Detecting Recurrent Activities in the Context of Obsessive-Compulsive Disorder using Sensors from Smart Devices and Deep Learning

Erkennung wiederkehrender Aktivitäten im Kontext von Zwangsstörungen mittels Sensoren von Smart Devices und Deep Learning

Martin Schlegel

Universitätsmasterarbeit
zur Erlangung des akademischen Grades

Master of Science
(M. Sc.)

im Studiengang
IT-Systems Engineering

eingereicht am 30. November 2020 am
Fachgebiet Digital Health - Connected Healthcare
der Digital-Engineering-Fakultät
der Universität Potsdam

Hauptbetreuer	Prof. Dr. Bert Arnrich
Weitere Gutachter	Prof. Dr. Christoph Lippert

Declaration of Authorship

I hereby declare that this thesis is my own unaided work. All direct or indirect sources used are acknowledged as references.

Berlin, 30th November 2020

Martin Schlegel

Abstract

Mental disorders are nowadays among the most common diseases. One prevalent representative among them is Obsessive-Compulsive Disorder (OCD) with a global lifetime prevalence of 1 to 3 % (announced 2014). Despite various possible treatment options, the disease has a poor chance of being cured. This is mainly due to the very long duration of untreated illness (on average 11 years) and thereby late onset of treatment. Hence, this work aims to develop a novel automatic detection system for OCD to address the early detection and diagnosis. The symptoms of the disease include so-called compulsions, which can appear as physical activities. The idea is to transfer well-researched technologies from the field of Human Activity Recognition into the context of OCD and thus detect the occurrence of compulsions and consequently the illness. A study was conducted collecting data from seven individuals performing simulated OCD activities using sensors in smart devices. Based on this, a system employing an LSTM recurrent neural network architecture was developed to distinguish compulsions from non-pathological behaviour. The deep learning network achieved an F1-Score of 0.53 on the collected data and thus proves that a distinction is possible. This work is the first in this specific topic and is meant to pose a foundation for future research focusing on automatic detection of OCD.

Keywords — obsessive-compulsive disorder | human activity recognition | machine learning | deep learning | recurrent neural network

Zusammenfassung

Psychische Störungen gehören heutzutage zu den häufigsten Erkrankungen. Ein weitverbreiteter Vertreter unter ihnen ist die Zwangsstörung mit einer globalen Lebenszeitprävalenz von 1 bis 3 % (stand 2014). Trotz verschiedener verfügbarer Behandlungsmöglichkeiten hat die Krankheit nur geringe Heilungschancen. Dies ist vor allem auf die sehr lange Dauer der unbehandelten Erkrankung (im Durchschnitt 11 Jahre) und den dadurch bedingten späten Behandlungsbeginn zurückzuführen. Ziel dieser Arbeit ist daher die Entwicklung eines neuartigen automatischen Erkennungssystems für Zwangsstörungen, um die Früherkennung und Diagnose zu verbessern. Zu den Symptomen der Krankheit gehören so genannte Zwänge, die als körperliche Aktivitäten auftreten können. Die Idee besteht darin, gut erforschte Technologien aus dem Bereich der Aktivitätserkennung in den Kontext der Zwangsstörungen zu übertragen und so das Auftreten von Zwängen und damit die Krankheit zu erkennen. Es wurde eine Studie durchgeführt, in der Daten mit Hilfe von Sensoren in Smart Devices von sieben Personen gesammelt wurden, welche simulierte Zwangsstörungsaktivitäten ausführten. Darauf aufbauend wurde ein System entwickelt, das eine LSTM rekurrente neuronale Netzwerk-Architektur verwendet, um Zwänge von nicht-pathologischem Verhalten zu unterscheiden. Das Deep Learning Netzwerk erreichte einen F1-Wert von 0.53 auf den gesammelten Daten und beweist damit, dass eine Unterscheidung grundsätzlich möglich ist. Diese Arbeit ist die erste in diesem speziellen Themengebiet und soll eine Grundlage für zukünftige Forschungen zur automatischen Erkennung von Zwangsstörungen darstellen.

Acknowledgments

I want to thank:

My sister, Alexa Schlegel, for giving me general advice and for helping me over the rock bottom as well as Gerardo Vitagliano, who gave me essential feedback on all of my work.

My parents, Ines Haferkorn and Karsten Schlegel, as well as my grandparents, for supporting my university studies and helping me realising it.

Yara Lochi, for her constant mental support throughout the whole length of the work.

Friends and all the people that even if they were not in my net of friends, made themselves available to help me.

And last but not least the past Martin Schlegel, who enjoyed all the advantages of studying and now let me pay for it.

Contents

List of Abbreviations

1 Introduction

Mental disorders, such as depressions, anxiety or bipolar disorders, are nowadays among the most common diseases. In the year 2014, a meta-analysis published in the International Journal of Epidemiology reviewed surveys providing population estimates of the combined prevalence of common mental disorders [1]. On average one in six adults (17.6 %) experienced a common mental disorder within the 12 months preceding the publication of [1] and almost the double (29.2 %) considering their whole lifetime. According to [2] there has been a constant increase of people living with a mental health disorder from 651 million in 1990 to 947 million in 2016. The share of mental health disorders among all disease burden almost doubled in the same period from 3.48 % to 5.24 %.

One prevalent representative among mental disorders is Obsessive-Compulsive Disorder (OCD), which is the focus of this work. The illness consists of two aspects: a) certain recurrent thoughts, called obsessions, and b) certain recurrent behaviours, named compulsions. Affected people are unable to control neither of them. Often the compulsions are performed to overcome and reassure the anxiety caused by the obsessions [3, p. 235].

In this thesis, I conducted a study to create a dataset containing activity data as well as simulated OCD compulsions. The data is collected by sensors in smart devices. Afterwards, this dataset gets prepared to serve as the basis for the development of a Machine Learning (ML) application. This application is able to distinguish compulsions from non-pathological behaviour.

The scientific contributions of this work are, besides the dataset and the trained model, especially the first insights that could be collected in this field of study.

1.1 Motivation

To be diagnosed with OCD a person must suffer under obsessions, compulsions or both [3, p. 237]. Obsessions are recurrent and intrusive thoughts, urges or images which are experienced unintentionally. They cause anxiety and fear in most individuals and the affected person is often not able to control or suppress them. Nevertheless, the individual tries to neutralise or decrease the effects of the obsession with other thoughts or actions. These actions are called compulsions and are characterised by exact defined (repetitive) behaviour, which can be of physical, e.g. hand washing, or mental, e.g. counting, nature. Sometimes the affected person feels driven to perform such compulsions. Although they are clearly excessive and not comprehensible from an objective point of view, they aim to reduce the effects of the obsessions.
The second criterion for OCD states that the summed up time of an individual suffering for obsessions and compulsions must be at least 1 hour per day [3, p. 238]. A common example for OCD is an individual with strong feelings of fear (obsessions) after touching a surface defined as contaminated, e.g. a door handle. This fear can lead up to thoughts about losing a dear person by an infection caused by the contaminated hands. To prevent this from happening, and to overcome the obsession, the affected person needs to wash their hands right away in a clear defined hyperbolic manner (compulsion).
In 2013, the worldwide 12-month prevalence of OCD was between 1.1 % and 1.8 %. A lifetime prevalence of 1 to 3 % of the world's population was announced in the year 2014 [4, p. 9]. The numbers are similar across different cultures and countries. Additionally, OCD was indicated as under-diagnosed and under-treated [4, p. 6], thus the estimated number of unreported or undiagnosed cases is even higher.
The impact of the illness on the life of an affected person is generally speaking a reduction in the quality of life. As previously mentioned, OCD is by definition time-consuming. The invested time varies across individuals and can reach from 1 to 3 hours per day for a moderate course of the disease up to a nearly constant disturbance in extreme cases [3, p. 238]. Plus, there are the caused fears and unpleasant feelings, which can include also panic attacks. Thereby, it is common for affected people to avoid situations (people, places) that can trigger or increase the illness. For example, the individual from the previous example could refuse to meet dear people in the fear of infecting them or could miss

important appointments by the need of washing the hands. All of this leads up to suicidal thoughts in every second person with OCD [3, p. 240].
If OCD remains untreated, the development is in most of the cases chronic including stronger and lighter episodes and the remission rates for adults are low [3, p. 239]. Therefore, the disease should be actively treated. Some studies reporting symptomatic remission rates ranging from 32 % to 70 % [5]. Another study including 213 adult participants over a course of five years observed a cumulative remission rate of 38.9 % [6], which translates to no improvements in 61.1 % of the cases. In total, only 16.9 % of the subjects achieved full remissions. Therefore, OCD is classified as a persistent disorder [5].
Even though it is widely known that the common kinds of treatments are way more effective the sooner after onset the disease is diagnosed and therefore the treatment is started [5], [7], the illness is in the majority of cases diagnosed very late. With an average time of 11 years between the occurrence of first evidence and the first appropriate treatment [7], OCD has one of the longest duration of untreated illness (DUI) among all psychiatric disorders [5]. This requires improvement, especially for the early detection and diagnosis. On the one hand, to improve the rates of remission in general, and on the other hand to be able to target full remission as the treatment goal in more cases [6]. Since OCD has high relapse rates ranging from 25 % to 60 % (higher for individuals who achieved only partial remission) [6], another area of application would be the detection of relapse [8].

1.2 Methods and Research Goals

The field of Human Activity Recognition (HAR) is very well researched and has made great progress in recent years [9]–[11]. Systems based on body-worn sensors are one of the most common modalities and are defined as: ‘Sensor-based activity recognition seeks the profound high-level knowledge about human activities from multitudes of low-level sensor readings’ ([12]). Former research has shown that it is possible to recognise and classify activities in the context of sport (e.g. walking, climbing stairs), activities of daily living (ADL) (e.g. folding clothes, brushing teeth) and eating (e.g. drinking from a cup, eating pasta) by using sensor data [13]. As stated before, compulsions are defined as exact defined (repetitive) behaviour, which can include also physical activities like hand washing or checking home appliances. Building on that, it is possible to

adapt the existing solutions of HAR to detect some of these compulsion related activities and to develop a basic Compulsion Related Activity Recognition (CAR) system.
The system to be developed is based on data and devices that occur in the real-world [14]. Smartphones and wearables, such as smartwatches, are ideal for this. On the one hand, because previous work has shown that this is possible [13]. On the other hand, the mean age at onset of OCD is 19.5 years [4, p. 37]. Therefore, it can be assumed that a large number of affected people are in ownership of these devices. Checking the same door several times consecutively could indicate pathological behaviour. While walking through and checking different doors when moving within a building can be classified as normal behaviour. Due to the strong similarity of the mere activities, it is a great challenge to distinguish normal from pathological behaviour. The Indoor Localisation (IL), e.g. on the basis of Wireless Local Area Network (WLAN), is related to HAR as well [10]. In the context of OCD, the IL is maybe a way to support and improve the differentiation.
Thus, I will create a sensor-based dataset including compulsions related activities, because to my knowledge there is no dataset fulfilling the requirements. As research in this area is still in its infancy, the main focus of this thesis is on developing a proof of concept (POC). Therefore the compulsions related activities will be simulated. Furthermore, I will examine how appropriate methods from the background of HAR are suitable for the detection of repetitive compulsion related activities. Following research questions are proposed to reach the goal:

Is it possible to develop an automated system to detect and address OCD by means of recognising the activity of patients using a dataset including sensor data from smart devices?

- *How to conduct a study and gather data for this purpose, especially without having access to affected people?*
- *What sensors, signals and devices should be used?*
- *How to create a usable dataset from the gathered data (data fusion from different devices)?*

- *What ML models are feasible to solve the problem and what parameters are necessary?*
- *How can IL be leveraged to improve such a system?*

This work is meant to serve as a POC by establishing a foundation for future research as well as giving first insights into the capability of adapting HAR to the context of CAR. The long term goal is to raise awareness and shorten the DUI of OCD.

The methodology of this work is outlined in Chapter 4 and consists of two major parts. Initially, the creation of a dataset is explained. The remainder deals with the development of a ML solution using the dataset of the first part.

1.3 Structure

The thesis is structured as follows. Chapter 2 explains the relevant theoretical background. Additionally, Chapter 3 provides an overview of related work on HAR. The methodology, reaching from study design over data collection and exploration to the development of a machine learning model, is detailed in Chapter 4. Particulars of the implementation are explained in Chapter 5. The results of the developed machine learning model are evaluated in Chapter 6. The achieved results are summed up and discussed in Chapter 7. Chapter 8 draws a conclusion including limitations and opportunities for future work.

2 Theoretical Background

This chapter provides the theoretical background regarding my thesis and gives needed knowledge for further understanding. Overall, the thesis spans across various areas, which are in detail explained in the following sections. Section 2.1 describes OCD, as it is the area of application of this thesis. Section 2.2 describes the use of smartphones and wearable devices for HAR. Section 2.3 describes IL as the technique used for collecting data, and Section 2.4 summarises ML and Deep Learning (DL) models used to solve the compulsion detection tasks.

2.1 Obsessive-Compulsive Disorder

OCD is a mental disorder and 'is characterised by the presence of obsessions and/or compulsions. Obsessions are recurrent and persistent thoughts, urges, or images that are experienced as intrusive and unwanted, whereas compulsions are repetitive behaviours or mental acts that an individual feels driven to perform in response to an obsession or according to rules that must be applied rigidly' ([3, p. 235]). In the past, the illness was considered as closely related to (or as a subgroup of) anxiety disorders. But with new scientific insights, OCD is nowadays seen as a representer of a novel group of disorders and was given an own category named 'Obsessive-Compulsive and Related Disorders' in the updated, fifth version, of the 'Diagnostic and statistical manual of mental disorders – DSM-5' [3]. Closely related disorders are for example the trichotillomania (hairpulling disorder), the excoriation (skin-picking) disorder, the body dysmorphic disorder or the hoarding disorder.
The criteria of diagnosis require the presence of obsessions, compulsions, or both [3, pp. 237–238], [4, pp. 3–5]. Obsessions are recurring, intrusive thoughts (e.g. contamination or sex-related), images (e.g. violent or horror phantasies) or impulses (e.g. to murder someone). They appear in an intrusive way, are unwanted and causing fear and distress in most individuals. Therefore, the affected person tries to ignore or suppress the obsessions, e.g. by other positive annotated thoughts or by executing compulsions. Compulsions can be seen as

rituals and are defined by recurring behaviours (e.g. cleaning (hand-washing, showering), checking) or mental acts (e.g. praying, counting). The individual feels the need to execute them, instantly and in a strictly defined way, as a reaction to an obsession. Although from an objective point of view compulsions are not directly linked to the obsessions (e.g. ordering items to prevent harm) and seem to be of excessive characteristic, the initial effects of the obsessions can be reduced or neutralised. Thereby, the acts of the compulsions are not pleasant itself.

The second part of the diagnostic criteria puts the focus on the time consumption of the disorder [3, pp. 237–238]. It states that the combined time of obsessions and compulsions is at least 1 hour per day and/ or they have a huge negative impact of important areas of life, like family or work. This helps to distinguish the disorder from normal behaviour, like the occasionally double-checking of doors, and add a more clear border around it. The hours per day spend for the illness can vary among affected individuals a lot and ranges from 1 to 3 hours per day for mild and moderate up to constant disturbances in extreme cases. In addition, the obsessive-compulsive symptoms must not be due to drugs or medication and not be better described by other mental disorders.

An important indication for the severity of the sickness is the degree of insight an individual has in the plausibility of the symptoms, especially in the purpose of the compulsions [3, p. 238]. The majority of affected people have a *good insight* (e.g. The house will probably not burn down when I do not check the stove 30 times.), whereas some individuals have a *poor insight* (e.g. The house will probably burn down when I do not check the stove 30 times.) or even an *absent insight* (e.g. The house will definitely burn down when I do not check the stove 30 times.).

The types of obsessions and compulsions can be very different among various individuals, but the majority of obsessions are contamination-related. Consequential, the associated obsessions are oftentimes of type cleaning and washing [4, p. 3]. Other types for obsessions can be superstition-, religion-, control-, harm- or perfectionism-related. Different typical kinds of compulsions are related to checking, repetition, or mental compulsion.

OCD can have a strong influence on the life of an affected person, and generally speaking reduces the quality of life [3, pp. 239–241]. Individuals have to suffer from the fear or strong feelings of disgust induced by obsessions, which can lead up to panic attacks. Moreover, many individuals have to deal with compulsions,

which can consume a lot of time or can bring the person in uncomfortable public situations. That is why many affected people try to reduce the triggers causing the symptoms. This can lead to the avoidance of public places like public restrooms or doctor's offices in case of contamination concerns. Others might avoid social interactions with loved persons or family members in the fear of causing harm to them. Symmetry or perfectionism related symptoms can lead to problems in school or work [15], for example when project deadlines can not be met because the project never feels 'just right'. This all leads to symptoms of depression [4, p. 1], and/ or suicidal thoughts in up to 50 % of the suffering people, which convert to actual attempt in half of the cases [3, p. 240].
The global 12-month prevalence of OCD was in the year 2013 between 1.1 % and 1.8 % [3, p. 239]. In the following year, a lifetime prevalence of 1 to 3 % was determined [4, p. 9]. No significant cultural differences were identified and OCD is diagnosed all over the world. It has been assumed that the estimated number of unreported cases is higher because the disease is most likely under-diagnosed and under-treated [4, p. 6]. Children can suffer from the illness as well, but the mean age at onset of OCD is about 19.5 years. Among children, males are more affected, whereas the reverse is true for adults.
If OCD remains untreated, the development of the sickness is in the majority of cases chronic and can have waxing and waning symptoms (also in the degree of insight) [3, p. 239]. Usually, the illness proceeds in an episodic manner, in exceptional cases in a deteriorating course. Overall, the remission rates under adults are low (20 %). Therefore, OCD is classified as a persistent disorder [5] and should be treated actively. First-line treatment options are serotonin reuptake inhibitors (SRIs) and cognitive-behavioural therapy (CBT), specifically exposure and ritual prevention (ERP), which can be applied also in combination. Poorer insights are correlated to worse long-term consequences and with an overall lower rate of remission [6]. An important point is a differentiation between the remission of symptoms and a full recovery, defined as the complete absence of symptoms. Full recovery is not always the main objective of therapy but is associated with a lower risk of relapse [5]. Section 1.1 contains a few more details and numbers.
Because of the chronic and oftentimes slowly degrading character of the illness, the common kinds of treatments are more effective the sooner therapy is initiated [5], [7]. In contrast to that, OCD has one of the longest DUI among all psychiatric disorders of 11 years [5]. Reasons for that are a still existing social

stigma for mental disorders in general [8], and senses of shame (especially in combination with sex- or religious-related obsessions) [4, p. 1], [16]. Throughout the sickness, individuals can get very skilled at hiding their symptoms [8]. Other reasons are that people often do not know where to find help [7] or simply that the health professional did not screen for OCD [4, p. 1].

