

Klassifikation von respiratorischen Ereignissen mit Earables und maschinellem Lernen

Bachelor Thesis
von

David Laubenstein

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

First Reviewer:
Second Reviewer:
Supervisor:

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

Project Period: 01/11/2019 – 01/02/2020

Contents

1	Introduction	1
2	Basics & Related Work	3
2.1	Respiratorische Ereignisse	3
2.2	Alternativen zur Aufzeichnung und Klassifizierung	3
2.3	Forschung von Klassifikation anhand von IMU-Daten	3
3	Schlafanalyse	5
3.1	Earable Plattform	5
3.1.1	Was wird aufgezeichnet?	5
3.1.2	Datenexport	6
3.2	Polysomnographie-Systeme	6
3.2.1	Datenexport	7
3.3	Kamera	7
3.4	Datensynchronisation	7
3.5	Zusatzinformationen der Nutzer	7
3.6	Maschinelle Lernverfahren	8
3.6.1	Datenaufbreitung für Klassifikation	8
4	Design	9
4.1	Studienplanung	9
4.2	Studienablauf	9
5	Implementierung	11
5.1	App	11
5.1.1	Plattform	11
5.1.2	Messungsablauf	12
5.1.3	Messung	12
5.2	Anbindung an Auswertungspipeline	13
5.2.1	Synchronisation der Daten	13
5.3	Verarbeitungspipeline zur Klassifikation	13
6	Evaluation	15
6.1	Vergleich verschiedener Klassifikationsverfahren	15
6.2	Gibt es passende Features?	15
6.3	Was kann bei respiratorischen Ereignissen klassifiziert werden, was nicht?	15
7	Zusammenfassung und Future Work	17

1. Introduction

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

2. Basics & Related Work

2.1 Respiratorische Ereignisse

Welche respiratorischen Ereignisse gibt es & wie unterscheiden sie sich (Apnoe, Hypopnoe, Hyperventilation, ...)?

2.2 Alternativen zur Aufzeichnung und Klassifizierung

Welche Alternativen gibt es zur Aufzeichnung und Klassifikation?

2.3 Forschung von Klassifikation anhand von IMU-Daten

Welche Forschung gibt es zur Klassifikation von respiratorischen Ereignissen mit IMUs

3. Schlafanalyse

3.1 Earable Plattform

Zur Erfassung der Daten werden eSense-Earpods der Firma “Nokia Bell Labs Cambridge” verwendet. Es ist ein Mikrofon und Lautsprecher verbaut, welche beide über Bluetooth angebunden werden können. Des weiteren ist das für diese Bachelorarbeit interessanteste Element, eine 6-Achsen IMU (Inertial Motion Unit) enthalten. Eine IMU ist eine inertielle Messeinheit, womit Gyroskop- und Beschleunigungsdaten aufgezeichnet und mittels BLE (Bluetooth Low Energy) auf das Smartphone übertragen werden können. Es handelt sich um einen 3-Achsen Beschleunigungssensor, sowie einen 3-Achsen Gyroskop. Die Messrate dieser Sensoren ist variabel einstellbar, wurde im folgenden auf 50Hz festgelegt.

TODO: Beschreibe noch die Filter, die auf die Daten angewandt werden per Default... steht in der Doku des eSense Kopfhörers

TODO: import picture of esense earpods

TODO: soll ich hier schreiben, dass die Kopfhörer noch nicht im Handel sind?

TODO: Welche Vor-/Nachteile gibt es diese zu nutzen? Was soll auf genommen werden?

