

Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Pfaffenwaldring 5a
70569 Stuttgart
Germany

Bachelor's Thesis Nr. 95

Implicit Night Sleep Monitoring Using Smartphone

Mohsen Zarei

Course of Study: Informatik, B.Sc.

Examiner: Jun.-Prof. Dr. Niels Henze

Supervisor: Alireza Sahami M.Sc.

Commenced: November 15, 2013

Completed: May 12, 2014

CR-Classification: H.1.2, I.5.2

Abstract

Recent studies show that many people nowadays suffer from sleep disorders, which can severely threaten the public health. Sleep monitoring could play an important role; since it makes it possible to recognize them at the early stages and prevent them.

Moreover, there are sort of methods, devices and special sensors as well as mobile phone applications, which try to realize the demand for sleep monitoring. Although all of these techniques require either a special device or sensor to be used or some user interactions, no approach has been proposed, that tracks sleep either in an unobtrusive way or without using any extra sensor.

To put it in a nut shell, we have tried in this work to figure out if it is viable, and if so, how efficient it could be to monitor the nightly sleep using smartphones without any need to interact with the phone or without using separate devices and/or sensors.

Kurzfassung

Bisherige Studien zeigen, dass immer mehr Menschen unter Schlafstörungen leiden, was sich negativ auf die Gesundheit auswirken kann. Schlafüberwachung bietet eine Möglichkeit, Schlafstörungen in einer früheren Entstehungsphase zu erkennen und ihnen vorzubeugen.

Es existieren bereits zahlreiche Methoden, Geräte und Sensoren, sowie Smartphone Anwendungen mit welchen die Schlafüberwachung durchgeführt werden kann. Der Nachteil aller dieser Techniken liegt darin, dass diese zur Schlafüberwachung entweder einen zusätzlichen Sensor benötigen oder das Eingreifen des Benutzers erfordern.

In dieser Arbeit wird die Frage untersucht, ob man mit Hilfe des Smartphones den Schlaf unauffällig überwachen kann, bzw. ob eine Schlafüberwachung ohne Bedarf von zusätzlichen Sensoren und/oder Interaktion mit dem Benutzer möglich ist.

Table of Contents

Abstract	3
Kurzfassung.....	4
Table of Contents	6
List of Figures	8
List of Tables.....	9
1 Introduction.....	11
1.1 Thesis Structure	11
2 Related Work	13
2.1 Commercial Devices.....	14
2.2 Mobile phone applications.....	15
3 Background	16
3.1 Machine Learning.....	16
3.1.1 Classification.....	16
3.1.2 Evaluation of the Classifier	16
3.2 Weka.....	16
3.2.1 ARFF.....	17
3.3 Fitbit Ultra	17
4 Data Acquisition.....	19
4.1 Procedure	19
4.2 Participants	19
4.3 Dataset	19
4.4 Ground Truth	20
5 Sleep Classification.....	22
5.1 Feature Extraction.....	22
5.2 Filtering Data	25
5.3 Model Development	26
5.3.1 Choosing the Classifier	26
5.3.2 Feature Anlysis.....	27
5.4 Discussion.....	38
5.4.1 Limitation.....	39
6 Conclusion	40
6.1 Future Work.....	41
7 Acknowledgement	42
Bibliography.....	43

Declaration 48

List of Figures

Figure 1: JSON file structure	21
Figure 2: heading of the ARFF file	24

List of Tables

Table 1: Tracked sensors by logeverything App.....	20
Table 2: List of features extracted.....	23
Table 3: Comparing the results of Naive Bayes Updatable and J48 Decision Tree using eight features with one-minute windows	26
Table 4: Comparing the results of using J48 Decision Tree using one single feature and one-minute windows	27
Table 5: Comparing the results of using J48 Decision Tree using combinations of two features and one-minute windows	29
Table 6: Comparing the results of using J48 Decision Tree using combinations of three features and one-minute windows.....	30
Table 7: Comparing the results of using J48 Decision Tree using combinations of four features and one-minute windows.....	31
Table 8: Comparing the best results of using J48 Decision Tree using one-minute windows	32
Table 9: Comparing the results of using J48 Decision Tree using one single feature and two-minute windows	33
Table 10: Comparing the results of using J48 Decision Tree using combinations of two features and two-minute windows.....	34
Table 11: Comparing the results of using J48 Decision Tree using combinations of three features and two-minute windows.....	35
Table 12: Comparing the results of using J48 Decision Tree using combinations of four features and two-minute windows.....	36
Table 13: Comparing the best results of using J48 Decision Tree using two-minute windows	37
Table 14: Comparing the results of using J48 Decision Tree using one-minute windows and two-minute windows having one single feature.....	38
Table 15: Comparing the results of using J48 Decision Tree with one-minute windows and two-minute windows having two features	38
Table 16: Comparing the results of using J48 Decision Tree with one-minute windows and two-minute windows having three features	38
Table 17: Comparing the results of using J48 Decision Tree with one-minute windows and two-minute windows having four features.....	39

1 Introduction

Adequate Sleep is an essential part of a healthy life [1]. Living longer, keeping the figure, having a better concentration, improving memory, spurring creativity and being in a great mood are just some advantages of a good sleep [2] [3]. On the other hand, sleep disorders can cause the increase of a heart disease, high blood pressure, stroke, diabetes and so forth [4].

Sleep disorders are unbelievable common in a modern life style, though we are well aware of the adverse effects of sleep disorders on people's health. For example 62 % of American adults experience a sleep problem a few nights per week [5] or more than half of Germans suffer more frequently from sleep disorders [6].

Therefore monitoring the sleep in order to prevent the sleep disorders and also measure the sleep quantity and quality has been an interesting topic for researchers and developers. Lots of devices have been produced; each of them uses different approaches, e.g. using physiological measurements, in order to achieve this goal. For realizing these measurements special sensors such as cameras, accelerometer or orientation sensors were needed.

Moreover, the number of smartphones increases constantly. More than one billion smartphones were shipped in 2013 according to market research firm IDC [7]. The strategic analysis predicts, that amount of purchased smartphones in 2015 will be doubled [8].

Smartphones have become part of our everyday lives. Considering a big range of prices nowadays, it is not a big deal to afford a smartphone. Regarding this fact, the developers came to this idea to use smartphone for tracking sleep.

Many mobile phone applications have been developed to track sleep manner. There is an advantage, which means that since smartphone sensors are used, therefore, no extra sensors were required. In these applications the user interaction is necessary, e.g. that the user should explicitly enter his sleep information. For example he/she should hold a button before going to bed and/or deactivate the sleep tracking after waking up and/or he/she should put the phone on/under the bed sheet, where he/she sleeps. This interaction may also lead the user not to act as he acts normally, because he should take care of monitoring sleep as well. However, more often users forget to track sleep, and therefore there is always some gap in sleep tracking.

All in all, in this work we are concerned with the question, whether smartphones can be harnessed to monitor the night sleep activity in an unobtrusive way? If so, what are relevant and important features to achieve this goal?

The idea is, on one hand, to utilize the ease of use and mobility of the smartphone, so that the users do not have to buy and carry or install extra sensors or devices, and on the other hand there is no need to worry about the interaction with the application, so that no behavior changes from the user is required.