2.2 Wearables and Smartphones

My proposal is to target early recognition of OCD with the use of smart devices, therefore I would like to give a brief overview of the topic. 'Wearable devices are defined as devices embedded within clothes, watches, or accessories' ([14]). There are several other elements, such as sensors, actuators and controllers, which are additionally required for being called wearable[17]. With this very general definition, the term wearables cover a wide range of tools/ devices including smart glasses, all kinds of smart fabrics, as well as contact lenses or hearing aids. The group of wrist-worn devices (WWDs), such as smartwatches or fitness wristbands, has the greatest popularity [14].
The market of wearables is quite new and has a large economic potential. The main sectors are health, sports and fitness, work safety, and fashion [17]. The field of wearables is becoming increasingly important in science as well, which gets reflected by the rising amount of conferences and research groups which deal with this topic. The principal fields of application are in the area of HAR and feeling or affect detection [14].
For the development of applications, a wide range of sensors can be accessed on the market. These are for instance, the accelerometer, the gyroscope, the magnetometer, the electric potential sensor, the ambient light sensor, the heart rate sensor or the barometer, just to name a few. When the sensors are read out over a period of time, a continuous sequence of values or readings ordered by a time parameter is generated. This type of data is called time series data [18, p. 1].

A smartphone is 'a mobile phone that can be used as a small computer and that connects to the internet' ([19]). The first device which was probably called 'smartphone' was developed in 1992 by Canova, but the real commercial success of the device group started in January 2007 with the introduction of the first iPhone. Nowadays, smartphones are characterised by a very high computing power, which is similar to that of conventional computers. In general,

they are equipped with a similar set of sensors like wearables, but often without sensors that require direct skin contact.

In the context of HAR, the biggest difference between wearables and smartphones is the position in which the devices are worn on the body in the real-world. This means that both device types are differently suitable for recognising various activities. Besides, smartphones can also perform complex real-time evaluations of the collected data due to their high computing power.

2.3 Indoor Localisation Systems

The data collected with smartphones and wearable devices can be integrated with the location information to improve the detection rate of systems. Therefore this chapter gives a short introduction to this area. 'IL is the process of obtaining a device or user location in an indoor setting or environment' ([20]). Due to the increasing spread of smartphones and other wireless devices in recent years, the demand and range of IL systems and services grew as well. Common areas of applications are the health sector, all kinds of industry, disaster management (e.g. IL systems for firefighters during operations), building management or surveillance.
The usage of outdoor localisation techniques, such as the satellite-based Global Positioning System (GPS) as the most common representative of this group, have a lack of precision or fail completely when it comes to its plain indoor application [21]. Buildings prevent a line-of-sight transmission between receivers and satellites and indoor environments are in general more complex and having therefore different requirement profiles. Indoor surroundings include obstacles, such as walls, furniture or human beings which can influence transmitted signals. Other sources of noise and inference can be electrical equipment or wireless networks.
To obtain an IL system, various signal metrics and techniques can be used:

Time-Based Approaches Several techniques are based on the measuring the time needed for the signals to travel between receiver and transmitter, for example, Time of Flight (ToF) or Time of Arrival (ToA), Time Difference of Arrival (TDoA) or Return Time of Flight (RToF). The approaches differ in the way which transmissions routes are used for time measuring and in how they are processed and calculated. Measurements between the device

and at least three reference nodes are needed to calculate a location with respect to the reference nodes in a two-dimensional space (four reference nodes in a three-dimensional space).

Angle of Arrival Another approach is named Angle of Arrival (AoA). Here, using multiple antennas on the receiver site, the time difference of arrival at the multiple antennas can be used to calculate the angle a signal is coming from. The angle of at least two reference nodes can be used to calculate the relative location in a two-dimensional space, while three angles are needed in a three-dimensional space.

Phase of Arrival Similar to the AoA approach, multiple antennas can be used to measure the phase difference at different antennas of the same signal. Techniques based on that are grouped under Phase of Arrival (PoA).

Received Signal Strength Systems based on Received Signal Strength Indication (RSSI) are amongst the simplest and most widely used approaches. The Received Signal Strength (RSS) is the actual signal power measured at the receiver and can be used to estimate the distance a signal travelled. The RSSI is a relative measurement of the RSS and defined by the hardware vendor. Similar to the time-based approaches the distances to several reference points can be used to obtain the relative location of a device.

Fingerprinting Fingerprinting is a technique which requires pre-knowledge before the system is capable to obtain a location. During an initial offline phase, a grid of locations to be learned is defined. For every defined location, multiple (usually) RSSI measurements to all available reference points are collected (called fingerprints). In the following online phase, the live gathered fingerprints are compared to the previously taken offline fingerprints and the user's location is estimated. Because online fingerprints are mapped to previously defined and learned positions, these systems provide a discrete estimation of locations rather than a continuous one. This is a major difference to all previously described techniques. Theoretically, a continuous localisation can be simulated using a very high granularity of defined locations. But there is an important tradeoff between the density of defined locations and the probability of successful location estimations, which should be kept in mind when locations to learn are chosen in the initial offline phase. Another drawback is the high

influence of changes in the environment over time, which can change the fingerprint at locations and complicate the measurements made between offline (old environmental setup) and online (new environmental setup) fingerprints. There exist different algorithms and methods to compare the offline and online fingerprints, for example, probabilistic methods, artificial neural networks (ANNs) or neural networks (NNs), k-nearest neighbours algorithm (k-NN), or Support Vector Machine (SVM).

Independently to the techniques, several technologies can bee used to gather the data needed for position determination [20], [21]. These are for example 1) Ultra-Wideband (UWB) which is more robust to noise from other signals, due to the usage of a different signal type and radio spectrum. 2) Ultrasound, which is inspired by bats. 3) Acoustic signals on the base of common speaker and microphone hardware. 4) Vision-based systems using cameras. 5) Magnetic or Infrared (IR) based systems, or 6) Radio Frequency (RF) technologies, such as Radio Frequency Identification (RFID) or WLAN and Wi-Fi. General advantages when using these technologies are that the radio waves can go through obstacles and therefore need less hardware and having a large coverage area in comparison to other systems. The systems can also reuse already existing hardware such as WLAN routers [20]. Time-based approaches and fingerprinting are commonly used in combination, whereby fingerprinting is an effective method when it comes to more complicated indoor environments.

2.4 Machine and Deep Learning

ML is 'defined as computational methods using experience to improve performance or to make accurate predictions' ([22, p. 1]). Thereby, experience means the information given to the machine to learn from. This information is typically previously collected data, preprocessed for this kind of analysis. The quality and quantity of this data are significant for the accuracy of the predictions made.
Typical fields of application are for example: 1) Text or document classification, such as automatic spam detection of e-mails or the identification of inappropriate contents of web pages. 2) Speech processing systems, which include speech recognition and speaker identification tasks. 3) Computer vision applications, such as object recognition, face detection or pose estimation of humans. 4) HAR systems, which includes automatic fall detection or the identification of ADL like walking, sitting or eating. 5) Other fields are fraud detection in the background

of online banking and payment, chess engines[1], or recommendation systems like search engines. This is only a small overview, many more problems can be encoded into ML tasks and the area of application is constantly increasing [22, p. 2].

One standard task for ML is for example *Classification*, which is about mapping a class to a given item. For instance, a task can be to assign the status spam or no spam to an e-mail (item). The number of different classes for classification tasks can vary from several hundred down to two. Tasks including only two categories are named binary classification tasks. *Regression* specifies tasks of predicting real values such as future stock values or the rain probability on the basis of previously made observations (e.g. air humidity or temperature). In *ranking*, the problem of ordering given items according to criteria, such as the return of a web search, is handled. *Clustering* specifies the problem of segmenting a bunch of items into subgroups and is useful for the analysis of very large datasets and social network analysis. In the end, the overall goal of all ML tasks is to make the best possible predictions for new unseen items. Thereby, they aim to be robust and general working, and not just for the data it was trained with [22, p. 3].

Depending on the data available during the learning, the way and order the data is given into the model, as well as how the process is evaluated, ML can be classified into various scenarios. These are: 1) In *supervised learning*, the given training samples are fully labelled. Predictions are made for all samples in the test set and a loss function calculates the deviation between the predicted and the actual result. The model is iteratively optimised to reduce the loss function. The general process and terms are explained in detail later. This scenario is especially used for classification and regression tasks. 2) In *unsupervised learning*, the training samples are not labelled at all. The process is similar to supervised learning but has the advantage that it can also work with unlabelled datasets. Sometimes it is too expensive to label samples or the classes included are unknown. Clustering is an example of unsupervised learning. 3) *Semi-supervised learning* is a combination of the supervised and unsupervised learning approaches, where the learning process is given labelled and unlabelled data. This can be the case if it is very expensive or time-consuming to label a complete dataset, which is why only a section is labelled. 4) In all three former described scenarios, the loss is calculated after seeing the whole training samples. In *on-line learning*, the calculation of the loss is done after each unseen example. [22, p. 6]. This

1 like https://stockfishchess.org/

represents only a selection of scenarios. In practice, there can also be hybrid approaches. For example, when the calculation of the loss does not happen after seeing one (on-line learning) or all (supervised learning) of the samples, but after a fixed amount in between, like 64 samples. This number is called *batch size* and can be another hyperparameter. In the case of the on-line learning scenario, the batch size is set to 1.
In the following, I will give an overview of the development and evaluation of ML models. Hereinafter is a list of basic terms and definitions commonly used in the field of ML [22, p. 4]:

- *Samples:* The dataset used for learning and evaluation, in the definition cited in the beginning of this section referred to experience.
- *Features:* Features are a set of attributes associated with and derived from an example. In the application of HAR, this could be the maximal or mean value of a sensor in a given period.
- *Labels:* Labels are values or identifiers of an example and represent the goal to be found. For a classification task, this is the class of the example. In case of a regression, this could be a real number.
- *Hyperparameters:* Hyperparameters are configuration options for the ML model. Normally, they depend on the problem to be solved and the chosen model itself.
- *Train, validation and test samples:* The training, validation and test samples are different subsets of the samples serving various purposes. The training samples are used as direct input during the learning phase of the model. They can differ for various kinds of learning types, as mentioned before. The validation samples are used to evaluate the actual performance of the model and to tune the hyperparameters. They are a subset of the test samples and only used when working with labelled data. The test sample is used after the learning stage is finished to evaluate the true performance. Therefore, the application has to predict the labels of new, unseen samples. The predictions are compared with the actual labels in the test samples.
- *Loss function:* The loss function measures the difference between a predicted label and the actual label. It is a hyperparameter and needs to be chosen depending on the task.

- *Hypothesis set:* The hypothesis set is the function or a set of functions taking as input the features of an example and calculating the predicted label. This is the solution under optimisation.
- *Layer:* A subgroup of ML models are called NNs. These networks are usually composed of different layers. The results of one layer are transferred to the next one as input. The first layer of a network is called input layer, the last one output layer, respectively. Layers that are located between these two are called hidden layer.
- *Optimiser:* The optimiser or optimisation function is a hyperparameter and is responsible for updating the parameters under training depending on the results of the loss function.
- *Learning rate:* The learning rate is a hyperparameter used to control the rate at which an optimiser updates the parameters under training.

The development of an ML model is an iterative process and starts with the dataset of samples. This dataset has to be cleaned and prepared if necessary before it is split up into subsamples of training and test samples. Depending on the actual problem, a bunch of training samples are grouped into a separate set of validation samples. For example, the size of the subsamples depends on the total amount of available data and the number of hyperparameters that one want to train and optimise. In the next step the features for each example are calculated. This is a very crucial step in ML because the set of chosen features can have a high impact on the overall performance of the model. Thereby, the selection of features is oftentimes left to the user, who needs to have deep knowledge and understanding of the problem and the available data to make good decisions. After the decision for a suitable scenario (depending on the task) and the choice of the ML model, the training phase is started. In this phase, the model learns and optimise its hypothesis set using the training samples. The current loss is calculated using the selected loss function and the performance is measured using the validation samples. This phase is repeated with a different combination of hyperparameters. The best performing hyperparameter setup on the validation samples is selected and finally evaluated using the test samples, defined in the beginning. Here, the model predicts the labels of the test samples using the optimised hypothesis set. Using again the loss function, the overall error between the predicted and true labels is calculated and used as the final

evaluation base [22, p. 5]. Figure 2.1 gives a broad overview of the process.

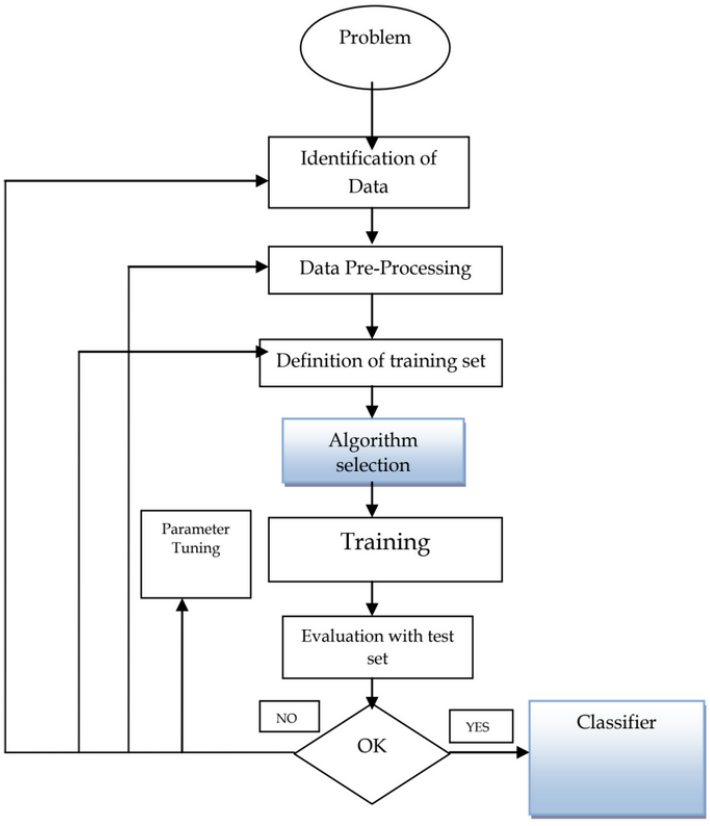

Figure 2.1: The fundamental overview of the training process of a machine learning model by the example of a supervised classification problem. This figure is taken from [23].

DL is a subfield of ML, since it has a similar principle process as described before. The networks used in DL have several hidden layers and are in general deeper and computational more complex than ML models. The idea behind that is to include the calculation of the features, which is done as a (manual) separate step before the training stage in ML, into the learned model. Starting from raw data as input, each layer tries to learn features with increasing abstraction and complication [24, p. 307]. Oftentimes, especially when working with a lot of data,

the user does not know the structure of the data and is not able to analyse all hidden dependencies and characteristics. In other words, it is very hard to gain enough domain knowledge of the problem and the data to design a reasonable set of features. That is why DL tries to automate this process, which reduces the required human contribution to a minimum [24, p. 308]. A downside of DL is the need for more data and the increased training time for the greater amount of parameters under training. In the recent years, DL has become more feasible and more widely used, partly because of the availability of more data and sufficiently cheap computing power [24, p. 309]. It is assumed that DL will gain further popularity and that it will be applied to even many more tasks and areas of application [24, p. 311].

Miscellaneous

Under this chapter, I summarise all other important aspects related to ML concerning this work.

Overfitting/ Generalisation

ML is fundamentally about generalisation, as it tries to establish general rules based on a limited set of samples to make predictions for unseen data [22, pp. 6–7]. If the created hypothesis set is too much adapted to the specifics of the samples used for training, it can happen that predictions for new samples are predominantly wrong. This effect is called overfitting. One way to recognise it is to use a validation set during training. The moment the result of the loss function for the validation set continuously increases, but the value of the training set decreases further, the model starts to overfit.

Imbalanced Datasets

Datasets do not always have a perfect distribution of samples for all included classes. If there is a strong imbalance between the individual classes, e.g. 90 % of class 'A' and only 10 % of class 'B', these datasets are named imbalanced datasets. This may lead to certain classes being underrepresented in the calculation of the hypothesis set. There are different ways to deal with it. One possibility is re-sampling, which is divided into under- and over-sampling. With under-sampling, samples of the over-represented classes are removed from the training set until

a balance is reached. The problem with this method is that available information is not used or that the resulting dataset can get too small. In over-sampling, samples from the underrepresented class are reused until there is a balance between the classes. This does not generate any new insights and noise may be amplified. Another option to work with imbalanced datasets is the usage of class weights. This allows the classes to be given weights which either increase or decrease the influence of the corresponding samples in the calculation according to the share of the classes.

Input Encoding

Some ML model can only consume data in form of numerical values. Therefore, features with different data types have to be mapped to numerical values. There are two possibilities for this. The first one is integer encoding, where each value of a feature is assigned an integer value. This can be used if the original values contain a natural order, which is maintained by the integer values. The second option is one-hot encoding. Here, the original feature is removed and for every value of the feature, a new separate feature is introduced. The values of the new features are either 0 or 1, depending on the value of the original feature for a sample.

Dropout

As stated before, overfitting and the lack of proper generalisation can be a problem in ML. This may in particular be the case for DL networks [25]. Dropout is a technique for addressing this problem. By dropping randomly units from the neural network during the training process, they get prevented from developing too much co-adaptions. This forces the units to learn useful features on their own without relying on others to correct them. The technique introduces an additional hyperparameter, called dropout rate, which indicates the percentage of units that are randomly dropped.

Metrics

The following metrics can be used to evaluate ML models in (binary) classification tasks.

The **accuracy** is defined as

$$accuracy = \frac{true\ samples}{total\ samples} = \frac{TP + TN}{FP + FN}^{2} \tag{2.1}$$

is the percentage of correctly classified samples. This metric gives a general insight but is not feasible for highly imbalanced datasets.

Precision is the percentage of predicted positives that were correctly classified, defined as

$$precision = \frac{TP}{TP + FP} \tag{2.2}$$

Recall is the percentage of actual positives that were correctly classified, defined as

$$recall = \frac{TP}{TP + FN} \tag{2.3}$$

The **F1-Score** is the harmonic mean of the precision and recall and combines both values in one. It is defined as

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.4}$$

Logistic Regression

Logistic regression is a statistical deterministic model that uses a logistic function to model a binary dependent variable. It is one of the most well-known algorithms for binary classification [22, p. 325].

2 In our problem, true positives (TP) and true negatives (TN) are the samples that were classified correctly as OCD and non-OCD. False positives (FP) and false negatives (FN) are the samples that were classified incorrectly.

3 Related Work – Human Activity Recognition

Since this work makes use of methodologies from the field of HAR, the following chapter gives a brief overview of the topic. In particular, the results and findings of a literature research are presented.

