoe wird in obstruktives und zentrales Apnoe unterschieden. 20% aller Erwachsenen haben 5 oder mehr obstruktive Ereignisse pro Schlafstunde [3]. Zentrales Apnoe hingegen tritt seltener auf, als obstruktives Apnoe, jedoch auffällig oft bei besonderen Patientengruppen. Ein Beispiel liefern Patienten mit Herzinsuffizienz und einer eingeschränkten kardialen Pumpfunktion. 72% dieser Patientengruppe leidet unter zentralem Schlafapnoe. Dies stellt die Bedeutung der Klassifikation deutlich klar, da eine genaue Erkennung eines zentralen Apnoes hier sehr wichtig ist.

### 2.2.1 Zentrales Schlafapnoe

Beim zentralen Apnoe steht der Luftfluss trotz offener Atemwege für mindestens 10s still. Dies kann vollständig (zentrales Apnoe), oder partiell (zentrales Hypopnoe) erfolgen. Ab einer Anzahl von 5 Apnoeereignissen wird zentrales Apnoe diagnostiziert. Die Ursachen gelten hierbei internistischer oder neurologischer Grundlage. Typische an- und abschwellige Muster sind durch eine chronische Herzinsuffizienz, durch eine verlängerte Kreislaufzeit, oder durch eine zentralnervöse Verstellung der sogenannten Apnoeschwelle zurückzuführen. Der  $CO_2$  Gehalt im Blut wird als Apnoeschwelle

bezeichnet. Am Tag sind die Symptome von zentralem Apnoe eher an deren Ursache, den internistischen und neurologischen Grunderkrankungen zu erkennen, da diese häufig nicht von den Symptomen der Schlafapnoe zu unterscheiden sind. In der Nacht wird zentrales Apnoe, ebenso wie obstruktives Apnoe an Atemaussetzern, häufig vom Partner des Patienten beobachtet. Zudem kann lautes und unregelmäßiges Schnarchen ein Indiz für ein Apnoe sein, ebenso wie ein Aufwachen in Atemnot.

Ein Schlafapnoe kann jedoch in unterschiedlichen Schweregraden auftreten. Es gibt Patienten, welche kaum bis keine Probleme haben und nur aufgrund von nächtlichen Erkenntnissen ihrer Ehepartner zum Arzt geschickt werden. Es gibt jedoch auch Patienten, die am Tag Probleme haben, bei monotonen Situationen wach zu bleiben.

Eine Diagnose würde mögliche Folgeerkrankungen schneller erkennbar machen und die Ursache dieser erklären.

## 2.3 Maschinelle Lernverfahren

Zur Klassifikation eines zentralen Apnoes werden im Rahmen dieser Bachelorarbeit verschiedene Klassifikationsverfahren verwendet, wobei Trainingsdaten gesammelt werden, welche dann zentrale Apnoeereignisse anhand dieser Trainingsdaten klassifizieren sollen.

Maschinelle Lernverfahren sind Algorithmen, welche die Performance des Algorithmus mittels Trainingsdaten verbessern [8]. Es wird zwischen folgenden Lernverfahren unterschieden:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Beim *Supervised Learning* sind die Trainingsdaten im Vergleich zum *Unsupervised Learning* markiert. So wird beim *Supervised Learning* anhand von markierten Trainingsdaten eine Entscheidung getroffen. Anhand der folgenden Tabelle wird bei *Supervised* und *Unsupervised Learning* zwischen verschiedenen Modellen unterschieden:

Supervised Learning	Unsupervised Learning
Regression	Clustering
Klassifikation	Dimensionsreduktion

Jeder Algorithmus, welcher in einem Maschinellen Lernverfahren eingesetzt ist, besteht aus 3 Teilen: Zum einen der *Representation*, der *Evaluation* und der *Optimization*. In der *Representation* unterscheidet man nach dem zugrunde liegenden Modell, welches beispielsweise ein Entscheidungsbaum, ein neuronales Netz oder eine Support-Vektor-Maschine sein kann. Die *Evaluation* behandelt die Frage, in welche Richtung die Entscheidung getroffen werden soll. Ein Beispiel hierfür wäre die Genauigkeit, den Precision & Recall, die Entropie oder den Likelihood-Schätzer.

Beim dritten Teil, der *Optimization*, wird eine Optimierung des Algorithmus angestrebt. Dies kann unter anderem mit Methoden 2. Ordnung, Zufälliger Suche, einem absteigenden Gradienten oder der Methode der kleinsten Quadrate versucht werden.

Das in dieser Bachelorarbeit verwendete Lernverfahren ist *Supervised Learning* mit dem zugehörigen Modell, die *Klassifikation*. Zur Evaluation wurden verschiedene Klassifikationsverfahren verwendet, welche im Folgenden genauer erläutert werden.

### 2.3.1 Random Forest

Random Forest ist ein Klassifikationsverfahren des *Supervised Learning* [15]. Zur Klassifikation werden markierte Daten vorrausgesetzt, woraus eine gelernt mathematische Representation erstellt wird (*model*). Dieses *model* wird anschließend genutzt, um auf unmarkierten Daten eine Vorhersage zu treffen. Random Forest basiert auf Entscheidungsbäumen. Entscheidungsbäume sind Bäume, bei welchen pro Ast eine Entscheidung getroffen wird. Sie lernen binäre Entscheidungsregeln, die einer logischen Abfolge entsprechen. Jedoch sind einzelne Entscheidungsbäume nicht robust genug und neigen zudem zu Overfitting. Random Forest nutzt deshalb viele Entscheidungsbäume und trainiert auf den selben Trainingsdatensatz. Der Zusammenschluss der Entscheidungen jedes Entscheidungsbaums repräsentiert die endgültige Entscheidung. Bei Random Forest werden bei jeder Verästelung eine zufällige Teilmenge der Features in Betracht gezogen, was den Einfluss von stark korrelierenden Features verringern soll. Entscheidungsbäume können allgemein durch verschiedene Arten zusammengetragen werden. Zum einen ist dies mit *Bagging*, zum anderen mit *Boosting* möglich. Bei *Bagging* werden zufällige Stichproben aus der Datenmenge für jeden Entscheidungsbaum genommen [7]. Je nach Wahl geschieht dies mit oder ohne zurücklegen dieser zufälligen Teilmenge für den nächsten Entscheidungsbaum. Bei Bagging werden die Entscheidungsbäume parallel trainiert, was sich positiv auf die Performance auswirkt. Bei *Boosting* werden ebenfalls Teilmengen pro Entscheidungsbaum aus der Datenmenge gepickt, jedoch nicht zufällig. Es werden schwierige Fälle höher gewichtet, wodurch diese häufiger in einen Entscheidungsbaum mit aufgenommen werden. Somit werden die darauffolgenden Entscheidungsbäume mit zurücklegen der Teilmenge ausgewählt. Falls eine falsche Vorhersage auf einen Datenbestand getroffen wurde, erhöht sich die Gewichtung und die Wahrscheinlichkeit ist nun höher, dass dieser Datenbestand im nächsten Entscheidungsbaum enthalten ist. Die Entscheidungsbäume werden bei Boosting sequenziell trainiert.

Random Forest verwendet *Bagging*. Die endgültige Entscheidung ist entweder der Durchschnitt der Vorhersagen aller Trainingsläufe wird berechnet, oder es findet ein Mehrheitsvotum statt. Bei Random Forest können zusätzlich einige Hyperparameter modifiziert werden, wie unter anderem die Anzahl zu kombinierenden Entscheidungsbäumen, die maximale Baumtiefe oder die maximalen Features pro Verästelung. Die Vorteile von Random Forest sind die Schnelligkeit, die Flexibilität und die Parallelisierbarkeit. Zudem gibt das Resultat die Wahrscheinlichkeit an und nicht nur die endgültige Entscheidung. Die Nachteile sind die schlechte Performance bei seltenen Klassen und die Anfälligkeit für Overfitting. Des Weiteren ist Random Forest ein Blackbox Modell, wodurch es kaum möglich ist, zu ermitteln, wie die Entscheidung getroffen wurde.

### 2.3.2 XGBoost

Bevor XGboost erklärbar ist, muss zuvor *Gradient Boosting* eingeführt werden. Gradient Boosting ist wie Random Forest ein Supervised Learning und kann für die Klassifikation oder Regression verwendet werden [4]. Zudem ist Gradient Boosting wie Random Forest ein Ensemble-Learner, d.h. sie setzen sich aus vielen einzelnen Modellen (den Entscheidungsbäumen) zusammen. Im Gegensatz zu Random Forest verwendet Gradient Boosting nicht Bagging, sondern *Boosting*. Wie schon bei Random Forest eingeführt, trainiert *Boosting* auf zufälligen gewichteten Stichproben aus einer Datenmenge mit Zurücklegen. Falsche Entscheidungen erhöhen die Gewichtung, richtige Entscheidungen verringern die Gewichtung jeder Vorhersage. Somit werden schwierige Fälle besonders beachtet, da die Anpassung der Gewichtungen nach jeder Vorhersage stattfindet, also bevor die neue Teilmenge anhand der Gewichtung bestimmt wird. Das Trainieren auf Teilmengen wird auch als Stochastisches Gradient Boosting bezeichnet und verringert die Wahrscheinlichkeit auf Overfitting. Zudem wird eine bessere Generalisierbarkeit erreicht. Zu Beginn wird ein Basismodell, bestehend aus den Entscheidungsbäumen gewählt, worauf anschließend Vorhersagen anhand der Trainingsdaten getroffen werden. Daraufhin werden die Gewichte angepasst, je nach der getroffenen Vorhersage. Aus dem Basismodell mit den nun angepassten Gewichten wird eine Teilmenge gewählt, womit erneut Entscheidungsbäume entstehen und Vorhersagen getroffen werden können. Gradient Boosting versucht nun, mittels Gradienten den Fehler der verschiedenen Ensembles zu minimieren. Auch hier gibt es Hyperparameter, welche angepasst werden können. Diese sind sehr ähnlich zu Random Forest, werden jedoch um diverse Parameter erweitert, wie zum Beispiel der Funktion, welche den Fehler berechnet, die Anzahl der Iterationen pro Baum oder auch die Anzahl der Instanzen pro Blatt. XGboost ist nun eine spezielle Version des Gradient Boosting, und zwar das *Extreme Gradient Boosting* [1]. Der Unterschied zu Gradient Boosting ist hierbei, dass bei der Fehlerfunktion (Loss-Funktion) die Gradienten 2. Ordnung gewählt werden, wodurch mehr Informationen für die Verringerung des Fehlers bereit stehen. Zudem können dadurch die besten Entscheidungsbäume approximiert werden. XGBoost verwendet außerdem L1 & L2 Regularisierung, dadurch kann besser generalisiert werden und Overfitting vermieden werden. XGBoost ist parallelisierbar, schnell und kann zudem für verteiltes Trainieren auf Clustern betrieben werden.

### 2.3.3 Support Vektor Maschine (SVM)

Eine Support Vektor Maschine ist ein Klassifikationsverfahren für Probleme, welche in zwei Klassen gruppiert werden [2]. Bei der linearen Support Vektor Maschine wird nun eine line „Trennlinie“ gesucht, welche die Menge der Daten in 2 Klassen gruppiert. Diese Trennlinie wird optimiert. Es gibt auch eine nichtlineare SVM, die eine nichtlineare Trennlinie sucht. Hierbei kann es jedoch zu Overfitting kommen, da eine nichtlineare Trennlinie durch potenzielle Outlier sehr komplex werden kann und zu „gut“ optimiert wird. Jeder Wert im Datensatz ist ein Support Vektor und somit ein repräsentativer Punkt, welcher bei der Klassifikation gruppiert werden soll. Die Trennlinie muss also zwischen allen gelabelten Daten unterscheiden können. Jedoch ist die Komplexität hierbei sehr hoch, da eine Trennlinie, bzw. eine Trennebene im Hochdimensionalen Raum nur mit sehr hohem Aufwand gefunden werden kann. Die Suche nach der Trennlinie kann durch mehrere Kernel verschieden bestimmt werden. Die beliebtesten Kernel Typen sind:

Kernel Name	Kernel Funktion
Linear Kernel	$K(x) = x \cdot y$
Polynomial Kernel	$K(x) = (x \cdot y + 1)^d$
Radial Basis Function (RBF) Kernel	$K(x) = e^{-\gamma(\ x-y\ )^2}$

Die korrekte Wahl des richtigen Kernels ist entscheidend für deren Performance und Genauigkeit, jedoch nicht so leicht zu wählen. Die Vorteile von SVM sind die Effektivität im Hochdimensionalen Raum, des weiteren ist SVM Speichereffizient und zudem können durch die verschiedenen Kerneltypen mehrere Datensätze variabler verarbeitet werden. Zu den Nachteilen zählt, dass Performance schlecht ist, wenn die Anzahl der Features größer als die Anzahl der Samples ist. Zudem neigt SVM gerade im nichtlinearen Bereich schnell zu Overfitting.