1.1 Thesis Structure

This thesis is structured as follows. First, in Chapter 2, an overview of related scientific work as well as commercial devices in the field of health and sleep monitoring is given. Next, in Chapter 3, we review the theoretical and technical background, which could be helpful to understand the rest of the work. Afterwards, in Chapter 4, the procedure of data acquisition through the user study is explained, followed by a detailed description of sleep classification

process in chapter 5. Finally, we come to the conclusion and hint at some possible future improvements in chapter 6.

2 Related Work

There is significant existing work on sleep monitoring.

In some of the existing works a system without using smartphones is presented, which not only specified for sleep, but also on other human activities. In [9] a “novel Context-Driven Activity Theory (CDAT) for recognizing complex activities” is developed and validated. An approach along with an evaluation for activity recognition is described by Kim et al. [10]. Moreover, Bulling et al. reviewed in their work [11] activity recognition methods using body-worn inertial sensors. Some other examples of activity recognition approaches are discussed in [12] and [13], as well as [14]. Although, almost all of the activity recognition systems are based on actigraphy, in “Automatic Sleep Detection Using Activity and Facial Electrodes” problems of sleep tracking using actigraphy, the measurement of body limb movements, is discussed. There is concluded that “the novel automated analysis of two self-applied electro-oculography (EOG) electrodes had considerably higher specificity” [15].

Some other approaches concentrated only on sleep tracking but without using smartphones. For example [16] described a work on sleep pattern recognition using a bed pressure mats. A combination of machine learning methods and statistical methods is used in this work. Moreover, in [17] a home-based system to monitor sleep and to predict algorithmically non-rapid-eye-movement (NREM) and rapid-eye-movement (REM) sleep states, using eye and body motility. Furthermore, a sleep quality extracting system, named EZwakeup, with an eTextile-based sensing system is introduced in [18]. Finally, in [19] an electrooculogram (EOG)-based sleep tracking system is proposed presented.

Idea of sleep tracking using smartphones for sleep tracking or activity recognition is presented in many articles. The advantage of using smartphone is, that there is no need for additional sensors.

Sahami Shirazi et al. dealt with the possibility of tracking sleep using a smartphone and concluded that “It is possible to monitor users’ sleep duration using only an application on the mobile phone instead of using wearable actigraphy devices.” [20] Inspired by this we wanted to use the ability of mobile phones to track sleep and improve this concept to work in an unobtrusive manner. D. Lane et al. presented the design, implantation and evaluation of an android app named “BeWell” to monitor and manage overall wellbeing [21]. Erichsen et al. described a method of using tri-axial accelerometer of smartphone and developed furthermore an application for Apple iOS called “KickSleep” to detect “repetitive leg movements in sleep which occur in a significant portion of the adult population and are associated with several sleep disorders”. To fulfil this, the user has to either attach his iPhone to his leg or place it under the bed sheet [22]. Moreover, the process of developing a model for sleep for detecting sleep, using phone sensor and sleep diary entries is described in “Toss 'n' turn: smartphone as sleep and sleep quality detector” [23]. Furthermore, [24], [25] discussed also human activity recognition systems using smartphones. The disadvantage of these approaches is, that they could not be used in an unobtrusively way.

The idea of doing the whole process of sleep tracking unobtrusively is presented in the work of Zhenyu Chen et al. Design and evaluation of “a sensor-based commotional model that provides daily automated sleep duration monitoring using commodity smartphones” named BES [26]. Milosevic, T.Shrove and Jovanov presented a concept for monitoring health in a ubiquitous way using the ability of smartphones, which cared not only about the sleep but also about the other health factors. For this purpose they equipped the user with some sensors-

Transmitters. Furthermore he/she needs to install the app “mUAHEALTH”, which is developed for iOS. The sensor-transmitters are able to communicate with the app and transfer the tracked data to the iPhone, where the iPhone itself is a part of a network. The data could later be accessed from users’ personal computer or other target computers and be used for medical purposes [27]. Furthermore, in “Validating a mobile phone application for the everyday, unobtrusive, objective measurement of sleep”, described the development and evaluation of a sleep tracking mobile phone monitoring in an unobtrusive way [28].

2.1 Commercial Devices

In this section we review some commercial devices available in the market.

M. Kelly et al. reviewed the various sleep monitoring devices, which can be used for home-based sleep tracking and categorized them in 6 separate main groups according to the approach used by them, as follows:

1. Brain Activity signals
2. Autonomic signals
3. Movement
4. Bed-based
5. Others

The devices, which are available in the market, are of type 1 and 3; i.e. they monitor sleep based on Brain activity signals or Movement [29]. Here we take a short look at some commercial devices:

- Zeo:
Zeo is a head-mounted sleep tracker. The Brain activity signal tracked by the Zeo headband will be sent to the smartphone app or another base station. Although Zeo, Inc., the producer of this tracker, is closed down since 2013, but the devices could still be purchased from internet [30].
- Fitbit:
Fitbit is a small activity tracker in several versions, which uses movement detector for sleep tracking. It can/should be clipped on the clothes, wristed on the armband or put in the pocket. The data can be synchronized with a mobile phone application or with an application on the PC [31].
- Lark/WakeMate:
Both of these devices are offered as wristwatches and work based on actigraphy. Lark was first designed for sleep tracking with iPhone Apple iOS. The android version is also developed later [32].
WakeMate was able to connect with the user’s Apple iOS, Blackberry or Android-based smartphone via Bluetooth. The production is ended in 2012 [33].
- Sleep Tracker:
Sleep Tracker is a wrist watch based on movement, which connects via USB to computer to transfer data to the user’s personal “Sleep Tracker Analytic” account [34].
- UP(Jawbone):
UP monitor by Jawbone is an activity tracker based on actigraphy, which is a bracelet-like device, with ability to connect to the iPhone [35].

Sleep tracking using smartphone accelerometer could be somehow compared to the sleep tracking of most commercial devices available on the market except Zeo, which are Movement-based. Additionally smartphones have some more sensors e.g. light sensor or audiolevel, which could provide us with very useful information. It could be considered as a good reason for ability of smartphones being used as sleep tracker.

2.2 Mobile phone applications

Also in the application market e.g. Apple App Store or Google Play Store, there are many applications available, which track sleep. “Sleep Cycle Alarm Clock” for iOS and “Sleep As Android” for android are two famous examples of these applications available in the market. Both of them use motion sensor for sleep tracking [36] [37]. “Sleepbot” is another application, which is available for both android and iOS. It tracks motion and sounds to paint a visual picture of the sleep [38].

Comparing to the activity tracker devices, the advantage of such mobile phone applications is that the user except his/her smartphone requires no extra device or sensor. However like the activity trackers, these mobile phone applications also need interaction of the user and the sleep behavior change of the user probably.

3 Background

In this chapter for a better understanding of this work some technical fundamentals are pointed out. First an introduction to machine learning is given. Next, we take a look at the Fitbit Ultra activity tracker, which is used in this work for the user study.

3.1 Machine Learning

The main part of this thesis is about machine learning, so in this part we review some important topics, which are relevant for this work.

3.1.1 Classification

„Given a set of classes, we seek to determine which class(es) a given object belongs to. This definition stipulates that an object is a member of exactly one class.” [39]

We speak from a two-class classification or a binary classification, when we have only two classes. In this work, we have also a two-class classification problem, since; we have only two classes, e.g. the class “sleeping” and the class “not sleeping”.