Activity recognition or Human Activity Recognition (HAR) is the research area which aims to automatically recognise physical activities [26]. Thereby, the recognition itself is never the final goal of an application but serves as one important step in a broader system. Typical work focuses for example on the detection of ADL, sports activities or sleep analysis. A current developing trend is to use HAR in medical and health areas such as trauma resuscitation, fall detection or elderly monitoring [9], [12].
The research field can be divided into two types: *Video-based* systems which analyse images or videos to detect the movements of humans and *Sensor-based* systems that use data from sensors like the accelerometer, the gyroscope or Bluetooth to achieve the same [12]. The *Sensor-based* systems can be divided according to different sensor modalities. *Object sensors* are attached to specific objects to detect the interaction with them. The interaction between humans and their environment can be captured using *Ambient sensors*, such as sound, pressure or temperatures sensors. One of the most common modality in HAR, especially in combination with DL architectures, is called *Body-worn sensor* [12]. Here, the sensors are placed on or near the human body and can infer activities directly. The literature research concentrates mainly on work of this modality since both, smartphones and smartwatches, belong to this category.

Studies that use inertial sensors, in particular sensors in smart devices such as smartphones, smartwatches and wearables, for HAR are relevant for my work. An additional medical reference and the focus on ML or DL approaches are favourable. Since HAR is a large field of study that has gained rapid popularity and made great progress in recent years, this literature review was focused on survey and review papers published in the last five years. Points

of interest were mainly the equipment and sensors used, the preprocessing of the raw data, the most used ML and DL architectures and the metrics used for performance measurements as well as general common and best practices:

1. **Devices and Sensors**
   All cited studies and papers refer to smartphones as the most frequently used devices. The waist and trouser pocket is seen as a good place to recognise full-body movements [9]. The most commonly used sensors are the accelerometer followed by the gyroscope [11], [12]. One problem is in the different positioning and orientation of the devices [10], [11], which can be solved by using the magnetometer and the magnitude [11]. Depending on the activities to be detected, other additional sensors are also suitable, such as Global Positioning System (GPS), Bluetooth or the microphone [10]. In general, the fusion of several sensors and the use of smartphone and wrist-worn devices in combination is advantageous, especially for the detection of more complex activities [9]–[11], because it cannot always be assumed that a smartphone will be carried in the pocket [13]. A sampling rate of the sensors between 20 and 50 Hz is seen as sufficient to avoid under-sampling of the data [9], [10], [13].

2. **Preprocessing**
   Since the data are recorded by sensors and contain errors and noise, cleaning methods such as the Kalman, lowpass or moving average filter should be used [9], [11]. The window lengths given to the networks depend on the length of the activities to be recognised [10], lengths of 2 to 5 seconds have proven effective [9].

3. **DL architectures**
   All papers put the focus on DL architectures and suggest the use of these more advanced technologies. It is reported that various techniques have been applied successfully, such as the Restricted Boltzmann Machine (RBM), the deep neural network (DNN), the convolutional neural network (CNN), the recurrent neural network (RNN), the deep autoencoder, sparse coding or also hybrid technologies [9]–[12]. CNN and RNN architectures, such as the long short-term memory (LSTM), were the most frequently used, with CNN being particularly suitable for long-lasting activities and RNN for short activities with natural order [9], [10], [12].

4. **Metrics**
   Suggested metrics for assessing the performance are the accuracy, the precision, the recall and the F1-Score [9], [10]. Besides, [9] advice to use the sensitivity and specificity as well.

In [13] a study with 17 subjects was conducted to perform smartwatch-based activity recognition using an ML approach. As activities, 18 mainly hand-focused ADL were defined. Each participant was equipped with a smartwatch on their dominant hand and a smartphone in their front-right trouser pocket. The data was gathered from the inertial acceleration and gyroscope sensors using a sampling rate of 20 Hz. Each activity was performed from every subject for two consecutive minutes. After trimming the raw data, higher-level features were generated for windows of ten seconds length. Using this data various ML models were trained. Finally, the achieved accuracy was compared between the different ML models, sensors, devices and model types (personal and impersonal). The personal models outperformed the impersonal ones and the models trained with the smartwatch sensor data achieved better results than the models using data from the smartphone sensors. The reason given was the hand focus of the activities.

In [27] the correlation between overlapping and non-overlapping windows in the context HAR using inertial sensors was investigated. Special reference was made to the evaluation techniques, namely to subject-dependent and subject-independent cross-validation (CV). They concluded that improvements with overlapping windows have to do with limitations in the subject-dependent CV. When a subject-independent CV is used overlapping window approaches do not result in improvements over non-overlapping window approaches, but require more resources.

4 Methodology

In the following chapter, I describe the methodology I used to achieve the research goals. In Section 4.1, the study design is described. Afterwards, I explain the data collection process in detail in Section 4.2 followed by the process needed to form a usable dataset in Section 4.3. In Section 4.4, I report on the data exploration process which serves as a foundation for the insights given in Section 4.5, which is about the development of a machine learning pipeline to recognise OCD events and corresponding models.

4.1 Study Design

The goal of this thesis was the implementation of a non-invasive application capable to identify OCD related activities or compulsions, therefore a dataset with special properties was needed which to my knowledge, does not exist at the time. Thus, the decision was made to collect data and to create a new dataset. As stated in Section 1.2, the approach in this thesis was to serve as a POC. Therefore, it was no strictly necessary to work with real data and it was not reasonable to gather data from real OCD patients before assessing the efficiency of the proposed solution. Additionally, there is much more room for adjustment and adaption when gathering simulated data. Hence, the decision was made to create a dataset including simulated activities. According to a study among OCD patients done in [15], 50% of the participants showed compulsions of type *Checking* and 39.8% of type *Cleaning/ Washing*. Based on these frequencies of main compulsions, the examples of common symptoms ([4, p. 5]), and the general gathered knowledge about OCD (Section 2.1), a subset of simulated activities has been defined:

Checking an Oven To check an oven, the oven is opened with one hand. The very same hand is used to reach into the oven and moved from left to right to feel and check the temperature. Afterwards, the oven is closed again with the same hand. Altogether this should take approximately 5 seconds.

One whole *Checking an Oven* activity consists of 3 rounds of single checks, hence a whole activity takes about 15 seconds.

Checking a Door To check a door the door handle is pushed down with one hand and the door is opened slightly. The door is closed again and the door handle is released. This is repeated 3 times, with the 3rd time the door is opened completely and the door is passed. The closing procedure is similar to the opening procedure, whereby the door is finally closed with the 3rd repetition. The opening and closing procedures are performed with the same hand. The total duration should need approximately 15 seconds.

Washing Hands The simulated hand washing activity consists of 7 different consecutively performed movements, each lasting approximately 3 seconds. These single movements are: 1) rubbing the palms of the hands together 2) rubbing the outsides of the hands together 3) washing the right thumb with the left hand 4) washing the left thumb with the right hand 5) rubbing the left fingertips on the right hand's palm 6) rubbing the right fingertips on the left hand's palm 7) rinsing of soap. The activity can be performed with or without soap or water and should take between 22 and 25 seconds.

Next to the simulated compulsions, the dataset had to include non-OCD related data, which was not defined in detail. Following the definition of OCD given in [3], the sum of the time used per day for obsessions and compulsions must be at least 1 hour. To reflect this relation between OCD and non-OCD periods the dataset had to consist of around 4% simulated OCD activities and around 96% non-OCD related activities.
Another important point was that the application had to work with data gathered in a real-world setting, using hardware available and used as in real-world scenarios. According to a forecast owning today (as of 2020) 3.5 billion people a smartphone [28], which corresponds to 44.85% of the world's population. As they are equipped with a great variety of sensors, used daily and often carried near the body, smartphones represent a valid tool for data collection in real-world contexts covering full body movements. As the former defined simulated OCD activities are heavily hand focused additional hand-related data is required. Wearables, such as smartwatches, are suitable devices to collect such data. At present (2020), their forecasted distribution of 600 million devices [29] corresponds to 7.9% of the

world's population. Besides, the diffusion of both device categories is predicted to rise in the next years. Considering such widespread diffusion, smartphone and smartwatch were used to gather data.
According to [11] the accelerometer and the gyroscope are feasible sensors for HAR. The magnetometer can help reduce the negative impact of gravity, offset slightly different sensor positions and provide orientation independence.
All of the former defined OCD activities are location-sensitive, e.g. *Washing Hands* has to happen near a sink, *Checking an Oven* must be performed near an oven and the *Checking a Door* activity implies a location change. The paper [30] has shown that it is possible to develop a HAR system based only on location data. Thus, the dataset should get enriched with location data to improve the capabilities of the application. In a real-world context, an IL system based on WLAN is a viable solution, since WLAN technology is supported by all smartphones. In addition, in 2014, more than 50% of the households in first world countries like the USA, Japan or French were equipped with WLAN [31].

4.2 Data Collection

This section explains the data collection process with focuses on the utilised devices and their setup.

4.2.1 Sensors – Smartphone and Smartwatch Setup

To collect the data an iPhone 7 Plus and an AppleWatch Series 5 were used. To access the raw sensor data on both devices, the application *SensorLog*[1] was used. The application offers a variety of configuration options, the setup for the most important adjustments were:

Logging Rate The logging rate of the used sensors was set to 100 Hz which was the maximal value. This option allows for further downsampling, in case it would be needed.

Fill Empty Data With Previous Data This setting is set to False. If a sensor returns no value for a certain time, an interpolation of the missing value with the surrounding values has been done.

1 https://apps.apple.com/us/app/sensorlog/id388014573

Sensors and Data For the iPhone all sensors have been disabled except the accelerometer, the gyroscope and the magnetometer. For the AppleWatch it was only possible to enable the accelerometer among the intended sensors.

Logging Option The AppleWatch application offers 2 different logging options. The option for recordings up to 1 hour was used, because the second option (for longer periods) stopped repeatedly recording in the data gathering process. This limits the duration of one data collection session to 1 hour.

The iPhone was carried in the front right trouser pocket, close to the center of the body. The AppleWatch was attached to the wrist of the non-dominant hand, which was used to perform the simulated OCD activities.

4.2.2 Indoor Localisation Setup

Because it was not possible to access the raw WLAN data of iPhone devices an additional Android device, namely a Google Pixel 3a, was used.
To enable an IL system using WLAN, the Framework for Internal Navigation and Discovery (FIND)[2] was used. This open source project provides a backend server and a corresponding Android application. The system has 2 working modes. During the first mode, called learning or offline mode, the system trains new or improves existing locations. The server is responsible to store received fingerprints with RSSI values and the label of the location where the signal was recorded at. This data is collected and send to the server by the FIND Android application. Based on this data, multiple ML models are trained on the server. Additionally, the server provides a website with information regarding the performance of the trained IL system. During the second mode, called localisation or online mode, the FIND Android application captures and sends only the fingerprint RSSI values to the server. By using the previously trained ML models, the server calculates to each received fingerprint a localisation guess plus corresponding probability and sends the result back to the Android application. The available FIND Android application was not working properly and was not supported by the Android version of the used device. Therefore the application was updated and improved[3]. Additionally, to let the application

2 https://www.internalpositioning.com
3 available at: https://github.com/MSCL/find3-android-scanner

work properly the option for *Wi-Fi scan throttling* needs to be disabled and the *stay awake* option needs to be activated in the developer options of the Android device. The improved application in combination with the used hardware was able to collect and send fingerprints every 2.8 seconds, which translates to a localisation sampling rate of 0.35Hz. This limitation was issued by the operating system, due to the fact that it was not possible to scan the WLAN environment more often.
Staff members of the Digital Health - Connected Healthcare department were selected as study participants, therefore the IL system was set up in their office environment in building/ floor G.2.1 of the Hasso-Plattner-Institut (HPI). Initially a reasonable set of locations had to be defined. For the reason that more locations lead to a longer offline/ learning phase the overall amount of locations were set to a reasonable minimum. The focus was set to locations were the simulated OCD activities were supposed to happen, sink and oven, and between different rooms to capture location changes while passing doors. The rest of the office area was covered by evenly distributed locations. For each room, serving as office, 3 location points were determined, one for each workplace, named *L* and *R*, and one in the center of the room, slightly shifted in the direction of the door, named *C*. In the area of the corridor 6 locations in total have been defined, each taken in the center of the hallway equispaced approximately every 3 meters. In the meeting room additional 2 locations were captured, named *L* and *R*, respectively. The office area did also include a kitchenette covered by 3 location points, one near the sink (*Sink*), one next to the dishwasher, which served as oven replacement (*Oven*), and one in the back of the room, named *B*. In total 26 locations were defined and for each one around 30 fingerprints were gathered. At the end of the offline phase the server indicated an accuracy of 86%. Figure 4.1 provides on overview of the installed IL system.

4.2.3 Additional Setup

For being able to receive and store the locations of a subject during a data gathering process, a computer was set up with an MQTT broker client, namely Mosquitto[4]. Every time the FIND server received a fingerprint the localisation result was send to the computer as well.

4 https://mosquitto.org

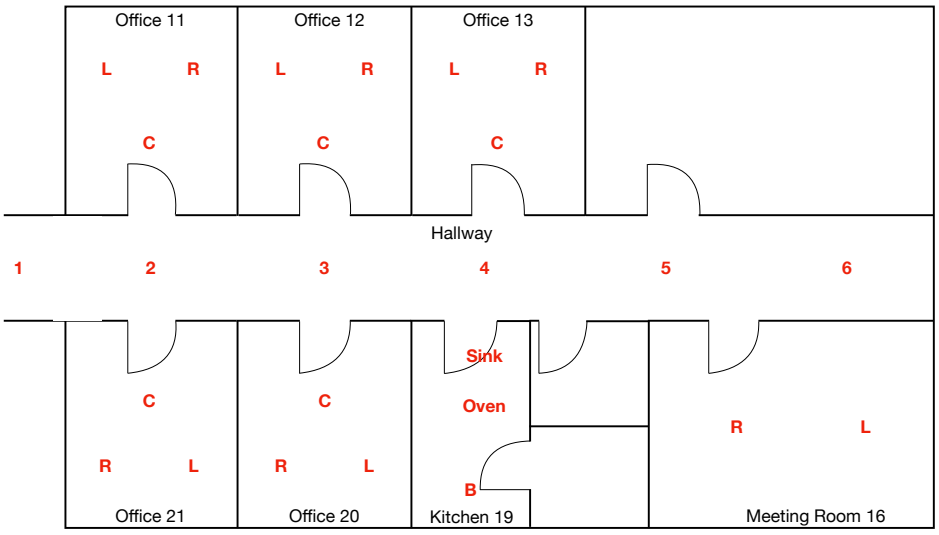

Figure 4.1: An overview of the captured IL locations in the office environment.

The subjects were not observed in person for the full time of a data gathering process, rather they were asked to follow their normal daily behaviour to gather data for the non-OCD class as well. In between they performed the simulated OCD compulsions. The data are in a continuous stream (time series), therefore OCD and non-OCD activities are not naturally separated from each other. In order to make this possible and to label the collected data later in a precise manner the subjects were additionally equipped with an action camera. The camera was mounted on their heads, capturing the area in front of their bodies. Figure 4.2 provides on overview of the complete setup of a subject during the data collection process.

4.2.4 Data Gathering

The biggest challenge during the actual gathering of data was the parallel gathering of data from 4, including the computer, 5 devices simultaneously. The action cam was started and mounted to the head of the subject. The FIND application on the Android device was started, set to the localisation mode and put into the pocket of the subject, respectively the Mosquitto script was started on the computer. Finally, the *SensorLog* applications on the iPhone and AppleWatch

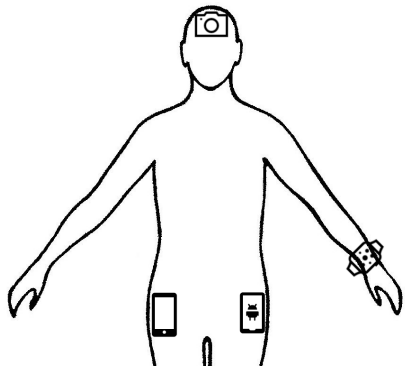

Figure 4.2: The complete setup of a subject during the data collection process. An action camera is mounted on the forehead. On the left wrist, an AppleWatch is carried. In the left trouser pocket is an Android device and in the right pocket an iPhone.

were started to gather sensor data and the watch was attached to the wrist of the subject. Before the iPhone was put in the pocket a shake gesture between the iPhone and AppleWatch was performed, therefore the iPhone was laid on top of the watch and both devices were shaken intensively for at least 10 times. Additionally, the gesture was captured by the action cam. This was needed to be able to properly synchronise the sensor data from the different devices in a later step as well as synchronise the captured video file for labelling purposes. After the iPhone was put in the pocket the subject was set up to gather data.
At the beginning of each run, the subject was shown how to perform the OCD related activities and was asked to execute each activity one time to check for correctness. Afterwards, the subject was not supervised further and was asked to continue with their normal daily behaviour and to perform the simulated OCD activities in between. The *Checking an Oven* and *Washing Hands* activities have been performed 4 times each, which leads to a summed up estimated duration of 60 seconds for *Checking an Oven* and 100 seconds for *Washing Hands*. The *Checking a Door* activity was executed 8 times which approximates to 120 seconds. The total approximated duration of all OCD activities per run is 280 seconds which corresponds to about 7.8% of the total gathered data. At the end of a run the previously specified shake gesture was performed again. In total 8 people participated, among them were 4 females and 4 males. 7 of the participants were right-handed and 1 left-handed. Each participant attended the

study 1 time except for 1 who attended 3 times. Each run took between 55 and 60 minutes.