## 2.4 Forschung von Klassifikation anhand von IMU-Daten

Es gibt bereits Ansätze, welche sich damit befassen, Alternativen zur Ermittlung von Schlafstörungen zu finden. Somit könnte ein Besuch im Schlaflabor durch einen bequemen Test ersetzt werden. Ein vielversprechender Ansatz ist es, mittels IMU-Daten die Bewegung des Körpers zu messen und anhand dieser Informationen die Atmung oder ähnliches herauszufiltern, um dann Rückschlüsse auf Schlafstörungen schließen zu können.

### 2.4.1 Accelerometer am Brustkorb

Bereits 2018 gab es Forschung in diesem Bereich. *Phan Duy Hung* hat sich damit befasst, einen Accelerometer an den Brustkorb zu fixieren [6]. Es wurde versucht, durch die Platzierung an genau dieser Stelle das Herzschlagen genauer erkennen zu können. Die Rohdaten zeigen hierbei bereits die Herzfrequenz und die Atmung des Probanden (siehe Abb. 2.2).

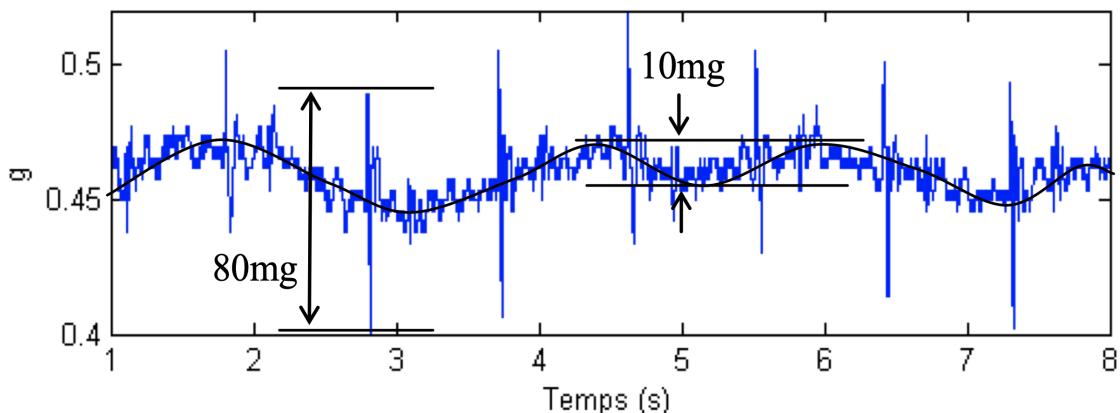


Abbildung 2.2: Rohdaten des Accelerometers in [6]

Die Daten wurden nach der Aufzeichnung durch einen Bandpassfilter mit adaptiver Apassung optimiert, um die SNR (*signal-noise-ratio*) zu verringern. Der Puls wurde ermittelt, indem Peaks (Max: *V*-Peak, Min: *R*-Peak) gefunden und als Herzschlag

interpretiert wurden. Des weiteren musste Signalrauschen, z.B von der Reibung des T-Shirts und Haut, aus dem Signal herausgefiltert werden. Zudem wurden 3 Features anhand der Amplitude über die Zeit berechnet. Das erste Feature ist die *Spektralverhältnis (spectral ratio)*. Hier wird ein Leistungsspektrum einminütigem Segments vom HR-Signal geschätzt. Der Frequenzbereich zwischen [0-1] Hz wurde untersucht. Dieser Bereich spiegelt die Frequenzvariation zwischen zentraler Schlafapnoe und normaler Aktivität wieder.

Im Bereich von [0,3-0,6] Hz zeigt sich das Maxima der spektralen Leistung, was immer in der Größenordnung abnimmt. Dieser Wert wird normiert, indem er durch den Mittelwert der spektralen Leistung im Bereich [0,1-0,3] Hz geteilt wird. Das Verhältnis der max(P) im Bereich von [0,3-0,6] Hz zum Mittelwert(P) in [0,1-0,3] Hz ist unabhängig von der Person.

Das zweite Feature sind die *Wavelet-Koeffizienten (wavelet coefficients)*, bei welcher die Analyse mit mehreren Auflösungen und guter Lokalisierungsfähigkeit im Zeit-Frequenz-Bereich häufig eine Wavelet-Transformation verwendet wird. Hierbei wurde das Atemsignal in 5 Ebenen mit der db2-Wavelet zerlegt. Danach wurden die Standardabweichungen der Detailkoeffizienten in den Ebenen 4 und 5 verwendet.

Das dritte Feature sind die *linear prediction coefficients*. Die zweite Ordnung der linearen Vorhersage

$$x(n) = a_1x(n - 1) + a_2x(n - 2) + e(n)$$

wurde verwendet, wobei x eine Zeitreihe ist, e(n) der Vorhersagefehler und a1, a2 die Vorhersagekoeffizienten, die durch die Least-Square-Optimierungsverfahren zu bestimmen sind.

Die folgenden Werte wurde zum Testen mit ANOVA ausgewählt:

ANOVA:  $a_1$ ,  $a_2$  und  $\sqrt{a_1^2 + a_2^2}$

Zudem wurden nichtlineare Features in Betracht gezogen, da bereits vorher festgestellt wurde, dass in komplexen Atemuntersuchungen lineare Funktionen nicht ausreichen. Die Herzfrequenz wird hierbei als Timeseries betrachtet. Es wurden diesbezüglich die nichtfunktionalen Features der *Poincaré-Plots (Poincare plot geometry)*, *Trendbereinigende Fluktuationsanalyse (Detrended Fluctuation Analysis)*, *Approximate-Entropie (Approximate Entropy)*, sowie der *Ljapunow-Exponent (Largest Lyapunov exponent)* verarbeitet.

Die Features wurden anschließend mit der *ANOVA-Toolbox* ausgewertet. Somit konnte ermittelt werden, ob in dem zeitlichen Intervall ein Apnoe stattgefunden hat, oder nicht.

Durch dieses Paper wurde eine Genauigkeit von 84.2% erreicht, ein zentrales Apnoe zu erkennen und mit 84.1% konnte ermittelt werden, dass in diesem Zeitrahmen kein zentrales Apnoe vorkam.

## 2.4.2 Detektion direkt am Kopf mittels der Google Glass Brille

2015 wurde erforscht, durch Informationen des Google Glass den Puls und das Atemsignal zu ermitteln [5]. Der Vorteil hierbei ist die Position der Brille. Da sie Am Kopf

platziert ist, liefert die Brille möglicherweise vielversprechende Werte im Vergleich zu IMU-Daten, die am Brustkorb aufgezeichnet worden sind. Die Google Glass wurde allerdings nicht entwickelt, um physiologische Daten zu sammeln, kann jedoch dafür verwendet werden, da alle nötigen Sensoren (Accelerometer, Gyroscope und Kamera) verbaut hat. Die Resultate liefern einen mittleren absoluten Fehler (*MAE*) von 0.82 Schlägen pro Minute (STD: 1.98) der Herzrate und 0.6 Atmungen pro Minute (STD: 1.19) bei der Atmung unter Betrachtung verschiedener Beobachtungsfenster und Kombinationen der Sensoren. Die Herausforderung lag zudem daran, stromspa-

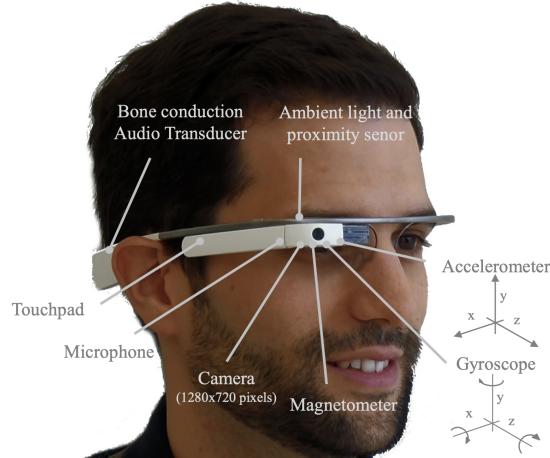


Abbildung 2.3: Google Glass Sensordaten [5]

rende Echtzeitberechnungen mit Algorithmen zu entwickeln, womit physiologische Parameter extrahiert werden. Der Nutzer soll das Gerät im Alltag normal weiternutzen können.

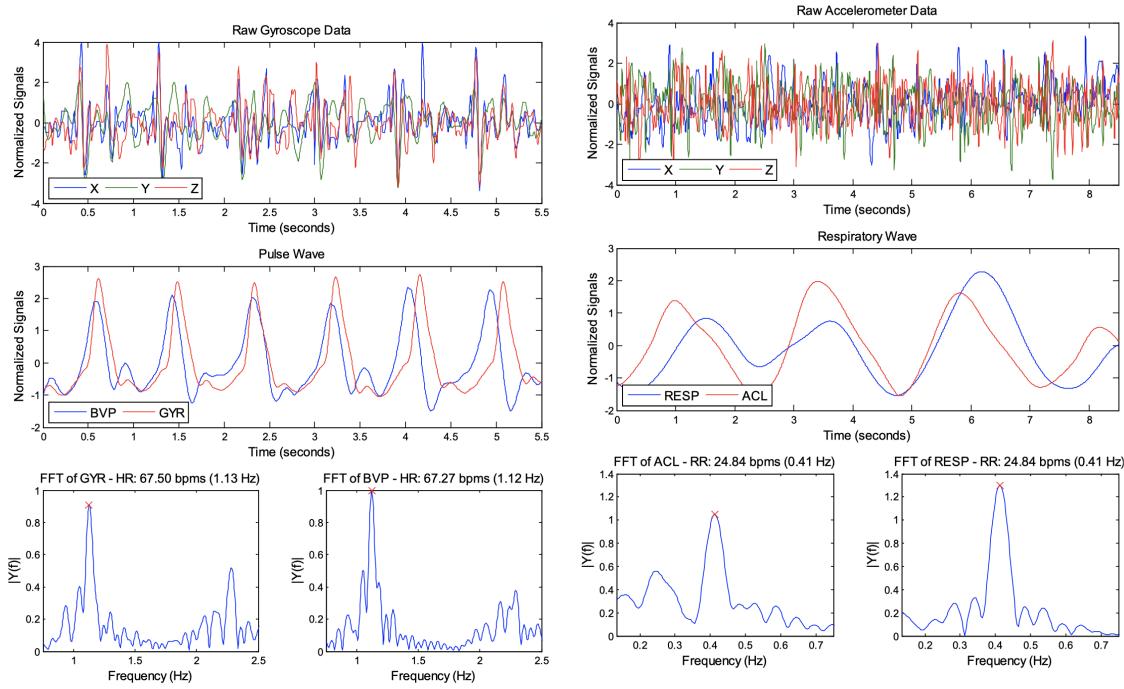
Es wurden 2 Techniken entwickelt, eine zur Ermittlung des Pulssignals und die andere für das Atemsignal.

#### 2.4.2.1 Pulssignal

Die Schätzung des Pulssignals wurde anhand einer Timeseries von Vektoren in mehrere Schritte aufgeteilt:

- Von jeder Dimension des Vektors wurde ein gleitendes Durchschnittsfenster von 3 Abtastwerten subtrahiert, wodurch Signalverschiebungen und Trends entfernt werden konnten.
- Ein Bandpass Butterworthfilter der 4. Ordnung mit Cut-Off Frequenzen von 10Hz und 13Hz wurde auf jede Dimension angewendet, um Veränderungen des BCG zu isolieren
- Zudem wurde ein Bandpass Butterworthfilter der 2. Ordnung mit den Cut-Off Frequenzen von 0.75Hz und 2.5Hz (entspricht 45 und 150 Schläge die Minute) angewandt, was schließlich das resultierende Pulssignal liefert.

