

Klassifikation von respiratorischen Ereignissen mit Earables und maschinellem Lernen

**Classification of Respiratory Events with Earables
and Machine Learning**

Bachelorarbeit

von

David Laubenstein

an der Fakultät für Informatik

Verantwortlicher Betreuer: Prof. Dr. Michael Beigl

Betreuernder Mitarbeiter: Tobias Röddiger

Bearbeitungszeit: 11/12/2019 – 15/04/2020

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und weiterhin die Richtlinien des KIT zur Sicherung guter wissenschaftlicher Praxis beachtet habe.

Karlsruhe, den 15.04.2020

Zusammenfassung

Weltweit leiden 15% der Menschen unter Atemaussetzern während des Schlafs, von denen 85% undiagnostiziert sind. Es gibt diverse Arten von Atemaussetzern, wie zum Beispiel der zentralen Apnoe, die ein wichtiger Bestandteil dieser Bachelorarbeit ist. Bei zentralem Apnoe setzt die Atmung bei geöffneten Atemwegen für mindestens 10 Sekunden aus. Der Grund für den Atemaussetzer ist, dass das Gehirn in dieser Zeit dem Körper keine Signale zum Atmen weitergibt. Heutige Methoden, die es zur Klassifikation solcher respiratorischer Ereignisse gibt, sind aufwendig und kostspielig. Spezialisierte Schlaflabore sind hier der Standard zur Ermittlung von Apnoeereignissen, jedoch ist dies mit einigen Nachteilen verbunden. Die durchschnittliche Wartezeit beträgt circa drei Monate und die Patienten verbringen meist nicht länger als zwei Tage im Schlaflabor. Dies führt dazu, dass der Datenbestand unzufriedenstellend ist und eine genaue Diagnose nur beschränkt möglich ist.

Diese Bachelorarbeit fokussiert sich darauf, mittels intelligenten Kopfhörern, sogenannten Earables, eine solche zentrale Apnoe zu klassifizieren. Die eSense Earpods dienen hier als Grundlage, da diese eine inertiale Messeinheit (IMU) beinhalten, womit die nötigen Signale aufgezeichnet werden können. Der Sensor ist im Gegensatz zu anderen Ansätzen nicht am Oberkörper, sondern im Ohr platziert und ohne weitere Verkabelung einsetzbar. Somit ist eine Möglichkeit geboten, ein Apnoeereignis im eigenen Bett zuhause zu erkennen. Als Ground-Truth dient ein Polysomnographie-Gerät, womit Vorhersagen anhand der IMU-Daten verifiziert werden können. Zu Beginn der Arbeit wurde eine App erstellt, welche Messergebnisse mit den eSense Earpods während einer Nutzerstudie aufzeichnet. Während der Nutzerstudie werden die Studienteilnehmer gebeten, an bestimmten Zeitpunkten die Luft für 10, 20 und 30 Sekunden anzuhalten. Dies simuliert ein zentrales Apnoeereignis und stellt die Grundlage dar, unter welchen der Klassifikator seine Entscheidung trifft.

In der Evaluation wird mit verschiedenen maschinellen Lernverfahren zur Klassifikation das Kreuzvalidierungsverfahren durchgeführt. Mit dem Kreuzvalidierungsverfahren „Within Subject“ wurde ein *score* von bis zu 95% erreicht, mit dem Kreuzvalidierungsverfahren „Leave One Subject Out“ ergab sich eine hohe Varianz in den Ergebnissen von Studienteilnehmer zu Studienteilnehmer. Hier wurde ein *f1-score* von bis zu 86% erreicht, vorherzusagen, dass in diesem Fenster kein Apnoeereignis stattfand. Ein Fenster als Apnoeereignis zu deklarieren ergab einen *f1-score* von bis zu 55%.

Im Mittel ergab das Kreuzvalidierungsverfahren „Within Subject“ einen *score* von **TODO: 94%**, das Kreuzvalidierungsverfahren „Leave One Subject Out“ einen *f1-score* von 79% bei der Klassifikation eines Fensters ohne Apnoeereignis und 47% eines Fensters mit Apnoeereignis. Aufgrund der hohen Anzahl an Features pro Fenster (circa 6000) neigt das Modell zu Overfitting, was die Varianz der Resultate erklärt.

Da bei der aktuellen Featureberechnung lediglich die Gyroskop- und Beschleunigungsdaten in Betracht gezogen wurden, wird bei einer erweiterten Featureberechnung, beispielsweise durch Betrachtung von Puls- und *SpO₂*-Signalen, eine Optimierung der Resultate erwartet. Zudem würde ein maschinelles Lernverfahren, welches

vorherige und nachfolgende Fenster in die Entscheidung mit einfließen lässt, eine Verbesserung der Signale liefern.

Abstract TODO: take summary and translate to engl

Inhaltsverzeichnis

1 Einführung	7
1.1 Problem	7
1.2 Idee	8
1.3 Struktur	8
2 Grundlagen & aktuelle Forschung	9
2.1 Schlafmedizin	9
2.2 Klassifizierung von Schlafstörungen	10
2.2.1 Zentrale Schlafapnoe	10
2.3 Maschinelle Lernverfahren	11
2.3.1 Random Forest	12
2.3.2 XGBoost	13
2.3.3 Support Vektor Maschine (SVM)	13
2.4 Forschung von Klassifikation anhand von IMU-Daten	14
2.4.1 Beschleunigungssensor am Brustkorb	14
2.4.2 Detektion direkt am Kopf mittels der Google-Glass Brille . . .	16
2.4.2.1 Pulssignal	16
2.4.2.2 Atemsignal	18
2.4.3 Überwachung von Puls und Atmung mittels eSense-Earpods .	18
3 Schlafanalyse	21
3.1 Earable Plattform	21
3.1.1 Was wird aufgezeichnet?	21
3.1.2 Datenexport	22
3.2 Polysomnographie-Systeme	22
3.2.1 Datenexport	24
3.3 Kamera	24
3.4 Datensynchronisation	24
3.5 Zusatzinformationen der Nutzer	25
4 Systementwurf	27
4.1 Studienplanung	27
4.2 Systemmodelle	27
5 Implementierung	31
5.1 App	31
5.1.1 Messungsablauf	32
5.1.2 Messung	32
5.2 Anbindung an die Auswertungspipeline	33

5.2.1	Synchronisation der Daten	33
5.3	Verarbeitungspipeline zur Klassifikation	34
6	Evaluation	37
6.1	Studienablauf	37
6.2	Gibt es passende Features?	39
6.3	Ablauf der Evaluierung	40
6.3.1	Data Labeling	40
6.3.2	Kreuzvalidierungsverfahren	40
6.3.2.1	Within Subject (<i>k-fold cross validation</i>)	40
6.3.2.2	Leave One Subject Out (LOSO)	41
6.3.3	Relevante Features	42
6.3.4	Vorverarbeitung durch einen Tiefpassfilter	43
7	Diskussion & Ausblick	47
8	Anhang	49
	Literatur	55

1. Einführung

Bevor die für das Verständnis dieser Arbeit erforderlichen Details erklärt werden, wird auf das Problem der aktuellen Diagnose eines Atemaussetzers eingegangen. Anschließend wird die Idee beschrieben, wie dieses Problem in dieser Bachelorarbeit nun behandelt wird. Zuletzt wird die Struktur der Vorangehensweise beschrieben, um einen Einblick zu geben, welchen Umfang die Bachelorarbeit hat.

1.1 Problem

Das in dieser Bachelorarbeit zu lösende Problem ist die Klassifizierung eines Atemaussetzers, welcher in verschiedenen Typen auftreten kann. In Abbildung 1.1 sind drei verschiedene Atmungsabläufe dargestellt. Darunter ist der jeweilige Atemfluss als Signal des zugehörigen Atemwegs abgebildet. Bei verengten und geschlossenen Atemwegen ist eine deutliche Signaländerung vorhanden. Der Standard zur Diagnose eines solchen respiratorischen Ereignisses während des Schlafs ist eine Analyse in einem Schlaflabor mittels vieler angeschlossener Sensoren. Da man oft bis zu drei Monate auf einen Termin warten muss und die Untersuchung sehr kostspielig ist, hätte eine bequem von Zuhause aus durchführbare Alternative viele Vorteile.

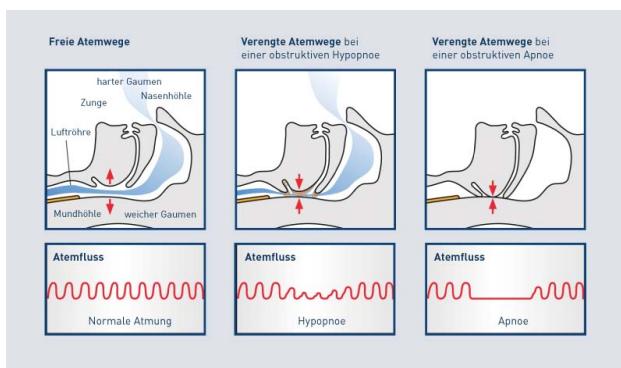


Abbildung 1.1: Die Darstellung zeigt die Querschnittsansicht einer Person und deren Atemwege und darunter der Atemfluss als Signal, welches Veränderungen während einer Atmungsstörung zeigt [4].

1.2 Idee

Die Idee ist nun solche Signaleinbrüche mit maschinellem Lernen zu erkennen (siehe Abbildung 1.1). Das Signal wird durch eine IMU (*inertial measurement unit*) der eSense-Earpods geliefert. Dieses Signal besteht aus 6-Achsen, den x -, y - und z -Achsen des Gyroskop- und Beschleunigungssensors. Durch eine Klassifikation, also ob in einem Zeitintervall ein Apnoeereignis vorgekommen ist, oder nicht, kann eine erste Diagnose eines respiratorischen Ereignisses bereits bequem von Zuhause erfolgen. Da der Proband eine Messung mit den eSense-Earpods über viele Nächte durchführen kann, sind die Signale über einen deutlich längeren Zeitabschnitt verfügbar, als sie in einem Schlaflabor aufgezeichnet werden.

1.3 Struktur

Zu Beginn wird eine App konstruiert, welche sich mit den eSense-Earpods über BLE (*Bluetooth Low Energy*) verbindet (siehe Abbildung 1.2). Anschließend wird eine Nutzerstudie mit 7 Personen durchgeführt, um einen Datensatz zu generieren. Den Probanden wird über die Lautsprecher der eSense-Earpods mitgeteilt, wenn sie die Luft anhalten sollen, um ein zentrales Apnoe zu simulieren. Anhand dieses Datensatzes kann schlussendlich ein Klassifikator trainiert werden, womit eine zukünftige Vorraussehung einer Messung getroffen werden kann. Im dritten Bild der Abbildung 1.2 ist das Signal des PSG-Geräts und die Rohdaten der eSense Earpods dargestellt. Zusammen bilden diese Signale neben den Kameraaufnahmen, dem Mikrofonsignal und den Nutzerinformationen einen Datensatz. Das vierte Bild zeigt, dass nach der Klassifikation der Rohdaten, also den Gyroskop- und Beschleunigungsdaten der eSense Earpods, eine Entscheidung darüber getroffen wird, ob in der Messung ein Apnoeereignis stattfand, oder nicht. Die Messung wurde hierbei in viele Intervalle aufgeteilt, worauf jeweils eine individuelle Entscheidung getroffen wurde.

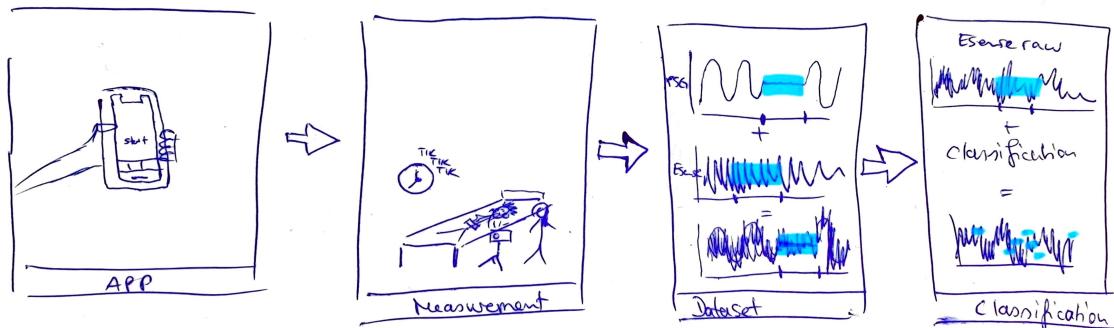


Abbildung 1.2: Darstellung des Ablaufs der Bachelorarbeit: Zu Beginn soll eine App erstellt werden, welche anschließend zur Datensammlung einer Nutzerstudie dient. Daraufhin wird ein Datensatz erstellt, welcher neben den Daten der eSense Earpods auch den Ground Truth beinhaltet. Anschließend wird eine Klassifikation anhand der Rohdaten der eSense Earpods getroffen, ob ein Apnoeereignis in einem Intervall vorkommt, oder nicht.

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 gewinnt Schlafmedizin erst seit dem letzten Jahrhundert an 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 diversen Bereichen bekannt, welche neben psychologischen Testverfahren überwiegend elektrophysiologisch untersucht und behandelt werden. Patienten werden mit ambulanten Hilfsmitteln oder stationär in einem Schlaflabor untersucht und diese dabei ermittelten Daten werden anschließend von technisch ausgebildetem Personal analysiert.

In einem Schlaflabor wird eine Polysomnographie durchgeführt, womit genauere Messergebnisse im Vergleich zu ambulanten Hilfsmitteln (z.B einer Langzeitbewegungsmessung) erreicht werden können. Während einer Polysomnographie werden Gehirnströme, Augenbewegungen und Muskelspannungen erkannt, wodurch die einzelnen Schlafphasen unterschieden werden können. Die verschiedenen Sensorwerte werden dann in Form eines Hypnogramms mittels der Kriterien des AASM (engl. *American Association for Sleep Medicine*) ausgewertet (siehe Abbildung 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.