4.3 Data Preprocessing

In this chapter I explain the data preprocessing, I outline the structure of the gathered raw data, the synchronisation of the data accumulated from different devices and the labelling process.
Every single run produced 4 different files: two comma-separated values (CSV) files, one text file and a video file. The CSV files contain the sensor data from the iPhone as well as the AppleWatch. Each row of a CSV file represents one gathered data point. The file from the iPhone includes columns for the accelerometer, the gyroscope and the magnetometer sensor. Each sensor provides 4 columns, one for each axis (X, Y and Z) as well as one with the timestamp when the sensor values were read out. The file from the AppleWatch is similar, but it contains only the accelerometer sensor data. The text file stores the locations in which each row represents a collected data point. One collected data point is represented via JavaScript Object Notation (JSON), including among others the topmost location guess of the IL system plus the timestamp and the probability of the guess. All timestamps were in *Unix time*, counting the milliseconds since the 1st January of 1970. The last file contains the video recorded from the action camera.
Because the files from the different devices cover slightly different periods of time, have different sampling rates and recorded data at different points of time, they needed to get synchronise to serve as a dataset for a ML application. Before the various files from the iPhone, AppleWatch and Android device were merged into one dataset, they were resampled to the same constant sampling rate (e.g. 100 Hz). Even though the sampling rate was set to 100 Hz in the *SensorLog* applications, the actual sampling rate of the gathered data was not precisely 100 Hz at every point of time. In particular, the average sampling rate of the phone magnetometer was only at around 17 Hz. Therefore, every sensor (phone and watch acceleration, phone gyroscope and phone magnetometer) was first moved in its distinct sub-dataset and was resampled to 100 Hz using the associated timestamps of each sensor. In case the local sampling rate was higher than 100 Hz the mean of the values in the time slot were taken. If instead the local sampling rate was lower than 100 Hz, which was the major circumstance for the magnetometer sensor data, missing values were interpolated weighted

according to the time distances to the surrounding values. In the next step, the 3 single datasets from the iPhone sensors were shifted to the same points of time and merged back into one dataset including one shared timestamp column (in the following named as **DS-1**). This is easily doable because their individual timestamps were based on the same clock.
This approach was not easily transferable to the merge between DS-1 and the watch acceleration dataset because their timestamps were based on different clocks. This means that two identical timestamps in the two datasets could still refer to two different points of time because their clocks were not necessarily synchronised perfectly or could have a time shift to each other. To synchronise across devices, using the shake gesture, the *Jointly*[5] framework was used. This framework can synchronise two sources with each other using simultaneously recorded signals including a characteristic pattern, which can be a shake gesture. The accelerometer data magnitude as an axes independent union of the single accelerometer axes defined as

$$sensor_{mag} = \sqrt{sensor_x^2 + sensor_y^2 + sensor_z^2} \tag{4.1}$$

were used as input for the framework. Additionally, *Jointly* offered several parameters to precisely define the kind of the utilised characteristic pattern:

Threshold A threshold value in percentage that defines how large an amplitude must be, compared to the maximum amplitude among the sources, to be considered as a peak. It was set to 10 %, 20 % or 30 % depending on the particular data of a run.

Distance The maximal distance in milliseconds in that two consecutive peaks can be counted as coherent. This value was set to 1000.

Min_Length The minimal amount of consecutive peaks needed to classify them as one shake gesture. This value was set to 10.

Window The duration of time in seconds, which is considered as a window for shake gestures, starting from the beginning and the end of the dataset. This value was set to 60.

The framework used the first shake gesture to synchronise the start of the datasets, while the final shake gesture was then used to detect a potential time

5 https://github.com/felixmusmann/jointly

shift, corrected by stretching or compressing. After DS-1 and the dataset of the watch acceleration sensor were properly synchronised, they were merged using the same procedure as for the single iPhone sensors (**DS-2**).
In the next step, the IL file was transformed into a dataset having in each row one data point consisting of the topmost location guess, the probability and the corresponding timestamp. This dataset was upsampled to a sampling rate of 100 Hz as well. In doing so empty rows were filled up with the closest existing value for a location guess. The IL dataset was merged with DS-2 in the same manner as the merge between the single iPhone sensors (**DS-3**). Although the timestamps of the two datasets were based on different clocks, there was no better solution to synchronise the datasets. For the reason that the original sampling rate of the IL dataset was much lower than 100 Hz, the accuracy of the merge was still sufficient.
Following, the term dataset refers to the DS-3. Since all sub-datasets had a different amount of rows, as well as various starting and finishing points in time (depending on the different devices which were started and stopped one after another to gather data), the dataset were cut off to the rows that refer to the timestamps for which all sub-datasets included data points. In other words, the temporal intersection of the single datasets was taken. The timestamp column was updated to contain the temporal difference to the first row's timestamp in milliseconds. This column was needed to synchronise the dataset with the recorded video file of the run to perform labelling. The dataset schema consisted of 15 columns, 1 column for each axe for every sensor, plus the IL guess and its corresponding probability, plus the updated timestamp column.
The next step was the labelling of the dataset. The tool EUDICO Linguistic Annotator (ELAN)[6] was used. It is an open-source annotation tool for audio and video recordings. By means of the captured shake gesture in the video and the dataset, and the former added timestamp column in the dataset, the video file and the dataset were synchronised. The dataset was extended with an additional *label* column. On the basis of the video, this column was filled with the corresponding label of the activity (either *Washing Hands*, *Checking an Oven*, *Checking a Door* or *No OCD*), that was executed at that time. Because the beginning and the end of an activity was difficult to define, the labelling process needed several iterations and was performed with a precision of 1 second. Questions like — 1) Is the initial grab to an oven door or a door handle part of

6 https://archive.mpi.nl/tla/elan

the associated core activity? or 2) Is the initial opening and closing of the water tap part of the *Washing Hands* activity, even though when some of the simulated *Washing Hands* activities (without water or soap) did not include the simulated opening and closing of the tap? — had to be answered. I decided that all initial movements should be considered as part of the core activities. Afterwards, the dataset was updated removing rows corresponding to the shake gestures and the no longer needed timestamp column.
The previously described process was conducted with the files of every single run. In the end, the final datasets of all runs were aggregated, excluding the dataset of one subject (right-handed, female) because the enclosed OCD related activities were not performed in the correct predefined way. During this process a new column, named *subject*, was added, storing an identifier of the subject the data row belongs. The combined total dataset of all utilised subjects contained 775 MB of data in 16 columns and about 2.9 million rows with a sampling rate of 100 Hz which translates to an approximate duration of 8 hours.

4.4 Data Exploration

This section gives an overview of the data exploration process to gain a deeper understanding of the dataset, which serves as a basis for the design of the ML solution. The exploration has been conducted in two subsequent phases, raw data exploration and activity length exploration.

4.4.1 Raw Data Exploration

The previously added *subject* column is needed for later identification purposes and is not going to be a direct input for ML applications. Therefore, it was not considered in the following exploration. A column including the magnitude (Mag), as previously defined in Equation (4.1), was added for each of the 4 sensors. The magnitude is a very popular method in HAR to reduce the error from rotation components [11]. Table 4.1 gives an overview of the dataset columns as well as their corresponding units. In the following tables, the abbreviations *acc*, *gyro* and *mag* refer to acceleration, gyroscope and magnetometer, respectively.

Table 4.1: Dataset columns and the units of their values.

Column	Unit
phoneAccX phoneAccY phoneAccZ phoneAccMag	acceleration in 9.81 $\frac{m}{s^2}$
phoneGyroX phoneGyroY phoneGyroZ phoneGyroMag	angular velocity in $\frac{rad}{s}$
phoneMagX phoneMagY phoneMagZ phoneMagMag	magnetism in μT (microtesla)
watchAccX watchAccY watchAccZ watchAccMag	acceleration in 9.81 $\frac{m}{s^2}$
location locationProp label subject	—

As the first explorational step, a basic column identification in terms of variable type (predictor or target variable), data type (character or numeric) and variable category (categorical or continuous) was performed:

Type of Variable All columns except for the *label* column have been defined as predictor variables. The column *label* was the only target variable.

Data Type Except for the columns *location* and *label*, which are of the data type character, all columns were of the data type numeric.

Variable Category The columns *location* and *label* were 'categorical', the remainder of the category 'continuous'.

In Table 4.2 the percentages of the data samples for each target class are represented. The non-OCD class was the most represented, with 91.39 %. The rows labelled as OCD represent a percentage of 8.61 % which splits up into 4.52 % for the *Washing Hands*, 2.33 % for the *Checking an Oven* and 1.76 % for the *Checking a Door* activity, respectively. As stated in Section 4.1, one requirement for the dataset was an imbalance of about 96 % to 4 % between the classes, which was met approximately. The total duration of the activities recorded in the whole dataset was about 8 hours and 8 minutes.

Table 4.2: Percentages of the different classes in the dataset.

Label	Absolute	Percentage	Duration
No OCD	2 680 270	91.39 %	07:26:42
OCD	252 527	8.61 %	00:42:04
Checking Door	132 370	4.52 %	00:22:03
Washing Hands	68 428	2.33 %	00:11:24
Checking Oven	51 729	1.76 %	00:08:37
Total	2 932 797	100.00 %	08:08,46

To obtain a first insight into the dataset, especially about the differences between the as OCD and as non-OCD labelled data, several basic statistical measures were calculated for the magnitudes of the sensor axes. The results are presented in Table 4.3. The mean values were, except for the magnetometer magnitude, greater for the OCD data. Except for the gyroscope magnitude, where the mean

value of the OCD data was 1.7 times greater than the mean value of the non-OCD data, the differences were not particularly significant. This is an expected result because the OCD class only consists of activity data. In contrast, the non-OCD data also includes periods in which participants for instance were only sitting at their desks. The standard deviation is defined as

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_{mean})^2}{n-1}} \tag{4.2}$$

where x_i is the value of the i^{th} point in the dataset, x_{mean} is the mean value of the dataset and n is the number of total data points. The standard deviation of the watch acceleration magnitude for the OCD data is 2 times greater than for the non-OCD data. This indicates that the values in the OCD class were spread out over a wider range than the non-OCD data. The standard deviation values for the particular phone sensors are close together. The global min and max values were more extreme in the non-OCD class. As shown in Table 4.2, this class contained about 10 times as much data as the OCD class. Therefore, it is more likely for it to include also more extreme values.

Table 4.3: Overview of several statistical variables calculated for the sensor magnitude columns. The left value in a cell is calculated from all rows of a column labelled as No OCD, the right from the data labelled as OCD, respectively. A boldly printed value identifies the more extreme value inside a cell.

Column	Mean	SD	Min	Max
watchAccMag	(1.02 | **1.06**)	(0.13 | **0.27**)	(**0.03** | 0.04)	(9.58 | **10.40**)
phoneAccMag	(1.00 | **1.01**)	(**0.10** | 0.07)	(**0.07** | 0.20)	(**6.19** | 2.94)
phoneGyroMag	(0.25 | **0.43**)	(**0.56** | 0.55)	(0.00 | 0.00)	(**10.42** | 6.43)
phoneMagMag	(**402** | 395)	(17.24 | **21.32**)	(**203** | 308)	(**517** | 449)

SD $\widehat{=}$ standart deviation; Min $\widehat{=}$ minimal value; Max $\widehat{=}$ maximal value

The following analysis aims to conclude whether the columns label and location of the dataset were correlated to each other or not. In Figure 4.3 the total number of data points gathered for every single location is illustrated. Additionally, the data points are colour-coded according to their activity. For all 26 defined locations data points were collected. The office room 13 held the most data points. That was because 2 subjects, including the subject which participated multiple

times in the study, worked in that office. In a perfect functioning IL system, all red points (*Washing Hands*) would have been located at the *kitchen.sink* and all orange points (*Checking Oven*) would have been located at the *kitchen.oven*. Most of the *Washing Hands* and *Checking Oven* data are located in one of the kitchen locations, whereby *hall.3* included most of the incorrectly located points. The majority of points coloured in blue (*Checking Door*) are located in the hallway, particularly in the more central area (*hall.3*, *hall.4* and *hall.5*). This is expected, because while performing a *Checking Door* activity the location gets changed, either from a room to the hallway or the other way around. Therefore, a hallway location was always crossed during that activity. Visually it is possible to derive a dependency between the location and the classification.

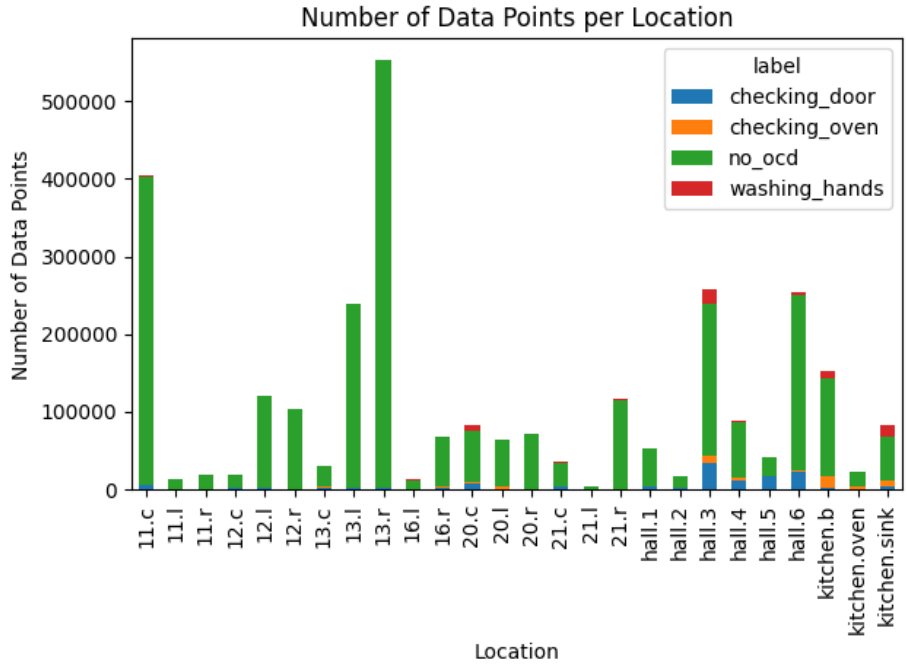

Figure 4.3: The number of data points per location. The shares per location were colour-coded according to their activity classification.

To support the visual impression with statistical evidence, Pearson's chi squared test[7] was performed. As input for the test a random subsample of 480 data points, which equals closely to one data point per minute, was drawn from the dataset. The working and null hypotheses were specified as follows:

Working Hypothesis H_1 There is a significant dependence between the categorical values of the *location* and *label* columns.

Null Hypothesis H_0 There is no dependence between the categorical values of the *location* and *label* (binary classification) columns.

The results of the conducted test are visualised in Table 4.4. The p-value was with a value of 0.0002 smaller than the chosen significance level of 0.01. Therefore, H_0 was rejected and H_1, stating that there is a significant dependence, was accepted. Cramer's V[8] result of 0.3449 is greater than 0.25. As also stated by [32], this indicates a very strong association between the values of the *location* and *label* columns.

Table 4.4: Overview of the results of Pearson's chi-squared test between the categorical data column *location* and *label*, whereby the different OCD activities were combined into one class.

Measurement	Result
Chi-Square	58.0507
Degrees of Freedom	25
P-Value	0.0002
Cramer's V	0.3449

4.4.2 Activity Length Exploration

As stated in Chapter 3, time series data need to be split into windows of distinct length before they can be consumed by NNs. The window represents the data that the network can consume at once. Therefore the size of a window is closely

7 Pearson's chi-squared test is intended to test how likely it is that an observed distribution is due to chance.

8 Cramér's V is a number between 0 and 1 that indicates how strongly two categorical variables are associated.

connected to the length of the activities the network is trained to detect. In this dataset, an activity is defined as one contiguous block of rows that have the same label. The whole dataset included 127 activities in total. Thereof 70, around 55 %, were from the activity *Checking Door*. The remainder was almost evenly divided between *Checking Oven* with 28 executions and *Washing Hands*, which was performed 29 times. The percentages are summarised in Table 4.5.

Table 4.5: Percentages of the different simulated OCD activities.

Label	Absolute	Percentage
Checking Door	70	55.12 %
Washing Hands	28	22.05 %
Checking Oven	29	22.83 %
Total	127	100.00 %

Table 4.6 exhibit various statistical measures calculated for the activity length. Since the activities were labelled to a precision of 1 second, as stated in Section 4.3, the lengths and measurements are in seconds. The activities lengths have a great variability and reach from 12 seconds, for the shortest, up to 37 seconds, for the longest activity. The standard deviation reflects this with a value of 5.7 seconds. The median and mean are close together with 18 seconds and 19.87 seconds, respectively. The histogram in Figure 4.4 depicts a mostly uniform distribution including a peak in the area around 17 seconds.
In Figure 4.5 the values of the sensors for one of the simulated *Checking a Door* activity are visualised. It includes the magnitudes of all sensors and the change of the location over time. In the topmost plot, the watch acceleration is displayed. The two times three peaks come from pressing the door handle several times and opening and closing the door, as described in Section 4.1. The three other plots show the magnitudes of the phone sensors. All three plots contain a peak in the middle, which corresponds to the action of passing through the door. The final plot includes the location change. Ideally, there would only be one peak in the middle, indicating the location change while passing the door. The three peaks show that the IL system has not worked perfectly in this specific case. Two more activities are similarly reported in Figure 4.6 and Figure 4.7.

Table 4.6: Overview of several statistical variables calculated for the lengths of all activities.

Activity Length Statistics	Result
Mean	19.87
Median	18.00
SD	5.70
Min	12.00
25 %	16.00
50 %	18.00
75 %	23.50
Max	37.00

SD ≙ standart deviation; Min ≙ minimal value; X % ≙ X % of the values are smaller; Max ≙ maximal value

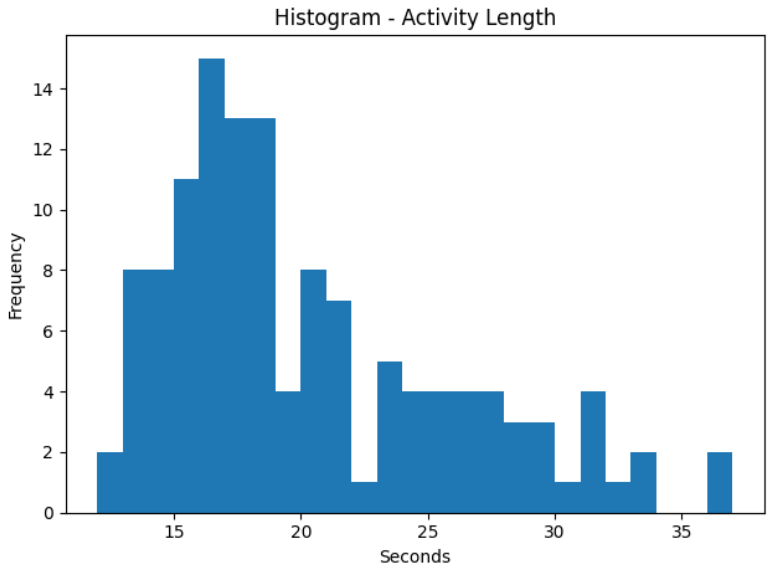

Figure 4.4: Histogram of the length of the various simulated OCD activities.