3.1.1 Was wird aufgezeichnet?

Zu vollständigen Aufzeichnung eines Datensatzes werden die IMU-Daten, welche via BLE auf das Smartphone übertragen werden, in einer Datenbank abgespeichert. Insgesamt werden hierbei pro empfangene Dateneinheit 6 Werte persistiert, die x , y und z Richtung vom Beschleunigungssensor, bzw vom Gyroskop. Des weiteren wird die aktuelle Zeit, die aktuell auszuführende Aktion des Studienablaufs und die Information, ob die LED des Smartphones an oder aus ist, zu jeder empfangenen Dateneinheit hinzugefügt. Das Mikrofon wird ebenfalls aufgezeichnet und nach der Messung abgespeichert. Vor dem Beginn einer Messung wird der Studienteilnehmer gebeten, ein paar Zusatzinformationen (siehe 3.5) anzugeben. Diese werden vor dem Start der Messung am Smartphone ausgefüllt und ebenfalls in der Datenbank gespeichert.

3.1.2 Datenexport

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

3.2 Polysomnographie-Systeme

Als Referenz zu den eSense-Earpods wird ein Polysomnographie-System (PSG-System) verwendet. Ein solches System zeichnet Messungen für physiologische Funktionen des Körpers während des Schlafs auf und kann somit mögliche Schlafstörungen diagnostizieren. Es werden kontinuierlich verschiedene Körperfunktionen überwacht, wodurch nach einer Messung ein umfangreiches und individuelles Schlafprofil erstellt werden kann.

Das Polysomnographie-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“.

TODO: erkläre, wie man das PSG-System konfigurieren kann, dass es ein programm gibt, wo man eine Montage definieren kann, was ich gewählt habe, warum

Im folgenden werden alle Sensoren aufgelistet, welche für die Studie aufgezeichnet wurden. Die nicht persistierten Daten werden im folgenden ignoriert.

TODO: übersetze tabelle und erkläre, was die sachen sind, wo sie genau gemessen werden, Licht dient als referenz

- Abdomen
- **Lichtsignal (light)** (128Hz): wird verwendet, um die Signale vom PSG-System und den eSense-Earpods zu synchronisieren
- **Drucksignal (Flow)** (inputHz)
- Movement
- Pleth
- Pulse
- Schnarch
- Spo2

- Thorax
- ThoraxAbdomen
- edfAnnotations

3.2.1 Datenexport

Im PSG-System befindet sich eine CF-Karte (*Compact-Flash*). Diese kann mithilfe der vom PSG-System bereitgestellten Software „**TODO: inser Name of software**“ ausgelesen werden. Die Software stellt eine Ansicht dar, womit man die Signale untereinander in einer Timeline betrachten kann. Die aufgezeichneten Signale können als *edf*-Datei exportiert werden. Mittels Python kann man *edf*-Dateien auslesen und weiterverarbeiten. Pro Studie wurden alle 3 Positionsabläufe in einem einzigen Messvorgang aufgezeichnet. Somit müssen die 3 Einzelmessungen aus der *edf*-Datei herausgezogen werden. Für weitere Details siehe Kapitel ?? **TODO: change ref to ref, where edf-analyzation is explained**

3.3 Kamera

TODO: describe the camera usage

3.4 Datensynchronisation

TODO: Beschreibe, wie die Daten synchron abgestimmt wurden Um sicherzugehen, dass das PSG-System, sowie die Daten der eSense-Earpods zeitlich exakt übereinstimmt, wurden mit der App kurze Lichtblitze gesendet (20ms). Durch einen 3D-Drucker wurde eine Vorrichtung angefertigt, welche das Smartphone auf das PSG-System platziert, sodass die Lichtblitze direkt auf den Lichtsensor zeigen. **TODO: insert pic from 3D-Printing** Die Lichtblitze lösen nach jeder Aktionsänderung aus, die der Studienteilnehmer erhält. Den genauen Ablauf der Lichtblitze kann man dem Kapitel ?? entnehmen. Die Lichtblitze der Messung können nun mit den Lichtblitzen der eSense-Daten synchronisiert werden (siehe ?? **TODO: insert ref**).

3.5 Zusatzinformationen der Nutzer

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

TODO: Beschreibe, welche Daten aufgezeichnet wurden, jedoch mit beschreiben, dass sie nur da sind, falls was erkannt und bestätigt werden sollte, wie z.B dass jmd krasser geatmet hat weil er dick ist, oder so...