Die Abbildung 2.4a zeigt ein Signal des Herzschlags, gesammelt von den Informationen der Gyroskopdaten. Diese Daten wurden von dem Google Glass aufgezeichnet,



(a) Beispiel eines Pulssignals mittels der Gyroskopdaten (rot) und des Ground-Truth Signals (blau). Die beiden unteren Graphen zeigen das Fouriespektrum von jedem Signal (FFT: Fourier Spectrum, GYR: Gyroscope, BVP: Blood Volume Pulse, HR: Heart Rate, bpm: beats per minute)

(b) Beispiel einer Schätzung des Atemsignals anhand der Beschleunigungsdaten (blau) und des Ground-Truthtsignals (rot). Die beiden unteren Graphen zeigen das Fouriespektrum von jedem Signal. (FFT: Fourier Spectrum, ACL: accelerometer, RESP: Respiration from chest band, RR: respiration rate, bpm: breaths per minute)

Abbildung 2.4: Herzrate- und Atemfrequenzanalyse

während der Person auf dem Rück lag. Der obige Graph zeigt ein 3-Achsen Gyroskop signal über eine Dauer von 5.5 Sekunden. Der mittlere Graph zeigt die Schätzung des Herzschlags, nachdem die vorgestellten Methoden angewandt wurden in rot, die des Referenzsignals in blau. Es ist sehr gut zu erkennen, dass die Schätzung sehr nahe an den Referenzsignalen ist.

#### 2.4.2.2 Atemsignal

Das Atemsignal wurde durch verschiedene Schritte berechnet:

- Ein gleitender Mittelwertfilter wurde auf jede Komponente angewandt. Die Fensterlänge wurde auf die Dauer eines Atemzyklus gesetzt, in diesem Fall 45 Atmungen pro Minute.
- Ein Bandpass Butterworthfilter der 4. Ordnung mit dem Cut-Off Frequenzen von 0.13Hz und 0.75Hz (entspricht 8-45 Atmungen pro Minute) wurde auf jede Dimension angewandt.
- Da die verschiedenen Dimensionen der Sensoren nicht in Relation zu den Körperpositionen stehen, wurde eine Principal Component Analyse angewandt, um einen derartigen Einfluss zu reduzieren. Nach einer Fast Fouriertransformation (FFT) auf jede Komponente wurde das Signal mit der Periode mit

der maximalen Größenordnung ausgewählt, welche innerhalb des betrachteten Frequenzbereichs liegt.

Die Abbildung 2.4b zeigt ein Beispiel einer Atemfrequenzschätzung der Beschleunigungsdaten eines Patienten. Wie zu sehen ist, sind die Daten sehr nahe an dem Referenzwert, welcher mit aufgezeichnet wurde.

### 2.4.3 Überwachung von Puls und Atmung mittels eSense-Earpods

2019 wurde von *Tobias Röddiger, Daniel Wolffram* und *David Laubenstein* nachgewiesen, dass es möglich ist, mittels den eSense-Earpods die Atmung und den Puls näherungsweise zu ermitteln [11]. Dies gelang etwas genauer, als das Monitoring von *J. Hernandez* mit dem Google Glass [5]. Es wurden hierbei eine Studie mit 12 Personen aufgezeichnet, welche in 3 Positionen (liegend, stehend, sitzend) jeweils vor, bzw. nach einer sportlichen Bewegungsphase einen einminütigen Atemablauf durchgeführt haben. Die Analyse der Daten erfolgte im Anschluss der Studie und wurde in einer Pipeline verarbeitet, welche zuerst das Rauschen reduziert, anschließend einen Triangle-Filter der Breite 2s anwendet und danach eine PCA (engl. *principal component analysis*) ausführt, um die Daten unabhängig von deren Achse zu bewerten. Nun wurden Windows mit der Größe von 20s extrahiert, welche die die Atmung und den Puls anhand dieses Windows berechnen. Die Resultate ergaben einen mittleren absoluten Fehler (engl. *mean absolute error*) von 2.62 CPM (acc) und 2.55 CPM (gyro), jedoch variieren diese von Proband zu Proband.

**TODO:** extend with more content from this paper

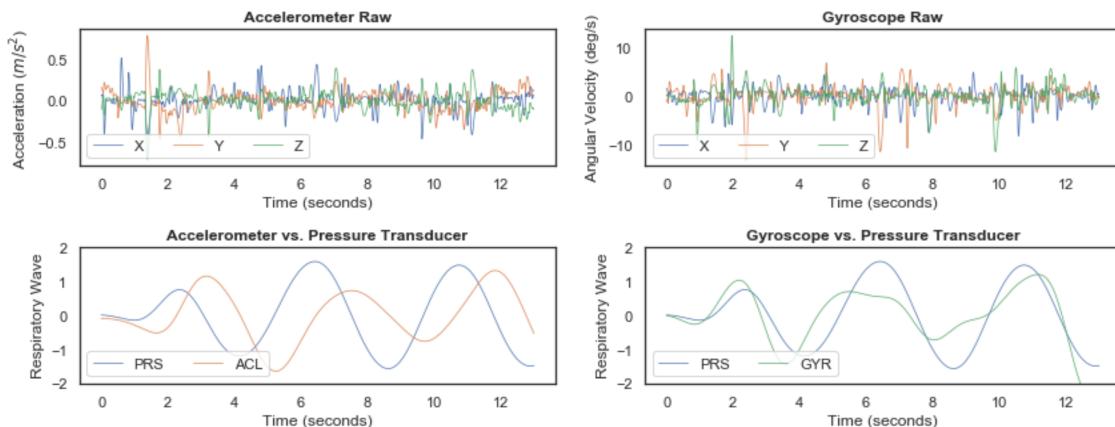


Abbildung 2.5: Rohdaten der Accelerometer und Gyroscope daten innerhalb von 12s, sowie die Atem- und Pulsschätzung, verglichen mit dem Ground-Truh (blau)

# **3. Schlafanalyse**

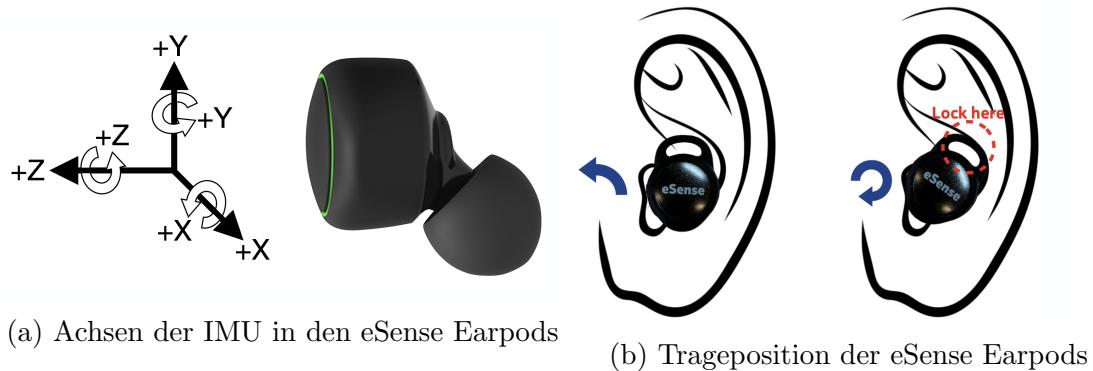
Zur Analyse eines respiratorischen Ereignisses findet im Zuge dieser Bachelorarbeit eine Nutzerstudie statt, um einen Datensatz zu erstellen, welcher daraufhin analysiert werden kann. Im Folgenden wird erläutert, welche Geräte verwendet werden und wie diese gesammelten Informationen zusammengetragen werden, damit der entstehende Datensatz in einem strukturierten Zustand vorliegt.

## **3.1 Earable Plattform**

Zur Erfassung der Daten werden eSense-Earpods der Firma “Nokia Bell Labs Cambridge” verwendet. Die Earpods beinhalten zwei Mikrofone und ein Lautsprecher, welche beide über Bluetooth angebunden werden können. Pro Earpod ist ein Mikrofon verbaut. Des weiteren ist das für diese Bachelorarbeit interessanteste Element, eine 6-Achsen IMU (Inertial Motion Unit) teil der Earpods. Eine IMU ist eine inertiale Messeinheit, womit Gyroskop- und Beschleunigungsdaten aufgezeichnet und mittels BLE (Bluetooth Low Energy) auf das Smartphone übertragen werden können. Es handelt sich um einen 3-Achsen Beschleunigungssensor, sowie einen 3-Achsen Gyroskop. Die IMU ist lediglich im linken Earpod verbaut. Die Messrate dieser Sensoren ist variabel einstellbar und wurde im folgenden auf 50Hz festgelegt. Die IMU ist ebenfalls konfigurierbar und wurde nicht verändert. Diese Messbereich ist beim Beschleunigungssensor  $\pm 4g$  und beim Gyroskop  $\pm 500deg/s$ . Die Messdaten der IMU sind bereits gefiltert, wenn sie per BLE übertragen werden. Hier ist die Konfiguration ebenfalls schon vorkonfiguriert, bei der ein Tiefpassfilter mit einer Bandbreite von 5Hz des Beschleunigungssensors, sowie ebenfalls 5Hz des Gyroskops angewandt wird.

### **3.1.1 Was wird aufgezeichnet?**

Zu vollständigen Aufzeichnung eines Datensatzes werden die Daten der IMU, in einer Datenbank abgespeichert. Insgesamt werden hierbei pro empfangene Dateneinheit 6 Werte persistiert, die  $x$ ,  $y$  und  $z$  Richtung des Beschleunigungssensors, bzw des Gyroskops. Des weiteren wird die aktuelle Zeit, die aktuell auszuführende Aktion des Studienablaufs und die Information, ob die LED des Smartphones an oder aus



ist, zu jeder empfangenen Dateneinheit hinzugefügt. Das Mikrofon wird ebenfalls aufgezeichnet und nach der Messung abgespeichert. Vor dem Beginn einer Messung wird der Studienteilnehmer gebeten, ein paar Zusatzinformationen (siehe Kapitel 3.5) anzugeben. Diese werden vor dem Start der Messung am Smartphone ausgefüllt und ebenfalls in der Datenbank gespeichert.

### 3.1.2 Datenexport

Zur weiteren Verarbeitung werden die Daten, nachdem sie von der App lokal in einer Datenbank gespeichert werden, exportiert. Zuerst werden die Datenbankeinträge der aktuellen Messung als *csv*-Datei exportiert und in einem temporären Ordner abgespeichert. Hierbei werden die Gyroskop einträge separat von den Beschleunigungsdaten exportiert, es entstehen folglich 2 *csv*-Dateien („*GyroData\_ID.csv*“, „*ACCDATA\_ID.csv*“). Das Mikrofon-Signal wird nach der Messung als *m4a*-Datei ebenfalls im temporären Ordner abgelegt. Die Zusatzinformationen, welche über den Studienteilnehmer hinterlegt wurden, werden als *csv*-Datei („*UserStudyPerson-Details\_ID.csv*“) ebenfalls in den temporären Ordner persistiert. Alle Dateien des temporären Ordners werden in einer *zip*-Datei verpackt und können über den Share-Screen von Apple über verschiedene Wege geteilt werden.

## 3.2 Polysomnographie-Systeme

Als Referenz zu den eSense-Earpods wird ein Polysomnographie-System (PSG-System) verwendet. Das Gerät SOMNOscreen™ plus bietet alle nötigen Sensoren und findet in der Wissenschaft Anerkennung. Ein solches System zeichnet Messungen für physiologische Funktionen des Körpers während des Schlafs auf und kann somit mögliche Schlafstörungen diagnostizieren. Im Gegensatz zur Polygraphie kann die Polysomnographie explizit zwischen obstruktivem und zentralem Apnoe unterscheiden [3]. Es werden kontinuierlich verschiedene Körperfunktionen überwacht, wodurch nach einer Messung ein umfangreiches und individuelles Schlafprofil erstellt werden kann.

Das PSG-System zeichnet während der Studie ebenfalls Daten auf und soll die Resultate, welche durch die eSense-Earpods gesammelt und analysiert werden, verifizieren. Somit dienen die Daten, welche durch das PSG-System gesammelt werden, als „Ground-Truth“.

<sup>1</sup>siehe Quelle: [12]



Abbildung 3.2: PSG-System im Einsatz und die visuelle Darstellung<sup>1</sup>

Mittels der „DOMINO Schlafdiagnostik“ Software kann eine Montage erstellt werden. Eine Montage ist eine Konfiguration, bei der man festlegen kann, welche Sensoren gemessen undpersistiert werden sollen. Zudem kann man die Abtastrate in Hz festlegen. Im folgenden werden alle Sensoren aufgelistet, welche für die Studie aufgezeichnet wurden und somit die Montage darstellen. Die nicht persistierten Daten werden im folgenden ignoriert.