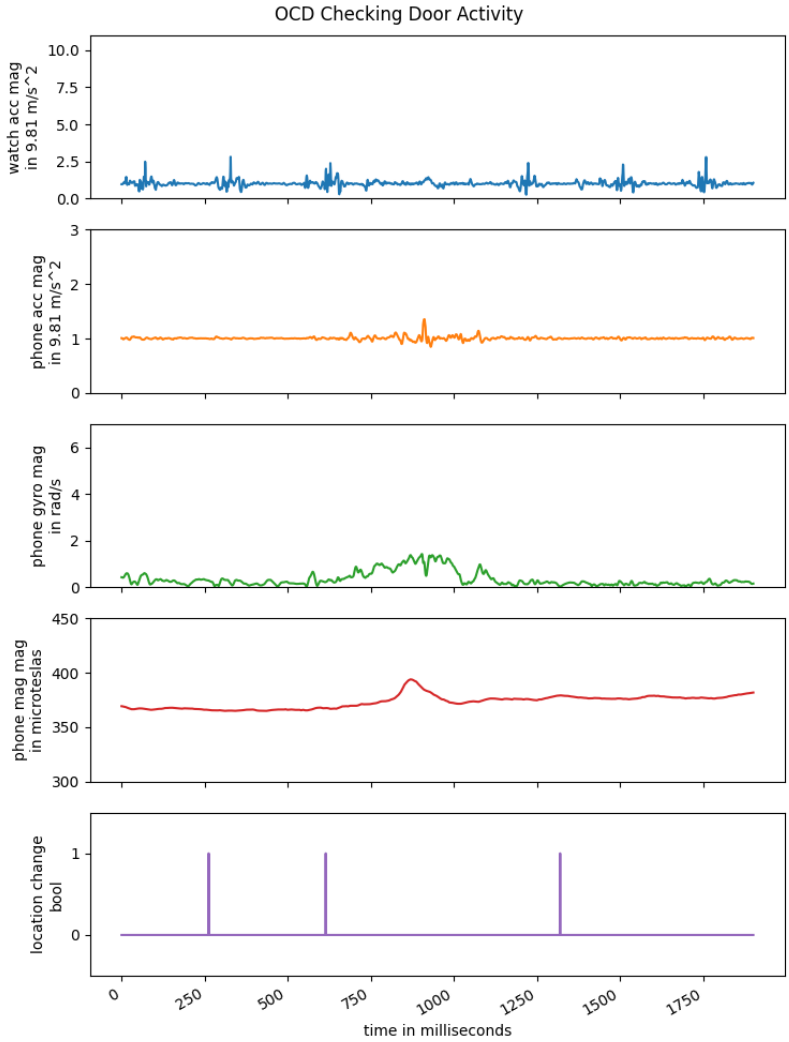

Figure 4.5: Sensor magnitudes and the location change during an *Checking Door* activity.

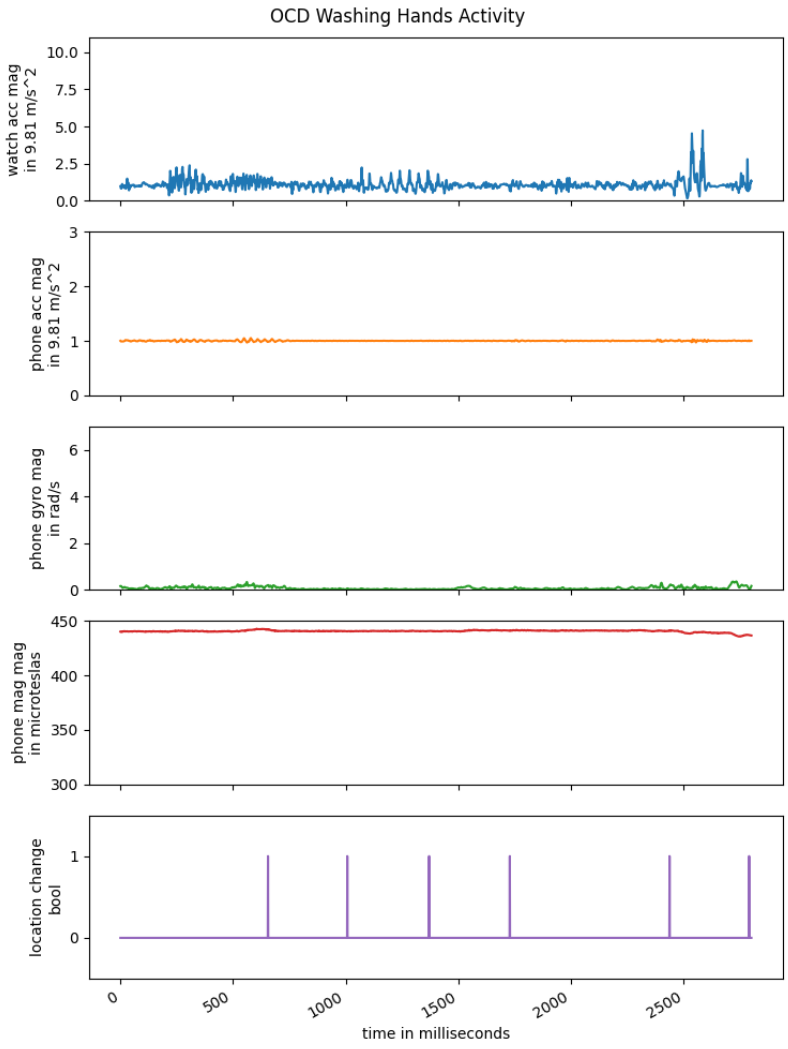

Figure 4.6: Sensor magnitudes and the location change during a *Washing Hands* activity. In the watch Acc magnitude curve, you can see the several movements of the handwashing process. For the reason that the activity has no full-body movements, the phone sensors include almost no amplitude changes. The changes in the location indicate an erroneous IL system.

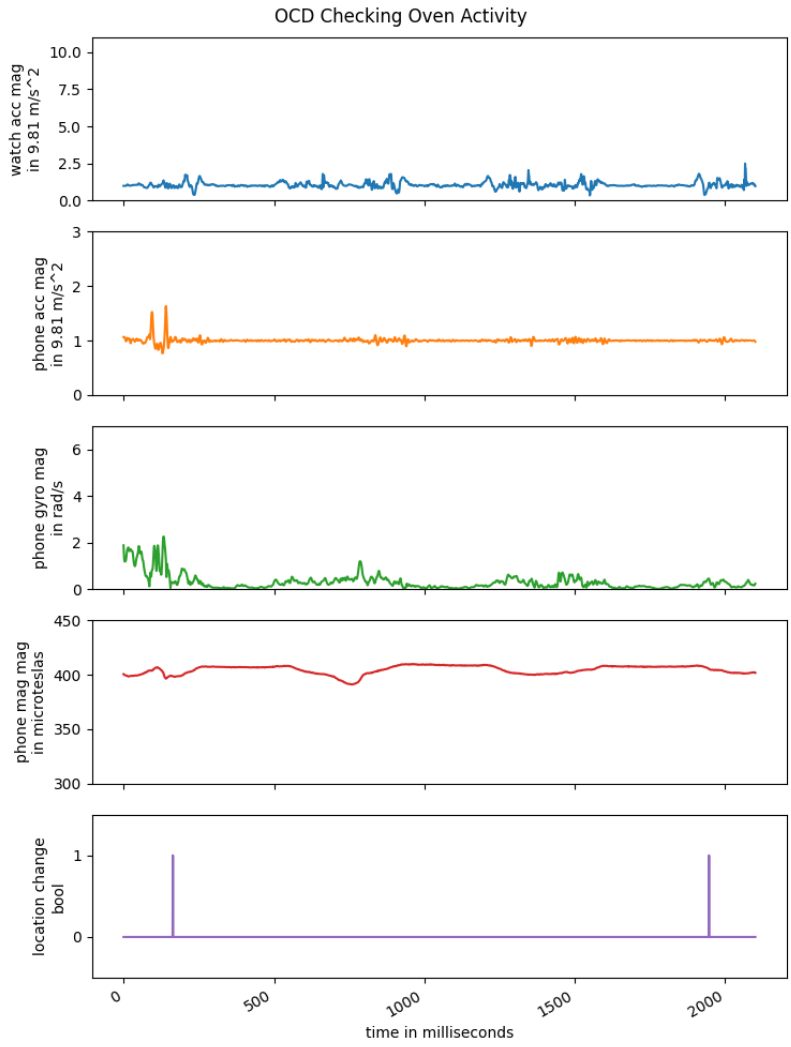

Figure 4.7: Sensor magnitudes and the location change during a *Checking Oven* activity. In the watch Acc magnitude curve, you can see the three repetitions of the checking process. One check includes two areas of amplitude changes centred by a more flat area. For the reason that the activity has nearly no full-body movements, the phone sensors include almost no amplitude changes. Additionally, there are two changes in the location at the beginning and the end. The peaks in the Acc and gyroscope magnitudes of the phone derive from the last step towards the oven.

4.5 Deep Learning Models

This section outlines the development of a ML system to automatically recognise OCD compulsions. In the following, the selection of a basic network structure, the initial setup and the hyperparameter tuning is discussed.

4.5.1 Network Architecture

The first step in the development of a machine learning system is to decide for a general network architecture. The task the network architecture has to solve is a binary classification problem, namely the classification between OCD and non-OCD. Because the dataset includes only fully labelled examples, the network will be trained in a supervised learning scenario. I decided to use a DL architecture because it offers one great advantage over classic ML architectures: DL does not require the manual specification of features. This reduces the need for specific domain knowledge and saves time. In general DL architectures offer more flexibility, robustness and achieve better or similar performance [11], [12]. They are considered as state of the art in areas such as computational vision, natural languages processing or HAR [9].
In the field of HAR, CNN and RNN architectures are the most common [12]. Both approaches have similar performances, although RNN is slightly better in the detection of short activities with natural order, whereas CNN is superior in the detection of long activities. In general, RNN architectures are well suited in classifying data with temporal dependencies (e.g. time-series data) [9], [10]. Since the dataset used in the OCD class to be recognised contains mainly short, repetitive activities, the RNN architecture was chosen. In the group of RNN networks there are several different architectures. I decided to use LSTM because it was reported to outperform other architectures in similar tasks [33].
For the development, the TensorFlow[9] framework with the included Keras API[10] was used.

4.5.2 Initial Setup

As the dataset consists of data from seven multiple subjects, leave-one-subject-out cross-validation (LOSO-CV) was used as the evaluation method.

9 https://www.tensorflow.org/
10 https://keras.io/

This means that one complete training and evaluation loop of the network consists of seven single loops, each one conducted with testing samples from a different subject. Finally, the average of the metrics of all seven single runs are computed to perform an overall evaluation. As every run of the network implies seven runs in total, one concern for the setup and selection of parameters was to shorten the calculation time. The process of hyperparameter tuning also included an extensive experimental determination of parameter combinations, which thereby was shortened and made feasible.

Following, the **most important parameters** are explained:

Network Structure Initially, the most basic LSTM network structure was used. This is: one LSTM layer followed by a fully connected dense layer. The dense layer gets the results from the LSTM layer and maps them to probabilities. Therefore, the sigmoid function, defined as

$$p = \frac{1}{1 + e^{-x}} \tag{4.3}$$

was used as the activation function in the dense layer. The first layer was trained with an output dropout rate [25] of 20 %.

Sampling Rate Following [9], [10] a sampling rate of 20 Hz to 30 Hz is sufficient to avoid under-sampling and to have sufficient information to detect human physical movements. Hence, the sampling rate of the dataset was reduced from 100 Hz to 33 Hz. This reduces the calculation time by about 66 % because only one-third of the data needs to be processed.

Overlap As described before, the network receives the samples in the form of time windows. Overlapping windows increase the amount of data the network needs to process and therefore the calculation time increases. [27] states that overlaps in subject independent cross-validation (as LOSO-CV) does not improve the performance of HAR systems but increases the needed resources. In the context of RNN, overlap would mean that information would flow into the calculation multiple times at the same time since several consecutive windows are consumed at once. Therefore, I decided not to use overlaps between the single windows.

IL encoding As RNN cannot consume variables of the data type 'character', all

values have to be mapped to numbers. For the reason that 1-hot encoding would add 25 new columns, one for each defined location minus the original location column, which would increase the calculation time, the locations were integer encoded. The encoding was designed to respect the spatial dependencies of the locations. The probability of a location guess was not considered.

Class Weights The dataset shows a great imbalance between the different classes (91.39 % to 8.61 % Table 4.2) to reflect a real-world scenario. To overcome the underrepresentation of the OCD class, the classes are fed into the network with different weights, over-valuing the contribution for the few examples of the OCD class in the calculation of the loss function. The class weights are calculated by:

$$\begin{aligned} nonOCD\ weight &= \frac{total\ examples}{2 \cdot negative\ examples} \\ OCD\ weight &= \frac{total\ examples}{2 \cdot positive\ examples} \end{aligned} \tag{4.4}$$

Scaling the weights by 0.5 helps to keep the loss to a similar magnitude and the sum of the weights of all examples staying the same as without class weights. The class weights were different for every single run, but close to 0.5 for the non-OCD and 5.4 for the OCD class. Class weights were preferred to the various resampling methods. With over-sampling, more data samples (copies) would have to be processed, which would increase the calculation time. Using under-sampling not all available data would be used and the dataset would become smaller.

Optimiser The Adaptive Moment Estimation (Adam) [34] with a learning rate of 0.001 was used as optimisation function. This is common practice.

Loss Function The binary cross-entropy (BCE), which is a common choice for the loss function in binary classification problems, was used as the loss function. It is defined as

$$bce = -\sum_{i=0}^{1} t_i \log(p_i) = -[t \log(p) + (1-t) \log(1-p)] \tag{4.5}$$

where t_i is the true value, either 0 or 1, p_i is the probability for the i^{th} class and p is the result of the sigmoid function of the dense layer, introduced in Equation (4.3).

Batch Size The batch size was set for every single run to the biggest reasonable value (BRV) which was the size of the smallest dataset (either validation or test). This was done because a bigger batch size refers to fewer updates of the weights – weight updates happen after the network processed all examples in a batch – to obtain a shorter calculation time.

Stop Function To stop the learning process to prevent the network from overfitting, a stop function was introduced. 20 % of the training samples for each run were used as validation samples. After each epoch, the loss value of the validation samples was calculated. If the value did not improve by at least 0.0001 in 100 consecutive epochs, the training was stopped and the weights of the best epoch were restored.

All other parameters were left at the default values of the used TensorFlow framework.

Accuracy, precision, recall and the F1-Score served as the basis for evaluation. The F1-Score is used to compare the following runs. They were introduced in Section 2.4. The accuracy was used because it is a common metric that provides a general insight into the performance. Precision and recall are well suited for the evaluation of imbalanced datasets. Both metrics are, in contrast to other often used metrics in binary classification tasks such as specificity or receiver operating characteristic (ROC) curves, not influenced by the true negatives and therefore focus on the positive/ minority class [35].

4.5.3 Hyperparameter Tuning

In this subsection the process of hyperparameter tuning is described in detail. The long calculation times had a great impact on the selection of the parameter combinations to be tested. One complete run took with the initial settings about one hour (with a sampling rate of 100 Hz and an overlap of 0.5 % it would have been 6 hours).

Units and Window Size

Before the network can consume the data, the data must be cut into time windows of certain lengths. In this process, a class must be assigned to each window. As the dataset is labelled (every row of the dataset contains the class information for that point of time), the window class is derived from the class which is most represented by the rows in a window. This means that activities with a duration of 12 seconds, like the shortest OCD activities in the dataset (Table 4.6), would be classified as non-OCD when a window size greater than 24 is used. The size of the windows should be related to the length of the activities to be identified. To decide for a feasible window size the number of units of the first LSTM layer of the network must also be considered. That is because the layer consists of multiple LSTM cells, where the units parameter indicates the number of cells. Each of these cells receives a time window. This means that the amount of data seen and calculated at once by the layer is made up of the number of cells (units) times the window size.
Because in [27] and [9], window sizes ranging of 0.25 seconds to 7 and of 2 seconds to 5 have been used for HAR, I decided to experiment with 2, 1 and 0.5 seconds. Referring to the length of the OCD activities (Table 4.6), I decided to cover 12 (as minimal activity length), 18 s (as the median), 24 s and 37 s (as maximal activity length). This is achieved by a different units parameter depending on the window size. A sum-up of the experiment is shown in Table 4.7. Regardless of the window size parameter, the best F1-Score is achieved covering 37 seconds of the input (using the most units). Using 37 units with a window size of 1 second achieved the greatest F1-Score of 0.367.
Because the parameter combination of the first experiment has been very broad the experiment was repeated with a parameter set close to the best parameter combinations of the first experiment. The seen input was extended to cover up to 43 seconds, the window sizes were tested in a range from 0.75 s to 1.75 s with a step size from 0.25 seconds. The experiment is summed up in Table 4.8. The experiment showed that there is a significant influence of the units and window size parameters and that the usage of a bad parameter combination can result in a poor F1-Score of 0.29 (units: 9, window size: 2 s, seen input: 18 s). The **best parameter combination** is a **window size of 1.25 s** with **30 units**, which refers to a seen input of 37 s which is exactly the length of the longest OCD activity in the dataset. This parameter combination is used in the following tuning steps. The best achieved **F1-Score** was **0.39**.

Table 4.7: The setup and results of the first units and window size tuning experiment. On the left, tested windows sizes are shown. The top refers to the seen data, which is obtained from the units times the window size parameter. The performance in terms of the F1-Score for each parameter combination is shown in each cell as well. The cell with the greatest F1-Score is marked in bold.

window size	seen data			
	12 s	**18 s**	**24 s**	**37 s**
2 s	u: 6 f: 0.33	u: 9 f: 0.29	u: 12 f: 0.32	u: 18 f: 0.35
1 s	u: 12 f: 0.31	u: 18 f: 0.366	u: 24 f: 0.33	**u: 37 f: 0.367**
0.5 s	u: 24 f: 0.31	u: 36 f: 0.32	u: 48 f: 0.32	u: 74 f: 0.33

u $\widehat{=}$ units; f $\widehat{=}$ F1-Score

Table 4.8: The setup and results of the second units and window size tuning experiment. On the left, tested windows sizes are shown. The top refers to the seen data, which is obtained from the units times the window size parameter. The performance in terms of the F1-Score for each parameter combination is shown in each cell as well. The cell with the greatest F1-Score is marked in bold. The underlined cell is the best setup from the previous experiment, see Table 4.7.

window size	seen data		
	31 s	**37 s**	**43 s**
1.75 s	u: 18 f: 0.365	u: 21 f: 0.34	u: 25 f: 0.33
1.5 s	u: 21 f: 0.36	u: 25 f: 0.3	u: 29 f: 0.29
1.25 s	u: 25 f: 0.32	**u: 30 f: 0.39**	u: 34 f: 0.37
1 s	u: 31 f: 0.34	u: 37 f: 0.367	u: 43 f: 0.369
0.75 s	u: 41 f: 0.32	u: 49 f: 0.33	u: 57 f: 0.38

u $\widehat{=}$ units; f $\widehat{=}$ F1-Score

Learning Rate and Batch Size

The learning rate is one of the most important hyperparameter to tune. Since [36] emphasises the influence of the batch size in combination with the learning rate, especially on the training times, it make sense to optimise both parameters together. A lower learning rate can increase the calculation times, but may improve the results, whereas a bigger batch size can reduce the training time, but may decrease the achieved results. A good combination of the parameters could increase the results using the same time or reduce the time needed for the same results.
The default learning rate inside of the Adam optimisation function is 0.001. The chosen values to be tested were 0.1, 0.01, 0.001 and 0.0001. The batch size was initially set for each run to the BRV, which was at around 2000. The tested values were 64, 256, 1024 and BRV. Table 4.9 gives an overview of the experimental setup and the results. A learning rate of 0.1 resulted in the worst F1-Score for all tested batch sizes. The **best F1-Score** of **0.44** was achieved with a **learning rate** of **0.01** and a **batch size** of **1024**. Additionally, the calculation time was reduced from 46 minutes for the run with the initial settings (learning rate: 0.001, batch size: BRV) to 34 minutes, which corresponds to a speedup of 26.1 %. This experiment proves that it is possible to increase the result and to reduce the calculation time.