In machine learning, the training of the set of rules is automatically possible through training data, which involve some typical examples for each class. To reach the goal of learning a classifier or a classification function, we use a learning method or learning algorithm. The output of the learning algorithm is the learned classifier, which assign the objects to the classes. To test the accuracy of classification function, it will be applied on new objects e.g. the test set. High accuracy in test data is the purpose in classification [39].

3.1.2 Evaluation of the Classifier

In a binary classification, as our case, there are four different possibilities for a prediction. The true positives (TP) and true negatives (TN) are correct classifications, which are desirable. A false positive (FP) is when the object is incorrectly mapped to the class “yes” (or positive) when it does not belong to it actually (negative). A false negative (FN) is when the object is incorrectly mapped to the class “no” when it is actually positive [40].

The measures of precision and recall concentrate the evaluation on the return of true positives, asking what percentage of the relevant objects has been classified and how many false positives have also been returned. A single measure that trades off precision versus recall is the F measure, which is the weighted harmonic mean of precision and recall [39].

$$\text{Precision: } P = \frac{TP_k}{TP_k + FP_k}$$

$$\text{Recall: } P = \frac{TP_k}{TP_k + FN_k}$$

$$\text{F-Measure: } F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.2 Weka

Weka is a free machine learning and data mining tool, which provide a sort of algorithms. The advantage of Weka is, that the algorithms can either be applied directly to a dataset or called

from the own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. [41]

In this work we used Weka 3.6 library in our own java code for classification.

3.2.1 ARFF

The name of the input file for using Weka is called ARFF. ARFF (Attribute-Relation File Format) file is an ASCII text file. It contains a list of instances sharing a set of attributes. ARFF files have two distinct sections, e.g. Header section and Data section [42].

The Header section consists the name of the relation an attribute declarations, which is a list of the attributes and their types. Every line in attribute declaration part begins with “@attribute” followed by the name of the attribute and its data type. Every attribute should be uniquely defined. The order the attributes in the header section is the same as the column position in the data section. [42].

The Data section of the ARFF file contains instance lines. Each line consists of an instance. Attribute values for each instance should be separated by commas. As mentioned, their order should be the order of declarations in header section [42].

3.3 Fitbit Ultra

In this work, we used a special model of Fitbit activity tracker, named Fitbit Ultra.

The Fitbit Ultra is a small clip-on device, which could be used to track the real-time activity stats. The Ultra's display shows the stats on the go, and the user can sync his tracker up to a free account online at Fitbit.com. The Ultra contains a 3D motion sensor that senses the motion in three dimensions. Fitbit Ultra can track the steps taken, distance traveled, calories burned, sleep time and quality, recent activity levels, clock and stopwatch and water Resistance. [43]

“The Fitbit Ultra tracker should hold up through normal use throughout the day. It can be worn on a bra or waistband or put in the pocket.” [43]

“For sleep tracking the user should place his tracker inside of the slot in the wristband that came with your Fitbit. The wristband should be worn on your non-dominant hand. when he is ready to go to sleep, he should press and hold his tracker's button for 2-3 seconds. A flashing stopwatch icon and counting numbers will be shown then. This indicates that he is in sleep mode.” [44]

“When he wakes up, he should remove his tracker from his wristband and hold down its button for 2-3 seconds until the icons stop flashing. The tracker should be synchronized to the Fitbit account to see information about the sleep quality the sleep log.” [44]

As result of the Fitbit Ultra we could have three different values:

- Value ‘1’ or asleep: It mentions that during sleep mode, the body is completely at rest and is not moving.

- Value '2' or awake: It happens, when the body of the user moves, for example when the user is turning over in the bed. This means the state, that the user is not awake but also not at rest.
- Value '3' or really awake: The value '3' appears in the log entries, when the user is awake, e.g. he/she is moving so much. [44].

In this work the value '1' was considered "sleeping" state of the user and the other values considered "no-sleeping" and were used as class attribute for classification in user's ARFF file.

4 Data Acquisition

To acquire the required data, a research study on detecting and monitoring sleep behavior using mobile phones was carried out. The raw data collected in the study was necessary for the further steps of the work, where we tried to build a model and mutually to develop the app.

4.1 Procedure

The study was planned and done in MCI group of “Institut für Visualisierung und Interaktive Systeme”. Seven participants were recruited for the study that last approximately two weeks. The study was conducted in the period from July to August 2013 at the University of Stuttgart. The main objective of the study was collecting data from smartphone using the android app called “logeverything”, which was developed by the MCI group. The app collects data from sensors of mobile phone. The data collected is saved on the mobile phone and could be accessed later.

Beside the app, each participant received an activity tracker named Fitbit Ultra for the study. They were asked to setup the Fitbit, carry it always with them, charge it every day by connecting it to the computer. The most important thing to do is turning it on/off before going to bed/after getting out of bed.

For the study, the participants needed to fill in questionnaires, which were necessary to get information about them.

4.2 Participants

Seven participants, who took part in the study, were all students between 23 and 29 years of age. Three of them were females and the other four were males. It was needed to have a smartphone, equipped with the Android operating system. There were four different models of smartphones which are used by participants including: Samsung Galaxy S2, Samsung Galaxy S3, Sony Xperia and LG Nexus S. Additionally, each user got a Fitbit activity tracker and worn it through the study period.

4.3 Dataset

The data collected by the logeverything app contained the information from 12 different sensors of the smartphone as listed in table 1. Actually the App tracked two more sensors, e.g. stillalive and gyroscope, which are ignored in data set, since; they did not include useful information for our purpose.

So, the data collected by the app included the values of each sensor accompanying by the corresponding time in UNIX time format. Every sampling period took 10 seconds and the values of the sensors were stored every second, during the 10 seconds of sampling period, in the external memory of the smartphone. Therefore, the participants were asked to keep around one Giga bytes of their smartphones memory card free.

Table 1: Tracked sensors by logeverything App

Sensor	Description
Accelerometer	Acceleration rate
App	Currently running app
Audiolevel	Audio volume level
Charging	Charging state of the device
Gyroscope	device's rate of rotation
Lightsensor	ambient light level (illumination)
Proximity	proximity of an object relative to the view screen of a device
Ringtone_volume	Ringtone volume level
Screen_on_off	State of screen
Stillalive	General state of the device
Wifi_ssid	Currently Wi-Fi SSID in use
Airplane_mode	Airplane mode of the device

4.4 Ground Truth

The data tracked by Fitbit Ultra activity tracker were used as ground truth in this work. Fitbit tracker provided information not only about sleeping state of the user, but also other activity information like calories burned, steps taken. As we were just interested in sleep tracking, the other information was ignored.

List of the sleeping data downloaded from account of the users in JSON format included user's sleep log entries as well as minute by minute sleep entry data. It provided following information for each tracking time: “start time”, “efficiency”, “duration”, “minutes to fall asleep”, “minutes asleep”, “minutes awake”, “minutes after wakeup”, “awakenings count”, “awake count”, “awake duration”, “restless count”, “restless duration”, “time in bed”, “minute data”. Furthermore, for each day, which may have contained more tracking times, the file supplied information about “total minutes asleep”, “total time in bed” and “total sleep records”.

