UNIVERSITÄT
LEIPZIG

Institut für Informatik
Fakultät für Mathematik und Informatik
Abteilung Datenbanken

Re-Identification Attacks on Smartwatch Health Data

Masterarbeit

vorgelegt von:
Tobias Schreieder

Matrikelnummer:
3763331

Betreuer:
Prof. Dr. Erhard Rahm
Lucas Lange

Abstract

Wearing a smartwatch enables efficient collection of health data, which can be used for research and comprehensive analysis given the large number of high-quality smartwatch sensors. The smartwatch manufacturers themselves also offer their users various applications, such as sleep monitoring, fall detection, and stress detection, with the goal of improving the health of the individual. However, in addition to numerous analysis and self-optimization options, ensuring privacy when handling health data is an important concern, as the collection and analysis of such data is now ubiquitous. In particular, health data contains sensitive information about the users of smartwatches, which makes it necessary to handle it in a particularly responsible way. In practice, this is often reflected in the use of a de-identification approach, which removes any information that directly identifies the user, such as name, address, IP address, etc., from the collected data. However, the data itself can also be exploited to reveal information and break the supposed anonymity. In this thesis, a novel modular attack framework with a total of four similarity-based re-identification attacks on time series health data is presented, the use of which reveals significant weaknesses in the de-identification approach. The data base is the WESAD data set, with multi-modal smartwatch health data from a total of 15 subjects, as well as two synthetic data sets generated with generative adversarial networks, each with up to 1000 subjects. Despite privacy-preserving measures, the attacks show that a short amount of different sensor data from a target person is sufficient to potentially identify them in a database of other samples, based solely on similarities at the sensor level. To compute the similarity between two samples, the use of dynamic time warping proved to be very useful. For the example scenario where data owners use health data from smartwatches, the results of this work show that the target data can be correctly matched in 100% of cases for the WESAD data set and in over 93% of cases for the two large-scale synthetic data sets. These results highlight that user privacy is already threatened by the data itself, even when personal information is removed. To address this privacy threat, the use of several different privacy models is discussed, and a case study is conducted in which random noise values of varying levels are added to the health data in addition to de-identification. The subject of the study is to compare the curve of the re-identification risk of the four attacks with the usability of the data for a stress detection application, under the influence of different noise levels. The results of this case study clearly demonstrate that noisy data results in a significantly lower re-identification risk for the users contained in the data, while still achieving solid stress detection classification results.

Tobias Schreieder
3763331

Preamble

This thesis builds on the research results of a previous project in which I myself played a key role. The results of this project have already been published as a preprint entitled "Privacy at Risk: Exploiting Similarities in Health Data for Identity Inference" [1]. This preprint addressed the question whether dynamic time warping is fundamentally suitable as a distance measure for the re-identification of individuals in a smartwatch health data set. For this purpose, an initial attack was developed, which is referred to in this thesis as "Single-DTW-Attack". The first version of a rank-based evaluation pipeline for the evaluation of similarity-based re-identification attacks was also presented. In this master thesis, the concepts presented in the preprint are adopted and transferred into a novel attack framework, which is able to integrate and combine various other attacks, new data sets, different steps of data preprocessing, signal processing and data visualization. In addition, new experiments are carried out with regard to the reduction of the re-identification risk by integrating a stress detection task with noisy data. Certain minor passages of this work, especially in chapter 2, chapter 3 and chapter 4, are taken in major parts from the preprint. Corresponding passages are marked with an asterisk (*) at the end of each paragraph.

Tobias Schreieder
3763331

Acknowledgements

Special thanks to Nils Wenzlitschke for the preliminary work on training generative adversarial networks on the WESAD data set in the master thesis *Privacy-Preserving Smartwatch Health Data Generation For Stress Detection Using GANs* and for providing his code for further experiments in this thesis. Thanks are also extended to Prof. Eamonn Keogh for providing useful tips on implementing and selecting efficient dynamic time warping methods. Finally, I would like to thank the Leipzig University for providing access to the scientific computing infrastructure, without which the extensive experiments in this thesis could not have been carried out.