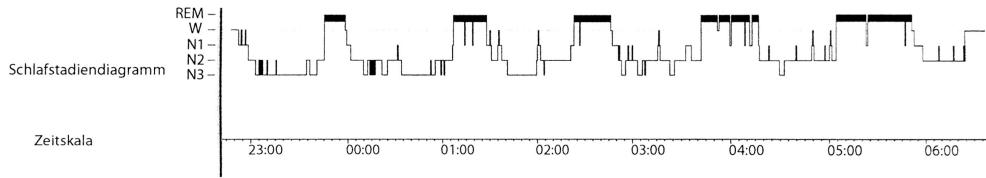


Abbildung 2.1: Beispielaufzeichnung, dargestellt in einem Hypnogramm [16]

2.2 Klassifizierung von Schlafstörungen

Zur Charakterisierung von Schlafstörungen werden viele Biosignale während des Schlafs registriert, die entscheidende Merkmale liefern [16]. Aufgrund dieser Annahme wurde eine Einteilung in 7 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* [16].

Diese Bachelorarbeit ist darauf fokussiert, eine zentrale Schlafapnoe zu klassifizieren, die Bestandteil der „schlafbezogenen Atmungsstörungen“ ist. Neben der zentralen Apnoe gibt es die obstruktive Apnoe. Hierbei sind die Atemwege geschlossen und eine Atmung ist für mehrere Sekunden unmöglich. Circa 20% aller Erwachsenen haben fünf oder mehr obstruktive Ereignisse pro Schlafstunde [3]. Zentrale Apnoe hingegen tritt seltener auf als obstruktive Apnoe, jedoch auffällig oft bei besonderen Patientengruppen. Ein Beispiel liefern Patienten mit Herzinsuffizienz und einer eingeschränkten kardialen Pumpfunktion. Unter einer zentralen Schlafapnoe leiden 72% dieser Patientengruppe. Dies macht die Bedeutung der Klassifikation deutlich, da eine genaue Erkennung einer zentralen Apnoe hier sehr wichtig ist.

2.2.1 Zentrale Schlafapnoe

Bei der zentralen Apnoe steht der Luftfluss trotz offener Atemwege für mindestens 10s still. Dies kann vollständig (zentrale Apnoe), oder partiell (zentrale Hypopnoe) erfolgen. Ab einer Anzahl von fünf Apnoeereignissen wird zentrale Apnoe diagnostiziert. Die Ursachen gelten hierbei als internistische oder neurologische Grundlage. Typische an- und abschwellige Muster sind auf eine chronische Herzinsuffizienz, auf

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 der zentralen 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 zentrale Apnoe ebenso wie obstruktive Apnoe in Form von Atemaussetzern, häufig vom Partner des Patienten beobachtet. Zudem kann lautes und unregelmäßiges Schnarchen ein Indiz für eine Apnoe sein, ebenso wie ein Aufwachen in Atemnot.

Eine Schlafapnoe kann jedoch in unterschiedlichen Schweregraden auftreten. Es gibt Patienten, welche kaum bis keine Probleme haben und nur aufgrund von nächtlichen Erkenntnissen ihrer Partner 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 einer zentralen Apnoe werden im Rahmen dieser Bachelorarbeit verschiedene Klassifikationsverfahren verwendet, welche dann zentrale Apnoeereignisse anhand der gesammelten Trainingsdaten klassifizieren sollen.

Maschinelle Lernverfahren sind Algorithmen, welche ihre Performance mittels Trainingsdaten verbessern [11]. Es wird zwischen folgenden Lernverfahren unterschieden:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Beim *Supervised Learning* sind die Trainingsdaten im Vergleich zum *Unsupervised Learning* markiert. Das heißt, dass zu jedem Eintrag ein Bit gespeichert wird, welches angibt, ob eine Apnoe zu dem Zeitpunkt stattfand, oder nicht. Falls ein Apnoeereignis simuliert wurde, wird eine 1 („positiv markiert“) eingetragen, andernfalls eine 0 („negativ markiert“). Die folgenden Tabelle zeigt die Unterschiede der verschiedenen Modelle bei *Supervised* und *Unsupervised Learning*:

Supervised Learning	Unsupervised Learning
Regression	Clustering
Klassifikation	Dimensionsreduktion

Jeder Algorithmus, welcher in einem dieser Modelle des Maschinellen Lernverfahrens 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. Beispiele hierfür

wären die Genauigkeit, der Precision & Recall, die Entropie oder der Likelihood-Schätzer. Beim dritten Teil, der *Optimization*, wird eine Optimierung des Algorithmus angestrebt. Dies kann unter anderem mit Methoden zweiter 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*. Aus dem Datensatz werden hierbei Features berechnet, welche dann einen wichtigen Teil der Klassifikation sind. Ein Feature ist ein Merkmal eines Datensatzes, welches ein individuelles, messbares Attribut der Rohdaten ist. Pro Klassifikation werden üblicherweise eine Vielzahl an Features gesammelt, anhand derer der Klassifikator seine Entscheidungen trifft.

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* [19]. Zur Klassifikation werden markierte Daten vorausgesetzt, woraus eine gelernt mathematische Repräsentation erstellt wird (*model*). Dieses Modell wird anschließend genutzt, um auf unmarkierten Daten eine Vorhersage zu treffen. Random Forest basiert auf Entscheidungsbäumen. Das 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 zu Overfitting. Random Forest nutzt deshalb viele Entscheidungsbäume und trainiert diese auf dem 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 Entscheidungsbäum genommen [9]. Je nach Wahl geschieht dies mit oder ohne Zurücklegen dieser zufälligen Teilmenge für den nächsten Entscheidungsbäum. Bei Bagging werden die Entscheidungsbäume parallel trainiert, was sich positiv auf die Performance auswirkt. Bei *Boosting* werden ebenfalls Teilmengen pro Entscheidungsbäum aus der Datenmenge gepickt, jedoch nicht zufällig. Es werden schwierige Fälle höher gewichtet, wodurch diese häufiger in einen Entscheidungsbäum 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. Die Wahrscheinlichkeit ist nun höher, dass dieser Datenbestand im nächsten Entscheidungsbäum enthalten ist. Die Entscheidungsbäume werden bei Boosting sequenziell trainiert.

Random Forest verwendet *Bagging*. Die endgültige Entscheidung ist entweder ein Mehrheitsvotum oder der Durchschnitt der Vorhersagen aller Trainingsläufe. Bei Random Forest können zusätzlich einige Hyperparameter modifiziert werden, wie unter anderem die Anzahl der zu kombinierenden Entscheidungsbäume, 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 nicht nur die entgültige Entscheidung, sondern auch die Wahrscheinlichkeit der Entscheidung an. Zu den Nachteilen zählt 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

Vor der Erklärung von XGboost muss zuvor *Gradient Boosting* eingeführt werden. Gradient Boosting ist wie Random Forest ein Teil des überwachten Lernens (*Supervised Learning*) und kann für die Klassifikation oder Regression verwendet werden [5]. 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 im Kapitel 2.3.1 eingeführt, trainiert *Boosting* zufällig gewichtete 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, insbesondere 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 je nach der getroffenen Vorhersage angepasst. Aus dem Basismodell mit den nun angepassten Gewichten wird eine Teilmenge gewählt, womit erneut Entscheidungsbäume entstehen und Vorhersagen getroffen werden können. Bei Gradient Boosting wird nun mittels Gradienten der Fehler der verschiedenen Ensembles minimiert. Auch hier gibt es Hyperparameter, welche angepasst werden können. Diese sind sehr ähnlich zu Random Forest, werden jedoch um weitere Parameter ergänzt, wie zum Beispiel einer 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 „Trennlinie“ gesucht, welche die Menge der Daten in zwei 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 markierten Daten unterscheiden können. Jedoch ist die Komplexität hierbei sehr hoch, da eine Trennlinie, beziehungsweise eine Trennebene im hochdimensionalen Raum, nur mit sehr hohem Aufwand gefunden werden kann. Die Suche nach der Trennline kann mit unterschiedlichen Kernel bestimmt werden. Die beliebtesten Kerneltypen 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 sehr einfach. 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 die 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 unter anderem die Atmung herauszufiltern, um dann Rückschlüsse auf Schlafstörungen machen zu können.

2.4.1 Beschleunigungssensor am Brustkorb

Bereits 2018 gab es Forschung in diesem Bereich. *Phan Duy Hung* hat sich damit befasst, einen Beschleunigungssensor (*Accelerometer*) an den Brustkorb zu fixieren [8]. Durch die Platzierung an genau dieser Stelle wurde der Herzschlag genauer erkannt. Die Rohdaten zeigen hierbei bereits die Herzfrequenz und die Atmung des Probanden (siehe Abb. 2.2).

Die Daten wurde 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, beispielsweise von der Reibung des T-Shirts und der Haut, aus dem Signal herausgefiltert werden. Zudem wurden 3 Features anhand der Amplitude über die Zeit berechnet. Das erste Feature ist das *Spektralverhältnis* (*spectral ratio*). Hier wird ein Leistungsspektrum eines einminütigen Segments vom HR-Signal geschätzt. Der Frequenzbereich zwischen [0,1] Hz

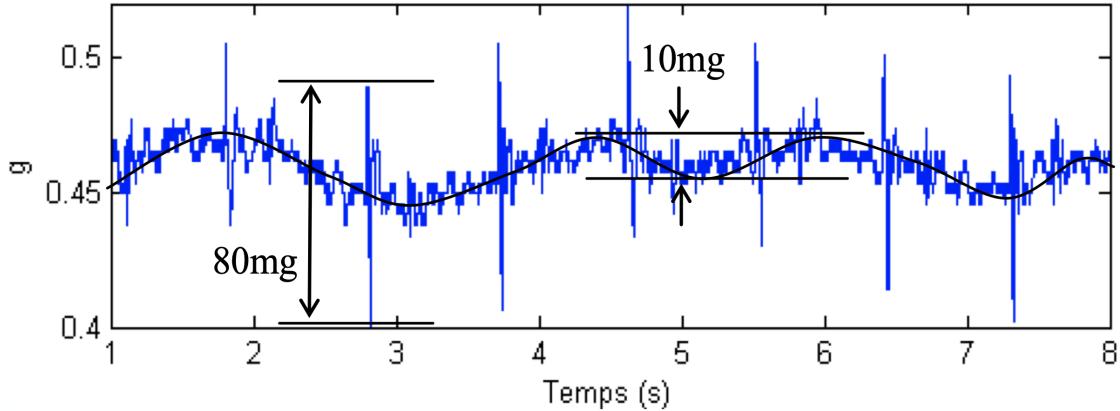


Abbildung 2.2: Rohdaten des Accelerometers in [8]

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 Maximum der spektralen Leistung. 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 welchen die Analyse mit mehreren Auflösungen und guter Lokalisierungsfähigkeit im Zeit-Frequenz-Bereich häufig eine Wavelet-Transformation verwendet wird. Hierbei wird das Atemsignal in 5 Ebenen mit der db2-Wavelet zerlegt. Danach werden die Standardabweichungen der Detailkoeffizienten in den Ebenen 4 und 5 verwendet.

Das dritte Feature sind die *linear prediction coefficients*. Die lineare Vorhersage zweiter Ordnung

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + e(n)$$

wird verwendet, wobei x eine Zeitreihe (*Timeseries*) ist, $e(n)$ der Vorhersagefehler und a_1, a_2 die Vorhersagekoeffizienten, die durch die Least-Square-Optimierungsverfahren zu bestimmen sind.

Nach einer Varianzanalyse (*ANOVA*) werden die folgenden Werte 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 wurde hierbei als Zeitreihe (*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 eine Apnoe stattgefunden hat, oder nicht.

Durch dieses Paper wurde eine Genauigkeit von 84.2% erreicht, eine zentrale Apnoe als solche zu erkennen. Zudem wurde zu 84.1% erkannt, dass in diesem Zeitrahmen keine zentrale Apnoe vorkam.

2.4.2 Detektion direkt am Kopf mittels der Google-Glass Brille

Bereits 2015 wurde versucht, durch Informationen des Google-Glass den Puls und das Atemsignal zu ermitteln [7]. Der Vorteil hierbei ist die Position der Brille. Da sie am Kopf platziert ist, liefert die Brille andere Werte im Vergleich zu IMU-Daten, die am Brustkorb aufgezeichnet werden. In Abbildung 2.3 ist die Google-Glass Brille zu sehen. Der Gyroskop- und Beschleunigungssensor ist am rechten Seite angebracht, ebenso wie eine Kamera, einen Magnetometer, einen Lightsensor und ein Mikrofon. Obwohl die Google-Glass Brille nicht entwickelt wurde, um physiologische Daten zu sammeln, kann sie dafür verwendet werden, da alle nötigen Sensoren darin verbaut sind. 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.