3.6 Maschinelle Lernverfahren

- Welche maschinellen Lernverfahren kommen in Frage?
- Welche Vor- und Nachteile können diese Verfahren bieten?
- Wie müssen Daten aufbereitet werden?

Klassifikation der Daten

- Random Forest
- SVM

Was bieten die beiden Verfahren, wie macht das Sinn, dass sinnvolle Ergebnisse herauskommen...

3.6.1 Datenaufbereitung für Klassifikation

5 sec Zeitschlitze, welche immer um 1 sec verschoben sind

Feature Extractor tsfresh auf 5 sec Zeitintervall angewandt...

4. Design

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

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

4.1 Studienplanung

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

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

4.2 Studienablauf

Dieser Ablauf (siehe Abb. 4.2) wurde nun pro Studienteilnehmer jeweils bei den 3 Positionen durchgeführt, womit alle Schlaflagen abgedeckt wären. Zu Beginn der

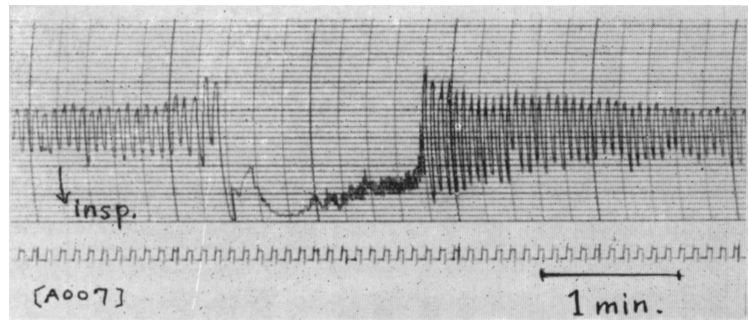


Figure 4.1: Regenerationsphase nach Luft anhalten [1].

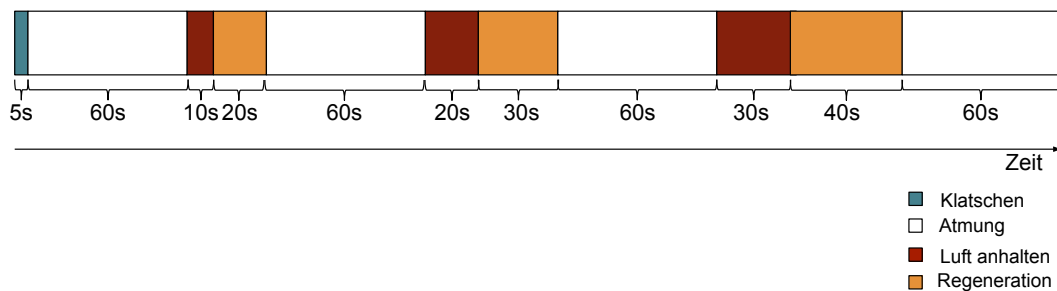


Figure 4.2: Ablauf der Studie mit einer Position

Studie fand eine kurze Einweisung statt, indem der Proband erfuhr, was er zu tragen hat und wie er Anweisungen erhält, um dem Ablauf folgen zu können. Die Kamera wurde auf dem Stativ platziert und so ausgelegt, dass sie das Ohr des Probanden filmt. Das PSG-System wurde am Studienteilnehmer angebracht, sowie alle nötigen Sensoren, die im Kapitel 3.2 beschrieben wurden. Nach der passenden Auswahl des Aufsatzes der eSense-Earpods war der Aufbau der Studie beendet.