- **Bauchsignal (abdomen) (32Hz)**: Gibt die Veränderung des Volumens am Bauch an.
- **Lichtsignal (light) (128Hz)**: Misst Umgebungslicht, was verwendet wird, um die Signale vom PSG-System und den eSense-Earpods zu synchronisieren.
- **Drucksignal (flow) (256Hz)**: Hier wird der Druck gemessen, der durch die Nase kommt. Der Sensor ist direkt an der Nase befestigt.
- **Bewegungssignal (movement) (32Hz)**: Misst bewegung anhand von Beschleunigungsdaten.
- **Sauerstoffsättigung (pleth) (128Hz)**: Gibt die Sauerstoffsättigung im Blut an.
- **Pulssignal (pulse) (32Hz)**: Gibt den Puls, gemessen am Finger des Patienten, an.
- **Schnarchmikrofonsignal (schnarch) (256Hz)**: Dieser Sensor wird am Kehlkopf befestigt und misst das Schnarchen.
- **Sauerstoffgehalt (spo2) (32Hz)**: Manschettenfreie kontinuierliche Blutdruckmessung
- **Brustkorbsignal (thorax) (32Hz)**: Gibt die Veränderung des Volumens am Brustkorb an.
- **Brustkorb & Bauchsignal (thoraxabdomen) (32Hz)**: Das kombinierte Signal von Brust- und Bauchsignal
- **EDF Informationen (edfAnnotations) (60Hz)**: Enthält Anmerkungen als char, z.B die Startzeit der Messung

### 3.2.1 Datenexport

Im PSG-System befindet sich eine CF-Karte (*Compact-Flash*). Diese kann mithilfe der vom PSG-System bereitgestellten „DOMINO Schlafdiagnostik“ Software ausgeleren werden. Die Software stellt eine Ansicht dar, womit man die Signale untereinander in einer zeitlich synchronisiert betrachten kann. Die aufgezeichneten Signale können als *edf*-Datei exportiert werden. Mittels Python kann man *edf*-Dateien auslesen und weiterverarbeiten. Pro Studie wurden alle 3 Positionsabläufe in einem einzigen Messvorgang aufgezeichnet. Somit müssen die 3 Einzelmessungen aus der *edf*-Datei herausgezogen werden. Für weitere Details siehe Kapitel 5.2.

## 3.3 Kamera

Während der Studie wurde zur vollständigen Erstellung eines Datensatzes eine Kamera mit Stativ aufgestellt, welche den Kopf des Studienteilnehmers fokussiert. Somit können eventuelle Unklarheiten im Datensatz, wie zum Beispiel ein Husten der Person, genau deklariert werden. Die verwendete Kamera ist eine „Canon EOS 6D Mk II“. Die Aufnahme wurde in *1080p* mit einer Abtastrate von *50fps* aufgenommen. Die Daten wurden nicht in die Klassifikation mit eingebunden und sind nur dazu gedacht, Unklarheiten aufzuräumen.

## 3.4 Datensynchronisation

Um sicherzugehen, dass das PSG-System, sowie die Daten der eSense-Earpods zeitlich exakt übereinstimmt, wurden mit der App kurze Lichtblitze gesendet (20ms). Mit einem 3D-Drucker wurde eine Vorrichtung angefertigt, welche das Smartphone auf das PSG-System platziert, sodass die LED des Smartphones direkt auf den Lichtsensor zeigt (siehe Abbildung 3.3).



Abbildung 3.3: 3D-Druck, welcher die LED des Smartphones auf den Lichtsensor zeigen lässt

Die Lichtblitze lösen nach jeder Aktionsänderung aus, die der Studienteilnehmer erhält. Den genauen Ablauf kann man der Abbildung 4.2 entnehmen. Die Lichtblitze der Messung können nun mit den Lichtblitzen der eSense-Daten synchronisiert werden (siehe Kapitel 5.2.1).

### 3.5 Zusatzinformationen der Nutzer

Vor dem Start der Datenaufzeichnung wurden Informationen über die Aufzeichnung und über den Teilnehmer gesammelt. Dies soll lediglich dazu dienen, spätere Unklarheiten im Datensatz erklären zu können. Es werden Informationen zum Körper der Person abgefragt (Alter, Größe, Gewicht, Geschlecht, Schlafrhythmus), den Ear-podaufsatzt, die Matratzenart, sowie Maße des Ohrs.



# 4. Design

Ein wichtiger Teil dieser Arbeit ist die Erstellung eines Datensatzes, welcher zur Klassifikation dienen soll. Mithilfe einer Smartphone-App soll ein Datensatz eines Studienteilnehmers erstellt und exportiert werden. Daraufhin liegen die Daten vor und können in einer Verarbeitungspipeline analysiert, bzw. klassifiziert werden.

Die Studie wurde so konzipiert, Atemaussetzer während des Schlafens zu klassifizieren. Während der Studie wurde jeder Datensatz im Bett des Teilnehmers aufgezeichnet, was das Wohlbefinden verschärken und somit auch die Qualität der Daten erhöhen soll.

Im Rahmen dieser Bachelorarbeit wurde ein Datensatz nach dem *Sample of Convenience* erstellt.

## 4.1 Studienplanung

Für die Studie wurde eine Teilnehmeranzahl von 10 Personen gewählt, welche die nötige Vielfältigkeit liefern soll. Des weiteren wurden pro Teilnehmer ein Datensatz an 3 verschiedene Positionen aufgezeichnet, auf dem Bauch, dem Rücken, sowie auf der Seite liegend.

Eine Fragestellung der Studie war, wie ein Atemaussetzer „simuliert“ werden soll. Es wurde entschieden, dass die Studie ein zentrales Schlafapnoe erkennen soll. Demzufolge soll der Studienteilnehmer in einer vordefinierten Reihenfolge einen Atemaussetzer „simulieren“, indem er die Luft für eine gewisse Zeit anhält. Um unterschiedliche Längen von Atemaussetzern aufzuzeichnen wurden 10s, 20s und 30s gewählt, in denen der Teilnehmer die Luft anhalten soll. Nun muss ein geeigneter Ablauf gewählt werden, wodurch sich die Ereignisse nicht überschneiden. Auf der Suche, wie lange die Regeneration dauere, nachdem eine Person die Luft angehalten hat, ergab sich durch das Schaubild 4.1, dass die Person ca. die gleiche Zeit zur Regeneration benötigt, wie sie die Luft zuvor angehalten hat. Diese Zeit wurde nun zusätzlich in der Studie mit eingebracht und daraus ergibt sich der Ablauf, welcher in Abbildung 4.2 zu sehen ist.

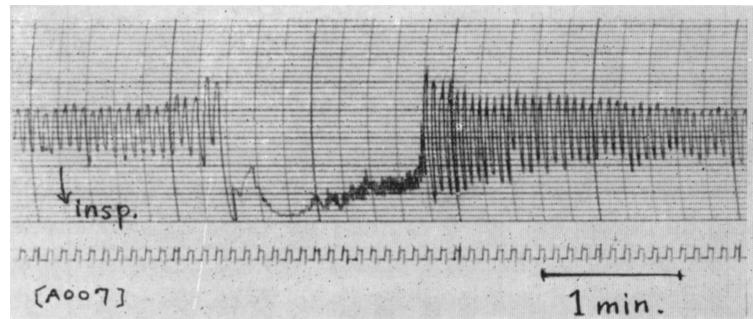


Abbildung 4.1: Regenerationsphase nach Luft anhalten [beath\_rebreathing].

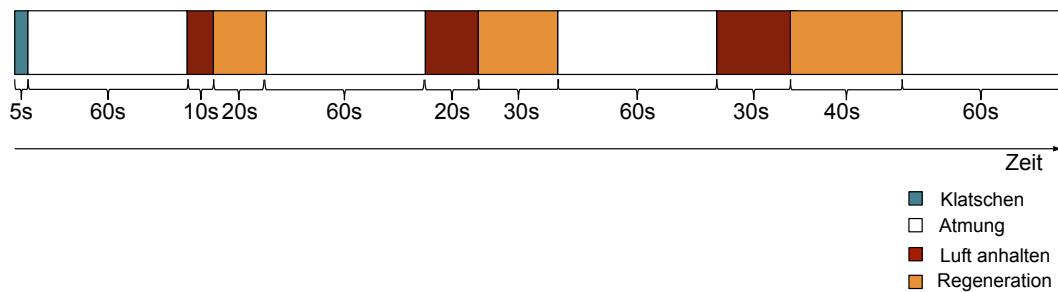


Abbildung 4.2: Ablauf der Studie mit einer Position

## 4.2 Studienablauf

Dieser Ablauf (siehe Abb. 4.2) wurde nun pro Studienteilnehmer jeweils bei den 3 Positionen durchgeführt, womit alle Schlaflagen abgedeckt wären. Zu Beginn der Studie fand eine kurze Einweisung statt, indem der Proband erfuhr, was er zu tragen hat und wie er Anweisungen erhält, um dem Ablauf folgen zu können. Die Kamera wurde auf dem Stativ platziert und so ausgelegt, dass sie das Ohr des Probanden filmt. Das PSG-System wurde am Studienteilnehmer angebracht, sowie alle nötigen Sensoren, die im Kapitel 3.2 beschrieben wurden. Nach der passenden Auswahl des Aufsatzes der eSense-Earpods war der Aufbau der Studie beendet.

Nun wird die Messung des PSG-Systems gestartet, sowie die Smartphone-App geöffnet. Nach Eingabe der Nutzerinformationen kann der erste Durchgang, welcher abhängig vom Ablauf der 3 Positionen war, begonnen werden. Durch den Start der Messung am Smartphone beginnt die Messung. Da zusätzlich das Mikrofon am eSense-Earpod mit aufgezeichnet wird, wird nach dem Start der Messung ein 4s Zeitfenster gewählt, indem der Teilnehmer in die Hände klatschen musste, um das Mikrofonsignal später synchronisieren zu können. Nun beginnt die Aufzeichnung. Der Leiter der Studie hat bereits den Raum verlassen und alle Anweisungen werden durch die Earpods per Audiosignal ausgesprochen. Sofern die Messung beendet ist, tritt der Leiter der Studie wieder in den Raum und die Messung kann exportiert werden. Der Export beinhaltet jegliche Smartphone-Daten. Die PSG-Daten werden als eine komplette Messung am Ende der Studie exportiert. Zudem wird die Kamera an gehalten und eine neue Aufnahme kann gestartet werden. Anschließend beginnt die nächste Position. Der Proband kann nun die neue Position einnehmen, anschließend wird per App die neue Messung gestartet. Zum Abschluss aller 3 Positionen wird die Messung am PSG-System gestoppt und mittels eines vom PSG-System bereit-

gestellten Programms lässt sich die Messung als „*edf*-Datei“ exportieren. Mehr zum Export der Daten und zur Synchronisation, siehe Kapitel 5.2 Pro Position dauert eine Messung ca. 7 Minuten, was bei 3 Personen eine Gesamtdauer von 21 Minuten bedeutet. Einschließlich der Instruktionszeit, dem Export nach der Messung und den Positionswechseln der Probanden war die durchschnittliche Gesamtdauer einer Aufzeichnung ca 45-60 Minuten.



# 5. Implementierung

In diesem Kapitel wird erläutert, wie die Daten des Datensatzes gesammelt, zur weiteren Verarbeitung vorbereitet und schließlich analysiert werden. Zur vollständigen Erarbeitung eines Datensatzes muss zuvor eine App erstellt werden, welche die zu sammelnden Daten der eSense Earpods empfängt und persistiert. Zudem soll die App den Ablauf einer Messung automatisch verwalten und danach einen einfachen Export der persistierten Daten ermöglichen. Anschließend müssen die persistierten Daten aufbereitet werden, sodass sie klassifiziert werden können.

## 5.1 App

### 5.1.1 Plattform

Die Smartphone-App wurde für Apple-Geräte konzipiert und mit der Sprache Swift entwickelt. Mit der Software XCode lässt sich eine mit Swift geschriebene App kompilieren und auf dem Smartphone installieren.

Zur Einbindung externer Frameworks werden die Dependency-Manager *Accio* und *Carthage* verwendet. Im Folgenden sind die relevanten Frameworks aufgelistet, welche in der App eingebunden worden sind:

Imperio	Strukturierung der App
Realm-Cocoa	Datenbank-Framework
Zip	Bietet die Möglichkeit, einen Ordner als „zip“-Datei zu komprimieren
Mongohealer	Fügt Logging zu der App hinzu

Durch das Framework *Imperio* ist es möglich, die View-Komponenten von der Logik zu trennen. Somit ändert sich die Struktur der App, indem jeder Ablauf in der App als *Flow* interpretiert wird. Pro Flow wird ein *FlowController* angelegt, welcher die Logik des Ablaufs kontrolliert. Ein Flow kann nun beliebig viele *ViewController* starten. Jede *View*, welche von einem Flow aufgerufen wird, hält ein *Delegate* Objekt. Ein *Delegate* ist ein Protocol, womit dem *Flow* eine Aktion auf der *View*

mitgeteilt werden kann. Somit wird bei jeder Aktion auf der View eine Funktion des FlowControllers aufgerufen, welcher die View gestartet hat.