The structure of the sleep part of JSON file is shown in figure 1 [45].

```
{
  "sleep": [
    {
      "isMainSleep":<value>,
      "logId":<value>,
      "efficiency":<value>,
      "startTime":<value>,
      "duration":<value>,
      "minutesToFallAsleep":<value>,
      "minutesAsleep":<value>,
      "minutesAwake":<value>,
      "minutesAfterWakeup":<value>,
      "awakeningsCount":<value>,
      "awakeCount":<value>,
      "awakeDuration":<value>,
      "restlessCount":<value>,
      "restlessDuration":<value>,
      "timeInBed":<value>,
      "minuteData": [
        {
          "dateTime":<value>,
          "value":<value>
        },
        <...>
      ]
    },
    <...>
  ],
  "summary": {
    "totalSleepRecords":<value>,
    "totalMinutesAsleep":<value>,
    "totalTimeInBed":<value>
  }
}
```

Figure 1: JSON file structure

Minute by minute sleep state of the user in the “minuteData” part of JSON file, contained three values, e.g. “1” for asleep, “2” for awake and “3” for totally awake, accompanying by the corresponding time. We considered the entries with values 2 and 3 as “not sleeping” state and the ones with value 1 as “sleeping” state.

5 Sleep Classification

The aim of this step was to develop a user-independent model for sleep monitoring, which could enable us to produce accurate predictions in new cases of smartphone sensor data. To achieve this objective, we used the data set and the ground truth, collected through user study.

5.1 Feature Extraction

The data acquired from the study needed to be processed in order to find out a relationship between our two pieces of information, e.g. the data from Fitbit tracker and from the logeverything app.

In this step, a list of features has been driven from the data collected by the app. To do so, we developed a java program, which was able to read the CSV files, produced from the app for each user, and provide the features for that user in ARFF format as output.

The app has saved the data of the user as CSV files, which included the tracked data of following 12 sensors:

- Accelerometer
- Airplane_mode
- App
- Audiolevel
- Charging
- Gyroscope
- Lightsensor
- Proximity
- Ringtone_volume
- Screen_on_off
- Stillalive
- Wifi_ssid

We used the data from all of the sensors except for Gyroscope and Stillalive sensors. These two sensors do not supply useful information for sleep tracking, e.g. gyroscope measures or maintains the orientation of a device, and stillalive indicates the general state of the smartphone. Therefore, we used the raw data from 10 sensors and tried to extract some features from them. The features extracted are listed in the table 1.

One main problem occurred during java programming, was synchronizing the different sensor data, since the sampling timestamps of the sensors were not simultaneous. To remedy this, we considered the timestamps of the accelerometer as our main sampling periods. So in these periods, other sensor data were checked, whether there is also tracking information from other sensors. If it was not the case, we ignored the information at that timestamp. After all we wrote all the acceptable timestamps accompanying by the corresponding sensor data in an ARFF file.

Table 2: List of features extracted

Feature Name	Description
Accelerometer Mean	$\frac{1}{N} \sum_1^N l_i$, where $l_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$
Accelerometer Standard Deviation	$\sqrt{\frac{1}{N} \sum_1^N (l_i - \mu)^2}$, where $\mu = \text{Mean}$
Airplane_mode Mode	switched/true/false
App Mode	switched/true/false
Audiolevel Mean	$\frac{1}{N} \sum_1^N x_i$
Audiolevel Standard Deviation	$\sqrt{\frac{1}{N} \sum_1^N (x_i - \mu)^2}$, where $\mu = \text{Mean}$
Charging Mode	switched/true/false
Lightsensor Mean	$\frac{1}{N} \sum_1^N x_i$
Lightsensor Standard Deviation	$\sqrt{\frac{1}{N} \sum_1^N (x_i - \mu)^2}$, where $\mu = \text{Mean}$
Proximity Mode	switched/0.0/0.5
Ringtone_volume Mean	$\frac{1}{N} \sum_1^N x_i$
Ringtone_volume Standard Deviation	$\sqrt{\frac{1}{N} \sum_1^N (x_i - \mu)^2}$, where $\mu = \text{Mean}$
Screen_on_off Mode	switched/on/off
Wifi_ssid Mode	switched/unchanged

Afterwards we had to read the ground truth data collected from Fitbit, which were downloaded from Fitbit accounts of the users in JSON files and add them as attribute to our ARFF files.

In the “minuteData” part of JSON file, three values, e.g. “1” for asleep, “2” for awake and “3” for totally awake were mentioned. We considered the entries with values 2 and 3 as “not sleeping” state and the ones with value 1 as “sleeping” state.

This information were added to the instances of the ARFF file, which then were used as class attribute for classification.

Therefore, for each user an ARFF file was produced which contains the data from the sensors collected by the app and the respective Fitbit value as ground truth with the adjusted time. The sampling period in our ARFF files were 1 minute. This was done using a java program, developed and tested in eclipse 4.3 Kepler.

An example for heading of such an ARFF file is shown in figure 2:

```
@relation sleep

@attribute timestamp date yyyy-MM-dd'T'HH:mm:ss
@attribute accelerometerMean numeric
@attribute accelerometerStandardDeviation numeric
@attribute airplane_modeMode {switched, on, off}
@attribute appMode {switched, true, false}
@attribute audiolevelMean numeric
@attribute audiolevelStandardDeviation numeric
@attribute chargingMode {switched, true, false}
@attribute lightsensorMean numeric
@attribute lightsensorStandardDeviation numeric
@attribute proximityMode {switched, 0.0, 5.0, 8.0}
@attribute ringtone_volumeMean numeric
@attribute ringtone_volumeStandardDeviation numeric
@attribute screen_on_offMode {switched, on, off}
@attribute wifi_ssidMode {switched, unchanged}
@attribute sleeping {yes, no}
@attribute fitbit_value {1, 2, 3}
```

Figure 2: heading of the ARFF file

5.2 Filtering Data

A further problem occurred in the ARFF file of the users, was that at some places there were missing lines either for Fitbit value, which caused when user forgot to track his sleeping, or for sensors due to the problems occurred in the smartphones. These sorts of data missing usually happened when the smartphone was turned off or ran out of battery. However, in some cases the app was not running or was not tracking the information from sensors correctly.

To solve this, the instances in ARFF file, where either Fitbit value or sensors data were missing, were omitted, since, it did not make any sense to use the instances of ARFF file, when either the class attribute was missing, or when we had just the class attribute and the other attributes were missing.

Furthermore, we checked the ARFF files to discover if there are any useless features. For example in this step the feature Wifi-ssid mode was ignored, since it contained no important information. So, in this part the six following features were eliminated:

- Airplane_mode Mode
- App Mode
- Charging Mode
- Ringtone_volume Mean
- Ringtone_volume Standard Deviation
- Wifi_ssid Mode

From this step onwards, we continued the classification with the eight remaining features:

1. Accelerometer Mean
2. Accelerometer Standard Deviation
3. Audiolevel Mean
4. Audiolevel Standard Deviation
5. Lightsensor Mean
6. Lightsensor Standard Deviation
7. Proximity Mode
8. Screen_on_off Mode

5.3 Model Development

In this step, we have tried to recognize and understand the relation between the features and physical state of the user, through appropriate classifier. The aim was to eliminate the irrelevant features from the features list in order to get a better performance and finally find the minimum number of features, which deliver the most important information.