Nun wird die Messung des PSG-Systems gestartet, sowie die Smartphone-App geöffnet. Nach Eingabe der Nutzerinformationen kann der erste Durchgang, welcher abhängig vom Ablauf der 3 Positionen war, begonnen werden. Durch den Start der Messung am Smartphone beginnt die Messung. Da zusätzlich das Mikrofon am eSense-Earpod mit aufgezeichnet wird, wird nach dem Start der Messung ein 4s Zeitfenster gewählt, indem der Teilnehmer in die Hände klatschen musste, um das Mikrofonsignal später synchronisieren zu können. Nun beginnt die Aufzeichnung. Der Leiter der Studie hat bereits den Raum verlassen und alle Anweisungen werden durch die Earpods per Audiosignal ausgesprochen. Sofern die Messung beendet ist, tritt der Leiter der Studie wieder in den Raum und die Messung kann exportiert werden. Der Export beinhaltet jegliche Smartphone-Daten. Die PSG-Daten werden als eine komplette Messung am Ende der Studie exportiert. Zudem wird die Kamera 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. Mehr zum Export der Daten und zur Synchronisation, siehe Kapitel ?? **TODO: linkREF**

5. Implementierung

In diesem Kapitel wird erläutert, wie die Daten des Datensatzes gesammelt, zur weiteren Verarbeitung vorbereitet und schließlich analysiert werden. Der Fokus hierbei liegt auf der Implementierung, für mehr Details, siehe Kapitel [TODO: link to ref](#)

5.1 App

5.1.1 Plattform

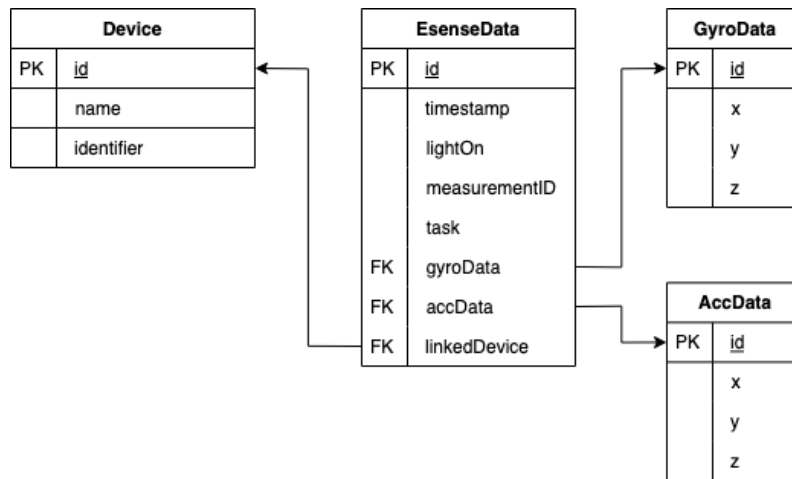
Die Smartphone-App wurde mit der Sprache Swift für Apple-Smartphones entwickelt. Mit der Software XCode lässt sich eine mit Swift geschriebene App kompilieren und auf dem Smartphone installieren.

Zur Einbindung externer Frameworks wird der Dependency-Manager *Accio* und *Carthage* verwendet. Folgende Frameworks sind in der App eingebunden worden: [TODO: insert table of frameworks with description](#)

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

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

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

Figure 5.1: ER-Diagramm der App-Datenbank **TODO: higher resolution**

5.1.2 Messungsablauf

Mit der Messungsansicht soll eine komplette Messung durchgeführt werden. Der *MeasurementFlow* wird gestartet und die erste *View* (Abb. 5.3a) wird geöffnet. Nach der erfolgreichen Verbindung mit den eSense-Earpods erfolgt eine Weiterleitung zur nächsten *View* (Abb. 5.3b) zum Ausfüllen der Nutzerinformationen. Mit dem Betätigen des Buttons: „Start Measurement“ bestätigt man die Eingabe der Daten und die Messung beginnt. Automatisch beginnt der erste Timer (Abb. 5.3c). Der Timer zeigt den aktuellen, sowie den nächsten *Task* an, sowie die Restzeit des aktuellen Tasks. Der genaue Ablauf der einzelnen Timer ist in Kapitel ?? **TODO: Add ref** 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.