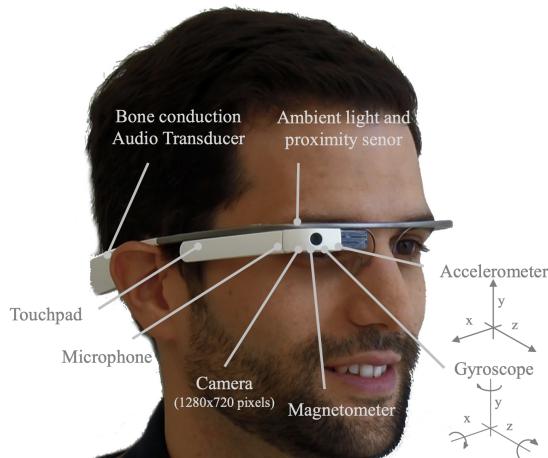


Abbildung 2.3: Darstellung der Google-Glass Brille: Die Brille hat einen Magnetometer, Gyroskop- und Beschleunigungssensor, eine Kamera, ein Mikrofon und einen Lichtsensor verbaut. Diese Sensoren werden genutzt, um die Atmung und den Puls zu ermitteln. [7]

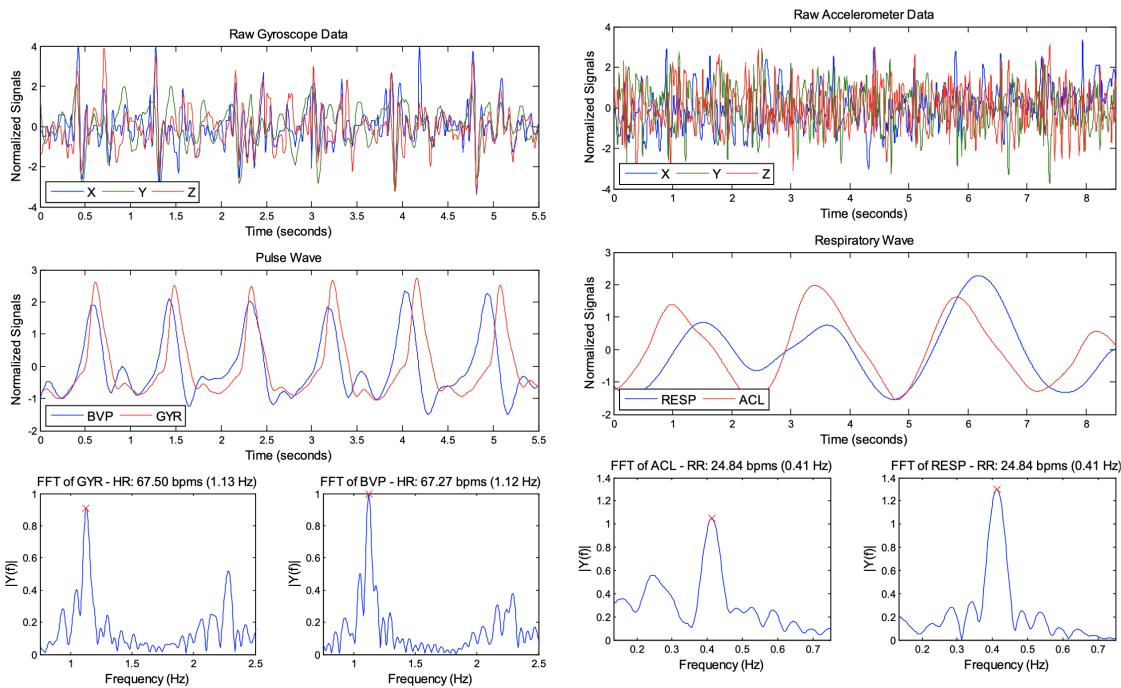
Die Herausforderung lag zudem daran, stromsparende Echtzeitberechnungen mit Algorithmen zu entwickeln, womit physiologische Parameter extrahiert werden. Der Nutzer soll das Gerät im Alltag normal weiternutzen können.

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

2.4.2.1 Pulssignal

Die Schätzung des Pulssignals wurde anhand einer Zeitreihe (*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 den Cut-Off Frequenzen von 10Hz und 13Hz wurde auf jede Dimension angewandt, um Veränderungen des BCG (*Ballistokardiogramm*) zu isolieren.
- Zudem wurde ein Bandpass Butterworthfilter der 2. Ordnung mit den Cut-Off Frequenzen von 0.75Hz und 2.5Hz (entspricht 45 and 150 Schläge pro Minute) angewandt, was schließlich das resultierende Pulssignal liefert.



(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-Truth-Signals (rot). Die beiden unteren Graphen zeigen das Fourierspektrum jedes Signal. (FFT: *Fourier Spectrum*, ACL: *accelerometer*, RESP: *Respiration from chest band*, RR: *respiration rate*, bpm: *breaths per minute*)

Abbildung 2.4: Herzrate- und Atemfrequenzanalyse

Die Abbildung 2.4a zeigt ein Signal des Herzschlags, gesammelt von den Informationen der Gyroskopdaten. Diese Daten wurden mit der Google-Glass Brille aufgezeichnet, während die Person auf dem Rücken lag. Der obige Graph zeigt ein 3-Achsen Gyroskopsignal 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 in mehreren Schritten 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 den 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 Größe ausgewählt, welche innerhalb des zu betrachtenden 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, einem Brustgurt, welcher als Ground Truth mit aufgezeichnet wurde.

2.4.3 Überwachung von Puls und Atmung mittels eSense-Earpods

Im Jahr 2019 wurde von *Tobias Röddiger, Daniel Wolfram und David Laubenstein* nachgewiesen, dass es möglich ist, mittels den eSense-Earpods die Atmung und den Puls annähernd zu ermitteln [14]. Dies gelang etwas genauer, als das Monitoring von *J. Hernandez* mit dem Google Glass [7]. Es wurde hierbei eine Studie mit 12 Personen aufgezeichnet, welche in 3 Positionen (liegend, stehend, sitzend) jeweils vor, beziehungsweise nach einer sportlichen Bewegungsphase einen einminütigen Atemablauf durchgeführt haben. Die Analyse der Daten erfolgte im Anschluss der Studie. Dabei wurden die Daten in einer Pipeline verarbeitet, welche zuerst das Rauschen reduzierte, anschließend einen Triangle-Filter der Breite 2s anwendete und danach eine PCA (engl. *principal component analysis*) ausführte, um die Daten unabhängig von deren Achse zu bewerten. Nun wurden Fenster mit der Größe von 20s extrahiert, in welche die Atmung und den Puls anhand dieses Fensters berechnen. Die Resultate ergaben einen mittleren absoluten Fehler (engl. *mean absolute error*) von 2.62 CPM (acc) und 2.55 CPM (gyro), jedoch variierten diese von Proband zu Proband.

Der Ablauf der Analyse begann damit, dass ein gleitendes Durchschnittsfenster (*sliding average window*) von 3 Samples von jeder Dimension abgezogen wurde. Zudem wurde auf jeder der Komponenten ein Mittelwertfilter angewandt. Diese haben eine Fenstergröße von 2 Sekunden. Das soll Signalverschiebungen und Signaltrends entfernen. Anschließend wurde eine kubische Splineinterpolation auf den Datensatz angewandt, um die Daten aufzublasen. Falls zuviel Bewegung in einem Fenster zu erkennen ist, wurde das Fenster verworfen [17]. Daraufhin wurde ein Bandpass-Butterworth-Filter der 4. Ordnung auf den Datensatz angewandt, um das Rauschen zu entfernen. Zum weiteren Glätten der Signale wurde ein Dreiecksfilter angewandt

[6], worauf schließlich eine Hauptkomponentenanalyse (PCA) durchgeführt wurde. Auf jede Hauptkomponente wurde nun eine Spektralanalyse mithilfe einer Fourier-Transformation (FFT) angewandt. Die Frequenz, welche dem Peak mit der höchsten Magnitude entspricht, resultiert schlussendlich als Atmungsfrequenz.

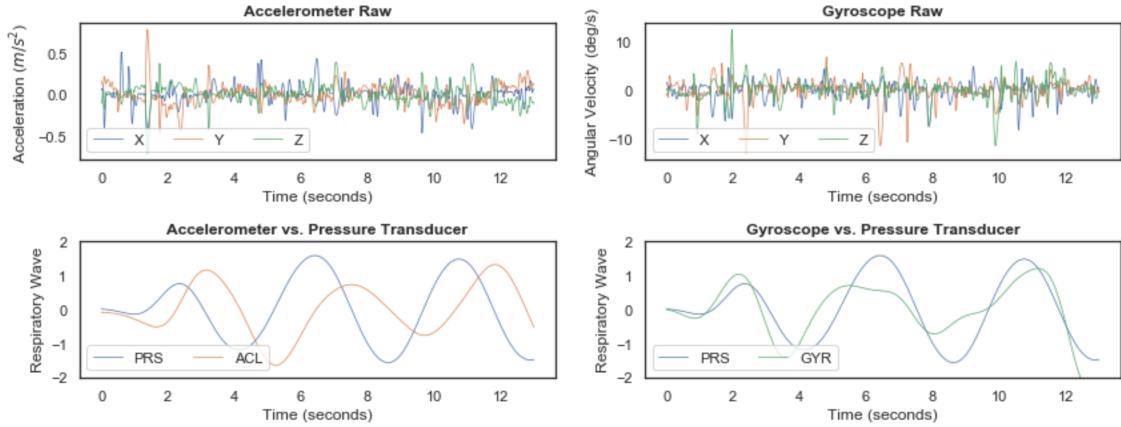


Abbildung 2.5: Rohdaten der Beschleunigungs- und Gyroskopdaten innerhalb von 12s, sowie die Atem- und Pulsschätzung, verglichen mit dem Ground-Truth (blau).

3. Schlafanalyse

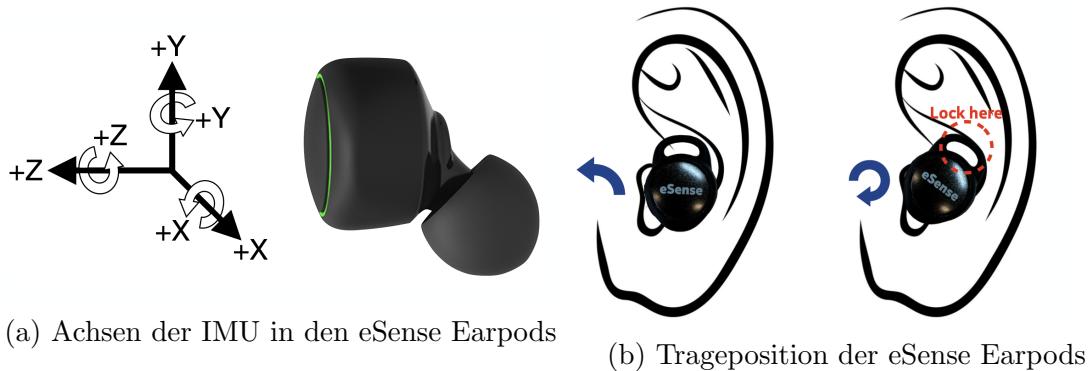
Zur Analyse eines respiratorischen Ereignisses findet im Zuge dieser Bachelorarbeit eine Nutzerstudie statt. Dazu wird ein Datensatz erstellt, welcher im Anschluss genauer betrachtet wird. Im Folgenden wird erläutert, welche Geräte verwendet werden und wie die gesammelten Informationen zusammengetragen 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 einen 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 ein 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, wurde aber nicht verändert. Dieser 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 wurde die Konfiguration ebenfalls auf den Standardwerten belassen, was bedeutet, dass ein Tiefpassfilter mit einer Bandbreite von 5Hz auf die Beschleunigungs- und Gyroskopdaten angewandt wird.

3.1.1 Was wird aufgezeichnet?

Zur vollständigen Aufzeichnung eines Datensatzes werden die Daten der IMU in einer Datenbank abgespeichert. Insgesamt werden hierbei pro empfangene Dateneinheit 6 Wertepersistiert, die x -, y - und z -Richtung des Beschleunigungs- und Gyroskopsensors. 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. Der Ton wird ebenfalls über das Mikrofon 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 in der App ausgefüllt und ebenfalls in der Datenbank gespeichert.

3.1.2 Datenexport

Zur weiteren Verarbeitung werden die Daten der aktuellen Messung, nachdem sie von der App lokal in einer Datenbank gespeichert werden, als *csv*-Datei exportiert und in einem temporären Ordner abgespeichert. Hierbei werden die Gyroskop- und Beschleunigungsdaten separat exportiert und es entstehen 2 *csv*-Dateien („*GyroData_ID.csv*“, „*ACCDATA_ID.csv*“). Das Mikrofon-Signal wird nach der Messung als *m4a*-Datei im selben temporären Ordner abgelegt. Die Zusatzinformationen, welche über den Studienteilnehmer hinterlegt wurden, werden als *csv*-Datei („*UserStudyPersonDetails_ID.csv*“) ebenfalls im gleichen 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 obstruktiver und zentraler 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“.

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. In der Abbildung 6.1 ist eine Person zu sehen, welche alle für die Studie



Abbildung 3.2: PSG-System im Einsatz und visuelle Darstellung **TODO: mehr beschreibung** [15]

relevanten Sensoren trägt. Im Folgenden werden alle Sensoren aufgelistet, welche für die Studie aufgezeichnet wurden und somit die Montage darstellen.

- **Bauchsignal (abdomen)** (32Hz): Gibt die Veränderung des Volumens am Bauch an.
- **Lichtsignal (light)** (128Hz): Misst das 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): Es misst die Bewegung anhand von Beschleunigungsdaten.
- **Sauerstoffsättigung (pleth)** (128Hz): Sie gibt die Sauerstoffsättigung im Blut an.
- **Pulssignal (pulse)** (32Hz): Gibt den am Finger gemessenen Puls an.
- **Schnarchmikrofonsignal (schnarch)** (256Hz): Dieser Sensor wird am Kehlkopf befestigt und misst das Schnarchen.
- **Sauerstoffgehalt (SpO_2)** (32Hz): Dieser wird anhand einer manschettenfreien kontinuierlichen Blutdruckmessung ermittelt.
- **Brustkorbsignal (thorax)** (32Hz): Es gibt die Veränderung des Volumens am Brustkorb an.
- **Brustkorb & Bauchsignal (thoraxabdomen)** (32Hz): Kombiniertes Signal von Brust- und Bauchsignal
- **EDF Informationen (edfAnnotations)** (60Hz): Sie enthalten 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 ausgelernt werden. Die Software stellt eine Darstellung bereit, womit man die Signale untereinander in einer zeitlich synchronisierten Ansicht 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 wird auf Kapitel 5.2 verwiesen.

3.3 Kamera

Während der Studie wurde zur vollständigen Erstellung eines Datensatzes eine Kamera mit Stativ aufgestellt, welche den Kopf des Studienteilnehmers fokussiert. So mit können eventuelle Unklarheiten im Datensatz, wie zum Beispiel ein Husten der Person, genau überprüft 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 auszuräumen.

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 Aktionsanweisung, die der Studienteilnehmer erhält, aus. 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 die Teilnehmer gesammelt. Dies soll lediglich dazu dienen, spätere Unklarheiten im Datensatz erklären zu können. Es werden Informationen zum Körper der Personen abgefragt (Alter, Größe, Gewicht, Geschlecht, Schlafrhythmus), zum Earpodaufsatzt, zur Matratzenart sowie zu den Maßen des jeweiligen Ohrs.

4. Systementwurf

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, beziehungsweise 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 verstärken 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 pro Teilnehmer ein Datensatz an 3 verschiedenen Positionen aufgezeichnet, auf dem Bauch, dem Rücken, sowie auf der Seite liegend.