Table 4.9: The setup and results of the learning rate and batch size tuning experiment. The value in each cell reflects the achieved F1-Score by the chosen parameter combination. The cell with the greatest F1-Score is marked in bold. The underlined cell is the best setup used in the previous experiment, see Table 4.8.

	learning rate			
batch size	**0.1**	**0.01**	**0.001**	**0.0001**
64	0.13	0.32	0.41	0.39
256	0.03	0.37	0.38	0.4
1024	0.32	**0.44**	0.39	0.36
brv	0.33	0.33	<u>0.39</u>	0.35

General Network Structure

The previous results correspond to the most basic network structure. More advanced LSTM architectures can be obtained by simply lining up layers, where deeper layers receive the results of the previous one, or using bidirectional layers [37]. In a bidirectional setup exist two layers, the first layer receives the input sequence, like the layer in a unidirectional structure, while the second bidirectional layer receives the input sequence in a reversed order. In this way, the network has access to information from the past and the future of a specific point of time.
In the experiments, I kept the final dense layer from the basic setup and changing the LSTM layers to: two, three and four consecutive unidirectional and one, two and three consecutive bidirectional layers. The experiment was conducted with and without the dropout layer before the final dense layer to avoid co-dependencies between the chosen dropout place/ rate and the network structure. The results are summed up in Table 4.10. The experimentally determined **best network structure** was **2 bidirectional LSTM layers without a dropout layer** and it accomplished an **F1-Score** of **0.5**.

Table 4.10: The setup and results of the general network structure tuning experiment. The value in each cell reflects the achieved F1-Score by the network and the used dropout configuration. The cell with the greatest F1-Score is marked in bold. The underlined cell is the best setup used in the previous experiment, see Table 4.9.

LSTM layers	**1L**	**2L**	**1BL**	**3L**	**4L**	**2BL**	**3BL**
with dropout	<u>0.44</u>	0.41	0.39	0.37	0.42	0.45	0.41
without dropout	0.43	0.39	0.39	0.39	0.38	**0.5**	0.39

L $\widehat{=}$ LSTM layer; BL $\widehat{=}$ bidirectional LSTM layer

Dropout and Learning Rate

Another important hyperparameter, especially for deep neural networks, is the use of dropout. It is advised to use a higher learning rate when using dropout in a network. Therefore the dropout and learning rate were tuned in combination. Referring to [38], four suitable positions exist where a dropout can be applied in the used network. These are, before the network (so-called input dropout), between the network LSTM layers (so-called between dropout), after the last

LSTM layer (known as output dropout) and between the LSTM cells within the layers (so-called recurrent dropout). For the input dropout, a rate up to 20 % is advised [25], typical values for dropout in the hidden units are in the range of 20 % to 50 %. Because of the high number of possible parameter combinations, including the dependency to the learning rate, a first experiment was conducted with a learning rate of 1, 0.1 and 0.01 in combination with an input dropout rate of 0.1 and 0.2. The other kinds of dropout (named other dropouts) were taken as one parameter, and rates of 0.2, 0.3, 0.4 and 0.5 were included in the experimental setup. For all parameter combinations, the combinations with the learning rate set to 0.01 achieved the greatest F1-Scores. That is why the learning rate was set to 0.01 for the following experiments. The results are presented in Table 4.11.

Table 4.11: The cells contain the F1-Scores achieved by the different dropout rate and learning rates combinations.

learning rate: **1.00**	other dropout			
input dropout	**0.2**	**0.3**	**0.4**	**0.5**
0.1	0.03	0.05	0.03	0.01
0.2	0.22	0.01	0.03	0.03

learning rate: **0.10**	other dropout			
input dropout	**0.2**	**0.3**	**0.4**	**0.5**
0.1	0.03	0.29	0.26	0.31
0.2	0.25	0.23	0.29	0.23

learning rate: **0.01**	other dropout			
input dropout	**0.2**	**0.3**	**0.4**	**0.5**
0.1	0.4	0.4	0.39	0.39
0.2	0.41	0.37	0.37	0.35

In a following experiment, parameter combinations of the input dropout (0.0, 0.1 and 0.2) and the other dropouts (0.0, 0.1, 0.2, 0.3, 0.4 and 0.5) were tested. The results show a trend, the fewer input dropout is used, the greater is the F1-Score

(Table 4.12). Therefore, the input dropout was removed from the experimental setup for the following experiments.
Because all previously tested dropout combinations resulted in a worse F1-Score than without dropout, the recurrent dropout was excluded from the experimental setup as well. The focus was set exclusively on the output dropout as well as on the between dropout. Subsequently, a sequence of many experiments was carried out with a great variety of different combinations of output and between dropout rates. Table 4.13 summarises the results. It was not possible to find a combination of parameters, which gives a better F1-Score than without. Therefore, **dropout** was **completely removed** from the network structure.

Table 4.12: The cells contain the F1-Scores achieved by the different input and other dropout rate combinations. The learning rate was set to 0.01. The cell with the greatest F1-Score is marked in bold. The underlined cell is the best setup used in the previous experiment, see Table 4.10.

	other dropout					
input dropout	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
0.0	**0.5**	0.44	0.43	–	–	–
0.1	0.39	0.4	0.4	0.4	0.39	0.39
0.2	0.39	0.42	0.41	0.37	0.37	0.35

The **finally** achieved mean **metrics** values over all runs with the LOSO-CV technique after the hyperparameter tuning were: **Precision: 0.42**, **Recall: 0.64**, **Accuracy: 0.88**, **F1-Score: 0.5** and **Time: 51 min**.

Table 4.13: The cells contain the F1-Scores achieved by the different between and output dropout rate combinations. The learning rate was set to 0.01. The cells with the greatest F1-Scores are marked in bold. The underlined cell is the best setup used in the experiment showed in Table 4.10.

	between dropout							
output dropout	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**
0.0	**0.5**	0.47	0.39	–	–	0.46	–	0.39
0.1	0.37	0.41	0.42	–	–	–	–	–
0.2	0.34	0.43	0.41	–	–	0.4	–	0.46
0.3	–	–	–	**0.5**	0.38	0.43	0.44	0.4
0.4	–	–	–	0.46	0.38	0.45	0.37	0.41
0.5	0.39	–	0.41	0.37	0.39	**0.5**	0.38	0.46
0.6	–	–	–	0.41	0.43	0.38	0.44	0.45
0.7	0.4	–	0.37	0.45	0.36	0.37	0.45	0.42
0.8	0.41	–	–	–	–	–	–	–
0.9	0.42	–	–	–	–	–	–	–

5 Implementation

This section describes the pipeline and steps for generating a dataset and the development of a machine learning application. The input for the pipeline are the data gathered during the data collection process described in Section 4.2 and a set of parameters previously described. It outputs deep learning networks in the TensorFlow SaveModel file format. The repository is available under https://gitlab.hpi.de/martin.schlegel/thesis.

I used Python[1] (v3.8.5) as programming language because it offers a great variety of frameworks for data processing and machine/ deep learning purposes. Pandas[2] (v1.1.3), NumPy[3] (v1.19.2) and jointly[4] (v0.1.5) were used for data analysis and manipulation. Researchpy[5] (v0.2.3) was used in the data exploration part. The development and evaluation of the deep learning models were done with the TensorFlow[6] (v2.3.1) and scikit-learn[7] (v0.0) frameworks. Tsfresh[8] (v0.17.0) was used to calculate features for time series data, which are used by the statsmodels[9] (v0.11.1) to perform a logistic regression. Matplotlib[10] (v3.3.2) was used to create plots.

1 https://www.python.org/
2 https://pandas.pydata.org/
3 https://numpy.org/
4 https://github.com/felixmusmann/jointly
5 https://researchpy.readthedocs.io/en/latest/
6 https://www.tensorflow.org/
7 https://scikit-learn.org/stable/index.html
8 https://tsfresh.readthedocs.io/en/latest/index.html
9 https://www.statsmodels.org/stable/index.html
10 https://matplotlib.org/

The pipeline consist of the following steps:

1. **Preparation**

    a) **Fuse**
    This step was performed for the data of each subject. Initially, the single phone, watch and wifi data was transformed in the same format using pandas dataframe. The sensors and the different devices do not capture the data synchronously and include also varying sampling rates. Therefore, the data were resampled to the same constant sampling rate (e.g. 100 Hz or 33 Hz). Doing this, missing values were interpolated as well. Besides, jointly was used to synchronise the data. Afterwards, it was merged into one dataset.

    b) **Cut and Label**
    For each subject the periods which include simulated OCD activities were identified with the captured video. Using this information, the datasets were labelled. Also, unnecessary data was cut at the beginning and the end.

    c) **Merge**
    Finally, the datasets of all single subjects were merged into one final dataset.

2. **Machine/ Deep Learning**

    a) **Preprocess**
    In this step, the dataset was prepared for the machine learning algorithm. This includes windowing, scaling, splitting in train, validation and test samples, the encoding of the IL information and batching.

    b) **Models and Training**
    In this phase, a TensorFlow deep learning model was designed, trained and its hyperparameter were tuned.

    c) **Evaluation**
    In the final step, the model was evaluated according to the metrics chosen.

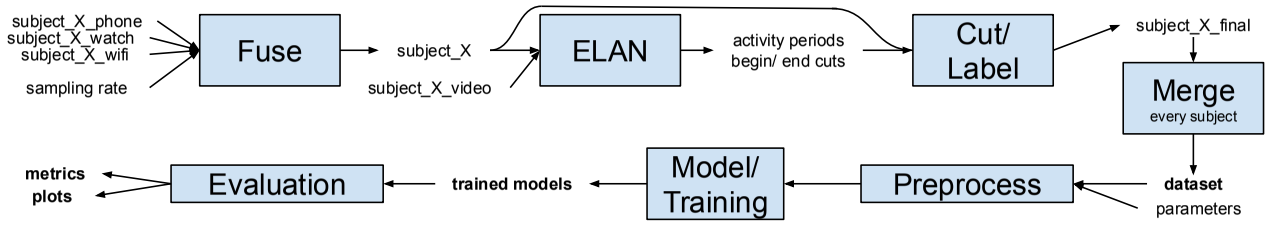

Figure 5.1: The overview of the implemented pipeline. The steps including subject_X as input were performed for all the single subjects before the results got merged to the total dataset. The dataset, trained models and the metrics with the corresponding plots are the important outcome.

6 Evaluation

In this chapter, the results of the former developed model are presented. In Section 6.1, I compare the performances of the model to a baseline composed of a simple logistic regression model. Further improvements of the DL model and the corresponding results are described in Section 6.2.

6.1 Logistic Regression Baseline

A logistic regression is a statistical deterministic model that uses a logistic function to model a binary dependent variable. By this, one can gain insight into the dataset and get a baseline.
The logistic regression is performed on features calculated for windows. Therefore, the dataset was cut into windows of 20 seconds length with an overlap of 50 % and a sampling rate of 100 Hz. For each sensor magnitude in each window the mean, median, sum, standard deviation, variance, min and max value was calculated. Thus, each window was represented by 28 features plus the IL guess (integer encoded).
The statsmodels[1] framework with the default parameter settings was used to perform a first logistic regression. The evaluation was done via the former described LOSO-CV method. The achieved F1-Score was 0.43. The coefficients and p-values of one exemplary model are displayed in Table 6.1. Using the TensorFlow framework a second logistic regression was simulated reusing the previously developed machine learning network with only one fully connected dense layer and the sigmoid function as activation function (basically simply the final layer of the previous network, described in Section 4.5.2). The achieved F1-Score was 0.14.
For NNs that use gradient descent as an optimisation technique, it is advised to bring the features in a similar range [24, p. 264]. The used dataset contains features which have a very different value range. For example, the phone magnetometer magnitude column has a mean of about 400 and covers values in the

1 https://www.statsmodels.org/stable/index.html

Table 6.1: Coefficients and p-values of the features of one exemplary logistic regression model calculated with the statsmodels framework.

Feature	Coefficient	P-Value
location	0.008	0.543
phoneAccMagMax	0.658	0.010
phoneAccMagMean	-76.670	1.000
phoneAccMagMedian	-101.324	0.003
phoneAccMagMin	-3.344	0.002
phoneAccMagSD	0.073	0.010
phoneAccMagSum	-27.166	1.000
phoneAccMagVar	-75.771	0.011
phoneGyroMagMax	-0.154	0.203
phoneGyroMagMean	0.326	1.000
phoneGyroMagMedian	-4.271	0.046
phoneGyroMagMin	31.850	0.000
phoneGyroMagSD	-1.253	0.650
phoneGyroMagSum	0.003	1.000
phoneGyroMagVar	1.282	0.439
phoneMagMagMax	0.021	0.103
phoneMagMagMean	0.017	nan
phoneMagMagMedian	-0.095	0.003
phoneMagMagMin	-0.018	0.077
phoneMagMagSD	-0.078	0.385
phoneMagMagSum	0.000	nan
phoneMagMagVar	-0.003	0.427
watchAccMagMax	-0.259	0.049
watchAccMagMean	0.000	1.000
watchAccMagMedian	44.403	0.000
watchAccMagMin	-1.940	0.033
watchAccMagSD	45.949	0.000
watchAccMagSum	-0.007	1.000
watchAccMagVar	-75.354	0.000

Acc ≙ Acceleration; Gyro ≙ Gyroscope; Mag ≙ Magnetometer/ Magnitude; Max ≙ Maximum; Min ≙ Minimum; SD ≙ Standard Deviation; Var ≙ Variance

range of 203 and 517. On the contrary, the values of the watch acceleration magnitude column are in the range between 0.03 and 10.4 with a mean at around 1.04, see Table 4.3. There exist two common scaling approaches to overcome this. One option is called normalisation (or Min-Max scaling) and is defined with

$$value_{normalised} = \frac{value - value_{min}}{value_{max} - value_{min}} \tag{6.1}$$

where $value_{min}$ and $value_{max}$ are the minimal and maximal values of the feature respectively. By this, the values are shifted and rescaled so that they end up ranging between 0 and 1.
Another method is standardisation. Here the values are centred around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. The mathematical equation is

$$value_{standardised} = \frac{value - value_{mean}}{value_{sd}} \tag{6.2}$$

where the $value_{mean}$ and $value_{sd}$ are the mean and the standard deviation of the feature values. The logistic regression using the TensorFlow framework was repeated with both scaling methods. Thereby, the logistic regression in combination with the standardisation technique achieved a greater **F1-Score** of **0.36**, which is the baseline value the advanced deep learning network will be compared with.

6.2 Further Improvements

In Section 4.5.2 I briefly described the evaluation and training process by using LOSO-CV with seven subjects and the mean of the metrics of every single run. For the final evaluation of the DL network, one would use unseen test data. The test dataset should consist of several subjects because there can be a great difference in performance between different subjects. Therefore, an evaluation based on data from one subject would not provide sufficient insight. Considering the scarcity of subject data, with our data consisting of only seven different subject, the mean metrics from the LOSO-CV are used as final evaluation base.

6.2.1 Final Network

At first, the standardisation approach explained in Section 6.1 was applied to the dataset.
The way how the IL guess was encoded in the previous Section 4.5.2 included a few inaccuracies. For the reason that the probability of the guess was not used, it was assumed to be 100 % correct, which did not correspond to the reality. Secondly, even though when the integer encoding was performed in a way that tried to reflect the spatial dependencies of the locations, the procedure is not completely right. The projection of locations in a two-dimensional space into a one-dimensional series of integer numbers is always associated with errors and inaccuracies. Also, the locations are not ordinal, which is assumed by the integer encoding. To overcome these problems, the IL guess was one-hot encoded and the probability of a guess was included.
The final network structure is presented in Figure 6.1. The input layer has a shape of [(?, 41, 43)]. The '?' refers to the batch size, 41 results from the number of rows a window consists of (33 Hz times 1.25 s) and 43 indicates the number of features (3 axes plus the magnitude for each of the four sensors plus the one-hot encoded IL guesses (26) plus the probability) of one row. The bidirectional LSTM layers have an output shape of (?, 41, 60) and (?, 60), where 60 results from 2 times the 30 units parameter. The whole network contains overall 39 661 trainable parameters. Table 6.2 pictures a summary of the setup and the final hyperparameter settings.
In binary classification tasks, machine learning networks oftentimes calculate a probability value from the given input example. This probability needs to be assigned to one of the two classes in a final step. This is done by comparing the probability with a defined threshold value, which is also a possible hyperparameter to tune. For every single run, the metrics were calculated for 1 000 equally distributed different threshold values between 0 and 1 using a step size of 0.001. The metrics achieving the greatest F1-Score were taken for each run. To sum up the values, the mean was taken.
In Figure 6.2 the precision-recall curve (PRC)s are plotted including the average curve. The PRC plots for all thresholds the precision and the recall against each other. It is a good measurement for binary classification tasks with highly imbalanced datasets. The **maximum mean F1-Score** achieved is **0.53**, with a **precision** of **0.42** and a **recall of 0.71**. It took **1 hour and 4 minutes to train** the network. The grey area marks the standard deviation around the mean curve.

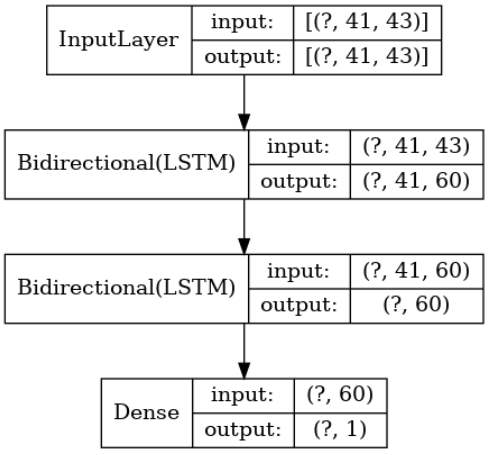

Figure 6.1: The final deep learning network structure.

Table 6.2: The overview of the hyperparameters and the final setup of the deep learning algorithm. The hyperparameters/ setup options above the line were affected by the former hyperparameter tuning process.

Parameter/ Setup	**Value**
Network Structure	2BL + Dense (activation: sigmoid)
Window Size	1.25 s
Units	30
Seen Data	37 s
Optimiser	Adam (learning rate: 0.01)
Batch Size	1024
Sampling Rate	33 Hz
Overlap	no Overlap
IL encoding	1-hot encoding with probability
Class Weights	Equation (4.4)
Loss Function	BCE
Stop Function	explained in Section 4.5.2
Feature Scaling	standardisation

BL $\widehat{=}$ bidirectional LSTM layer

It reaches, with a value of 0.23, nearly the half of the corresponding area under the curve (AUC) value, which indicates a big variance between the single runs.