5.1.3 Messung

Eine Messung wird im Code in einem *Measurement* Objekt persistiert. Durch ein Observer-Pattern wird der *MeasurementFlow* über jegliche Änderung informiert und kann entsprechende Handlungen durchführen. Durch die Funktionen **startMeasurement** und **stopMeasurement** kann eine Messung gestartet, bzw gestoppt werden. Mit der Funktion **startMeasurement** wird das IMU-Sampling per BLE, sowie die Audioaufnahme gestartet. Ebenso wird der erste Timer gestartet und ein doppeltes Lichtsignal gesendet. Durch die Funktion **stopMeasurement** werden die Datenströme gestoppt, ebenfalls ein doppeltes Lichtsignal gesendet und der Timer wird beendet. Der nächste Task wird gestartet, wenn der Timer abgelaufen ist. Der Timer startet mit der Länge des nächsten Tasks. Sofern der nächste Task „Hold_breath“ ist, also die Person im folgenden Task die Luft anhält, wird dem Teilnehmer kurz vor Ablauf mitgeteilt, wann der nächste Task startet. Mit der Instruktion „Bitte die Luft anhalten in 3, 2, 1“ weiß der Teilnehmer, wann er die Luft anhalten soll. Durch die Anweisung „Stopp“ wird dem Nutzer das Ende des Tasks mitgeteilt. Die Instruktionen liegen als Audiodatei vor und werden jeweils vor dem jeweiligen Task abgespielt und mittels Bluetooth über die Lautsprecher der eSense-Earpods ausgegeben.

TODO: extend with diagrams like class-diagram

5.2 Anbindung an Auswertungspipeline

Die App liefert beim Export die Daten der eSense-Earpods, was die IMU-Daten, sowie die Nutzerinformationen und die Mikrofonaufnahme beinhaltet. Zum aktuellen Stand liegen somit die Daten der eSense-Earpods und vom PSG-System vor. Zu Beginn müssen die PSG-Daten, welche als eine Messung für alle 3 Positionen pro Studienteilnehmer persistiert wurde, in 3 einzelne Messungen aufgeteilt werden. Die Daten des PSG-Systems liegen als *edf*-Datei vor. Diese können mittels python und der Library *edfrd* (siehe [TODO: verlinke zu tools](#)) ausgelesen werden. Mittels der Funktion `find_peaks` aus `scipy.peak` können die Peaks des Lichtsensors ermittelt werden. Da eine Messung mit 2 Lichtblitzen beginnt und endet, kann nun der Start- und Endzeitpunkt einer Position ermittelt und in die 11 verfügbaren Signale 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 zusammengeführt. Die Ornderstruktur kann der Abb. 5.2 entnommen werden und ist nun vollständig.