Eine Frage, die sich bei der Studie gestellt hat, war, wie ein Atemaussetzer „simuliert“ werden soll. Es wurde entschieden, dass die Studie eine zentrale 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, wurde der Teilnehmer dazu aufgerufen, über eine Zeitspanne von 10s, 20s und 30s die Luft anzuhalten. Nun muss ein geeigneter Ablauf gewählt werden, damit sich die Ereignisse nicht überschneiden. Schaubild 4.1 zeigt, dass eine Person circa die gleiche Zeit zur Regeneration benötigt, wie sie zuvor die Luft angehalten hat. Diese Zeit wurde mit etwas Puffer nun zusätzlich in der Studie mit eingebbracht und daraus ergibt sich der Ablauf, welcher in Abbildung 4.2 zu sehen ist.

4.2 Systemmodelle

Zur Datenaufzeichnung während der Studie wurde eine Smartphone-App für Apple-Geräte konzipiert und mit der Sprache Swift entwickelt. Mit der Software XCode

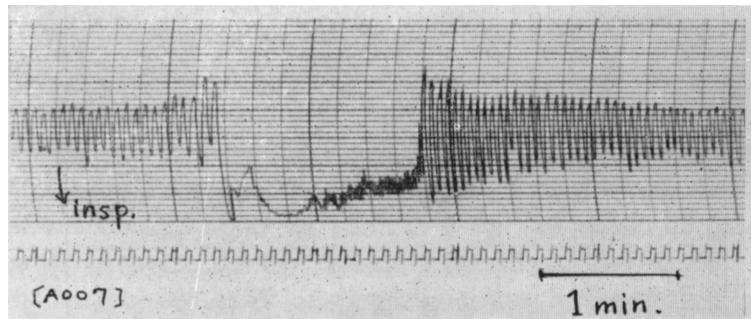


Abbildung 4.1: Regenerationsphase nach dem Luftanhalten [10].

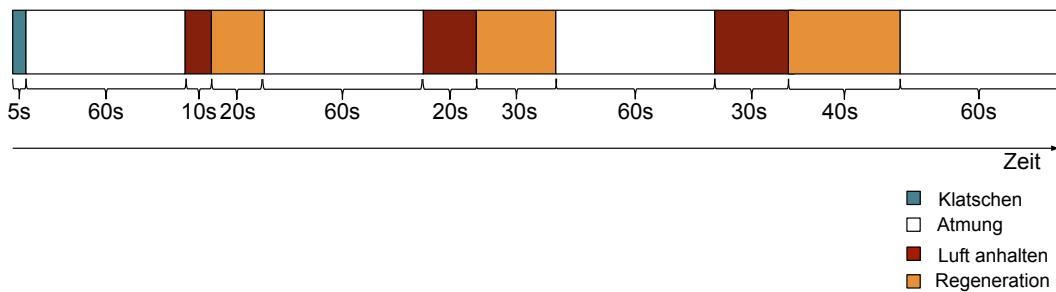


Abbildung 4.2: Ablauf der Studie mit einer Position

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 Protokoll, 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)

Das Framework *Mungohealer* stellt eine Möglichkeit dar, bessere Fehlerbenachrichtigungen erstellen zu können. Der Standard, welcher von Apple bereitgestellt wird (*NSError*), wird durch dieses Framework erweitert. Somit können Fehler in der Konsole genauer beschrieben werden. In dieser App wird dies unter anderem verwendet, um Benachrichtigungen der Bluetoothverbindung zu den eSense-Earpods zu erhalten.

In Abbildung 4.3 ist der Aufbau der App beschrieben. Die Grundlage ist eine Tab-bar, welche 3 Sektionen der App verfügbar macht, die *Chartansicht*, die *Einstellungsansicht* und die *Messungsansicht*. Der Messablauf wird detailliert in Kapitel 5.1.1 beschrieben.

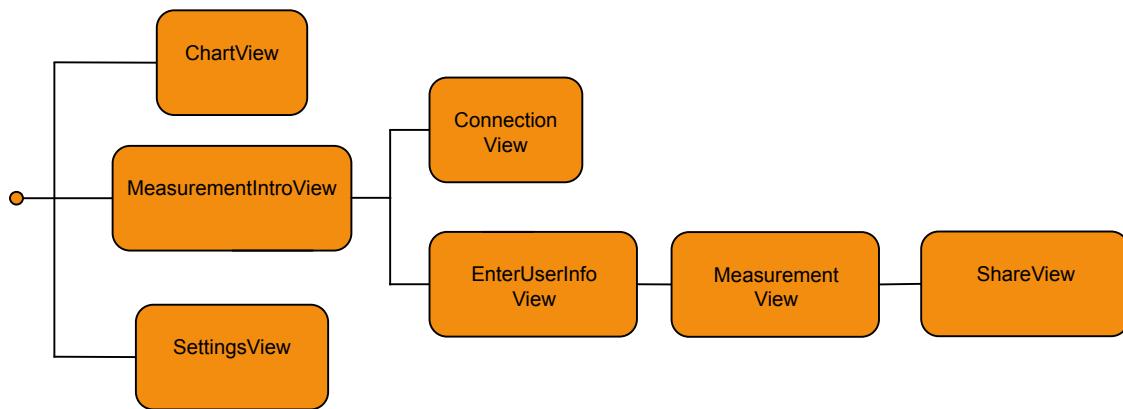


Abbildung 4.3: Ablauf der App-Screens: Nach dem Verbinden mit den eSense Earpods kann eine Messung gestartet werden, sofern die Nutzereingaben zuvor erfolgreich eingegeben wurden. Im Anschluss startet die Messung. Schlussendlich bietet sich in der ShareView die Möglichkeit, die Messung zu teilen.

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 gespeicherten Daten ermöglichen. Anschließend müssen diepersistierten Daten aufbereitet werden, sodass sie klassifiziert werden können.

5.1 App

Zur Erstellung des Datensatzes wurde eine App entwickelt. Diese startet eine Messung, nachdem eine Verbindung zu den eSense-Earpods hergestellt wurde. Jeder Messwert wird in der Datenbank abgespeichert. In Abbildung 5.1 ist das ER-Diagramm der Datenbank beschrieben. Das Gerät wird bei der ersten Kopplung mit dem Namen und dessen Identifikator in der Tabelle „Device“ abgespeichert. Der Identifikator ist hierbei eine Repräsentation der MAC-Adresse, welche unter Swift nicht erreichbar ist. Für jeden Messwert wird ein Eintrag in der Tabelle „EsenseData“ abgelegt sowie eine Referenz zu den x -, y - und z -Achsenwerten der Gyroskop- und Beschleunigungsdaten, welche in der entsprechenden Tabelle abgelegt werden. Zudem wird für jede Messung ein Zeitstempel (`timestamp`), eine Messungs-ID (`measurementID`), die aktuell auszuführende Aufgabe (`task`) und die Information, ob die LED des Smartphones an ist oder nicht (`lightOn`), abgespeichert.

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.

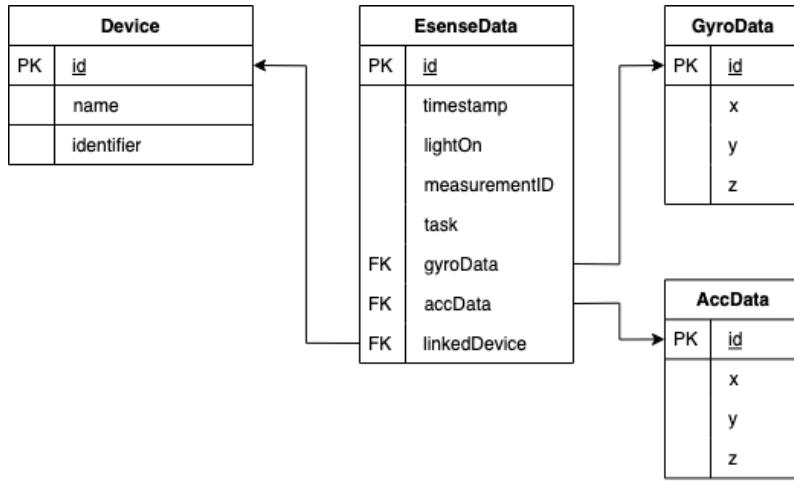


Abbildung 5.1: ER-Diagramm der App-Datenbank

5.1.1 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. 6.4b) zum Ausfüllen der Nutzerinformationen. Mit dem Betätigen des Buttons „Start Measurement“ bestätigt man die Eingabe der Daten und die Messung beginnt. Automatisch startet der erste Timer (Abb. 5.3c). Der Timer zeigt den aktuellen, den nächsten *Task* und die Restzeit des aktuellen Tasks an. 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.

5.1.2 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, beziehungsweise 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 wird 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 die Auswertungspipeline

Die App liefert beim Export die Daten der eSense-Earpods. Diese beinhalten die IMU-Daten, die Nutzerinformationen und die Mikrofonaufnahme. Zum aktuellen Stand liegen somit die Daten der eSense-Earpods und die des PSG-Systems 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. Je doch 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. 5.4a). Da eine Messung mit 2 Lichtblitzen beginnt und endet, kann nun der Start- und Endzeitpunkt einer Position ermittelt werden. Die einzelnen Positionen können dadurch voneinander 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 Ordnerstruktur kann der Abbildung 5.2 entnommen werden und ist nun vollständig.

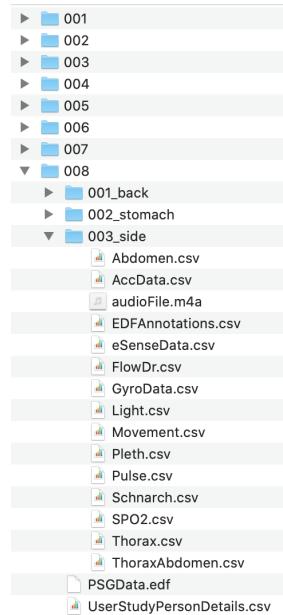


Abbildung 5.2: Ordnerstruktur des Datensatzes

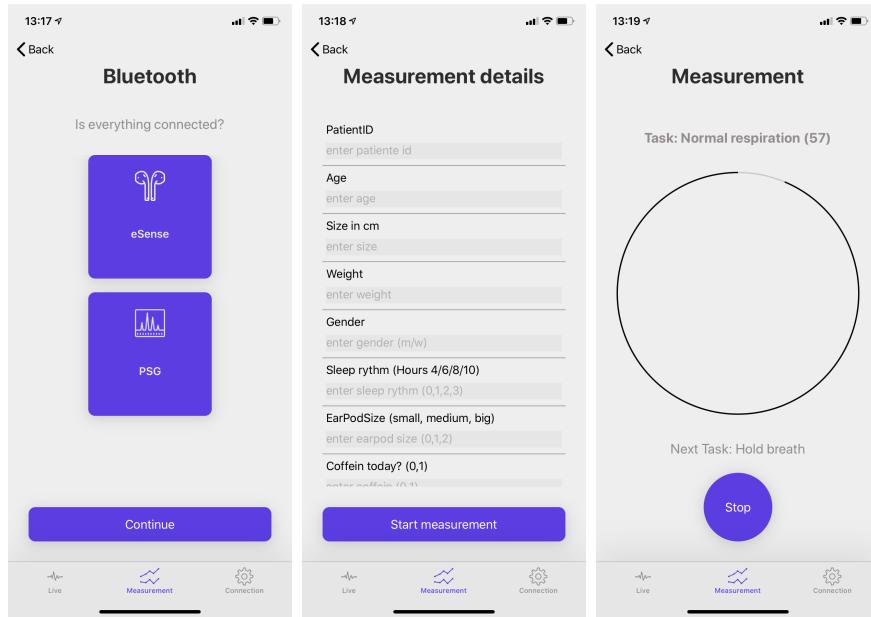
5.2.1 Synchronisation der Daten

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 ebenso wie die durchschnittliche Distanz der Peaks. In der Abbildung 5.4b ist der Vergleich eines Lichtblitzes von Smartphone (blau) und dem PSG-System (orange) im zeitlichen Verlauf dargestellt (`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 um die durchschnittliche Distanz der Peaks verschoben. Dies ist ausreichend, da die PSG-Daten in der Bachelorarbeit lediglich dem visuellen Vergleich dienen und in der Klassifikation nicht mit betrachtet werden. Die zeitliche Verschiebung wird nun in jede *csv*-Datei mit der Spalte `new_time` eingearbeitet. In Abbildung 5.4c ist klar zu erkennen, dass mit dem neuen Zeitwert (`new_time`) der Lichtwert des PSG-Signals (orange) direkt über dem Lichtwert des Smartphonesignals (blau) liegt. Nun ist garantiert, dass die Zeitwerte von PSG-System und dem Smartphone 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 grenzende 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.

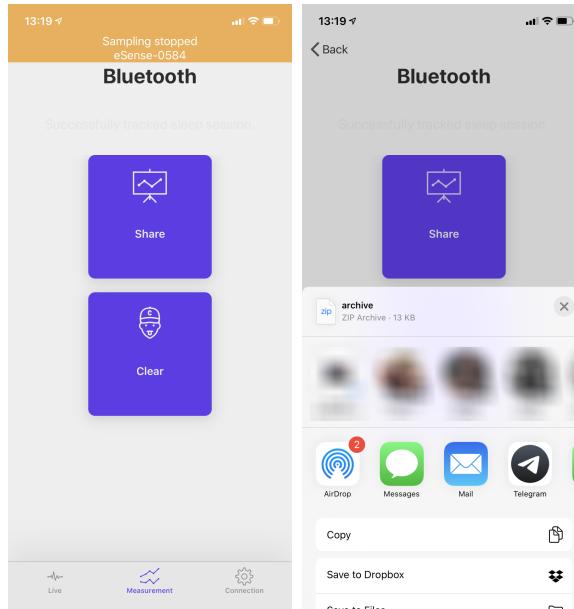
Da nun die Features für eine Fenstergröße von 5s und 10s mit der Verschiebung von 1s pro Fenster vorliegen, kann mit der Klassifikation begonnen werden.