5.3.1 Choosing the Classifier

For building our user-independent model, we decided to use two kinds of classifiers. As Naive Bayes and Decision tree especially C4.5 were the most commonly used classification algorithms for sleep classification, we chose them as our classifiers. Therefore, we used J48 Decision Tree, the Weka implementation of the C4.5, as well as “Naïve Bayes Updatable” and compared their performance.

To evaluate the classifier performance, we modified the k-cross-validation of Weka, since the normal k-cross-validation could not give us an accurate evaluation for our user-independent model.

The normal k-fold-cross-validation in Weka works as follows: All the labeled data will be taken and randomly divided in k equal sized set $\frac{k-1}{k}$ part of each set will be used for training and the remaining $\frac{1}{k}$ part for testing. It produces a classifier from the training part of first set and then tests it with its testing part. The same sequence will be done for all sets. This validation averages the performance of k classifiers and delivers it as a result.

In our modified version of cross-validation, which is developed and used in this work, each iteration included the dataset of one user considered as the test set and the rest of data considered as the training set. We wanted to prevent a random division of the data set into k disjoint sets of equal size. Therefore this modified version of cross-validation is called user-cross-validation. As there were seven users, the number of sets is also seven; such that a modified 7-user-cross-validation is applied. The modified 7-cross-validation is developed in java with the aid of Weka API. For this part Eclipse 4.3 Kepler and Weka 3.6.10 are used.

First we applied the J48 Decision Tree and naïve Bayes Updatable on our data set, e.g. on our data with the eight remaining features. The results of the validations of these two classifiers are shown in the table 3. Here we used one second window for our data set.

Table 3: Comparing the results of Naive Bayes Updatable and J48 Decision Tree using eight features with one-minute windows

	Precision	Recall	F-Measure
Naïve Bayes Updatable	0.71	0.79	0.73
J48 Decision Tree	0.73	0.89	0.80

Regarding the obviously better performance of J48, it was selected and from that time the work was continued using this algorithm.

5.3.2 Feature Analysis

The next goal was to improve the results of the evaluation and possibly to increase the performance by removing some features. We tested two different sampling windows length. First we used one-minute long windows, e.g. we considered our whole data set ARFF files, and applied the J48 Decision Tree with different combinations of the features on it. Second we do the same, but with two-minutes windows. In contrast to one-minute windows sampling, where we had sampling data for each minute, in two-minute windows they were needed every second minute.

5.3.2.1 Using one-minute Windows

In this section our sampling period was one-minute windows, so we had sensor data and ground truth in every minute. On this data, we applied J48 Decision Tree algorithm and modified 7-cross-validation.

Initially using feature selection we used a single feature for classification. The results are shown in table 4:

Table 4: Comparing the results of using J48 Decision Tree using one single feature and one-minute windows

	Precision	Recall	F-Measure
Accelerometer Mean	0.73	0.95	0.82
Accelerometer Standard Deviation	0.70	1.0	0.82
Audiolevel Mean	0.74	0.82	0.77
Audiolevel Standard Deviation	0.73	0.86	0.78
Lightsensor Mean	0.79	0.92	0.85
Lightsensor Standard Deviation	0.73	0.97	0.83
Proximity Mode	0.70	1.0	0.82
Screen_on_off Mode	0.70	1.0	0.82

Respecting the evaluations, the best option for developing a model using a single feature is using Lightsensor Mean. In other words, the most important sensor could be Lightsensor, since, it provided the best result.

Furthermore, in this step, we tried to eliminate the features, which were driven from the same sensor, e.g. accelerometer mean and accelerometer standard deviation that provided similar information.

Regarding the presented results, it was decided to eliminate Audiolevel Mean. It was similar to Audiolevel Standard Deviation, since they were both driven from the same sensors. However, Audiolevel Standard Deviation yields better outputs. The same reasoning was also applicable to the Accelerometer Mean and Lightsensor Standard Deviation measures. Therefore, these measures were also omitted.

Therefore, the process was continued with the remaining five features:

1. Accelerometer Standard Deviation
2. Audiolevel Standard Deviation
3. Lightsensor Mean
4. Proximity Mode
5. Screen_on_off Mode

In the next step, as shown in table 5, all possible combinations of two features out of our five remaining features were analyzed.

Table 5: Comparing the results of using J48 Decision Tree using combinations of two features and one-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD 	0.75	0.85	0.79
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor Mean 	0.80	0.92	0.85
<ul style="list-style-type: none"> • Accelerometer SD • Proximity Mode 	0.70	0.99	0.82
<ul style="list-style-type: none"> • Accelerometer SD • Screen_on_off Mode 	0.70	0.99	0.82
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor Mean 	0.76	0.85	0.80
<ul style="list-style-type: none"> • Audiolevel SD • Proximity Mode 	0.73	0.82	0.77
<ul style="list-style-type: none"> • Audiolevel SD • Screen_on_off Mode 	0.74	0.82	0.77
<ul style="list-style-type: none"> • Lightsensor Mean • Proximity Mode 	0.78	0.94	0.85
<ul style="list-style-type: none"> • Lightsensor Mean • Screen_on_off Mode 	0.78	0.93	0.85
<ul style="list-style-type: none"> • Proximity Mode • Screen_on_off Mode 	0.70	1.0	0.82

So for the double combination of features provided “Lightsensor Mean & Proximity Mode” the best performance, though, “Accelerometer Standard Deviation & Lightsensor Mean” or “Lightsensor Mean & Screen_on_off Mode” performed almost as good as “Lightsensor Mean & Proximity Mode”.

Then we tried to develop our model using three sensors. Table 6 presents the results:

Table 6: Comparing the results of using J48 Decision Tree using combinations of three features and one-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Lightsensor M. 	0.78	0.85	0.81
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Proximity Mode 	0.75	0.85	0.79
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Screen. Mode 	0.74	0.84	0.78
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor M. • Proximity Mode 	0.79	0.94	0.86
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor M. • Screen. Mode 	0.79	0.91	0.84
<ul style="list-style-type: none"> • Accelerometer SD • Proximity Mode • Screen. Mode 	0.70	0.99	0.82
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor M. • Proximity Mode 	0.76	0.87	0.80
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor M. • Screen. Mode 	0.75	0.85	0.79
<ul style="list-style-type: none"> • Audiolevel SD • Proximity Mode • Screen. Mode 	0.74	0.82	0.77
<ul style="list-style-type: none"> • Lightsensor M. • Proximity Mode • Screen. Mode 	0.78	0.93	0.84

The best trio-combination of the features is “Accelerometer Standard Deviation & Lightsensor Mean & Proximity Mode”.

Finally, we tested as follows, which features could produce accurate results.

Table 7: Comparing the results of using J48 Decision Tree using combinations of four features and one-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Lightsensor Mean • Proximity Mode 	0.77	0.87	0.81
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Lightsensor Mean • Screen_on_off Mode 	0.77	0.85	0.80
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Proximity Mode • Screen_on_off Mode 	0.73	0.85	0.78
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor Mean • Proximity Mode • Screen_on_off Mode 	0.79	0.93	0.85
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor Mean • Proximity Mode • Screen_on_off Mode 	0.75	0.87	0.80

Results showed, we achieved the best results for combinations of four features, when we used “Accelerometer SD, Lightsensor Mean, Proximity Mode, Screen_on_off Mode”.