Das Framework *Realm* ist eine Datenbank für mobile Systeme, die vollständig auf dem mobilen Endgerät läuft. Die Daten können direkt als *Objekt* ausgelesen und verarbeitet werden. In der App wird die Datenbank verwendet, um eine Messung abzuspeichern (siehe Abbildung 5.1))

Die App ist in 3 Sektionen aufgeteilt, einer *Chartansicht*, einer *Messungsansicht*, sowie einer *Einstellungsansicht*. (Abb. 5.3 Tabbar) Im weiteren wird nur die *Messungsansicht* genauer erläutert, da die anderen Ansichten im Rahmen der Bachelorarbeit nicht relevant sind.

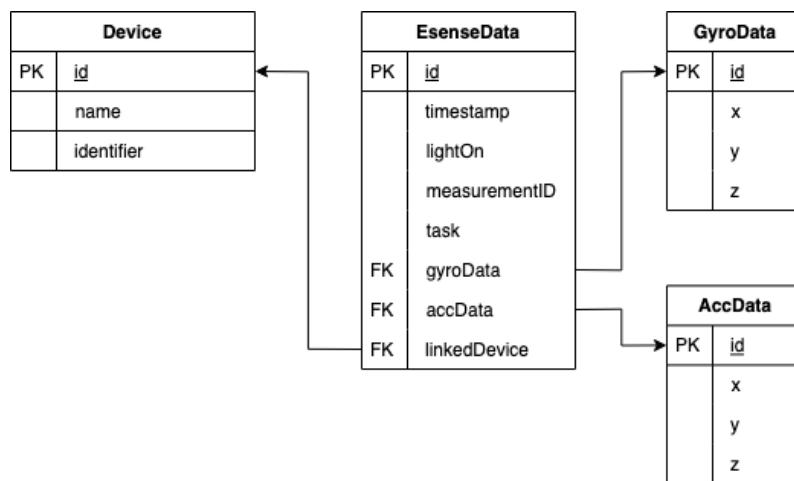


Abbildung 5.1: ER-Diagramm der App-Datenbank

### 5.1.2 Messungsablauf

Mit der Messungsansicht soll eine komplette Messung durchgeführt werden. Der *MeasurementFlow* wird gestartet und die erste *View* (Abb. 5.3a) wird geöffnet. Nach der erfolgreichen Verbindung mit den eSense-Earpods erfolgt eine Weiterleitung zur nächsten *View* (Abb. 6.2b) zum Ausfüllen der Nutzerinformationen. Mit dem Betätigen des Buttons: „Start Measurement“ bestätigt man die Eingabe der Daten und die Messung beginnt. Automatisch beginnt der erste Timer (Abb. 5.3c). Der Timer zeigt den aktuellen, sowie den nächsten *Task* an, sowie die Restzeit des aktuellen Tasks. Der genaue Ablauf der einzelnen Timer ist in Abbildung 4.2 detailliert beschrieben. Nach dem Ablauf des letzten Timers wird eine View geöffnet, welche die Möglichkeit zum Teilen der aktuellen Messung bietet (Abb. 5.3d, 5.3e). Zudem kann die Datenbank vollständig geleert werden. Während dieses Ablaufs wird ein Objekt erstellt, welches die Messung verwaltet.

### 5.1.3 Messung

Eine Messung wird im Code in einem *Measurement* Objekt persistiert. Durch ein Observer-Pattern wird der *MeasurementFlow* über jegliche Änderung informiert und kann entsprechende Handlungen durchführen. Durch die Funktionen **startMeasurement** und **stopMeasurement** kann eine Messung gestartet, bzw gestoppt werden. Mit der Funktion **startMeasurement** wird das IMU-Sampling per BLE, sowie die

Audioaufnahme gestartet. Ebenso wird der erste Timer gestartet und ein doppeltes Lichtsignal gesendet. Durch die Funktion `stopMeasurement` werden die Datenströme gestoppt, ebenfalls ein doppeltes Lichtsignal gesendet und der Timer wird beendet. Der nächste Task wird gestartet, wenn der Timer abgelaufen ist. Der Timer startet mit der Länge des nächsten Tasks. Sofern der nächste Task „Hold\_breath“ ist, also die Person im folgenden Task die Luft anhält, wird dem Teilnehmer kurz vor Ablauf mitgeteilt, wann der nächste Task startet. Mit der Instruktion „Bitte die Luft anhalten in 3, 2, 1“ weiß der Teilnehmer, wann er die Luft anhalten soll. Durch die Anweisung „Stopp“ wird dem Nutzer das Ende des Tasks mitgeteilt. Die Instruktionen liegen als Audiodatei vor und werden jeweils vor dem jeweiligen Task abgespielt und mittels Bluetooth über die Lautsprecher der eSense-Earpods ausgegeben.

## 5.2 Anbindung an Auswertungspipeline

Die App liefert beim Export die Daten der eSense-Earpods, was die IMU-Daten, sowie die Nutzerinformationen und die Mikrofonaufnahme beinhaltet. Zum aktuellen Stand liegen somit die Daten der eSense-Earpods und vom PSG-System vor. Zu Beginn müssen die PSG-Daten, welche als eine Messung für alle 3 Positionen pro Studienteilnehmer persistiert wurde, in 3 einzelne Messungen aufgeteilt werden. Die Daten des PSG-Systems liegen als *edf*-Datei vor. Diese können mittels python und der Library `edfrd` ausgelesen werden. Jedoch sind die Einträge des jeweiligen Signals ohne einen Zeitwert abgespeichert. Es muss nun aufgrund der Abtastrate in Hz die Zeit manuell berechnet werden. Mittels der Funktion `find_peaks` aus `scipy.peak` können die Peaks des Lichtsensors am PSG-System ermittelt werden (siehe Abb. ??). Da eine Messung mit 2 Lichtblitzen beginnt und endet, kann nun der Start- und Endzeitpunkt einer Position ermittelt und die einzelnen Positionen können von einander unterschieden werden. Daraufhin können die 11 verfügbaren Signale einzeln ausgelesen werden. Pro Position wird nun jedes der Signale als *csv*-Datei im jeweiligen Ordner abgelegt. Die Daten der eSense-Earpods liegen getrennt in *AccData\_\$.ID\$.csv* und *GyroData\_\$.ID\$.csv* vor. Diese werden ausgelesen und zu der Datei *eSenseData.csv* zusammengeführt. Die Ornderstruktur kann der Abb. 5.2 entnommen werden und ist nun vollständig.

### 5.2.1 Synchronisation der Daten

Da es nicht garantiert ist, dass der Timer des PSG-Systems zuverlässig arbeitet, wird jeder Peak des Lichtsensors mit dem Lichtsignal der Smartphone-Daten verglichen. Der Abstand jedes einzelnen Peaks wird nun ermittelt und die durchschnittliche Distanz der Peaks wird ermittelt. In der Abbildung 5.4b ist der Vergleich eines Lichtblitzes von Smartphone (blau) und PSG-System (orange) im zeitlichen Verlauf (*time*). Da die Verschiebung des Timers des PSG-Systems während einer 7-minütigen Messung im Mittel bei 0.3s lag, wurde der Timer des PSG-Systems

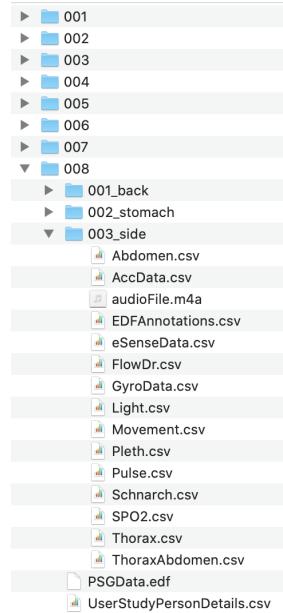


Abbildung 5.2: Ornderstruktur des Datensatzes

um die durchschnittliche Distanz der Peaks verschoben. Dies ist ausreichend, da die PSG-Daten in der Bachelorarbeit lediglich zum visuellen Vergleich dienen und in der Klassifikation nicht mit betrachtet werden. Die zeitliche Verschiebung wird nun in jede *csv*-Datei eingearbeitet mit der Zeile `new_time`. In Abbildung 5.4c ist klar zu erkennen, dass der neue Zeitwert (`new_time`) des PSG-Signals (orange) direkt über dem Zeitwert des Smartphonesignals (blau) liegt, was an den Lichtpeaks zu erkennen ist. Nun ist garantiert, dass die Lichtsignale von PSG-System und dem Smartphone annähernd synchron sind. Die Daten sind nun bereit zur Analyse.

### 5.3 Verarbeitungspipeline zur Klassifikation

Zur Klassifizierung der Daten müssen nun Features berechnet werden. Diese werden jedoch nicht auf das ganze Zeitintervall einer Messung berechnet, sondern aufgeteilt. Eine Messung wird nun in Fenster der Größe von 5, bzw 10 Sekunden aufgeteilt, wobei jedes Fenster um eine Sekunde verschoben wird. Somit überlappen sich zwei aneinander liegende Fenster immer um 4, bzw 9 Sekunden. Die Trennung der Fenster wurde anhand des Zeitwertes jedes Messergebnisses entschieden, da die Messergebnisse der eSense Earpods beim Senden per BLE an das Smartphone keine Uhrzeit beinhalteten. Dadurch liegt kein genauer Abstand der einzelnen Messdaten von 50Hz vor. Anschließend wurden mittels des Moduls `tsfresh` Features für jedes Fenster einzeln berechnet und es entstand eine Datei `df$patient_id$$_position_id$$_feature.csv`, welche alle Features der einzelnen Fenster beinhaltet.

Da nun die Features für eine Fenstergröße von 5s und 10s mit der Verschiebung von 1s pro Fenster vorliegen, kann mit der Klassifikation begonnen werden.