Tobias Schreieder
3763331

Contents

List of Abbreviations

ACC	Accelerometer
BVP	Blood Volume Pulse
cGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
DBA	Dynamic Time Warping Barycenter Averaging
DGAN	DoppelGANger
DSF	Downsampling Factor
DTW	Dynamic Time Warping
EDA	Electrodermal Activity
GAN	Generative Adversarial Network
HMAC	Hash-Based Message Authentication Code
HR	Heart Rate
HRV	Heart Rate Variability
IoT	Internet of Things
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory Neural Network
MLP	Multilayer Perceptron Neural Network
NLAAF	Nonlinear Alignment and Averaging Filters
NP	Noise Multiplier
PCA	Principal Component Analysis
PPG	Photoplethysmography
PPRL	Privacy-Preserving Record Linkage
PSA	Prioritized Shape Averaging
RNN	Recurrent Neural Network
RQ	Research Question
TEMP	Skin Temperature
WESAD	Wearable Stress and Affect Detection

List of Figures

List of Tables

1. Introduction

Motivation. The term Internet of Things (IoT) covers a wide range of application areas, such as smart cities, agriculture, transportation, medicine, industry and manufacturing [2]. One area of application, which has grown considerably in recent years, is the area of wearable devices such as smartwatches. Sales of smartwatches already amounted to 149 million units worldwide in 2022, while 206 million smartwatches are forecast to be sold in 2027 [3]. Wrist-worn smartwatches are used for a variety of activities. These include reading messages and using the calendar, but also tracking and recording personal health data, such as sleep tracking, stress detection, recording heart rate and steps. For the purpose of health monitoring, smartwatches are equipped with various high-quality sensors that make it possible to record a person's sensitive health data over the long term. Sikder et al. [4] have shown what serious threats this can lead to and what attacks can be carried out on smart devices. While suitable devices are becoming more and more widespread and the amount of data collected is increasing rapidly as a result, the issue of data protection is becoming increasingly important and user awareness is growing. Ernst and Ernst [5] found that the perceived data protection risk has a direct influence on device acceptance and could ultimately prove to be a decisive factor for interested users. In addition to general privacy concerns regarding personal data, there is also a direct correlation between perceived risk and trust in a data owner's privacy promise. The primary reason for this may be the fact that as soon as a user's data is collected, the responsibility for privacy protection is completely transferred to the collecting institution, which is why the users should be fully informed about threats and defensive measures.

De-identification is a common mechanism for preserving privacy, with the aim of preventing a user's personal identity from being revealed. For example, the privacy policy of a smartwatch distributor states that they "... may share non-personal information that is aggregated or de-identified so that it cannot reasonably be used to identify an individual" [6, 7]. At first glance, de-identification may conceal the identity of the user, but it does not remove the inherent characteristics of an individual encoded in the data. For this reason, de-identification may not be an effective protection against identity inference [8].

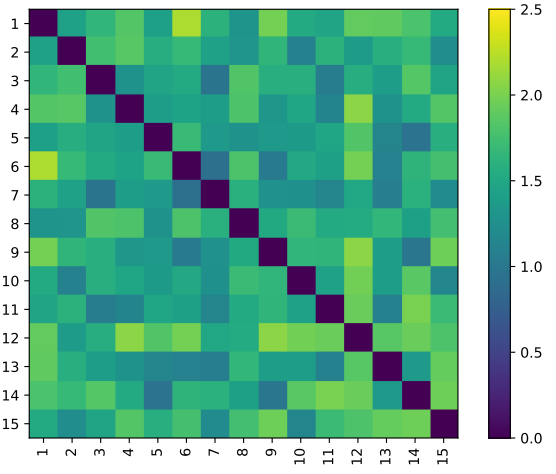


Figure 1.1.: DTW distance heatmap for 15 subjects of the WESAD data set, where smaller scores indicate stronger similarity.

In this work, this risk will be demonstrated by four similarity-based re-identification attacks. The attacks exclusively use a short time series of a target user's health data to reconnect their de-identified data samples within a data set. All four attacks are based on the distance measure dynamic time warping (DTW) [9] to compare the time series with each other and exploit the common characteristics of the provided multimodal sensor data. Figure 1.1 shows a matrix with calculated DTW distances between 15 subjects of the WESAD data set [10], where the diagonal line compares each subject with itself. It can be seen from the matrix

that the distances between subjects vary constantly. However, the proposed attack may show that even the often only small differences in distance, which can be recognized by a very similar color, offer the potential to distinguish the original individuals. This thesis investigates whether the presented re-identification attacks can effectively break the de-identification, especially in the example scenario in which institutions collect and use health data from smartwatches.

This thesis provides the following four research contributions. (1) An attack framework with four novel re-identification attacks based on DTW distances for smartwatch time series data is presented. (2) For the first time, data-specific optimization strategies that exploit the multi-modal and biological characteristics of the underlying health data are evaluated. (3) The results of this work reveal the inherent re-identification threats that may be present in personal smartwatch health data. (4) The derived findings are of practical relevance for data collection with smartwatches, for which user privacy is currently being implemented on a large scale through de-identification.