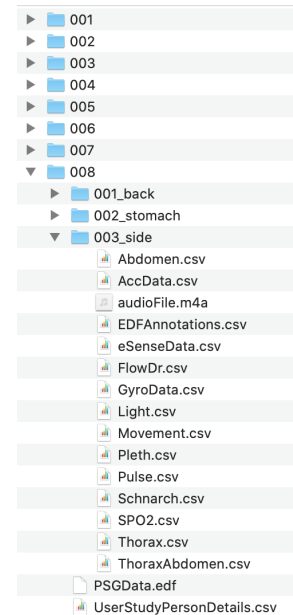


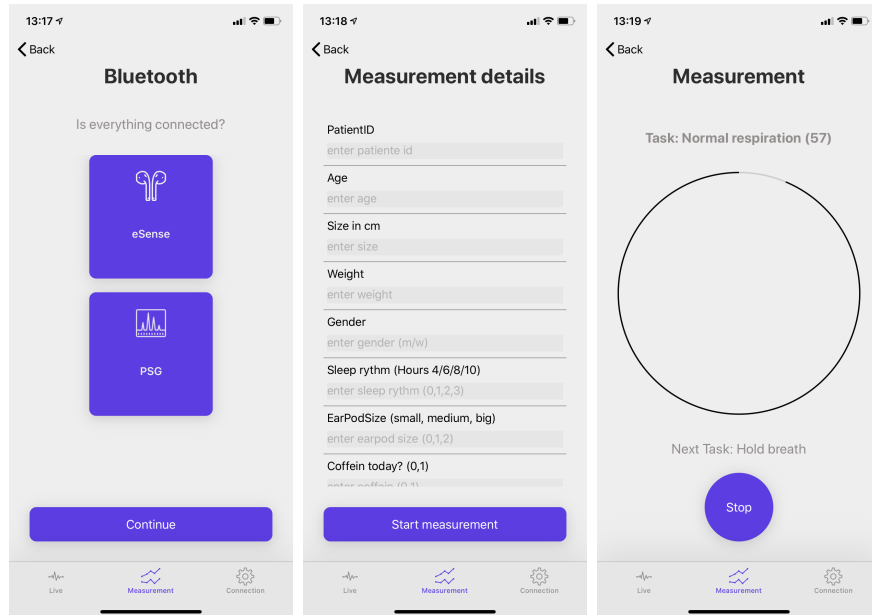
Figure 5.2: Ornderstruktur des Datensatzes

5.2.1 Synchronisation der Daten

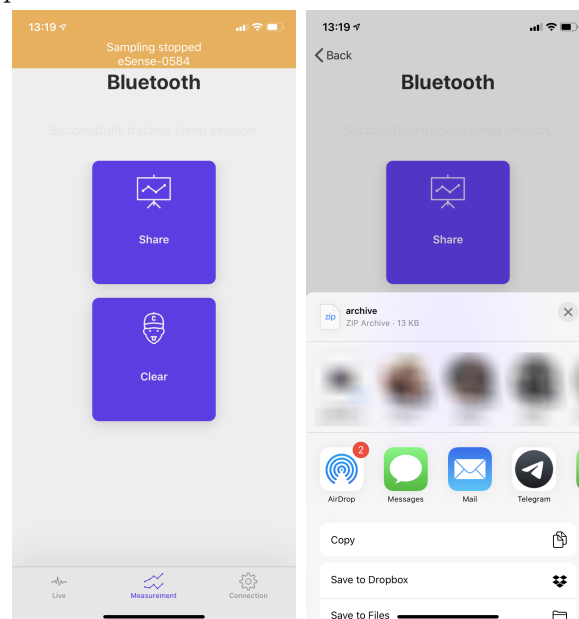
Da es nicht garantiert ist, dass der Timer des PSG-Systems zuverlässig arbeitet, wird jeder Peak des Lichtsensors mit dem Lichtsignal der Smartphone-Daten verglichen. Es wird ein Mittelwert aller Differenzen gebildet und das PSG-Signal wird um diesen verschoben. Nun ist garantiert, dass die Lichtsignale von PSG-System und dem Smartphone synchron sind. Die Daten sind nun bereit zur Analyse

5.3 Verarbeitungspipeline zur Klassifikation

Wie werden Daten aufgeteilt, wie wird trainiert? [TODO: Fill section](#)



(a) Verbinden der eSense-Earpods (b) Eingabe der Nutzerinformationen (c) Messung, aktueller und nächster Task



(d) Messung beendet (e) Messung teilen

Figure 5.3: Verlauf einer Messung mit der App

6. Evaluation

- 6.1 Vergleich verschiedener Klassifikationsverfahren
- 6.2 Gibt es passende Features?
- 6.3 Was kann bei respiratorischen Ereignissen klassifiziert werden, was nicht?

7. Zusammenfassung und Future Work

Bibliography

- [1] Chieko Sasaki Syoiti Kobayasi. “Breaking point of breath holding and tolerance time in rebreathing”. In: (Apr. 25, 1966). URL: https://www.jstage.jst.go.jp/article/jjphysiol1950/17/1/17_1_43/_pdf (visited on 01/21/2020).