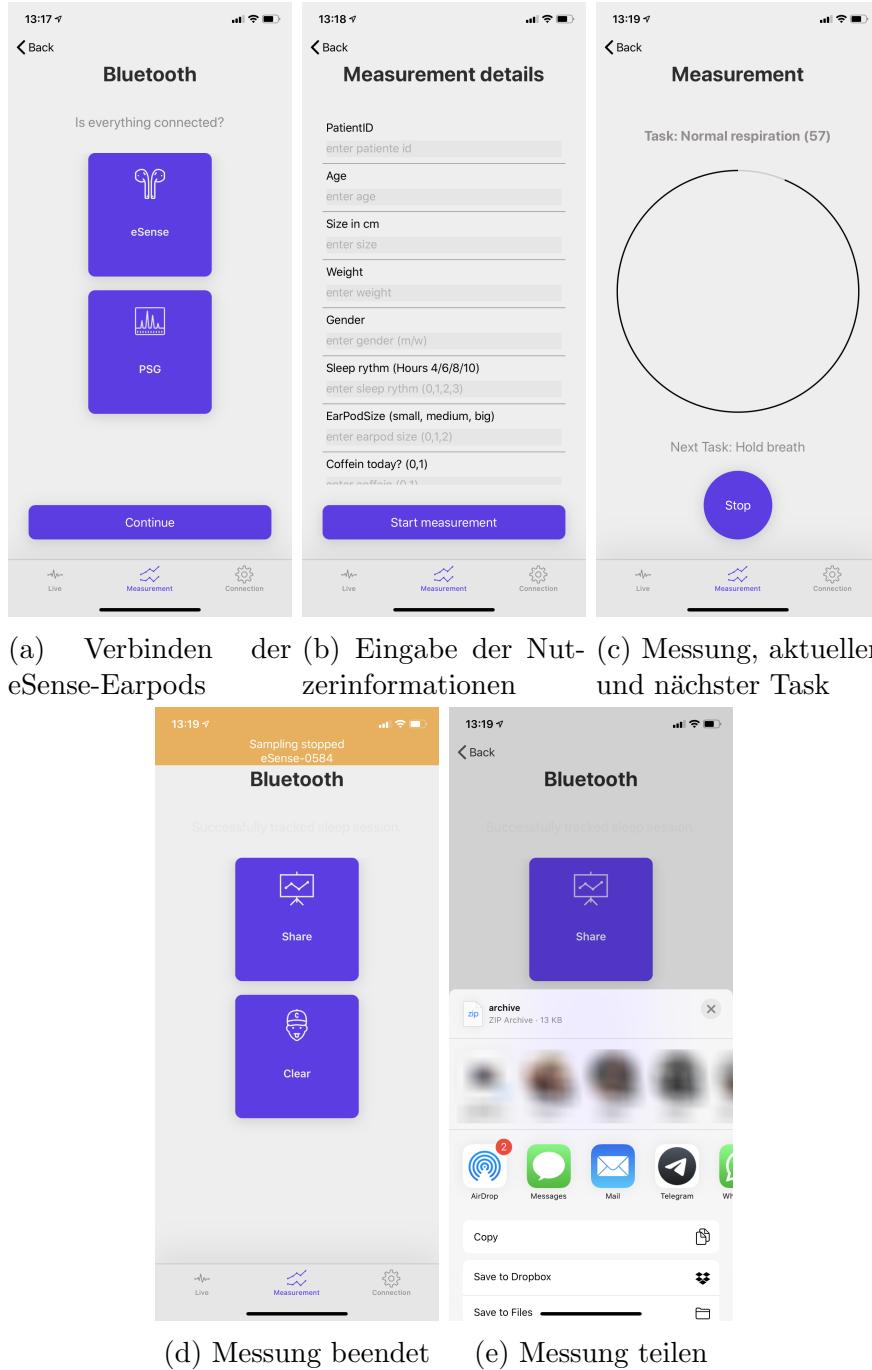
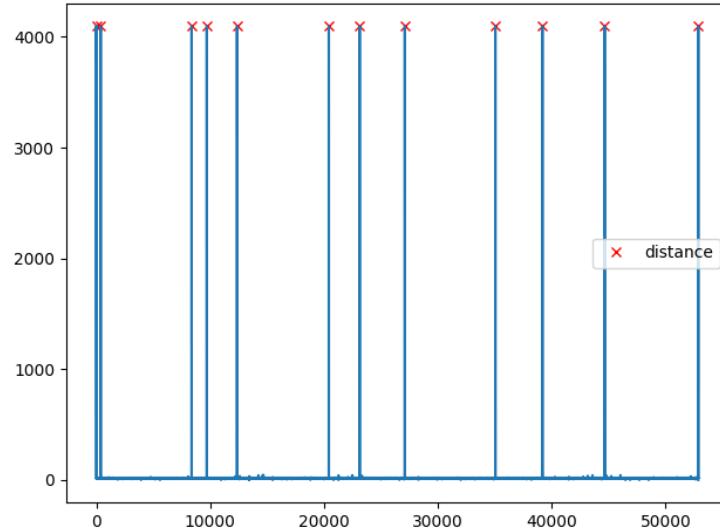
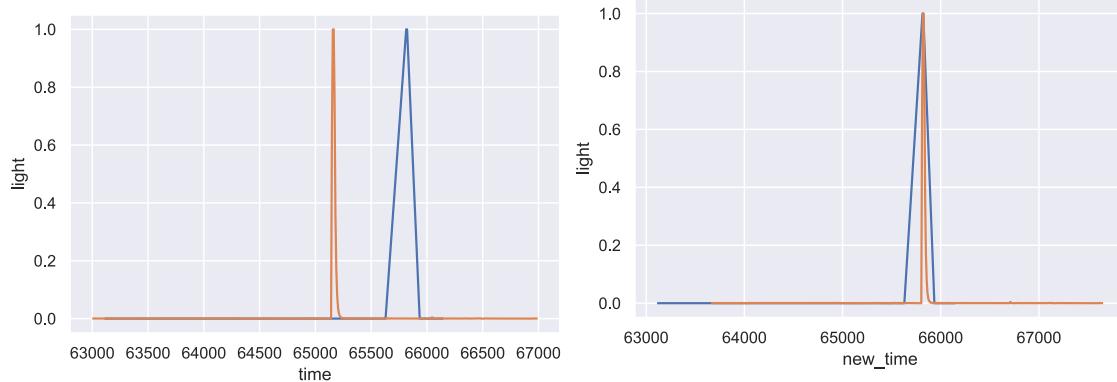


Abbildung 5.3: Verlauf einer Messung mit der App



(a) Peaks des Lichtsignals vom PSG-Gerät. Somit können die PSG-Daten und die Daten, gesammelt von den eSense Earpods zeitlich synchronisiert werden. Der Ablauf der Lichtpeaks ist identisch zum Ablauf der Nutzerstudie (siehe Abb. 4.2). Die x-Achse zeigt den Lichtwert des Sensors, die y-Achse den Zeitwert.



(b) Zeitlicher Lichtpeakvergleich vor der Synchronisation. Das Lichtsignal des PSG-Geräts (orange) ist weniger als 1s vom Signal der eSense Earpods (blau) entfernt.

(c) Zeitlicher Lichtpeakvergleich nach der Synchronisation. Das Lichtsignal des PSG-Systems (orange) ist nun zeitlich sehr nahe über den Lichtpeaks der eSense Earpods (blau)

# 6. Evaluation

Die Evaluation der Ergebnisse beginnt damit, die Daten der eSense Earpods in Fenster (*windows*) einzuteilen. Auf diese Fenster werden anschließend *Features* berechnet, welche die Grundlage für die Klassifizierung sind.

## 6.1 Gibt es passende Features?

Die Suche nach passenden Features wird mit dem Python Package `tsfresh` angegangen. `tsfresh` berechnet automatisch Charakteristiken, sowie deren Relevanz anhand eines Zeitintervalls (*Feature*) [14]. Diese Charakteristiken werden fortan als Features bezeichnet.

Wie in Kapitel 5.3 bereits erklärt, wird eine Messung in viele sich überlappende Fenster aufgeteilt. Eine Messung wird in eine Fenstergröße von 5s, bzw. 10s und einer Verschiebung des nächsten Fensters von 1s in viele Teile aufgeteilt. Nun wird für jedes Fenster eine Featureberechnung ausgeführt, womit durch `tsfresh` ( $\sim 6000$ ) Features für jedes Fenster entstehen. Als Eingabe bekommt `tsfresh` die Daten der eSense Earpods, also die  $x$ ,  $y$  und  $z$ - Achse der Gyroskop und Beschleunigungsdaten, versehen mit einem Zeitstempel.

## 6.2 Ablauf der Evaluierung

Im Kapitel 5.3 wurde beschrieben, wie die Features berechnet und persistiert wurden. Nun sind pro Studienteilnehmer für alle 3 Positionen Features berechnet worden. Es sind jeweils die Features für ein Fenster von 5 Sekunden und 10 Sekunden berechnet worden. Zur Erinnerung: Jede Messung wurde in Fenster der Länge von 5, bzw. 10 Sekunden aufgeteilt, wobei jedes Fenster um eine Sekunde verschoben ist.

### 6.2.1 Data Labeling

Zur Klassifikation müssen die Daten bereits markiert sein, um eine Entscheidung treffen zu können. Beim Abspeichern der eSense Daten ist dies bereits geschehen. Da die Messung genau vorgibt, wann eine Person die luft anhalten soll, beziehungsweise nicht, wird diese Information mit einem *Boolean* als zusätzliches Attribut

vermerkt. Da bei der Klassifikation Features anhand der eSense Daten berechnet werden, darf dieses Attribut offensichtlich nicht Teil der Featureberechnung sein. Da nun bei einem 5 Sekunden Intervall circa 250 Einträge, bei einem 10 Sekunden Intervall circa 500 Einträge in eine Featureberechnung zusammenfließen, muss die Markierung dieses Features gewählt werden. Ab 50% der markierten Einträge wird das ganze Intervall als markiert gesetzt. Dies ist eine essenzielle Entscheidung, da beim Training des Modells nun anhand dieser Markierung entschieden wird, ob ein Intervall ein Atemaussetzer repräsentiert, oder nicht. **TODO:** genauer beschreiben, warum hier 50% gewählt wurde... ich wollte bei 90% z.B die Übergänge nicht als 0 markieren, weil dann teile der 0 markierten features eigentlich ein Atemaussetzer gewesen wäre... dann trainiert es kacke... Nun können die Resultate mit den Klassifikatoren verglichen werden.

## 6.2.2 Kreuzvalidierungsverfahren

### 6.2.2.1 Within Subject (*k-fold cross validation*)

Das erste Kreuzvalidierungsverfahren ist die *K-fold cross validation*. Bei diesem Verfahren werden alle Daten in  $k$  Partitionen aufgeteilt. Eine Partition wird als Testdatensatz und  $k - 1$  Partitionen werden als Trainingsdatensatz gewählt [8]. So mit werden alle Personen als ein Datensatz zusammengefasst und mit der Methode `test_train_split` in Trainingsdaten und Testdaten aufgeteilt. Hier wird eine Aufteilung von 70% für die Trainingsdaten und 30% für die Testdaten gewählt. Daraus ergibt sich ein *score*, welcher **TODO:** was gibt mit der score Die Resultate sind in Abbildung 6.1 zu sehen.

Verfahren	Positionen	Score $w = 5s$ $d = 1s$	Score $w = 5s$ $d = 5s$	Score $w = 10s$ $d = 1s$	Score $w = 10s$ $d = 10s$
Random Forest	Alle	0.87	0.74	0.89	0.71
	Rücken	0.85	0.67	0.93	0.57
	Seite	0.86	0.72	0.89	0.75
	Bauch	0.90	0.85	0.92	0.66
XG Boost	Alle	0.88	0.70	0.92	0.76
	Rücken	0.89	0.67	0.95	0.62
	Seite	0.90	0.78	0.92	0.48
	Bauch	0.90	0.75	0.95	0.63
SVM	Alle	0.54	0.48	0.65	0.42
	Rücken	0.60	0.67	0.72	0.46
	Seite	0.57	0.38	0.63	0.5
	Bauch	0.61	0.41	0.67	0.4

Tabelle 6.1: Ergebnisse des Kreuzvalidierungsverfahrens innerhalb eines Subjects mit einer Aufteilung von 70% Trainings- und 30% Testdaten. Alle Personen der Nutzerstudie wurden verwendet.  $w$  steht für die Fenstergröße,  $d$  für die Verschiebung der aneinanderfolgenden Fenster.

Die Tabelle zeigt die Ergebnisse verschiedener Klassifikationsalgorithmen mit verschiedenen Datensätzen. Pro Klassifikationsalgorithmus wurden alle Positionen, so-

wie jede Position einzeln betrachtet, d.h jede Position jeder Person. Jede Kombination aus Klassifikationsalgorithmus und gewähltem Datensatz wurde nun mit den Fenstergrößen von 5s und 10s mit einer Verschiebung von 1s und einer Verschiebung der Fenstergröße evaluiert. Somit wurden die Fenstergrößen mit und ohne überlappende Fenster getestet. Die Resultate zeigen, dass der *score* mit überlappenden Fenstern deutlich höher ist. Dies könnte daran liegen, dass deutlich mehr Fenster pro Datensatz existieren, falls eine Überlappung der aneinanderfolgenden Fenster stattfindet. Jedoch ist es auch möglich, dass Teile der überlappenden Fenster bereits erkannt und markiert wurden. Dadurch ist der überlappende Teil eventuell in den Trainingsdaten vorhanden und eine Klassifizierung ist einfach für den Klassifikationsalgorithmus.

#### 6.2.2.2 Leave One Subject Out (LOSO)

Neben dem k-fachen Kreuzvalidierungsverfahren wird das das *Leave One Subject Out (LOSO)* Kreuzvalidierungsverfahren herbeigezogen. Hierbei wird ein Modell anhand der Daten der Studienteilnehmer trainiert, jedoch wird ein Datensatz eines Studienteilnehmers ausgelassen. Dieser eine Datensatz wird schließlich verwendet, um eine Vorhersage anhand der Trainingsdaten zu treffen, ohne durch interne Trainingsdaten optimiert sein zu können. Im Folgenden wurde das LOSO Kreuzvalidierungsverfahren mit den beiden Klassifikationsalgorithmen *Random Forest* und *XGBoost* durchgeführt. Es wurde jede Person einmal ausgelassen und auf den Trainingsdatensatz angewandt, welcher diese Person nicht enthält. Somit entstehen 7 Resultate, von denen nun der Mittelwert gebildet wird. Die Resultate sind in den Abbildungen 6.1 und 6.2 beispielhaft an Person 6 dargestellt. Im Anhang sind weitere Abbildungen anderer Personen zu finden. Die Abbildungen listen die Häufigkeit der Vorhersagen für jede einzelne Sekunde. Da eine Sekunde aufgrund der überlappenden Fenster in mehreren Fenstern vorkommt, summieren sich die Vorhersagen pro Sekunde maximal auf die Fenstergröße. Zu erkennen ist, dass die Fenstergröße von 10s deutlich bessere Ergebnisse liefert, als eine Fenstergröße von 5s. XGBoost neigt bei einer Fenstergröße von 5s zum Overfitting, ebenso wie Random Forest. Da bei der Nutzerstudie die 3 Phasen des Luftanhaltens 10s, 20s, bzw 30s betrag, könnte dies **TODO: ein möglicher Grund sein, kp warum aber :D** Der Mittelwert aller Ergebnisse des LOSO Kreuzvalidierungsverfahrens sind im *Precision* und *Recall* der unterliegenden Tabelle der Beispiele bei Person 6 mit dem Mittelwert zusammengetragen worden. Zu sehen ist, dass XGBoost bessere Ergebnisse liefert als Random Forest. **TODO: schreibe darüber, was die ergebnisse von LOSO aussagen**  
Des weiteren, um mögliche Positionsabhängigkeiten zu erkennen, wurden das selbe erneut evaluiert, jedoch nur mit den einzelnen Positionsdaten der jeweiligen Person. In der Tabelle 6.2 sind die Resultate zu sehen. **TODO: schreibe darüber, was die ergebnisse der einzelnen personen aussagen**

#### 6.2.3 Relevante Features

Jede Klassifikation ergab ein Resultat, welches durch einzelne Features entschieden wurde. Das Python-Framework `scikit-learn` beinhaltet `feature_importances_` zeigt die Relevanz der einzelnen Features in einem Plot. Um einen kleinen Überblick zu bekommen, welche Features entscheidend für die Evaluation waren, sind nun einige aufgelistet.