The best combinations are shown again in Table 8:

Table 8: Comparing the best results of using J48 Decision Tree using one-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Lightsensor Mean 	0.79	0.92	0.85
<ul style="list-style-type: none"> • Lightsensor Mean • Proximity Mode 	0.78	0.94	0.85
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor M. • Proximity Mode 	0.79	0.94	0.86
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor Mean • Proximity Mode • Screen_on_off Mode 	0.79	0.93	0.85

As the final result, the best choice of all tested combinations was picked, which is the triple combination of “Accelerometer Standard Deviation, Light Mean and Proximity Mode”, which had Precision of 79%, Recall of 94% and F-Measure of 86%.

5.3.2.2 Using two-minutes Windows

After developing the sleep monitoring model for one-minute windows data, we repeated the whole process for two-minute windows data. The reason for this was, that in two-minute windows we had less effort for sampling and wanted to find out, whether it is possible to develop a model with comparably good results as the model built using one-minute data.

In two-minute windows sampling we needed a sample every second minute, therefore we could do the sampling with less effort.

Similar as for the classification using one-minute windows, we began the classification for two-minute windows using one single feature:

Table 9: Comparing the results of using J48 Decision Tree using one single feature and two-minute windows

	Precision	Recall	F-Measure
Accelerometer Mean	0.70	1.0	0.82
Accelerometer Standard Deviation	0.73	0.96	0.83
Audiolevel Mean	0.73	0.85	0.78
Audiolevel Standard Deviation	0.73	0.85	0.78
Lightsensor Mean	0.78	0.93	0.85
Lightsensor Standard Deviation	0.73	0.97	0.83
Proximity Mode	0.70	1.0	0.82
Screen_on_off Mode	0.70	1.0	0.82

As expected, in case that only a single feature is available, we could get the best results from “Lightsensor Mean”.

As before, in this part the sensors Audiolevel Mean, Accelerometer Mean and Lightsensor Standard Deviation were eliminated and the process was continued with the remaining five features:

- Accelerometer Standard Deviation
- Audiolevel Standard Deviation
- Lightsensor Mean
- Proximity Mode
- Screen_on_off Mod

In the next step all possible combinations of double combinations out of our five features were analyzed:

Table 10: Comparing the results of using J48 Decision Tree using combinations of two features and two-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD 	0.71	0.94	0.81
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor M. 	0.74	0.96	0.83
<ul style="list-style-type: none"> • Accelerometer SD • Proximity Mode 	0.72	0.94	0.81
<ul style="list-style-type: none"> • Accelerometer SD • Screen. Mode 	0.73	0.94	0.82
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor M. 	0.77	0.85	0.80
<ul style="list-style-type: none"> • Audiolevel SD • Proximity Mode 	0.73	0.84	0.77
<ul style="list-style-type: none"> • Audiolevel SD • Screen. Mode 	0.73	0.84	0.78
<ul style="list-style-type: none"> • Lightsensor Mean • Proximity Mode 	0.78	0.93	0.85
<ul style="list-style-type: none"> • Lightsensor Mean • Screen. Mode 	0.78	0.93	0.85
<ul style="list-style-type: none"> • Proximity Mode • Screen. Mode 	0.70	1.0	0.82

Here the best combination could be the one of “Lightsensor Mean & Screen_on_off Mode”, though “Lightsensor Mean & Proximity Mode” provided almost the same accuracy.

Then we tested the trio-combination of the features to get the best evaluation results:

Table 11: Comparing the results of using J48 Decision Tree using combinations of three features and two-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Acc. SD • Audiolevel SD • Lightsensor M. 	0.75	0.94	0.84
<ul style="list-style-type: none"> • Acc. SD • Audiolevel SD • Proximity Mode 	0.71	0.93	0.80
<ul style="list-style-type: none"> • Acc. SD • Audiolevel SD • Screen. Mode 	0.71	0.91	0.80
<ul style="list-style-type: none"> • Acc. SD • Lightsensor M. • Proximity Mode 	0.74	0.95	0.82
<ul style="list-style-type: none"> • Acc. SD • Lightsensor M. • Screen. Mode 	0.74	0.95	0.83
<ul style="list-style-type: none"> • Acc. SD • Proximity Mode • Screen. Mode 	0.72	0.96	0.82
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor M. • Proximity Mode 	0.78	0.86	0.81
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor M. • Screen. Mode 	0.76	0.85	0.80
<ul style="list-style-type: none"> • Audiolevel SD • Proximity Mode • Screen. Mode 	0.74	0.82	0.77
<ul style="list-style-type: none"> • Lightsensor M. • Proximity Mode • Screen. Mode 	0.78	0.93	0.84

As expected, the combination of the features “Lightsensor Mean & Proximity Mode & Screen_on_off Mode” provided the best results. The results of the trio-combination “Accelerometer Standard Deviation & Audiolevel Standard Deviation & Lightsensor Mean” were also acceptable.

Finally, we tested, which features could produce accurate results for four- feature combinations, as follows:

Table 12: Comparing the results of using J48 Decision Tree using combinations of four features and two-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Lightsensor Mean • Proximity Mode 	0.75	0.95	0.83
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Lightsensor Mean • Screen_on_off Mode 	0.75	0.93	0.82
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Proximity Mode • Screen_on_off Mode 	0.71	0.93	0.80
<ul style="list-style-type: none"> • Accelerometer SD • Lightsensor Mean • Proximity Mode • Screen_on_off Mode 	0.72	0.95	0.81
<ul style="list-style-type: none"> • Audiolevel SD • Lightsensor Mean • Proximity Mode • Screen_on_off Mode 	0.76	0.86	0.80

The best evaluation outputs we could get from the combination of “Accelerometer Standard Deviation, Lightsensor Mean, Proximity Mode, Screen_on_off Mode”.

The results of the classification with two-minute windows data set were almost the same as from last section, but with slightly worse outputs.

Table 13: Comparing the best results of using J48 Decision Tree using two-minute windows

	Precision	Recall	F-Measure
<ul style="list-style-type: none"> • Lightsensor Mean 	0.78	0.93	0.85
<ul style="list-style-type: none"> • Lightsensor Mean • Screen. Mode 	0.78	0.93	0.85
<ul style="list-style-type: none"> • Lightsensor M. • Proximity Mode • Screen. Mode 	0.78	0.93	0.84
<ul style="list-style-type: none"> • Accelerometer SD • Audiolevel SD • Lightsensor Mean • Proximity Mode 	0.75	0.95	0.838

5.4 Discussion

We applied J48 Decision Tree on the different combinations of features as well as different sampling windows. Regarding the evaluation, the results of the two-minute windows data set were slightly worse than the results of the one-minute windows. Considering the fact, that for two-minute windows we need to put less effort in sampling period, the results were also acceptable. In other words, there is a trade-off between reducing the effort taken for sampling and increasing the accuracy of the model.

In following tables we compared the evaluation results. As a conclusion it can be said that, if one wants to get better results, he may choose one-minute windows. But if less sampling effort is desirable, then using two-minute windows could be a better choice.