(a) Verbinden eSense-Earpods

(b) Eingabe der Nutzerinformationen

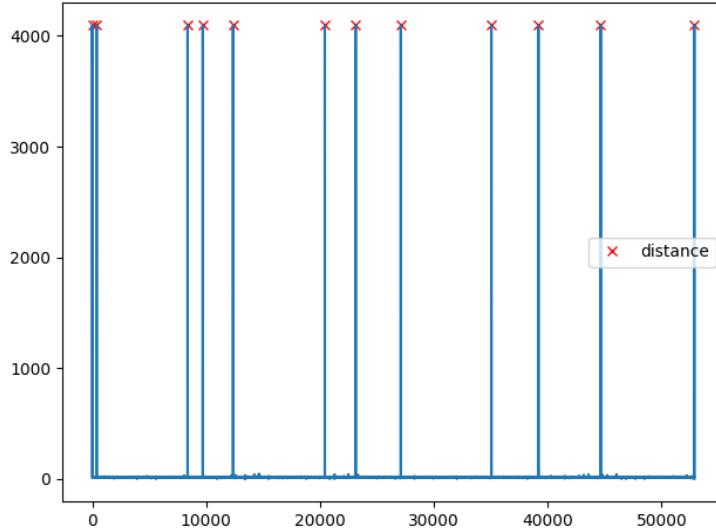
(c) Messung, aktueller und nächster Task



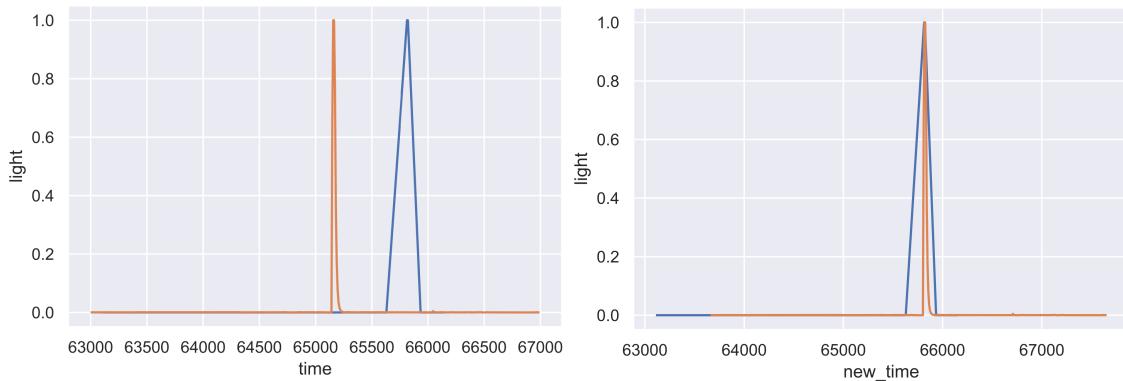
(d) Messung beendet

(e) Messung teilen

Abbildung 5.3: Verlauf einer Messung mit der App: Zu Beginn wird eine Verbindung mit den eSense-Earpods hergestellt. Anschließend werden die Nutzerinformationen des Probanden abgefragt. Mit dem Klick auf *Start Measurement* wird eine Messung gestartet. Der erste Timer läuft ab und über die Lautsprecher werden alle Information an den Probanden weitergegeben. Nach Ablauf des Timers beginnt der nächste Timer. Sobald der letzte Timer abgelaufen ist, kann die Messung geteilt werden. Zudem kann die Datenbank gelöscht werden und die nächste Messung kann beginnen. Der Ablauf einer Messung ist in Abbildung 4.2 nachzusehen.



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



(b) Zeitlicher Lichtpeakvergleich vor der Synchronisation. Der Lichtpeak des PSG-Geräts (orange) ist weniger als 1s vom Si-Systems (blau) entfernt. Der Zeitwert ist in ms angegeben. Die Lichtsignale wurden skaliert, um diese vergleichen zu können.
(c) Zeitlicher Lichtpeakvergleich nach der Synchronisation. Der Lichtpeak des PSG-Geräts (orange) ist nun zeitlich sehr nah an den eSense Earpods (blau) entfernt. Der Zeitwert ist in ms angegeben. Der neue Zeitwert ist in der Spalte 'new_time' jeder Datei zu entnehmen. Der Zeitwert ist in ms angegeben.

6. Evaluation

Zu Beginn der Evaluation wurde eine Nutzerstudie durchgeführt, womit ein Datensatz generiert wurde. Anhand dieses Datensatzes wird anschließend ein Klassifikationsmodell erstellt, womit eine Vorhersage darüber getroffen wird, an welcher Stelle ein Apnoeereignis stattgefunden hat.

6.1 Studienablauf

In Kapitel 4.1 wurde eine einzelne Messung einer Studie detailliert beschrieben. Es wurden nun 7 Probanden nach dem Prinzip des *Sample of Convenience* ausgewählt.

Dieser Ablauf (siehe Abb. 4.2) wurde nun pro Studienteilnehmer jeweils in den 3 Positionen durchgeführt, womit fast alle Schlaflagen abgedeckt sind. Zu Beginn der Studie fand eine kurze Einweisung statt, bei der 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. Nachdem die passenden eSense-Earpodaufsätze gefunden wurden, war der Aufbau der Studie beendet.

Nun wurde 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 der Ton durch das Mikrofon am eSense-EarPod mit aufgezeichnet wird, wird nach dem Start der Messung ein 4s Zeitfenster gewählt, in dem der Teilnehmer in die Hände klatschen muss, 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 angehalten 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 bereitgestellten Programms lässt sich die Messung als „*edf*-Datei“ exportieren. Weitere Informationen zum Export der Daten und zur Synchronisation sind in Kapitel 5.2 zu finden. 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 beträgt die durchschnittliche Gesamtdauer einer Aufzeichnung circa 45-60 Minuten.



Abbildung 6.1: Darstellung von Beispielprobanden der Nutzerstudie. Das PSG-System wurde mitsamt den Sensoren am Körper angebracht. Zudem wurde das Handy am PSG-System befestigt. Die Person ist nun bereit für die erste Messung.

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.

6.2 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 [18]. Diese Charakteristiken werden fortan als *Feature* bezeichnet.

Wie in Kapitel 5.3 bereits erklärt, wird eine Messung in viele sich überlappende Fenster aufgeteilt. Eine Messung wird in Fenster der Größe von 5s, bzw. 10s eingeteilt. Diese Fenster sind jeweils um 1s verschoben, d.h. die aneinander grenzenden Fenster überlappen sich um 4s, bzw. 9s. Nun wird für jedes Fenster eine Featureberechnung ausgeführt, womit durch `tsfresh` (~ 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, die mit einem Zeitstempel versehen sind. Die Abbildung 6.2 zeigt einen Ausschnitt einer Messung. Die in Blau eingefärbte Linie stellt den absoluten Wert aus den Gyroskopdaten der x , y und z -Achse dar. In Grün ist das Signal des Drucksensors, direkt unter der Nase befestigt, zu sehen, welches die Atmung repräsentiert. Der orange markierte Bereich ist jener Bereich, in dem der Proband die Luft angehalten hat. Man erkennt bereits, dass die Signale während der Phase des Luftanhaltens wenig Ausschlag in einem Bereich haben. Wenn der Proband nicht die Luft anhält, erkennt man deutliche Bewegungen der Gyroskopdaten.

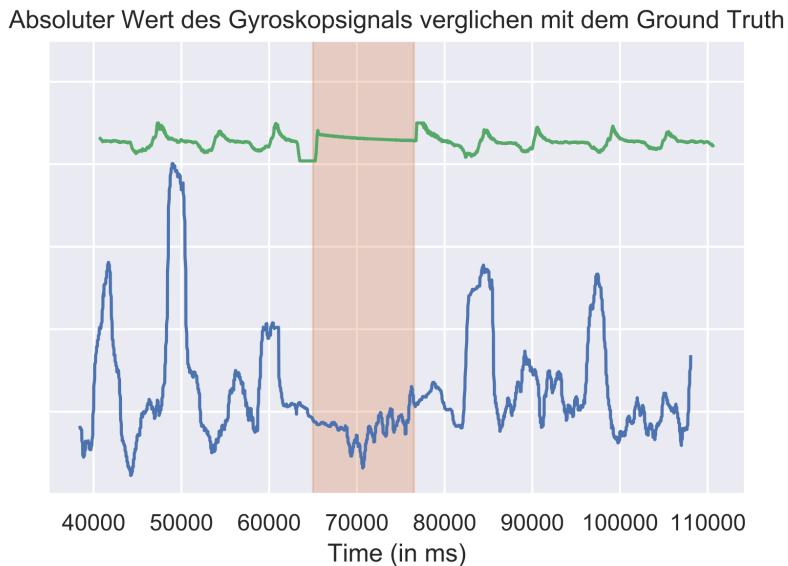


Abbildung 6.2: Darstellung des Signals eines Probanden während einer Messung. In Grün ist das Signal des Drucksensors vom PSG-Gerät zu sehen, welcher an der Nase befestigt ist. Zudem ist der absolute Wert, welcher sich aus allen 3 Achsen des Gyroskopsignals berechnen lässt, in blau und das Apnoeereignis in orange zu sehen. Circa zwischen Sekunde 65 und 75 wurde ein Apnoeereignis simuliert. Es ist deutlich zu sehen, dass ein Apnoeereignis auch anhand der IMU-Werte zu erkennen ist. Die Berechnung des absoluten Werts ist $\sqrt{x^2 + y^2 + z^2}$. Beide Signale wurden normalisiert und so in der y-Achse verschoben, sodass sie vergleichbar sind.

6.3 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, bzw. 5 und 10 Sekunden verschoben ist.

6.3.1 Data Labeling

Bei Supervised Learning mit 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 und wann 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 nicht Teil der Featureberechnung sein. Bei einem 5 Sekunden Intervall werden circa 250 Einträge, bei einem 10 Sekunden Intervall circa 500 Einträge in eine Featureberechnung zusammenfließen, weshalb eine Markierung dieses Fensters gewählt werden muss. Ist mehr als die Hälfte der Einträge positiv markiert, gilt das ganze Intervall als positiv markiert. Dies ist eine essenzielle Entscheidung, da beim Training des Modells nun anhand dieser Markierung entschieden wird, ob ein Intervall einen Atemaussetzer repräsentiert, oder nicht. Mit einer Grenze von 50% wurde eine gleichwertige Aufteilung gewählt. Nun können die Resultate mit den Klassifikatoren verglichen werden.

6.3.2 Kreuzvalidierungsverfahren

6.3.2.1 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 [11]. Somit werden alle Personen zu einem 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 die mittlere Genauigkeit (*mean accuracy*) auf den gegebenen Testdaten, welche 30% des Datensatzes sind, angibt.

Die Resultate sind in Abbildung 6.1 zu sehen. Die Tabelle zeigt die Ergebnisse verschiedener Klassifikationsalgorithmen mit verschiedenen Datensätzen. Pro Klassifikationsalgorithmus wurden alle Positionen, sowie 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 positiv markiert wurden. Dadurch ist der überlappende Teil eventuell in den Trainingsdaten vorhanden und eine Klassifizierung ist einfach für den Klassifikationsalgorithmus.

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.

6.3.2.2 Leave One Subject Out (LOSO)

Neben dem k-fachen Kreuzvalidierungsverfahren wird das *Leave One Subject Out (LOSO)* Kreuzvalidierungsverfahren hinzugezogen. 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 vorhergesagt, welcher diese Person nicht enthält. Somit entstehen 7 Resultate, von denen nun der Mittelwert gebildet wurde. Die Resultate sind in den Abbildungen 6.3 und 6.4 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 auf. 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. Alle 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. Des Weiteren, um mögliche Positionsabhängigkeiten zu erkennen, wurde dasselbe erneut evaluiert, jedoch nur mit den einzelnen Positionsdaten der jeweiligen Person. In der Tabelle 6.2 sind die Resultate zu sehen.

6.3.3 Relevante Features

Jede Klassifikation ergab ein Resultat, welches durch einzelne Features entschieden wurde. Das Python-Framework `scikit-learn` beinhaltet `feature_importances_` und 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 [18].

Die partielle Autokorrelationfunktion (`partial autocorrelation`) beschreibt die Abhängigkeit 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 weniger verändert und ähneln somit eher den positiv markierten Trainingsdaten.

Die *Schnelle Fourier-Transformation* berechnet effizient die diskrete Fourier-Transformation. Damit können zeitdiskrete Signale in Frequenzanteile zerlegt werden. Es wird aus einer Darstellung, bestehend aus einem Zeit- und Abtastwert in einer Darstellung, bestehend aus einem Frequenzanteil, einer Amplitude und einer Phase überführt. Die Funktion `fft_coefficient` in `tsfresh` berechnet die Fourier Koeffizienten einer eindimensionalen diskreten Fouriertransformation für 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 ermöglicht wird.

Die Aggregation `agg_autocorrelation` berechnet den Wert der Autokorrelationsfunktion f_{agg} (beispielsweise die Varianz oder den Mittelwert) über die Autokorrelation $R(l)$ für 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 Zeitreihe (*Timeseries*), n deren Länge. σ und μ sind Schätzwerte für die Varianz und den Mittelwert.

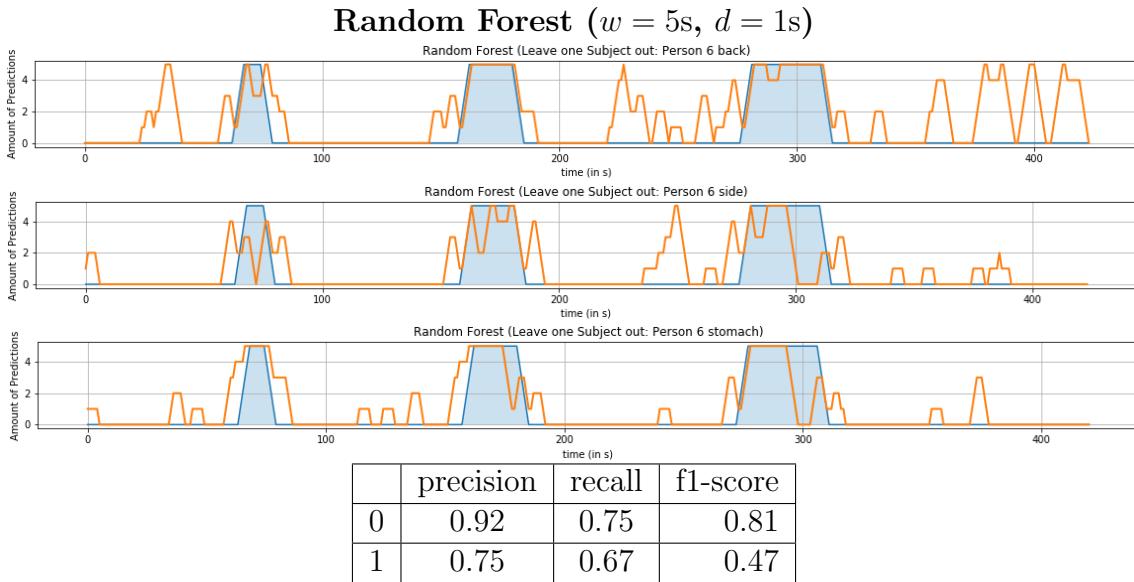
Ein weiteres Feature, welches ausschlaggebend für die Klassifikation ist, ist der Wechsel der Quantilwerte (`change_quantiles`). Pro Fenster wird ein q_l niedriges Quantil und ein q_h hohes Quantil berechnet. Anschließend wird der durchschnittliche, absolute Wert der aufeinanderfolgenden Änderungen der Reihe x innerhalb dieser Quantile berechnet. Dadurch werden starke Veränderungen deutlich und zeigen, dass in diesem Fenster kein Apnoe stattfindet.