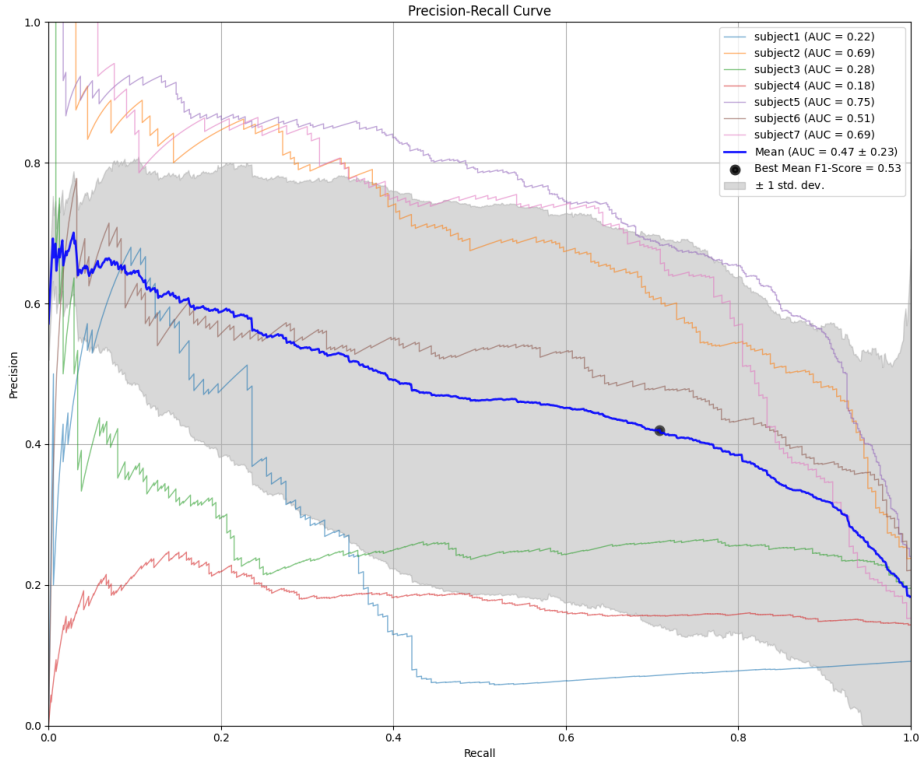

Figure 6.2: The PRC of all runs including the mean curve. The grey area marks the standard deviation around the mean curve. The point indicates the best mean F1-Score.

6.2.2 Additional Networks

Because the sampling rate was reduced to 33 Hz during the tuning process, an additional network was trained with a sampling rate of 100 Hz. It achieved a mean F1-Score of 0.54 and is thereby not significantly higher than the network based on 33 Hz. The PRC is visualised in Figure 6.3.

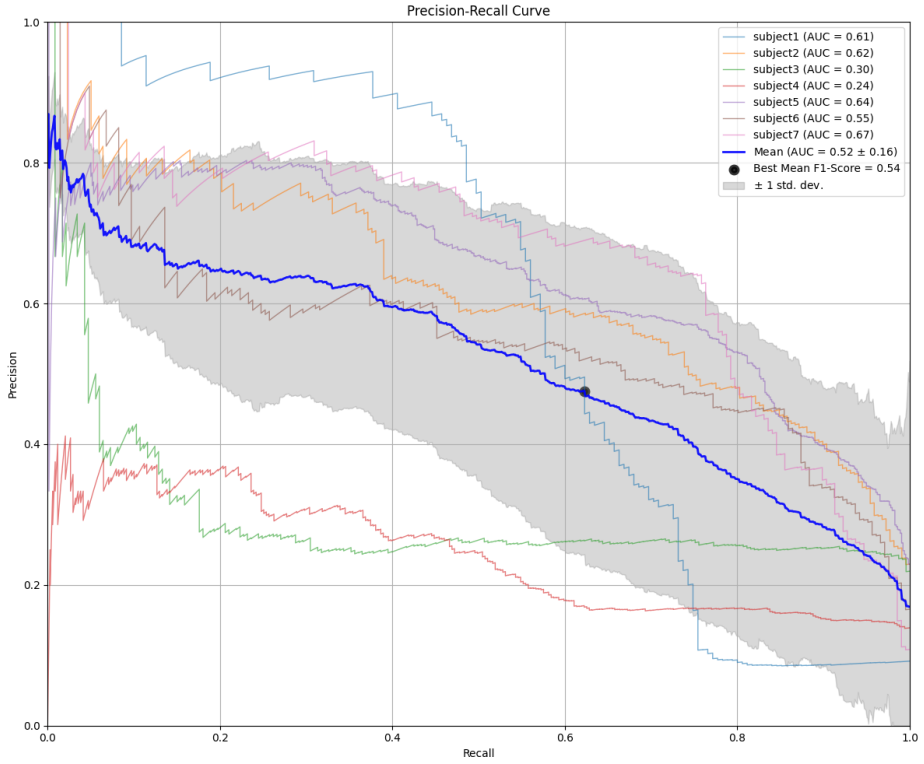

Figure 6.3: The PRC of all runs including the mean curve with a sampling rate of 100 Hz. The grey area marks the standard deviation around the mean curve. The point indicates the best mean F1-Score.

Another network was trained with a sampling rate of 33 Hz and without the IL information to get an insight into the final influence of the IL. It achieved an F1-Score of 0.57, which is by 0.04 greater than the network with the IL information. This suggests that the IL informations are of insufficient quality or that there is no

significant connection between the simulated OCD activities and the locations. Figure 6.4 shows the PRC.

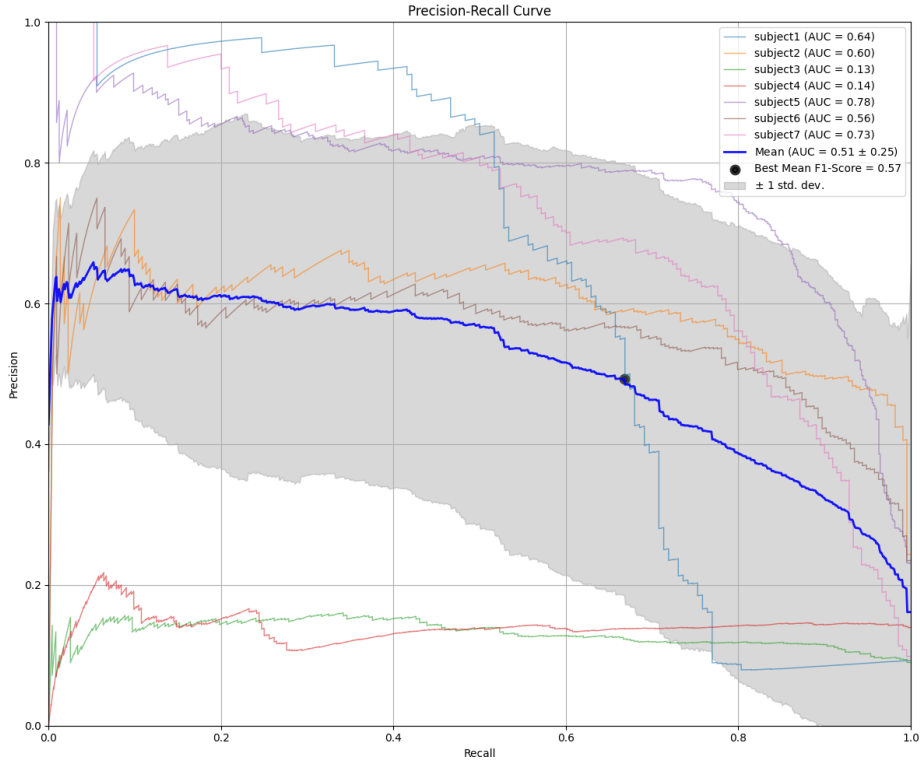

Figure 6.4: The PRC of all runs including the mean curve without the IL information. The grey area marks the standard deviation around the mean curve. The point indicates the best mean F1-Score.

Figure 6.2 shows that Subject 4 is, with an AUC value of 0.18, the worst classified subject. This is the only left-handed subject in the dataset. Therefore, another network excluding this particular subject was trained. It achieved an F1-Score of 0.55 which is slightly higher than the network trained including the subject. The PRC is detailed in Figure 6.5.

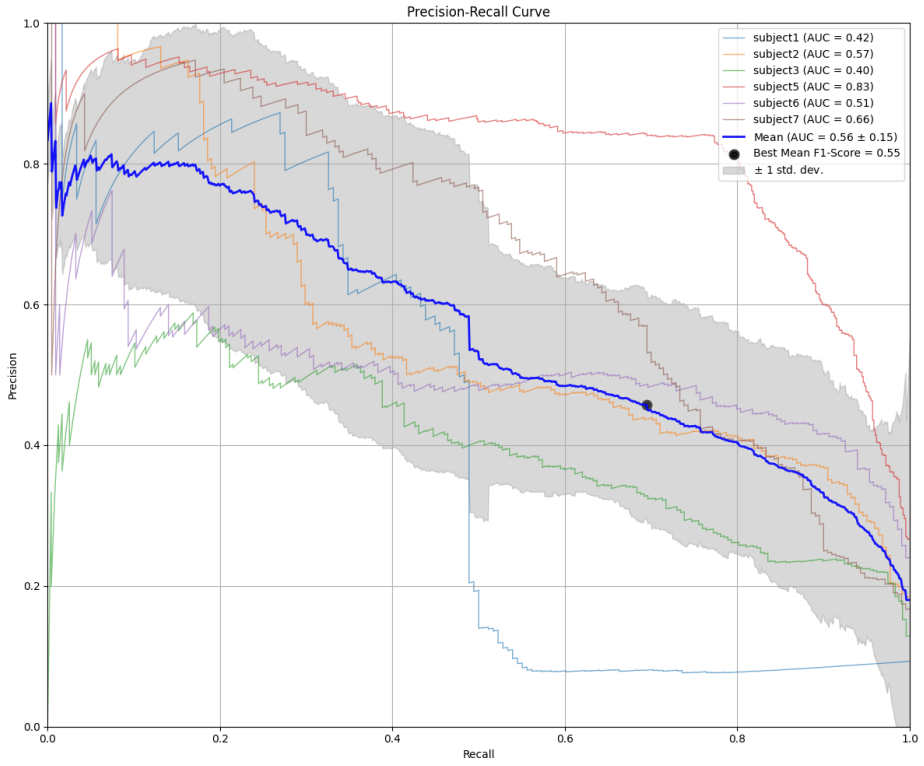

Figure 6.5: The PRC of all runs including the mean curve without subject4. The grey area marks the standard deviation around the mean curve. The point indicates the best mean F1-Score.

7 Discussion

In the following chapter, I summarise and discuss my results regarding the research questions, listed in Section 1.2, and the applied methodologies, Chapter 4. Firstly, I utilise the data gathering process and the created dataset. Afterwards, I evaluate the results of the deep learning algorithm in relation to the outcome of the logistic regression. A further focus will be on the evaluation of the additional networks created in Section 6.2.2.

7.1 Dataset

The first part of the thesis was about the creation of a dataset. To my knowledge, this is the first dataset particularly created in the context of OCD. It includes data from several sensors from different devices (smartphone and smartwatch), plus location information. To collect the data, hardware, devices and a setup was chosen which could also occur in the real-world. It should also be emphasised that the dataset is imbalanced with a strong weighting towards non-OCD activities in order to correspond to the occurrence in the real-world.
In contrast to other mental illnesses, in the case of OCD it was possible to conduct an initial study without approaching patients. The study covers only a part of the physical symptoms (compulsions) of the disease that were simulated. This makes sense in the context of the development of a POC, as one can start in a more ideal world. If a system does not work in these improved circumstances, neither will it with real patients.
The big disadvantage of the dataset is its small size, which is reflected in different aspects. In total the dataset only contains data from seven different subjects and only three different simulated OCD activities, which are all predominantly hand focused. This excludes full-body or feet-focused movements and covers in general just a limited scope. This also increases the influence of handedness, as seen from the poor results obtained in the case of the one left-handed person. Another facet is the overall amount of performed activities, which is particularly low for the *Washing Hands* and

Checking an Oven with 28 and 29 executions, see Table 4.5. Assuming 10 % each are used as test and validation samples, a machine learning algorithm is only trained on approximately 23 examples and validated or tested on only 3 each.

Research Questions:

- *How to conduct a study and gather data for this purpose, especially without having access to affected people?*
  I rate the study as successful, even though it uses simulated data. An important takeaway to note for future data collection, when asking subjects to simulate activities, is the explanation and the degree of freedom given to the subjects in performing them. On the one hand, multiple subjects should perform the same activity differently. On the other hand, the activity has to be similar enough to be recognisable. In this study, subjects were given great freedom. This effect can for example be seen in the different duration of activities. Duration of activities of the same subject are in the same range but across subjects, the time ranges from 12 to 37 seconds, see Table 4.6.

- *What sensors, signals and devices should be used?*
  The devices (smartphone and smartwatch) and integrated sensors (accelerometer, gyroscope and magnetometer) used have proven to be suitable. From Chapter 3 it was possible to derive a justified set of sensors to use. Here is the main take away, that you have to use state-of-the-art standalone smart devices to measure data as unobtrusive as possible to minimise the Hawthorne effect [39]. The influence of IL remains the subject of future research and could not be conclusively assessed in this work.

- *How to create a usable dataset from the gathered data (data fusion from different devices)?*
  The used methodology seemed to be feasible to create a dataset and to fuse the data from different devices. Timestamps were used for the synchronisation, in the special case of the sensor data, additional shake gestures were used. Nevertheless, there are open questions and room for improvements. Sensor data are known to be affected by noise. Therefore, it is advised to use an additional filtering method, such as low/ high pass or Kalman filtering, to clean the data [11].

7.2 Automated OCD Recognition

The later part of the methodology and thesis dealt with the development of a DL model for binary classification of activity data into OCD and non-OCD. The network was trained in a supervised scenario. LOSO-CV was used as the evaluation technique. This means that the data of six of the seven subjects were used for training and one for calculating the metrics. This was repeated until each subject was used once as the test subject. In total seven networks were trained with the same settings, with different input data and evaluated with different test samples. To receive an overall insight of one particular setting setup, the metric values of the seven individual networks were combined using the mean. Due to the small size of the dataset, all subjects were used during the hyperparameter tuning process, explained in Section 4.5.3, and thus influenced the selection of the parameters and the network structure. This has the consequence that in the end, no new data was available for a final evaluation, which should have been carried out with several subjects as well. Thus, the values shown may present the network in a more positive light than a new unseen dataset would.

Research Questions:

- *What ML models are feasible to solve the problem and what parameters are necessary?*
  The plot in Figure 6.2 indicates a mean[1] AUC[2] of 0.47 and a best mean F1-Score of 0.53 witch stems from a precision of 0.42 and a recall of 0.71. The mean AUC has, with a value of 0.23, a very high standard deviation (nearly 50 % of the mean AUC value). This means that there is a high difference between the metrics values of individual networks, which is further reinforced by the small number of subjects, as already described in Section 7.1.
  The recall of 0.71 means that over two-thirds of the 1.25 seconds long windows labelled as OCD were correctly recognised. Just over half (58 %) of all windows classified as OCD were incorrectly classified as OCD. The ratio between these values can be easily adjusted using the classification threshold. For example, the network can be optimised so that more of the

1 The term mean in relation to the plots always refers to the values of the mean curve.
2 Since PRCs are examined, the AUC values relate to the area under the PRCs.

windows labelled as OCD are recognised as such, for the price of more false positives. Of course, this also works in the other direction and the percentage of windows classified wrongly as OCD can be reduced, but at the expense of the recall. The different possible ratios can be viewed using the PRC, see Figure 6.2. At the moment, the threshold is set for each network (one per subject to perform LOSO-CV) individually to maximise the F1-Score, marked by the black dot in the plot.
These values only reflect the detection rates to the windows. In some cases, it is unclear how windows map to individual activities, which extend over several windows. For example, activities may be recognised, but not classified over their entire length. Maybe only two-thirds of an activity is classified as such, or only two-thirds of all activities are recognised, but if so, then completely. Another possibility is that there are areas in which windows classified as OCD and non-OCD alternate. Especially in the latter case, a reasonable post-processing can increase the values of the metrics by an additional step recognising outlier windows that differ from the surrounding.
Even if the F1-Score of 0.53 is not very high, the performance of the machine learning algorithms should not be viewed too critically. The performances are likely to improve if more training samples (subjects) would be available. Additionally, the chosen LOSO-CV technique resulted in the development of so-called impersonal models, which can be worse in comparison to personal models (where models are trained and tested with data from the same subject). Impersonal models are especially bad for hand focused activities [13]. The results of the network are also supported by the results of the performed logistic regression, which reached F1-Scores of 0.36 and 0.43, respectively. These values are also low and were exceeded especially during the hyperparameter tuning process by the recurrent network model (F1-Score went from 0.367 to 0.53). Even if the values of the logistic regression were calculated with changed parameter settings (window size, sampling rate), the same dataset was used and thus it gave an insight into which value ranges should be realistically reached or exceeded by the deep learning algorithm. The performances of the baseline are inline with the behaviour expected, and conform to how the DL network acts.

- *How can IL be leveraged to improve such a system?*
  To answer this question an additional network was trained, removing the IL information from the used dataset. Against the expectations, supported by Figure 4.3 and the results of the chi-squared test (Table 4.4), the plot in Figure 6.4 (without the IL information) shows an improvement of the metrics values. The mean AUC and the best F1-Score are, with 0.51 and 0.57 (this is the maximum achieved value over all networks), greater than the achieved values of the network trained with the IL information. Possible reasons for this are that the quality of the IL information is insufficient or that there is simply no significant correlation between the class of the activity and the IL information. This is supported by the p-value of 0.543 (greater than 0.01) for the location feature in the logistic regression summary presented in Table 6.1. Considering the poor data quality, these results are hard to generalise to future models.

The network trained with a dataset of a sampling rate of 100 Hz achieved slightly better results (AUC: 0.52, best F1-Score: 0.54) than the original network, see Figure 6.3. Noteworthy is the significantly lower standard deviation of 0.16 from the original 0.23, which shows that the results of the individual subjects are closer together using the higher amount of available data. Generally speaking, both networks are similar and support the statement in [9], [10] that a low sampling rate is sufficient for HAR.

Removing the only left-handed subject, increased the achieved results to AUC: 0.56 and best F1-Score: 0.55, see Figure 6.5. The standard deviation was reduced to 0.16. Excluding subject4 only from the calculations of the metrics for the original network results in a mean AUC of 0.52 with a standard deviation of 0.21. The results for the networks trained without subject4 are also better than these. This indicates that the mean metrics of the original networks are not only worse because of the direct influence of subject4 as a test subject during the calculations of the mean metrics, but also that subject4 had a negative influence during the training when networks were trained for another test subject.