Table 14: Comparing the results of using J48 Decision Tree using one-minute windows and two-minute windows having one single feature

	Precision	Recall	F-Measure
One-minute windows	0.79	0.92	0.85
Two-minute windows	0.78	0.93	0.85

Table 15: Comparing the results of using J48 Decision Tree with one-minute windows and two-minute windows having two features

	Precision	Recall	F-Measure
One-minute windows	0.78	0.94	0.85
Two-minute windows	0.78	0.93	0.85

Table 16: Comparing the results of using J48 Decision Tree with one-minute windows and two-minute windows having three features

	Precision	Recall	F-Measure
One-minute windows	0.79	0.94	0.86
Two-minute windows	0.78	0.93	0.84

Table 17: Comparing the results of using J48 Decision Tree with one-minute windows and two-minute windows having four features

	Precision	Recall	F-Measure
One-minute windows	0.79	0.93	0.85
Two-minute windows	0.75	0.95	0.83

5.4.1 Limitation

Attempts were made to develop an accurate sleep tracking model using two different classifiers and with different sampling periods. This was achieved as well, but from the first step we had to take some limitations into account.

First of all, in our user study, we had seven participants available, which all were students and between 23 and 29 in age. We carried out our study for just around two weeks. It could have been possible to get a better data set, if we could have more participants from a wide range of age groups and professional field, and over a longer period of time.

Furthermore, we used the logeverything app for get information from smartphone sensors. Unfortunately it led to a fail in some cases, since, the smartphones of the users were out of charge or the app was not running. Moreover, the tracking data from app also produced some noise.

As ground truth, the data from the commercial activity tracker was used. This information could also not be always reliable, since the tracker could not deliver information without errors. Moreover, in some cases we had no information in the dataset from the Fitbit i.e. missing data. This occurred, when the participants of the user study forgot to use the tracker during the night sleep.

Finally, our models were the best options after testing only two classifiers. One can imagine getting probably better results using some other classifiers. Due to lack of time, we were not able to apply other classifiers during this work.

6 Conclusion

As sleep plays a vital role in overall health and well-being throughout the life. “Yet millions of people do not get enough sleep and many suffer from lack of sleep. For example, surveys conducted by the NSF(National Science Foundation)” [46]. As people are beginning to understand the importance of good sleep, the sleep tracking methods like activity tracker commercial devices and sleep tracking smartphone application are becoming more and more common.

In this work, we tried to find a way to monitor the sleep using smartphone, so that the user does not have to buy another tracking device or install extra sensors. The objective was also to do the whole sleep monitoring process unobtrusively, e.g. without any need to interact with the smartphone, which is not the case when using applications available in the market.

First of all, we carried out a research study with seven participants, where we collected raw data using the android app “logeverything” and Fitbit Ultra activity tracker. The app provided us with data tracked from 12 different sensors of the smartphone and Fitbit supplied information about the sleeping state of the participants. We used the collected data from Fitbit as our ground truth.

Then we developed a java program, which enabled us to extract 14 features from the tracked data from the “logeverything” app. Moreover, we had to download the ground truth data from the Fitbit accounts of the users in JSON format and read the sleeping data from the files. Afterwards we combined these features with the ground truth data from Fitbit in adjusted time and saved them as ARFF files.

In next stage, we tried to find out a relation between the features driven from the sensors of the smartphone and our ground truth data from Fitbit activity tracker. Thereby, we used some filtering methods for unusable instances of the ARFF files, followed by elimination of irrelevant features. After filtering, we tested two different classifiers, e.g. naive Bayes and J48 Decision Tree from the machine learning tool Weka. Due to the better results, we decided to use J48 Decision Tree as our classifier.

Afterwards, we tested different combinations of combinations of features on the one-minute window data set. We found out, if one wants to develop a model using only one sensor, he should use “Lightsensor”. If two sensors values could be available, the performance is the one from using “Lightsensor” and “Proximity”. If a third sensor, “Accelerometer” is the best choice. Screen_on_off is the fourth best sensor, which provides relevant information for sleep tracking. As final result, we could have the best performance using three features “Accelerometer Standard Deviation”, “Lightsensor Mean” and “Proximity Mode” with Precision of 79%, Recall of 94% and F-measure of 86%.

In next step, we repeated the whole procedure again for two-minute windows. We wanted to figure out, if we could get the same accurate models with having longer sampling periods, since, for two-minute windows less effort on sampling phase is needed. The results of two-minute windows were slightly worse as the result of one-minute windows, but still acceptable. So the choice between the better results and less effort for sampling seemed to be a trade-off.

6.1 Future Work

It is conceivable that the built model for sleep tracking in this work may improve further in future, because of the lack of time and little number of participants; it was not possible to carry out the research study with more participants and for a longer period. As mentioned, the research study was done with seven people during two weeks.

Maybe we could have some better results, if we could have applied and tested more classifiers on the data set. In this workset, we just considered two classifiers.

Furthermore, developing an android application using the models built in this work, as well as evaluating this during a user study, could be considered as further steps.

7 Acknowledgement

I wish to thank my professor Jun.-Prof. Dr. Niels Henze for giving me the opportunity to perform this work in his institute.

I would also like to express my gratitude to my supervisor Alireza Sahami M.Sc. for his continuous support, patience, motivation and engagement during the length of this work. His guidance helped me in all the time of this work and writing this thesis.

Also, I like to thank my family, friends, classmates in the University of Stuttgart, and those, who helped me throughout this work.

Bibliography

- [1] M.-Y. Chen, E. K Wang and Y.-J. Jeng, "Adequate sleep among adolescents is positively associated with health status and health-related behaviors," *BMC Public Health*, 2006.
- [2] "Body&Soul," [Online]. Available: <http://www.bodyandsoul.com.au/health/health+advice/10+benefits+of+a+good+nights+sleep,17681>. [Accessed 7 May 2014].
- [3] "Health," [Online]. Available: http://www.health.com/health/gallery/0,,20459221_2,00.html. [Accessed 7 May 2014].
- [4] "WebMD," [Online]. Available: <http://www.webmd.com/sleep-disorders/excessive-sleepiness-10/10-results-sleep-loss>. [Accessed 7 May 2014].
- [5] "Statistic Brain," [Online]. Available: <http://www.statisticbrain.com/sleeping-disorder-statistics/>. [Accessed 7 May 2014].
- [6] "DAK Gesundheit," [Online]. Available: http://www.dak.de/dak/download/Vollstaendiger_bundesweiter_Gesundheitsreport_2010-1319238.pdf. [Accessed 7 May 2014].
- [7] "International Data Corporation (IDC)," [Online]. Available: <http://www.idc.com/getdoc.jsp?containerId=prUS24645514>. [Accessed 7 May 2014].
- [8] "Bloomberg," [Online]. Available: <http://www.bloomberg.com/news/2012-10-17/smartphones-in-use-surpass-1-billion-will-double-by-2015.html>. [Accessed 7 May 2014].
- [9] S. Saguna, A. Zaslavsky and D. Chakraborty, "Complex activity recognition using context-driven activity theory and activity signatures," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 20, no. 6, 2013 .
- [10] E. Kim, S. Helal, C. Nugent and J. W. Lee, "Assurance-oriented activity recognition,"] *Proceedings of the 2011 international workshop on Situation activity & goal awareness*, pp. 15-24, 2011.
- [11] A. Bulling, U. Blanke and B. Schiele, "A tutorial on human activity recognition using] body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, 2014 .
- [12] J. Yin, Q. Yang, D. Shen and Z.-N. Li, "Activity recognition via user-trace] segmentation," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 4 , 2008 .
- [13] M. Yoshizawa, W. Takasaki and R. Ohmura, "Parameter exploration for response time] reduction in accelerometer-based activity recognition," *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 653-664.