6.3.4 Vorverarbeitung durch einen Tiefpassfilter

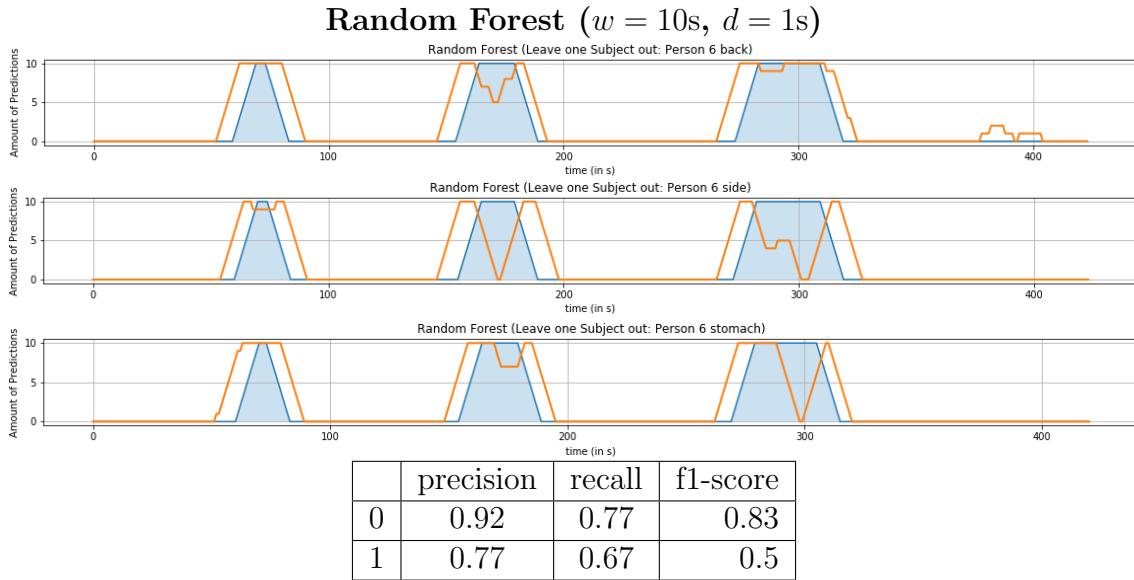
Als weiterer Versuch wurde ein Tiefpassfilter auf die Gyroskop- und Beschleunigungsdaten angewandt. Im Anschluss wurden Features mittels `tsfresh` auf jedem Fenster berechnet. Da `tsfresh` ebenfalls Tiefpassfilter zur Featureberechnung nutzt, ergaben sich mit *Random Forest* und *XGBoost* schlechtere Resultate. Dies ist darauf zurückzuführen, dass durch den Tiefpassfilter Informationen verloren gehen, welche in der Featureberechnung anschließend nicht mehr zur Verfügung stehen.

Klassifikation	Fenstergröße	Position		precision	recall	f1-score
Random Forest	5s	Seite	0	0.86	0.76	0.81
			1	0.76	0.57	0.48
		Rücken	0	0.90	0.79	0.83
		Bauch	1	0.83	0.52	0.39
			0	0.92	0.72	0.70
			1	0.72	0.69	0.52
XG Boost	5s	Seite	0	0.91	0.67	0.71
			1	0.67	0.61	0.35
		Rücken	0	0.90	0.74	0.80
		Bauch	1	0.75	0.57	0.38
			0	0.94	0.71	0.74
			1	0.71	0.72	0.55
Random Forest	10s	Seite	0	0.89	0.77	0.83
			1	0.80	0.67	0.5
		Rücken	0	0.89	0.83	0.85
		Bauch	1	0.82	0.52	0.41
			0	0.93	0.71	0.74
			1	0.71	0.70	0.49
XGBoost	10s	Seite	0	0.90	0.78	0.78
			1	0.78	0.71	0.47
		Rücken	0	0.90	0.85	0.86
		Bauch	1	0.85	0.50	0.44
			0	0.94	0.72	0.75
			1	0.72	0.70	0.53

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

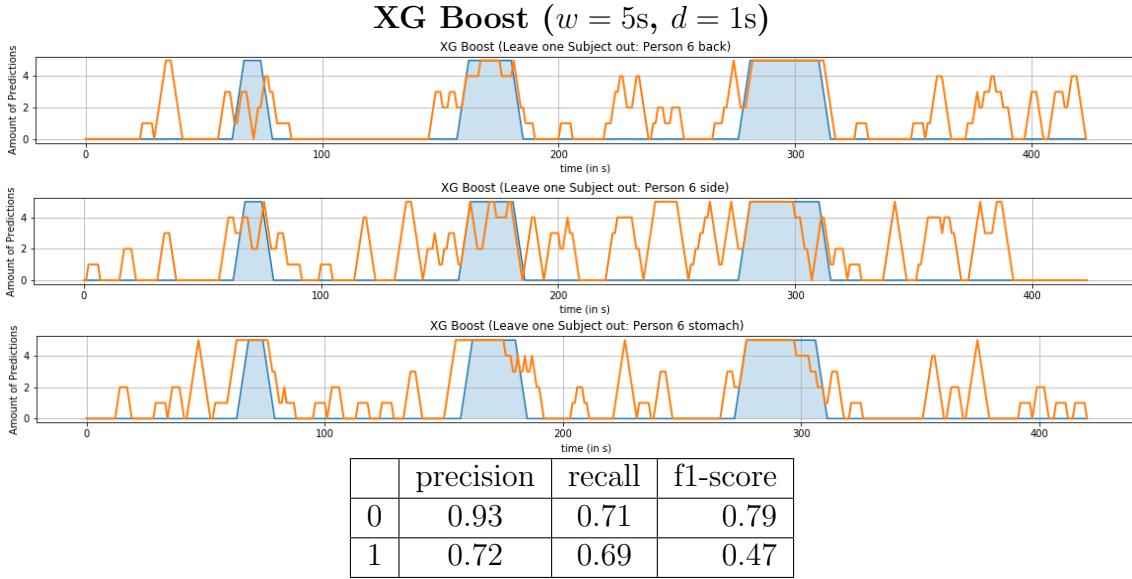


(a) Random Forest mit dem Kreuzvalidierungsverfahren: Die Tabelle zeigt den Mittelwert aller Vorhersagen der einzelnen Personen.

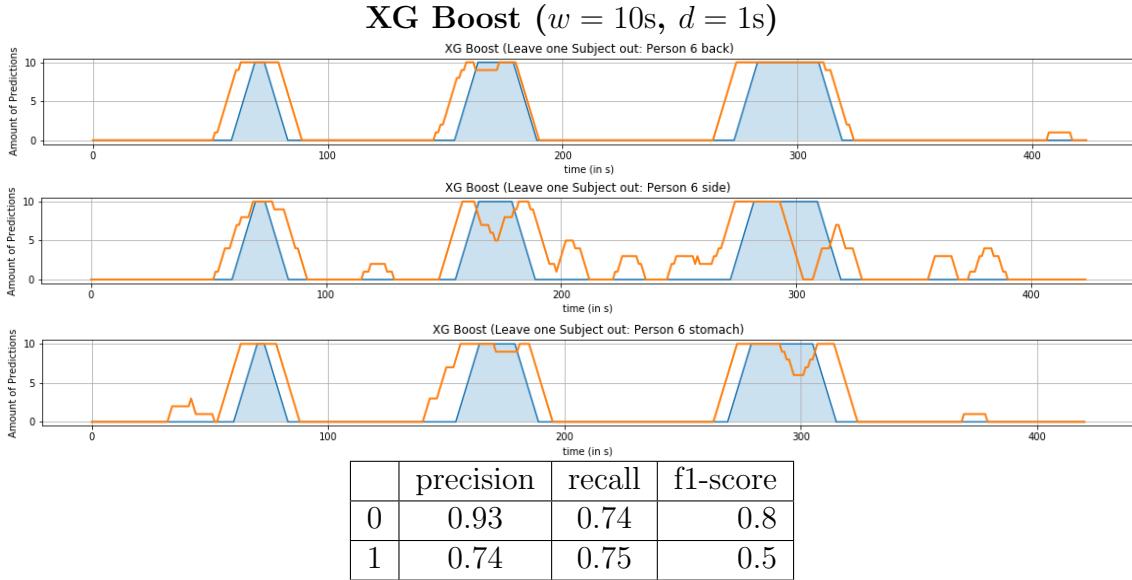


(b) Random Forest mit dem Kreuzvalidierungsverfahren (LOSO): Die Tabelle zeigt den Mittelwert aller Vorhersagen der einzelnen Personen.

Abbildung 6.3: Das Kreuzvalidierungsverfahren (LOSO) mit dem Klassifikationsalgorithmus Random Forest. Das Modell wurde auf allen Personen, exklusive einer Person trainiert. 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 orange Kurve zeigt die Vorhersage ($w = \text{Fenstergröße}$, $d = \text{Verschiebung der Fenster}$).



(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 6.4: Das Kreuzvalidierungsverfahren (LOSO) mit dem Klassifikationsalgorithmus XG Boost: Das Modell wurde auf allen Personen, exklusive einer Person trainiert. 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 orangefarbene Kurve zeigt die Vorhersage ($w =$ Fenstergröße, $d =$ Verschiebung der Fenster).

7. Diskussion & 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.

Ein Problem liegt möglicherweise daran, ab welcher Grenze ein Fenster als positiv markiert gilt. Es wurde ein Fenster als positiv markiert deklariert, falls mehr als 50% der Werte in diesem Fenster positiv markiert sind. Somit wurde das Modell auch auf Fenster trainiert, welche knapp über 50% positiv markierte Daten beinhalten. Das betrifft die Regionen eines Übergangs zu einem Apnoeereignis oder von einem solchen weg. Eine Lösung wäre es, diese Fenster, welche den Übergang in ein Apnoeereignis beinhalten, zu verwerfen, wenn ein Modell trainiert wird. Das wurde in dieser Bachelorarbeit nicht getan, da dies auch bedeutet, dass die Anzahl an positiv markierten Fenstern kleiner wird, worauf das Modell trainiert wird.

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 einer Schlafapnoe liefern. Mittels des Tonsignals direkt am Ohr können unter anderem die Atmung und auch den Puls herausgefiltert werden [12]. Diese Informationen können in die Auswertung mit einfließen und somit können 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 einfließen lassen. Somit wird die Fehlerquote verringert, da potenzielle Outlier vermieden werden.

Zudem gibt es die Möglichkeit, die weiteren 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 der Abbildung 7.1 ist eine Beispieldarstellung einer Messung eines Probanden zu sehen. In Blau sind die Phasen zu sehen, in denen der Proband die Luft angehalten hat. Hier ist der Wert auf 1 gesetzt, wenn die Person die Luft angehalten hat, ansonsten ist der Wert auf 0. In Grün ist das SpO_2 Signal zu sehen, in Orange das Pulssignal. Das SpO_2 Signal zeigt einen Abstieg nach dem Luftanhalt, ebenso steigt der Puls nach dem Lufanhalt. Wenn dieses Signal mit in die Features mit aufgenommen wird, könnte ebenfalls die Genauigkeit der Klassifikation steigen.

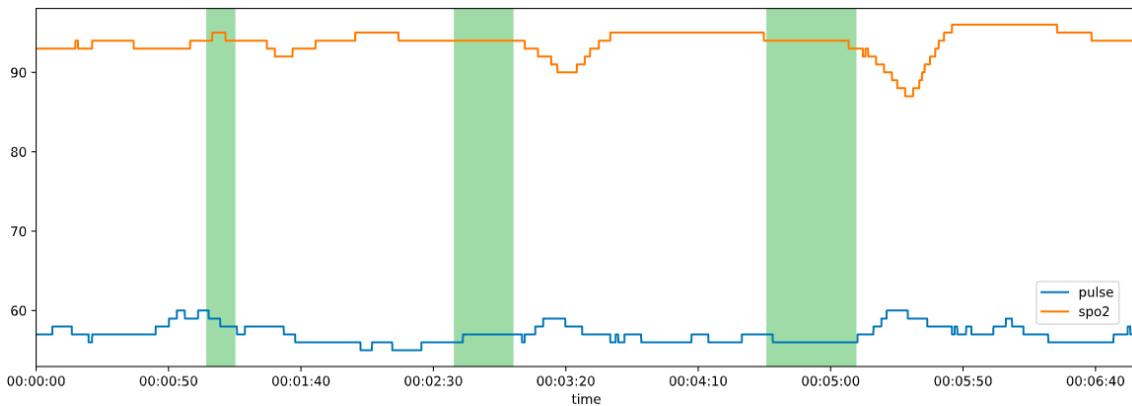


Abbildung 7.1: Darstellung des Puls und SpO_2 Signals einer Messung an einem Probanden. Beide Signale wurden aus den IMU-Daten der eSense-Earpods herausgefiltert. Das SpO_2 Signal zeigt Einbrüche an genau den 3 Stellen, an denen der Proband die Luft angehalten hat. Der Puls ist ebenfalls minimal gestiegen in diesen Bereichen. In grün sind die Bereiche markiert, in denen der Proband die Luft angehalten hat.