- gyroZ partial autocorrelation
- gytoX fft coefficient
- gyroX agg autocorrelation
- accY autocorrelation
- gyroY change quantiles
- gyroZ fft coefficient

Diese Features wurden mittels `tsfresh` berechnet und waren bei der Klassifikation des Kreuzvalidierungsverfahrens entscheidend [14]. Die partielle Autokorrelationfunktion (`partial autocorrelation`) beschreibt die Abhangigkeit zwischen den Werten der Zeitreihen. Zudem wird ein zeitlicher linearer Zusammenhang der verschiedenen Werte gemessen. Somit haben sich die Werte in diesem Fenster zwischen den Werten vermutlich weniger verandert und ahneln somit eher den positiv markierten Trainingsdaten.

Die *Schnelle Fourier-Transformation* berechnet effizient die diskrete Fourier-Transformation. Damit konnen zeitdiskrete Signale in Frequenzanteile zerlegt werden. Es wird aus einer Darstellung, bestehend aus Zeit- und Abtastwert in eine Darstellung, bestehend aus Frequenzanteil, Amplitude und Phase uberfuhrt. Die Funktion `fft_coefficient` in `tsfresh` berechnet die Fourier Koeffizienten einer eindimensionalen diskreten Fouriertransformation fur eine reale Eingabe mit dem Algorithmus der Fast-Fourier-Transformation:

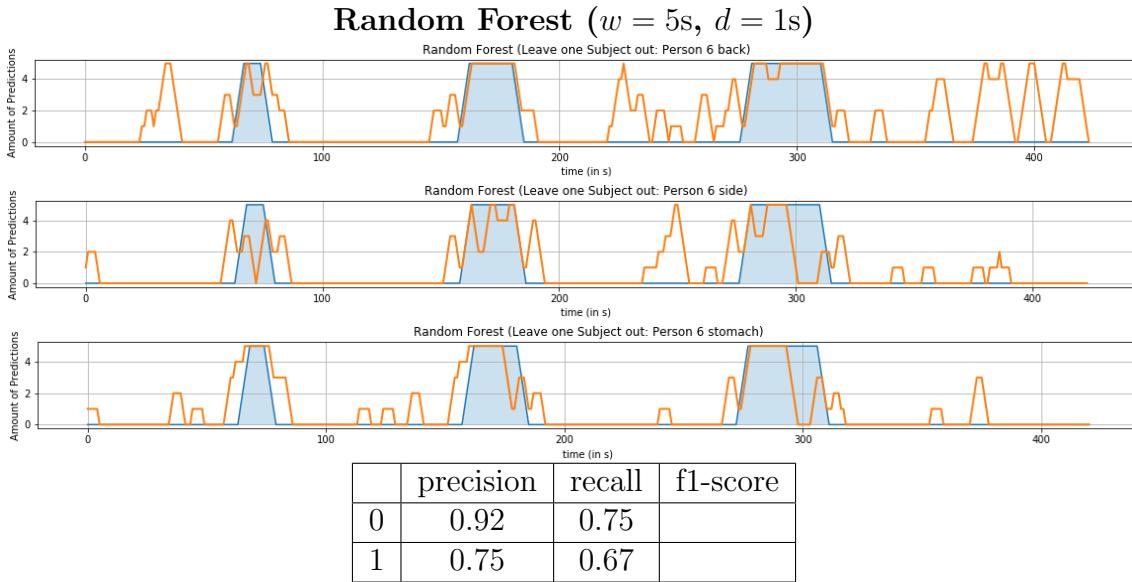
$$A_k = \sum_{m=0}^{n-1} a_m \exp^{-2\pi i \frac{mk}{n}}, \quad k = 0, \dots, n-1$$

Die bei der Apnoephase geringe Amplitude ist hier erkennbar und detektierbar, womit eine Klassifikation ermoglicht wird. Die Aggregation `agg_autocorrelation` berechnet den Wert der Autokorrelationsfunktion  $f_{agg}$  (beispielsweise die Varianz oder den Mittelwert) uber die Autokorrelation  $R(l)$  fur die verschiedenen Fenster.

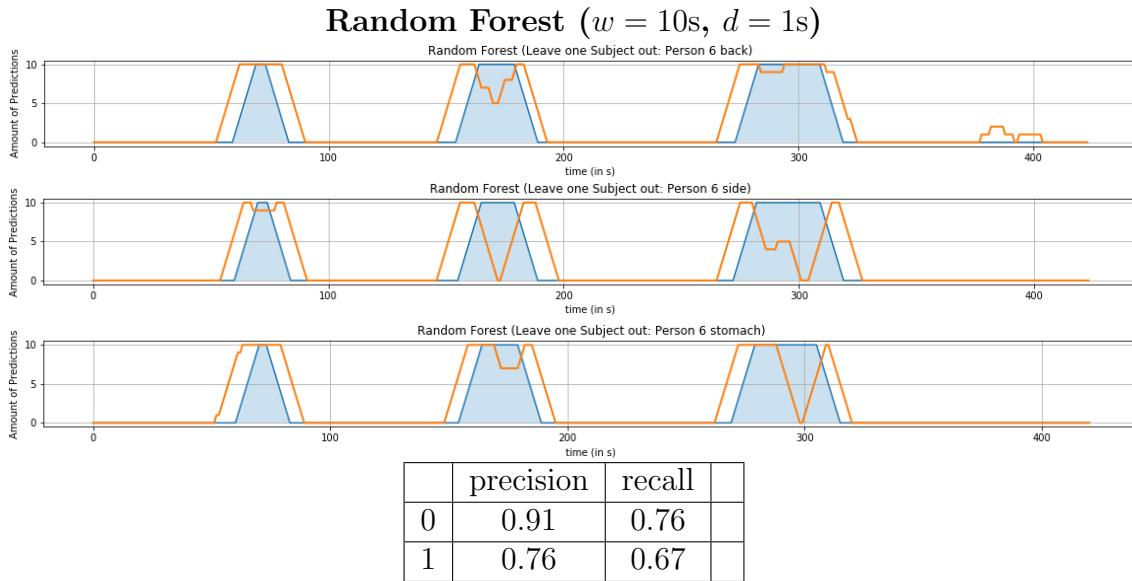
$$R(l) = \frac{1}{(n-l)\sigma^2} \sum_{t=1}^{n-l} (X_t - \mu)(X_{t+l} - \mu)$$

$X_i$  sind hierbei die Werte der Timeseries,  $n$  deren Lange.  $\sigma$  und  $\mu$  sind Schatzwerte fur die Varianz und den Mittelwert.

Ein weiteres Feature, welches ausschlaggebend fur die Klassifikation ist, ist der Wechsel der Quantilwerte (`change_quantiles`). Pro Fenster wird ein  $ql$  niedriges Quantil und ein  $qh$  hohes Quantil berechnet, anschlieend wird der durchschnittliche, absolute Wert der aufeinanderfolgenden anderungen der Reihe x innerhalb dieser Quantile berechnet. Dadurch werden starke Veranderungen deutlich und stellen klar, dass in diesem Fenster kein Apnoe stattfindet.

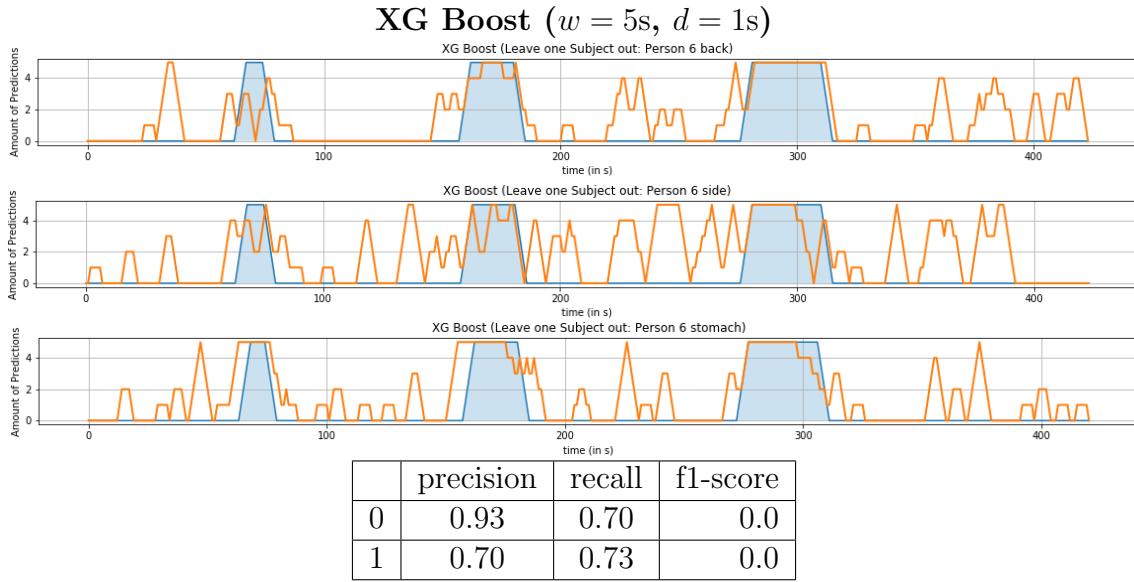


(a) Random Forest mit dem Kreuzvalidierungsverfahren. Die Tabelle zeigt den Mittelwert aller Vorhersagen der einzelnen Personen.

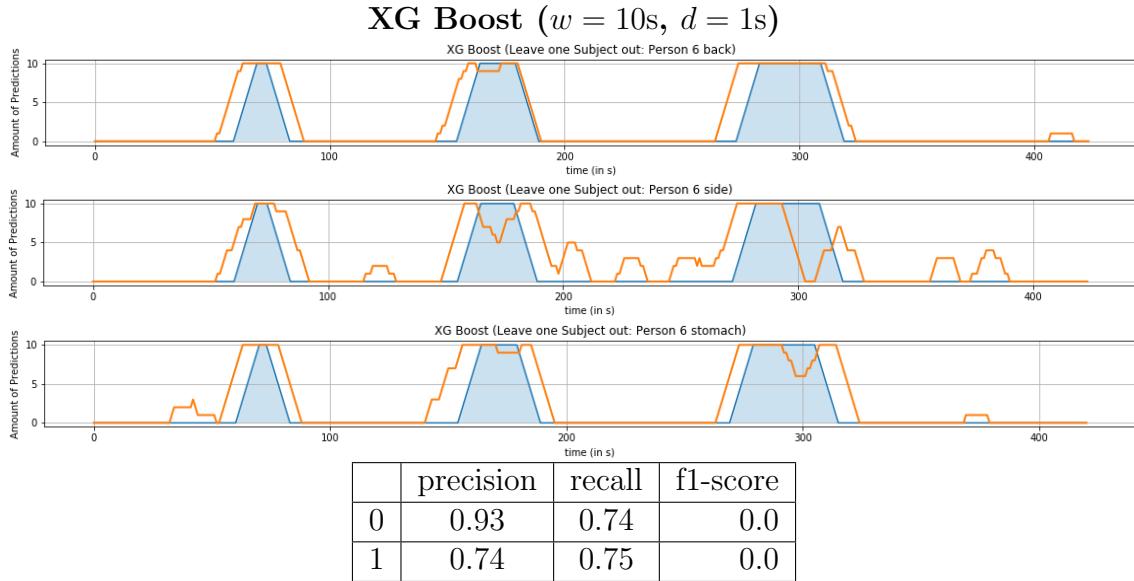


(b) Random Forest mit dem Kreuzvalidierungsverfahren (LOSO). Die Tabelle zeigt den Mittelwert aller Vorhersagen der einzelnen Personen.