- [14 Y. Nam, S. Rho and C. Lee, "Physical activity recognition using multiple sensors embedded in a wearable device," *ACM Transactions on Embedded Computing Systems (TECS) - Special issue on embedded systems for interactive multimedia services (ES-IMS)*, vol. 12, no. 2, 2013.
- [15 J. Virkkala, J. Hasan, R. Velin, S.-L. Himanen, A. Värri and E. . J. Van Someren ,] "Automatic sleep detection using activity and facial electrodes," *30th Annual International IEEE EMBS Conference*, 2008.
- [16 V. Metsis, G. Galatas, A. Papangelis, D. Kosmopoulos and F. Makedon, "Recognition of sleep patterns using a bed pressure mat," *PETRA '11 Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments*, 2011.
- [17 A. Mamelak and J. Hobson, "Nightcap: a home-based sleep monitoring system," *Journal Article, Research Support, Non-U.S. Gov't*, 1989.
- [18 M.-C. Huang, X. Zhang, W. Xu, J. Liu and M. Sarrafzadeh, "EZwakeup: a sleep environment design for sleep quality improvement," *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, pp. 2389-2394 , 2014.
- [19 C.-E. Kuo, S.-F. Liang, Y.-C. Li, F.-Y. Cherng, W.-C. Lin, P.-Y. Chen, Y.-C. Liu and F.-Z. Shaw, "An EOG-based Sleep Monitoring System and Its Application on On-line Sleep-stage Sensitive Light Control".
- [20 A. Sahami Shirazi, J. Clawson, Y. Hassanpour, . M. J. Tourian, A. Schmidt, . E. H. Chi,] M. Borazio and K. Van Laerhoven, "Already up? using mobile phones to track & share sleep behavior," *International Journal of Human-Computer Studies*, vol. 71, no. 9, pp. 878-888, 2013.
- [21 N. D. Lane, M. Mashfiqui , L. Mu , Y. Xiaochao , L. Hong , A. Shahid , A. Doryab, E. Berke, T. Choudhury and A. T. Campbell, "Bewell: A smartphone application to monitor, model and promote wellbeing," in *Intl. ICST Conf. on Pervasive Computing Technologies for Healthcare*, 2011.
- [22 R. Bhopi, D. Nagy and D. Erichsen, "Can a novel smartphone application detect periodic limb movements?," *Studies in health technology and informatics*, 2012.
- [23 J.-K. Min, A. Doryab, J. Wiese, S. Amini, J. Zimmerman and J. I. Hong, "Toss 'n' turn: smartphone as sleep and sleep quality detector," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 477-486 , 2014.
- [24 S. Dernbach , B. Das, N. C. Krishnan, B. L. Thomas and D. J. Cook, "Simple and] Complex Activity Recognition Through Smart Phones," *Intelligent Environments (IE), 2012 8th International Conference*, pp. 214 - 221, 2013.
- [25 D. Anguita, A. Ghio, L. Oneto, X. Parra and J. L. Reyes-Ortiz, "Human Activity] Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine," *International Workshop of Ambient Assited Living (IWAAL 2012)*, 2012.
- [26 C. Zhenyu, N. Lane, G. Cardone, . M. Lin, T. Choudhury and A. Campbell, "Unobtrusive

-] Sleep Monitoring Using Smartphones," *Proceedings of Pervasive Health*, 2013.
- [27 M. Milošević, M. T. Shrove and E. Jovanov, "Applications of Smartphones for Ubiquitous Health Monitoring and Wellbeing Management," *Journal of Information Technology and Applications*, vol. 1, no. 1, p. 7–15, 2011.
- [28 S. Lawson, S. Jamison-Powell, A. Garbett, C. Linehan, E. Kucharczyk, S. Verbaan, D. A. Rowland and K. Morgan, "Validating a mobile phone application for the everyday, unobtrusive, objective measurement of sleep," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2497-2506, 2013.
- [29 J. M. Kelly, R. E. Strecker and M. T. Bianchi, "Recent Developments in Home Sleep-Monitoring Devices," *International Scholarly Research Network*, Vols. 2012,, 2012,.
- [30 "Fellnr," [Online]. Available: http://fellnr.com/wiki/Zeo_Sleep_Monitor. [Accessed 7 May 2014].
- [31 "Fitbit," [Online]. Available: <https://www.fitbit.com>. [Accessed 7 May 2014].
- [32 "WakeMate," [Online]. Available: <http://www.theurbanshogun.com/2011/06/wakemate-digitizing-your-sleep.html>. [Accessed 7 May 2014].
- [33 "Lark," [Online]. Available: <http://lark.com/Samsung/index.html>. [Accessed 7 May 2014].
- [34 "Sleep Tracker," [Online]. Available: <http://www.sleeptracker.com/>. [Accessed 7 May 2014].
- [35 "UP(Jawbone)," [Online]. Available: <https://jawbone.com/up/>. [Accessed 7 May 2014].
- [36 "Sleep Cycle," [Online]. Available: <http://www.sleepcycle.com/>. [Accessed 7 May 2014].
- [37 S. A. Android. [Online]. Available: <https://sites.google.com/site/sleepasandroid/>. [Accessed 7 May 2014].
- [38 "My Sleepbot," [Online]. Available: <http://mysleepbot.com/>. [Accessed 7 May 2014].
- [39 C. D. Manning, P. Raghavan and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
- [40 I. H. Witten, E. Frank and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, 2011.
- [41 "Weka," [Online]. Available: <http://www.cs.waikato.ac.nz/ml/Weka/>. [Accessed 7 May

] 2014].

[42 "Weka - ARFF," [Online]. Available: <http://www.cs.waikato.ac.nz/ml/Weka/ARFF.html>.
] [Accessed 7 May 2014].

[43 "Fitbit Ultra," [Online]. Available:
] <https://help.fitbit.com/customer/portal/articles/231440-about-the-fitbit-ultra>. [Accessed 7
May 2014].

[44 "Fitbit Ultra Sleep," [Online]. Available:
] [https://help.fitbit.com/customer/portal/articles/176094-how-do-i-use-the-ultra-to-monitor-
my-sleep-](https://help.fitbit.com/customer/portal/articles/176094-how-do-i-use-the-ultra-to-monitor-my-sleep-). [Accessed 7 May 2014].

[45 P. Risenberg and D. Chen, "Fitbit wiki," [Online]. Available:
] <https://wiki.fitbit.com/display/API/API-Get-Sleep>. [Accessed 7 May 2014].

[46 "American Psychological Association(APA)," [Online]. Available:
] <http://www.apa.org/topics/sleep/why.aspx?item=1>. [Accessed 7 May 2014].

[47 "Weka Wikispaces," [Online]. Available:
] <http://Weka.wikispaces.com/ARFF+%28book+version%29>. [Accessed 7 May 2014].

[48 "Android Developers," [Online]. Available:
] http://developer.android.com/guide/topics/sensors/sensors_overview.html. [Accessed 7
May 2014].

Declaration

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.

The electronic copy is consistent with all submitted copies.

Stuttgart, May 13, 2014