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 Schlafrythmusses in einem Modell trainiert werden. Dies könnte mögliche Einflussfaktoren eliminieren und das Ergebnis ebenfalls verbessern.

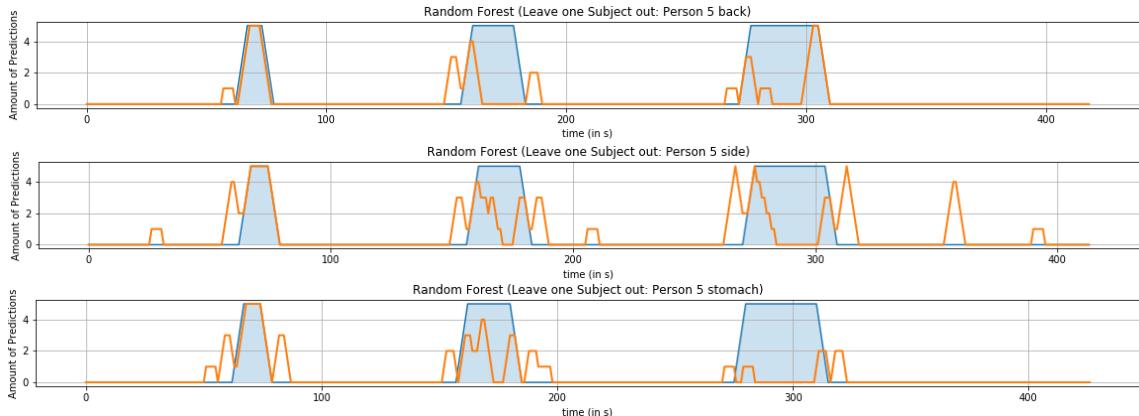
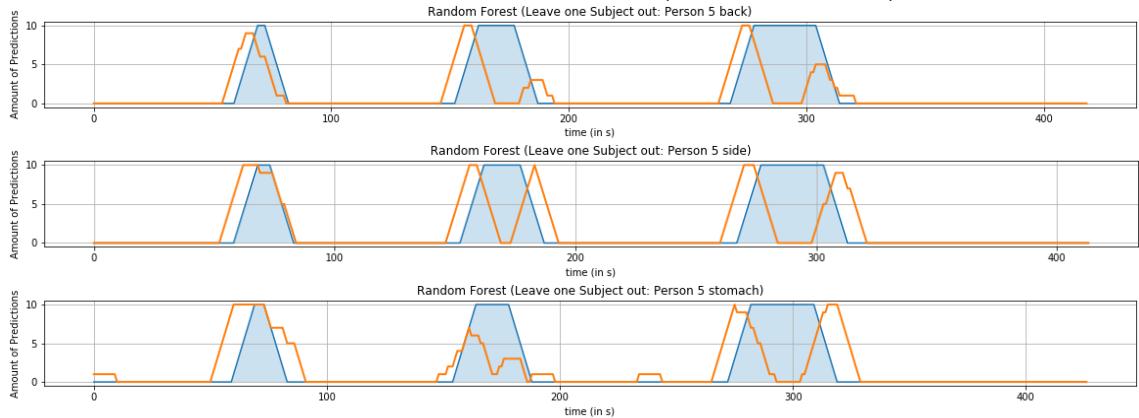
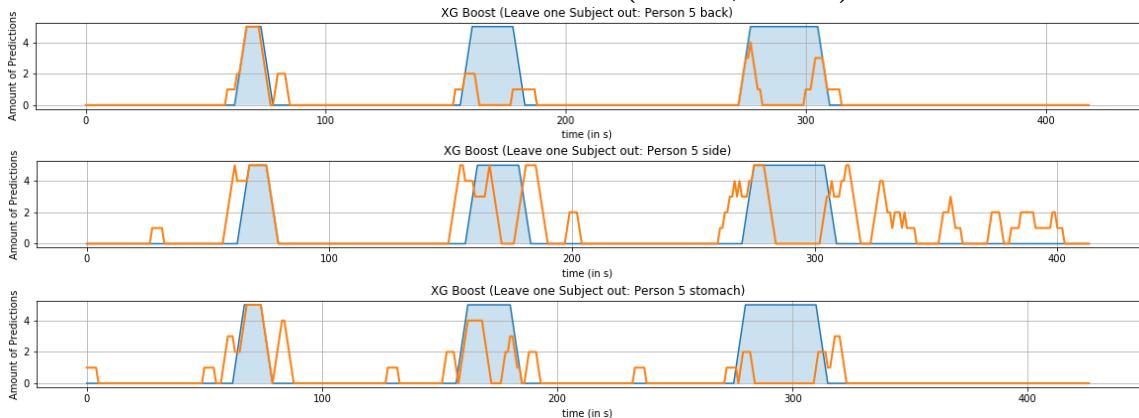
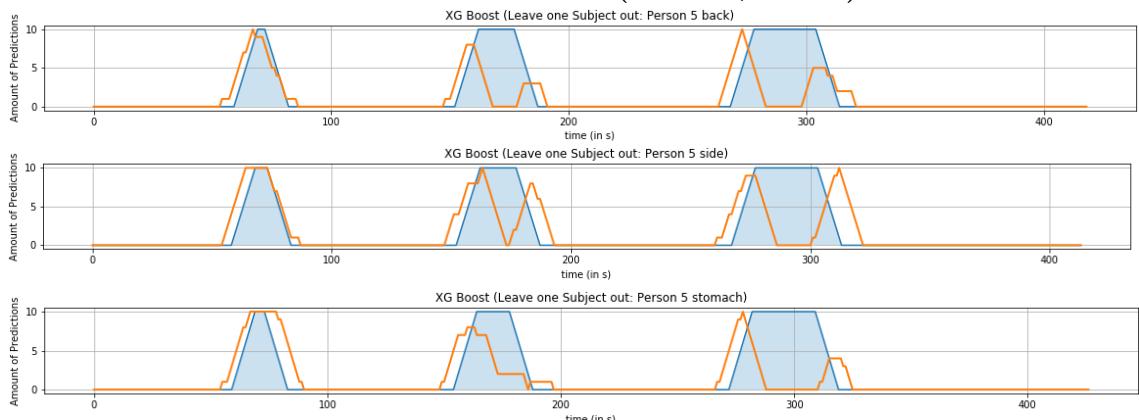
Ein weiterer Ansatz zur Weiterarbeit an diesem Thema wird in [13] behandelt. Dort wird der Ansatz, Convolutional Layer und Recurrent Layer zu kombinieren und daraus sinnvolle Features zu extrahieren. Zudem werden zeitliche Beziehungen ermittelt, was jedoch schwer zu ermitteln ist. Das Paper veröffentlicht *U-Time*, einen zeichlichen, vollständiges Deep Learning Ansatz zur Analyse von physiologischen Zeitreihensegmenten. *U-Time* wurde explizit zur Analyse von Schlafdaten entwickelt. Jeder Zeitpunkt des Eingangssignals wird implizit klassifiziert, anschließend wird durch Aggregation dieser Klassifikation über feste Intervalle eine endgültigen Vorhersage gebildet.

8. Anhang

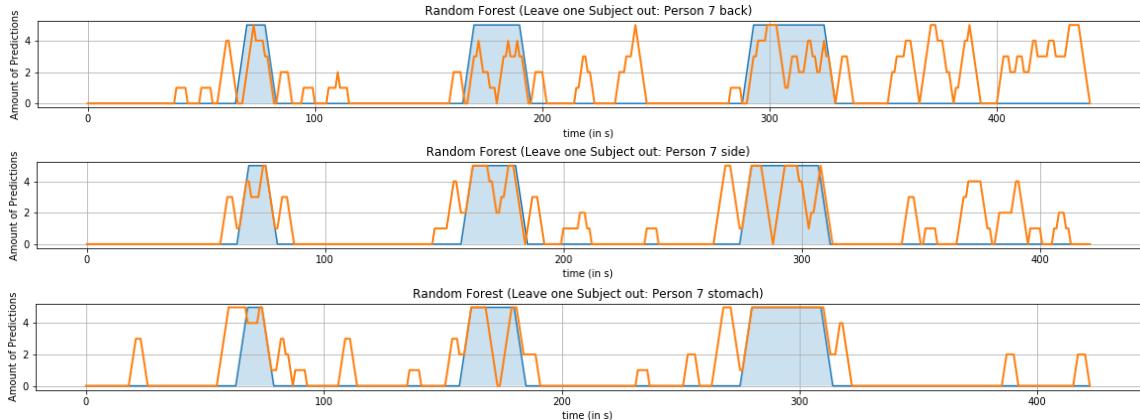
Im Anhang sind weitere Resultate des Kreuzvalidierungsverfahrens „Leave One Subject Out“ aufgelistet. Es wurde zuerst die Resultate der Person 5, 7 und 8 aufgelistet, wobei hier alle Positionen im Modell mittrainiert wurden und eine Vorhersage auf jede Position getroffen wurde. In orange sind die Vorhersagen der Fenster für jede Sekunde aufaddiert, in blau sind die Markierungen der Fenster aufaddiert.

Es ist erkennbar, dass bei Person 5 kaum Bereiche als positiv markiert vorhergesagt wurden, in denen kein Apnoeereignis simuliert wurde. jedoch wurden nicht alle Fenster der Apnoeereignisse als solche klassifiziert.

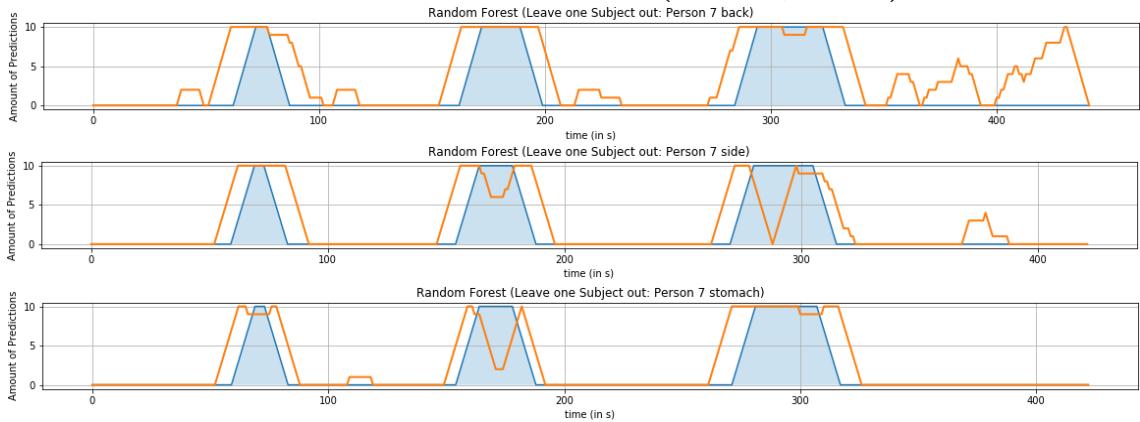
Bei Person 7 ist klar zu erkennen, dass eine Fenstergröße von 10 Sekunden deutlich bessere Resultate liefert im Vergleich zu einer Fenstergröße von 5 Sekunden. Hier ist die Variation der Vorhersage von Proband zu Proband gut zu sehen. Person 8 zeigt jedoch auch, dass die Ergebnisse bei einer Fenstergröße von 10s auch nicht immer gut sind.

Person 5: Random Forest ($w = 5s, d = 1s$)**Person 5: Random Forest ($w = 10s, d = 1s$)****Person 5: XG Boost ($w = 5s, d = 1s$)****Person 5: XG Boost ($w = 10s, d = 1s$)**

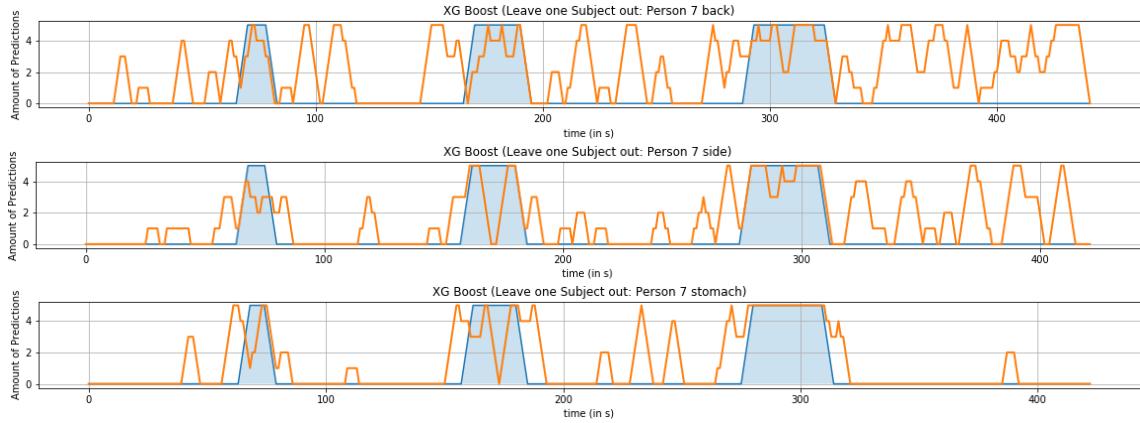
Person 7: Random Forest ($w = 5s, d = 1s$)



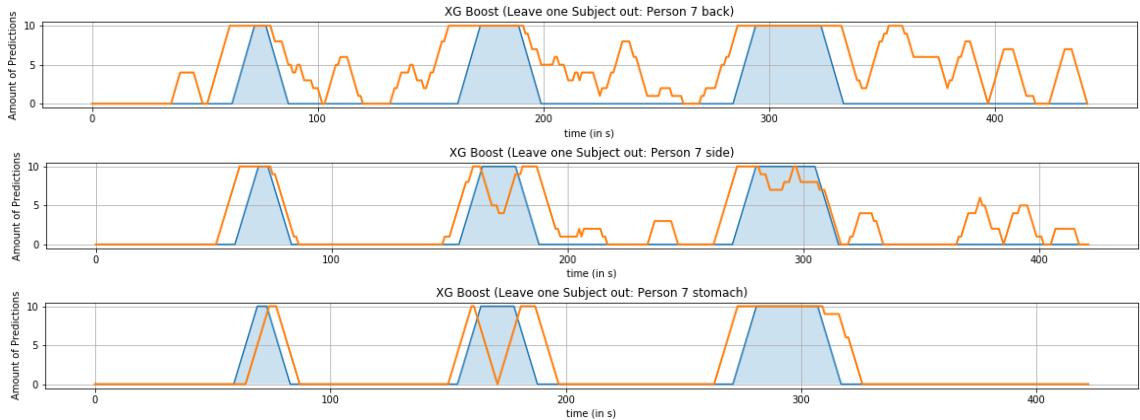
Person 7: Random Forest ($w = 10s, d = 1s$)