Is it possible to develop an automated system to detect and address OCD by means of recognising the activity of patients using a dataset including sensor data from smart devices?
In the end, this question is very difficult to answer, as the dataset created and

used only contains a small subset of simulated compulsion-related activities and thus only offers a small insight. In principle, there is an almost infinite number of different compulsion-related activities, which can differ greatly between individuals. This thesis proved that for specific activities, under certain assumptions, it is possible to detect their happening. A deeper evaluation is required, which provides precise insights into the relationship between the classified windows and the OCD related activities. At the moment the evaluation is limited to the window level.
To address the issue of having a large number and individual-specific compulsive activities, a possible direction is to try an unsupervised approach focused on the recognition of repetitive activities. Since there are also a lot of natural ADL who have repetitive characters (e.g. eating-related activities or brushing teeth).

8 Conclusion

This chapter gives a summary of the work in relation to the introduction and the contributions of this work in light of the experimental results reported. In particular, the limitations of the developed system are discussed and directions for further research are given.

The purpose of this thesis was to develop a POC for a system which can automatically detect OCD. Thereby, a foundation for future research as well as first insights into the capability of adapting HAR to the context of CAR should be established. To achieve this, relevant data was collected and a system with the goal of automatically detecting OCD compulsions was built.
The created dataset contains data from seven subjects, including one left-handed. Approximately one hour of data was collected from each participant, with about five minutes of pre-set activities being performed. Three different activities were defined (washing hands, checking door, checking oven), all of which have a repetitive character and are, as the literature has shown (Section 2.1), exemplary for OCD patients. In order to have a real-world setup, inertial sensors (accelerometer, gyroscope and magnetometer) from smartphones and smartwatches were used to collect data. In addition, the IL of the subject was recorded with the assumption that it would improve the detection rate.
After preprocessing the raw data and calculating a set of relevant features, a DL model was trained and evaluated. The created network is a binary classifier capable to distinguish between OCD and non-OCD behaviour. It was trained in a supervised scenario and evaluated using LOSO-CV. The neural network architecture is a RNN, using LSTM layers.

With an achieved F1-Score of 0.53, the developed system is able to detect some of the simulated OCD activities. The temporal aspect of the OCD definition states that compulsions and obsessions must take at least one hour per day. This means that an affected person produces a lot of data over the course of one day, which in turn means many opportunities for detection.

Ultimately, this thesis gives a small insight into the novel topic of automatic detection of OCD and shows directions and ways in which the area can be further explored.

8.1 Limitations

The following section outlines the limitations that arose in carrying out this study.
The data that can be classified by the developed network must correspond to the data that was used for training. This means that the hardware setup must be similar, which results in several conditions. First of all, a person must own and carry the equipment to be used in the correct way as defined within the study. For example, the smartphone, from which the sensors were read out, must be carried in the right trouser pocket and the smartwatch (or another device with an acceleration sensor) has to be on the left wrist. This is a limitation that can affect left-handed users.
The required IL information limits the system to indoor environments with IL system capability. Simultaneously, these conditions are only met by a small group of people, often only partially or for a limited time.
Compulsions do not appear as a symptom of OCD in all affected persons. In some cases, the disease is confined to obsessions. Also, not all compulsions are characterised by a repetitive character and they can differ greatly between the individuals. Thus not all affected people could be helped even with a perfectly functioning system.
Currently, the system is limited to the detection of the three defined simulated OCD activities. Even if this were not the case and other activities could be identified, the system would be restricted to a maximum activity length. That is because the trained DL system is bounded by the units (30) and window size (1.25 seconds) parameters to process only 37.5 seconds of data at ones.

8.2 Future Work

Concerning the dataset, size has a direct effect on the performances of DL networks. This study was conducted with a narrow focus on certain activities and a limited number of subjects. Increasing the number of training subjects, therefore increasing the information seen by the NNs in the training phase,

can lead to improved performances. Besides, the IL system did not produce satisfactory results.
A new study should contain more activities (possibly not hand-focused ones). In any case, it would benefit from including more participants, to enable a proper final evaluation of the developed system and to reduce the influence of individual participants. More than one hour of data should be collected from each subject to be able to carry out each activity appropriately often (especially if more activities are defined).
Even if IL did not have a positive influence on this work, the approach sounds promising in theory. For this purpose, a reliably functioning IL system should first be set up and evaluated independently before including it into the system. To develop and collect more knowledge and expertise in this area, simulated data should be used in the next study before one start to collect data from affected people. In the study carried out as part of this thesis, certain simulated activities were defined to the best of my knowledge. In order to make these activities as good and as real as possible, an expert in this field should be consulted in the future.

The developed method could be used to train personal models, as the thesis has shown that even the execution of the same activities can vary greatly between different people, even in a simulated case. This could be of particular interest in the context of a relapse detection system, by collecting data from affected individuals before/ at the beginning of treatment. Using this data a personalised detection system could be developed and used for relapse detection after successful treatment.
By using a supervised scenario, the system developed so far is limited to pre-defined OCD activities. This could be avoided by using unsupervised or semi-supervised scenarios and trying to focus more on recognising the general repetitive character of compulsions. Due to the performance of the current network, this should not be the focus of future work.
Another opportunity is the current limitation of the evaluation to the window level. An evaluation at the activity level is important because the system is aimed to recognise whole activities and not just parts of them. As described in Section 7.2, an artefact of the window-level classification is the possibility that successive windows alternate in their predicted classes, creating classification profiles that could never occur in the real-world. A post-processing mechanism

could be developed to prevent this.
The used DL algorithm is based on recurrent structures (LSTM). Alternatively, is worthwhile exploring convolutional architectures (CNN), which are frequently used in the field of HAR, as their design is intended to detect long activities. Convolutional and recurring structures have also been used in conjunction in a model called convolutional LSTM (ConvLSTM) [40], which tries to combine the advantages of the two structures. It uses LSTM to recognise the general repetitive structure of overall activities, in which individual sub-activities are detected using the convolution technique. The recognition of compulsions with their repetitive character seems to be a perfect field for applying this model.
A final important point to be considered is the optimisation of the classification threshold described in Section 7.2. This is currently set to maximise the F1-Score. In the future, this value should be more focused on increasing the recall. This may result in a reduction of the precision, but it is less bad to classify activities wrongly as OCD than to overlook OCD behaviour.

The work has shown that a distinction between simulated compulsion-related activities and non-pathological behaviour in a simulated environment is partially possible. The thesis gives a first insight into the topic and points out directions for reasonable future research.

Bibliography

[1] Z. Steel, C. Marnane, C. Iranpour, T. Chey, J. W. Jackson, V. Patel and D. Silove, **The global prevalence of common mental disorders: A systematic review and meta-analysis 1980–2013**, *International Journal of Epidemiology*, vol. 43:no. 2, 476–493, ISSN: 1464-3685, 0300-5771. DOI: 10.1093/ije/dyu038. [Online]. Available: https://academic.oup.com/ije/article-lookup/doi/10.1093/ije/dyu038 (visited on 01/10/2020) (see page 2).

[2] H. Ritchie and M. Roser. (2018). 'Mental Health', [Online]. Available: https://ourworldindata.org/mental-health (visited on 28/09/2020) (see page 2).

[3] A. P. Association and A. P. Association, Eds., *Diagnostic and statistical manual of mental disorders: DSM-5*, 5th ed, Washington, D.C: American Psychiatric Association, 2013, 947 pp., ISBN: 978-0-89042-554-1 (see pages 2–4, 7–9, 25).

[4] J. E. Grant, S. R. Chamberlain and B. L. Odlaug, **Clinical Guide to Obsessive Compulsive and Related Disorders**. Oxford, UK: Oxford University Press, Aug. 2014. DOI: 10.1093/med/9780199977758.001.0001. [Online]. Available: https://oxfordmedicine.com/view/10.1093/med/9780199977758.001.0001/med-9780199977758 (see pages 3, 5, 7–10, 24).

[5] E. Burchi, E. Hollander and S. Pallanti, **From treatment response to recovery: A realistic goal in OCD**, *International Journal of Neuropsychopharmacology*, vol. 21:no. 11, 1007–1013, ISSN: 1461-1457, 1469-5111. DOI: 10.1093/ijnp/pyy079. [Online]. Available: https://academic.oup.com/ijnp/article/21/11/1007/5090057 (visited on 12/05/2020) (see pages 4, 9).

[6] J. L. Eisen, N. J. Sibrava, C. L. Boisseau, M. C. Mancebo, R. L. Stout, A. Pinto and S. A. Rasmussen, **Five-year course of obsessive-compulsive disorder: Predictors of remission and relapse [CME]**, *The Journal of Clinical Psychiatry*, vol. 74:no. 3, 233–239, ISSN: 0160-6689. DOI: 10.4088/JCP.12m07657. [Online]. Available: http://article.psychiatrist.com/?ContentType=START&ID=10008205 (visited on 12/05/2020) (see pages 4, 9).

[7] U. Albert, F. Barbaro, S. Bramante, G. Rosso, D. De Ronchi and G. Maina, **Duration of untreated illness and response to SRI treatment in obsessive-compulsive disorder**, *European Psychiatry*, vol. 58, 19–26, ISSN: 0924-9338, 1778-3585. DOI: 10.1016/j.eurpsy.2019.01.017. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0924933800009354/type/journal_article (visited on 12/05/2020) (see pages 4, 9, 10).

[8] D. Veale and A. Roberts, **Obsessive-compulsive disorder**, *BMJ*, vol. 348, g2183–g2183, ISSN: 1756-1833. DOI: 10.1136/bmj.g2183. [Online]. Available: http://www.bmj.com/cgi/doi/10.1136/bmj.g2183 (visited on 12/05/2020) (see pages 4, 10).

[9] W. Sousa Lima, E. Souto, K. El-Khatib, R. Jalali and J. Gama, **Human activity recognition using inertial sensors in a smartphone: An overview**, *Sensors*, vol. 19:no. 14, 3213, ISSN: 1424-8220. DOI: 10.3390/s19143213. [Online]. Available: https://www.mdpi.com/1424-8220/19/14/3213 (visited on 12/05/2020) (see pages 4, 21–23, 45, 46, 49, 72).

[10] G. Yuan, Z. Wang, F. Meng, Q. Yan and S. Xia, **An overview of human activity recognition based on smartphone**, *Sensor Review*, vol. 39:no. 2, 288–306, ISSN: 0260-2288. DOI: 10.1108/SR-11-2017-0245. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/SR-11-2017-0245/full/html (visited on 12/05/2020) (see pages 4, 5, 22, 23, 45, 46, 72).

[11] H. F. Nweke, Y. W. Teh, G. Mujtaba and M. A. Al-garadi, **Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions**, *Information Fusion*, vol. 46, 147–170, ISSN: 15662535. DOI: 10.1016/j.inffus.2018.06.002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1566253518304135 (visited on 12/05/2020) (see pages 4, 22, 26, 34, 45, 69).

[12] J. Wang, Y. Chen, S. Hao, X. Peng and L. Hu, **Deep learning for sensor-based activity recognition: A survey**, *Pattern Recognition Letters*, vol. 119, 3–11, ISSN: 01678655. DOI: 10.1016/j.patrec.2018.02.010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016786551830045X (visited on 12/05/2020) (see pages 4, 21, 22, 45).

[13] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda and A. J. Schreiber, **Smartwatch-based activity recognition: A machine learning approach**, in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, presented at the 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), Las Vegas, NV, USA: IEEE, Feb. 2016, 426–429, ISBN: 978-1-5090-2455-1. DOI: 10.1109/BHI.2016.7455925. [Online]. Available: http://ieeexplore.ieee.org/document/7455925/ (visited on 07/10/2020) (see pages 4, 5, 22, 23, 71).

[14] R. M. Al-Eidan, H. Al-Khalifa and A. M. Al-Salman, **A review of wrist-worn wearable: Sensors, models, and challenges**, *Journal of Sensors*, vol. 2018, 1–20, ISSN: 1687-725X, 1687-7268. DOI: 10.1155/2018/5853917. [Online]. Available: https://www.hindawi.com/journals/js/2018/5853917/ (visited on 10/10/2020) (see pages 5, 10).

[15] V. Starcevic, D. Berle, V. Brakoulias, P. Sammut, K. Moses, D. Milicevic and A. Hannan, **Functions of compulsions in obsessive–compulsive disorder**, *Australian & New Zealand Journal of Psychiatry*, vol. 45:no. 6, 449–457, ISSN: 0004-8674, 1440-1614. DOI: 10.3109/00048674.2011.567243. [Online]. Available: http://journals.sagepub.com/doi/10.3109/00048674.2011.567243 (visited on 12/05/2020) (see pages 9, 24).

[16] P. Seibell and E. Hollander, **Management of obsessive-compulsive disorder**, *F1000Prime Reports*, vol. 6, ISSN: 20517599. DOI: 10.12703/P6-68. [Online]. Available: http://f1000.com/prime/reports/m/6/68 (visited on 12/05/2020) (see page 10).

[17] S. Wilson and R. Laing, **Wearable technology: Present and future**, 16 (see page 10).

[18] C. W. J. Granger and P. Newbold, **Forecasting economic time series**, 2nd ed, ser. Economic theory, econometrics, and mathematical economics. Orlando: Academic Press, 1986, 338 pp., ISBN: 978-0-12-295183-1 (see page 10).

[19] (2020). 'Definition of smartphone from the Cambridge Advanced Learner's Dictionary & Thesaurus', Cambridge University Press, [Online]. Available: https://dictionary.cambridge.org/dictionary/english/smartphone (visited on 11/10/2020) (see page 10).

[20] F. Zafari, A. Gkelias and K. Leung, **A survey of indoor localization systems and technologies**, *arXiv:1709.01015 [cs]*. arXiv: 1709.01015. [Online]. Available: http://arxiv.org/abs/1709.01015 (visited on 05/08/2020) (see pages 11, 13).

[21] Y. Gu, A. Lo and I. Niemegeers, **A survey of indoor positioning systems for wireless personal networks**, *IEEE Communications Surveys & Tutorials*, vol. 11:no. 1, 13–32, ISSN: 1553-877X. DOI: 10.1109/SURV.2009.090103. [Online]. Available: http://ieeexplore.ieee.org/document/4796924/ (visited on 23/10/2020) (see pages 11, 13).

[22] M. Mohri, A. Rostamizadeh and A. Talwalkar, **Foundations of machine learning**, Second edition, ser. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2018, 486 pp., ISBN: 978-0-262-03940-6 (see pages 13–15, 17, 18, 20).

[23] T. Oladipupo, 'Types of machine learning algorithms', in *New Advances in Machine Learning*, Y. Zhang, Ed., InTech, 1st Feb. 2010, ISBN: 978-953-307-034-6. DOI: 10.5772/9385. [Online]. Available: http://www.intechopen.com/books/new-advances-in-machine-learning/types-of-machine-learning-algorithms (visited on 22/10/2020) (see page 17).

[24] E. Alpaydin, **Introduction to machine learning**, ser. Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 2014, 616 pp., ISBN: 978-0-262-02818-9 (see pages 17, 18, 59).

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, **Dropout: A simple way to prevent neural networks from overfitting**, 30 (see pages 19, 46, 53).

[26] A. Bulling, U. Blanke and B. Schiele, **A tutorial on human activity recognition using body-worn inertial sensors**, *ACM Computing Surveys*, vol. 46:no. 3, 1–33, ISSN: 0360-0300, 1557-7341. DOI: 10.1145/2499621. [Online]. Available: https://dl.acm.org/doi/10.1145/2499621 (visited on 25/11/2020) (see page 21).

[27] A. Dehghani, O. Sarbishei, T. Glatard and E. Shihab, **A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors**, *Sensors*, vol. 19:no. 22, 5026, ISSN: 1424-8220. DOI: 10.3390/s19225026. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/5026 (visited on 20/06/2020) (see pages 23, 46, 49).

[28] (Sep. 2019). 'Number of smartphone users worldwide from 2016 to 2021', Statista, [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide (visited on 18/09/2020) (see page 25).

[29] (Feb. 2019). 'Number of connected wearable devices worldwide by region from 2015 to 2022', Statista, [Online]. Available: https://www.statista.com/statistics/490231/wearable-devices-worldwide-by-region (visited on 18/09/2020) (see page 25).

[30] L. Liao, D. Fox and H. Kautz, **Location-based activity recognition**, 8 (see page 26).

[31] (Nov. 2014). 'Anteil der Haushalte mit WLAN in den führenden Ländern weltweit 2014', Statista, [Online]. Available: https://de.statista.com/statistik/daten/studie/222242/umfrage/anteil-der-haushalte-mit-w-lan-in-ausgewaehlten-laendern/#professional (visited on 18/09/2020) (see page 26).

[32] H. Akoglu, **User's guide to correlation coefficients**, *Turkish Journal of Emergency Medicine*, vol. 18:no. 3, 91–93, ISSN: 24522473. DOI: 10.1016/j.tjem.2018.08.001. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2452247318302164 (visited on 23/09/2020) (see page 39).

[33] A. Shewalkar, D. Nyavanandi and S. A. Ludwig, **Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU**, *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9:no. 4, 235–245 (see page 45).

[34] D. P. Kingma and J. Ba, **Adam: A method for stochastic optimization**, *arXiv:1412.6980 [cs]*. arXiv: 1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 27/10/2020) (see page 47).

[35] P. Branco, L. Torgo and R. Ribeiro, **A survey of predictive modelling under imbalanced distributions**, *arXiv:1505.01658 [cs]*. arXiv: 1505.01658. [Online]. Available: http://arxiv.org/abs/1505.01658 (visited on 24/11/2020) (see page 48).

[36] S. L. Smith, P.-J. Kindermans, C. Ying and Q. V. Le, **Don't decay the learning rate, increase the batch size**, *arXiv:1711.00489 [cs, stat]*. arXiv: 1711.00489. [Online]. Available: http://arxiv.org/abs/1711.00489 (visited on 27/10/2020) (see page 51).

[37] M. Schuster and K. Paliwal, **Bidirectional recurrent neural networks**, *IEEE Transactions on Signal Processing*, vol. 45:no. 11, 2673–2681, ISSN: 1053587X. DOI: 10.1109/78.650093. [Online]. Available: http://ieeexplore.ieee.org/document/650093/ (visited on 28/10/2020) (see page 52).

[38] T. Bluche, C. Kermorvant and J. Louradour, **Where to apply dropout in recurrent neural networks for handwriting recognition?**, 681–685 (see page 52).

[39] P. Sedgwick and N. Greenwood, **Understanding the hawthorne effect**, *BMJ*, h4672, ISSN: 1756-1833. DOI: 10.1136/bmj.h4672. [Online]. Available: https://www.bmj.com/lookup/doi/10.1136/bmj.h4672 (visited on 23/11/2020) (see page 69).

[40] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong and W.-c. Woo, **Convolutional LSTM network: A machine learning approach for precipitation nowcasting**, 802–810 (see page 77).

List of Tables

List of Figures