Abbildung 6.1: Das Kreuzvalidierungsverfahren (LOSO) mit dem Klassifikationsalgorithmus Random Forest. Das Modell wurde auf allen Personen, exklusive einer Person trainiert und auf alle Positionen dieser einen Person wurde eine Vorhersage getroffen. Am Beispiel hier sind die Resultate von Person 6 zu sehen. Die blauen Bereiche sind die, in denen die Luft angehalten wurde, die orangene Kurve ist die Vorhersage. ( $w = \text{FenstergröÙe}$ ,  $d = \text{Verschiebung der Fenster}$ )



(a) XGBoost mit dem Kreuzvalidierungsverfahren (LOSO). Die Tabelle zeigt den Mittelwert aller Vorhersagen der einzelnen Personen.



(b) XGBoost mit dem Kreuzvalidierungsverfahren (LOSO). Die Tabelle zeigt den Mittelwert aller Vorhersagen der einzelnen Personen.

Abbildung 6.2: Das Kreuzvalidierungsverfahren (LOSO) mit dem Klassifikationsalgorithmus XG Boost. Das Modell wurde auf allen Personen, exklusive einer Person trainiert und auf alle Positionen dieser einen Person wurde eine Vorhersage getroffen. Am Beispiel hier sind die Resultate von Person 6 zu sehen. Die blauen Bereiche sind die, in denen die Luft angehalten wurde, die orangene Kurve ist die Vorhersage. ( $w = \text{Fenstergröße}, d = \text{Verschiebung der Fenster}$ )

	precision	recall	f1-score		precision	recall	f1-score		precision	recall	f1-score
0	0.90	0.72	0.00	0.89	0.83	0.00	0.92	0.72	0.72	0.0	
1	0.72	0.57	0.010	0.83	0.45	0.010	0.72	0.66	0.66	0.0	

Random Forest (only side, Random Forest (only back,  
*window: 5s*) Random Forest (only stomach, *window: 5s*)

	precision	recall	f1-score		precision	recall	f1-score		precision	recall	f1-score
0	0.91	0.67	0.00	0.90	0.75	0.00	0.94	0.71	0.71	0.0	
1	0.67	0.61	0.010	0.75	0.54	0.010	0.71	0.72	0.72	0.0	

XG Boost (only side, XG Boost (only back, XG Boost (only stomach,  
*window: 5s*) *window: 5s*) *window: 5s*)

	precision	recall	f1-score		precision	recall	f1-score		precision	recall	f1-score
0	0.89	0.79	0.00	0.89	0.79	0.00	0.94	0.71	0.71	0.0	
1	0.79	0.53	0.010	0.79	0.52	0.010	0.71	0.73	0.73	0.0	

Random Forest (only side, Random Forest (only back, Random Forest (only stomach, *window: 10s*) *window: 10s*) *window: 10s*)

	precision	recall	f1-score		precision	recall	f1-score		precision	recall	f1-score
0	0.90	0.78	0.00	0.90	0.75	0.00	0.94	0.71	0.71	0.0	
1	0.78	0.61	0.01	0.75	0.58	0.01	0.71	0.72	0.72	0.0	

XG Boost (only side, XG Boost (only back, XG Boost (only stomach,  
*window: 10s*) *window: 10s*) *window: 10s*)

Tabelle 6.2: Klassifikationsergebnisse mit dem Kreuzvalidierungsverfahren bei einer Fenstergröße von 5, bzw. 10 Sekunden, welche um 1 Sekunde verschoben wurden. Es wurde immer eine Person aus dem Trainieren des Modells ausgelassen und auf diese Person wurde anschließend eine Vorhersage getroffen. Jede Person wurde einmal in dem Modell außen vorgelassen und auf dieses Modell vorhergesagt. Das zu sehende Resultat ist der Mittelwert aller Resultate des Kreuzvalidierungsverfahrens der einzelnen Positionen der Position.



## 7. Ausblick

Wie in der Evaluation zu sehen, ist noch nicht die perfekte Lösung gefunden worden, ein Schlafapnoe mittels der eSense Earpods zu ermitteln. Es gibt Bereiche, in denen gute Resultate erzielt wurden, jedoch auch Bereiche, in denen Verbesserungspotenzial steckt.

Der Ausblick wird in zwei Bereiche aufgeteilt. Der erste Bereich sind die Signale, welche zusätzlich in die Analyse mit aufgenommen werden können. Im zweiten Bereich wird die Datenanalyse und Klassifikation genauer betrachtet.

Zum ersten Bereich zählt das Mikrofonsignal, welches bei der Nutzerstudie zwar aufgezeichnet undpersistiert wurde, jedoch in der Auswertung nicht mit eingeflossen ist. Das Mikrofonsignal kann weitere Informationen zur Detektion eines Schlafapnoes liefern. Mittels des Tonsignals direkt am Ohr können unter anderem die Atmung und auch den Puls herausgefiltert werden [9]. Diese Informationen können in die Auswertung mit einfließen und somit die Erkenntnisse aus den IMU-Daten mit den Erkenntnissen des Mikrofonsignals zusammengeführt werden.

Im zweiten Bereich des Ausblicks wird erläutert, wie die Auswertung der zur Verfügung stehenden Daten verbessert werden kann. Ein großer Nachteil der aktuellen Klassifikation eines Schlafapnoe-Ereignisses ist, dass die Fenster nicht in Abhängigkeit zueinander stehen. Bei der aktuellen Klassifikation wird jedes Fenster separat und unabhängig von den anderen Fenstern klassifiziert. Ein LSTM (*Long short-term Memory*) würde hier die vorherigen, bzw. nachfolgenden Fenster ebenfalls in Betracht ziehen und deren Entscheidung mit in die aktuelle Entscheidung mit einfließen lassen. Somit wird die Fehlerquote verringert, da potenzielle Outlier vermieden werden.

Zudem gibt es die Möglichkeit, die weitere Features der IMU-Daten zu berechnen. In dieser Bachelorarbeit wurden alle Features aus `tsfresh` mit in Betracht gezogen. Es könnten weitere Features in die Auswertung mit einfließen. Es ist möglich, anhand der IMU-Daten das Puls- und  $SPO_2$ -Signal zu extrahieren. Diese beiden Signale können dann als Feature pro Fenster weitere Erkenntnisse liefern. In Abbildung

Eine weitere Möglichkeit bietet sich, indem die Zusatzinformationen der Nutzer mit in die Klassifikation eingebunden werden. Hier können Personen des gleichen Geschlechts, Gewichts oder des gleichen Schlafrhythmus in einem Modell trainiert

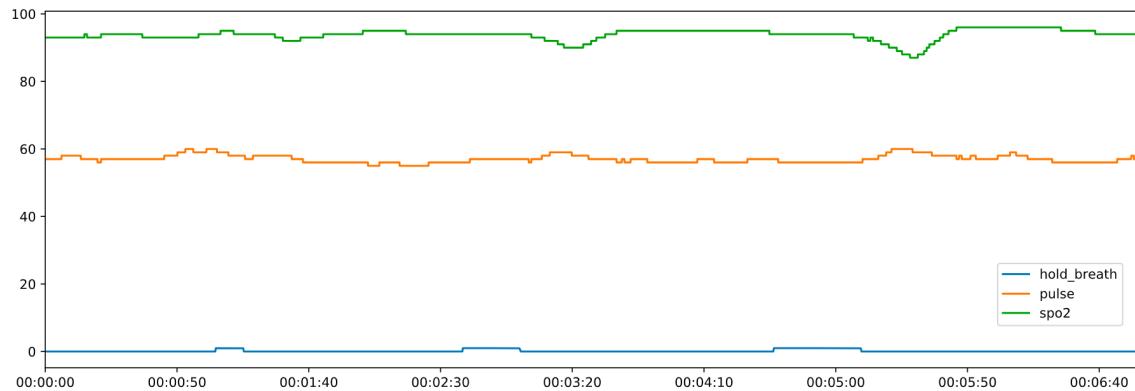


Abbildung 7.1: TODO: Add description

werden. Dies könnte mögliche Einflussfaktoren elmininieren und das Ergebnis ebenfalls verbessern.

TODO: Paper von Tobi hinzufügen [10]

TODO: können bei der Klassifikation outlier entfernt werden? kann overfitting mindern

# Literatur

- [1] Tianqi Chen und Carlos Guestrin. „XGBoost: A Scalable Tree Boosting System“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16* (2016), S. 785–794. DOI: 10.1145/2939672.2939785. arXiv: 1603.02754. URL: <http://arxiv.org/abs/1603.02754> (besucht am 31.03.2020).
- [2] Corinna Cortes und Vladimir Vapnik. „Support-Vector Networks“. In: *Machine Learning* 20.3 (Sep. 1995), S. 273–297. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/BF00994018. URL: <http://link.springer.com/10.1007/BF00994018> (besucht am 31.03.2020).
- [3] Tatjana Crönlein, Wolfgang Galetke und Peter Young. *Schlafmedizin 1x1: Praxisorientiertes Basiswissen*. Berlin Heidelberg: Springer-Verlag, 2017. ISBN: 978-3-662-49788-3. DOI: 10.1007/978-3-662-49789-0. URL: <https://www.springer.com/de/book/9783662497883> (besucht am 31.01.2020).
- [4] Jerome H. Friedman. *Stochastic Gradient Boosting*. 1999.
- [5] J Hernandez u. a. „Cardiac and Respiratory Parameter Estimation Using Head-Mounted Motion-Sensitive Sensors“. In: (), S. 10.
- [6] Phan Duy Hung. „Central Sleep Apnea Detection Using an Accelerometer“. In: *Proceedings of the 2018 International Conference on Control and Computer Vision - ICCCV '18*. The 2018 International Conference. Singapore, Singapore: ACM Press, 2018, S. 106–111. ISBN: 978-1-4503-6470-6. DOI: 10.1145/3232651.3232660. URL: <http://dl.acm.org/citation.cfm?doid=3232651.3232660> (besucht am 04.02.2020).
- [7] Gareth James u. a. *An Introduction to Statistical Learning*. Bd. 103. Springer Texts in Statistics. New York, NY: Springer New York, 2013. ISBN: 978-1-4614-7137-0 978-1-4614-7138-7. DOI: 10.1007/978-1-4614-7138-7. URL: <http://link.springer.com/10.1007/978-1-4614-7138-7> (besucht am 31.03.2020).
- [8] Gerhard Neumann. *Maschine Learning KIT*. 2020.
- [9] H. Noma u. a. „Wearable Data Acquisition for Heartbeat and Respiratory Information Using NAM (Non-Audible Murmur) Microphone“. In: *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*. Ninth IEEE International Symposium on Wearable Computers (ISWC'05). Okt. 2005, S. 210–211. DOI: 10.1109/ISWC.2005.56.
- [10] Mathias Perslev u. a. „U-Time: A Fully Convolutional Network for Time Series Segmentation Applied to Sleep Staging“. In: (24. Okt. 2019). arXiv: 1910.11162 [cs, eess, stat]. URL: <http://arxiv.org/abs/1910.11162> (besucht am 03.04.2020).

- [11] Tobias Röddiger u. a. „Towards Respiration Rate Monitoring Using an In-Ear Headphone Inertial Measurement Unit“. In: *Proceedings of the 1st International Workshop on Earable Computing*. UbiComp '19: The 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing. London United Kingdom: ACM, 9. Sep. 2019, S. 48–53. ISBN: 978-1-4503-6902-2. DOI: 10.1145/3345615.3361130. URL: <https://dl.acm.org/doi/10.1145/3345615.3361130> (besucht am 24.02.2020).
- [12] *Somnomedics Somnoscreen Plus Broschüre*.
- [13] Boris Stuck u. a. *Praxis der Schlafmedizin: Diagnostik, Differenzialdiagnostik und Therapie bei Erwachsenen und Kindern*. 3. Aufl. Berlin Heidelberg: Springer-Verlag, 2018. ISBN: 978-3-662-54382-5. DOI: 10.1007/978-3-662-54383-2. URL: <https://www.springer.com/de/book/9783662543825> (besucht am 31.01.2020).
- [14] *Tsfresh — Tsfresh 0.12.0 Documentation*. URL: <https://tsfresh.readthedocs.io/en/latest/> (besucht am 24.03.2020).
- [15] *Wayback Machine*. 17. Apr. 2016. URL: <https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf> (besucht am 28.03.2020).