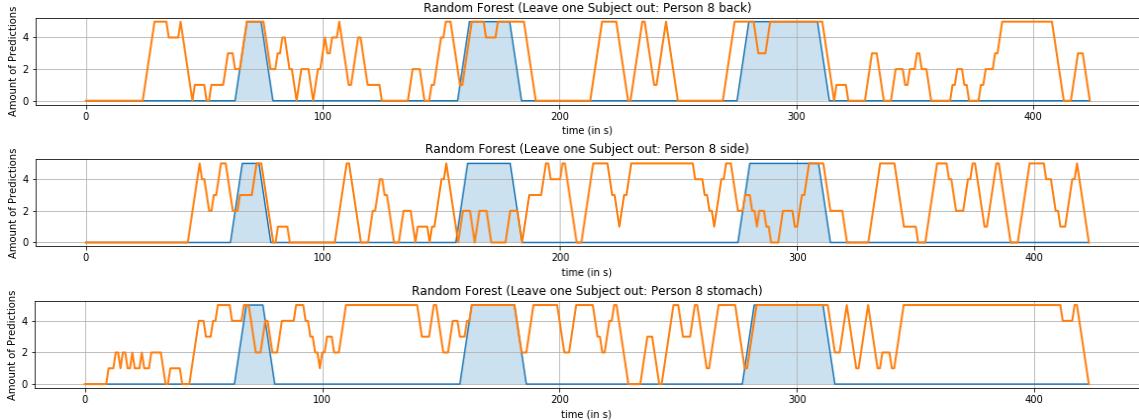
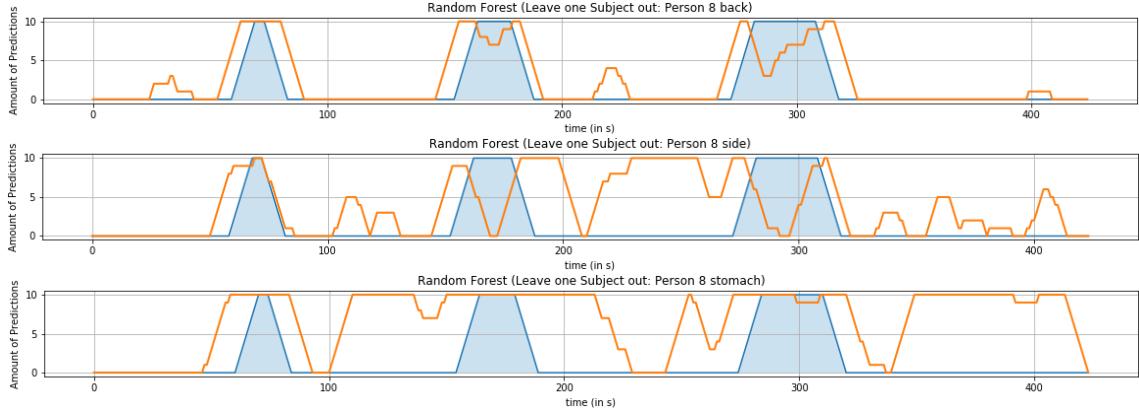
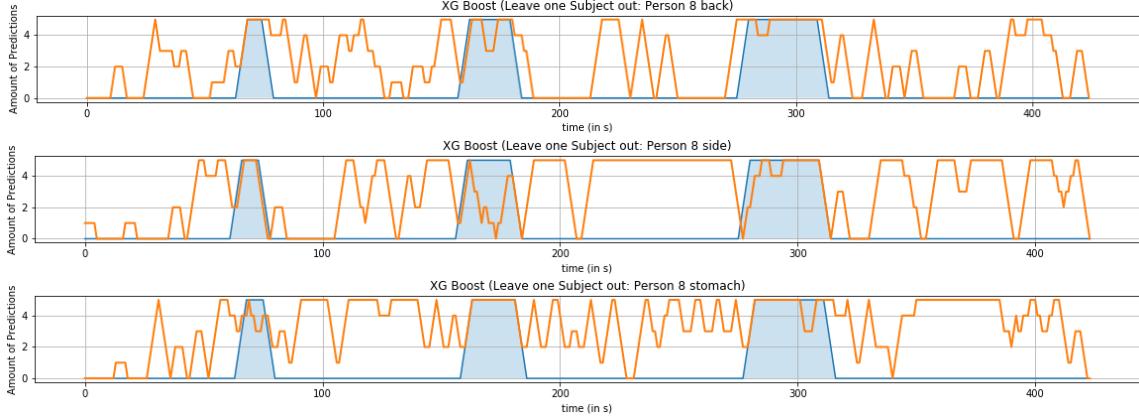
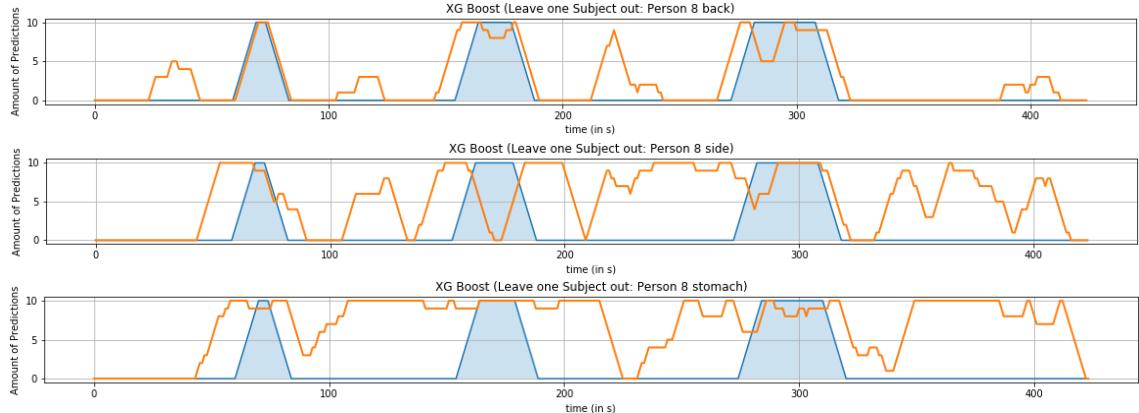


Person 7: XG Boost ($w = 5s, d = 1s$)

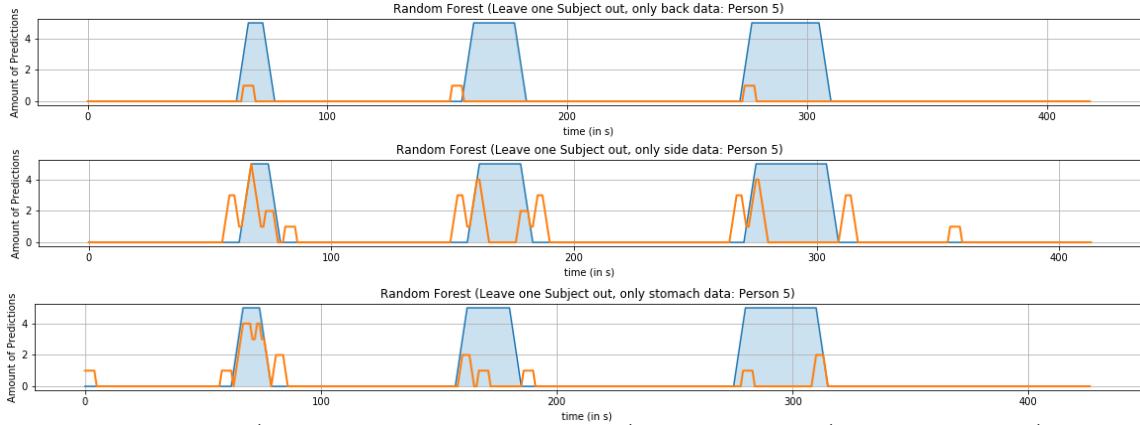


Person 7: XG Boost ($w = 10s, d = 1s$)



Person 8: Random Forest ($w = 5s, d = 1s$)**Person 8: Random Forest ($w = 10s, d = 1s$)****Person 8: XG Boost ($w = 5s, d = 1s$)****Person 8: XG Boost ($w = 10s, d = 1s$)**

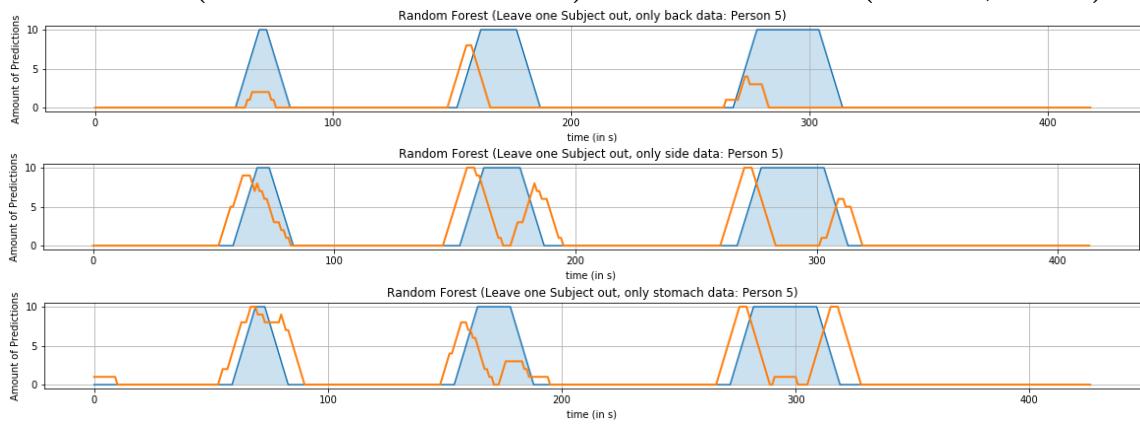
Person 5 (nur einzelne Positionen): Random Forest ($w = 5s, d = 1s$)



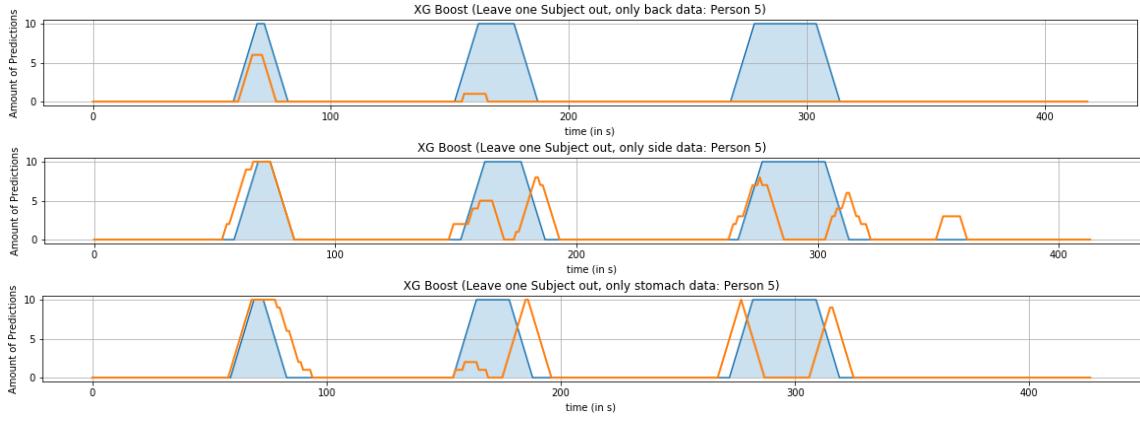
Person 5 (nur einzelne Positionen): XGBoost ($w = 5s, d = 1s$)



Person 5 (nur einzelne Positionen): Random Forest ($w = 10s, d = 1s$)



Person 5 (nur einzelne Positionen): XGBoost ($w = 10s, d = 1s$)



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] *Deutsche Familienversicherung - Das Schlafapnoesyndrom*. URL: <https://www.deutsche-familienversicherung.de/ratgeber/artikel/das-%20schlafapnoesyndrom/> (besucht am 01.11.2020).
- [5] Jerome H. Friedman. *Stochastic Gradient Boosting*. 1999.
- [6] HaoTian u. a. „MindfulWatch“. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (11. Sep. 2017). PUB27 New York, NY, USA. URL: <https://dl.acm.org/doi/abs/10.1145/3130922> (besucht am 05.04.2020).
- [7] J Hernandez u. a. „Cardiac and Respiratory Parameter Estimation Using Head-Mounted Motion-Sensitive Sensors“. In: (), S. 10.
- [8] 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).
- [9] 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).
- [10] Syoiti KOBAYASI und Chieko SASAKI. *BREAKING POINT OF BREATH HOLDING AND TOLERANCE TIME IN REBREATHING*. 1967. URL: https://www.jstage.jst.go.jp/article/jjphysiol1950/17/1/17_1_43/_pdf.

- [11] Gerhard Neumann. *Maschine Learning KIT*. 2020.
- [12] 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.
- [13] 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).
- [14] 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).
- [15] *Somnomedics Somnoscreen Plus Broschüre*.
- [16] 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).
- [17] Xiao Sun u. a. „SleepMonitor: Monitoring Respiratory Rate and Body Position During Sleep Using Smartwatch“. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (11. Sep. 2017), S. 1–22. ISSN: 2474-9567, 2474-9567. DOI: 10.1145/3130969. URL: <https://dl.acm.org/doi/10.1145/3130969> (besucht am 05.04.2020).
- [18] *Tsfresh — Tsfresh 0.12.0 Documentation*. URL: <https://tsfresh.readthedocs.io/en/latest/> (besucht am 24.03.2020).
- [19] *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).