Research Questions. The following research questions (RQs) in particular will be addressed in detail in this thesis. A final evaluative observation of the RQs is provided in chapter 7.

- *RQ1:* How severe is the actual threat level that is reached in the attack scenario?
- *RQ2:* Are synthetic data sets generated by generative adversarial networks suitable for determining the scalability of re-identification attacks?
- *RQ3:* To what extent are the re-identification attacks tailored to precisely this task? Can they be adapted to other data sets?
- *RQ4:* Are there also beneficial use cases for such similarity searches?
- *RQ5:* What are possible defense mechanisms that are more appropriate than de-identification? How much do these affect the usability of the data for other use cases?

Structure. Following this brief introduction, chapter 2 provides an overview of the key concepts and methods to ensure a good understanding. These include health data for stress detection, the generative adversarial network (GAN), re-identification and dynamic time warping. Chapter 3 then provides a brief literature review of previously published re-identification attacks in comparison to the attacks in this thesis. The methodology is presented in chapter 4. The underlying attack scenario is first outlined here. This is followed by an overview of all the components of the attack framework, which are examined in detail in the subsequent steps. The signal preprocessing and possible complexity reduction methods are also shown. Finally, the architecture of the four novel DTW re-identification attacks is explored. Chapter 5 presents the experimental setup. The WESAD data set, which is used to test the attacks, and two synthetic GAN data sets for testing the scalability are presented. The basic data model and the choice of a rank-based evaluation are then briefly described. Furthermore, the self-developed evaluation pipeline, which is integrated into the attack framework and uniformly evaluates DTW attacks, is introduced. Afterward, the focus is shifted to privacy experiments in order to show possible countermeasures. All presented experiments are evaluated in detail in chapter 6. The RQs are addressed again in chapter 7. The chapter attempts to answer them and the limitations of the DTW attacks are analyzed. Finally, a brief summary of the key findings of this thesis is provided in chapter 8.

2. Background

Various fundamental concepts are used in this work, which are not explained in full in the relevant chapters. To ensure a good understanding of this thesis, the relevant topics are presented in detail in this chapter. Section 2.1 generally deals with health data for stress detection tasks. In this context, the benefits of wearables in the healthcare sector will be examined, the possibility of creating health data with the Empatica E4 smartwatch will be demonstrated and an overview of various stress detection methods will be given. Next, section 2.2 presents the functionality of a generative adversarial network, which is to be used to create large synthetic data sets. The two concepts of re-identification and identity inference are then briefly introduced in section 2.3. Finally, section 2.4 deals with the alignment algorithm dynamic time warping for which, in addition to a brief temporal context, the functionality, the advantages over other distance measures and possible optimization strategies are presented. In addition, a brief comparison of the performance between different Python implementations of dynamic time warping is provided.

2.1. Smartwatch Health Data for Stress Detection

The following section describes the role of wearable devices such as smartwatches in the healthcare sector. The use of such wearable devices opens up a wide range of different applications in the field of personal analysis or self-diagnosis [11], which are examined in more detail below. There are numerous smartwatches on the market today that serve the target group of private individuals. These smartwatches contain various sensors that can for example record movement or health data. The Empatica E4 wristband is a smartwatch that enables researchers to access sensor data for instance to develop health applications. One specific application for smartwatches is stress detection, which attempts to recognize whether a person is currently stressed and suggests appropriate recommendations for action. Various methods of stress detection are presented in more detail below as a well-researched exemplary application for smartwatches.

2.1.1. Using Wearables in Healthcare

The increasing use of wearables for the purpose of self-diagnosis and personal analytics was demonstrated by Piwek et al. [12]. Because of this, various scenarios for the use of wearables were initially analyzed for healthy individuals. Here, wearables are used by individuals who already live a healthy lifestyle but want to maintain or further improve it. Wearable manufacturers rely on various digital persuasive techniques and social influence strategies, such as gamification of activities, competitions and challenges with other wearable users or virtual rewards, to encourage users to continue using wearables [12]. Possible self-optimization goals in this context include improving sleep, dealing with stress, or increasing personal productivity [12]. The second step was to consider how patients with a defined illness or comorbidity can use wearables. For this purpose, wearables can serve as a second diagnostic tool that records detailed longitudinal data without great effort for patients [12].

Alongside the many positive effects of the use of wearables in healthcare, there are also some concerns. King and Sarrafzadeh [13] analyzed 27 studies that used smartwatches for applications in the field of healthcare for activity monitoring, chronic disease self-management, nursing or home-based care and healthcare education. All the studies examined worked with a very limited number of study subjects and no randomized clinical trial research was found. King and Sarrafzadeh [13] conclude that significantly larger populations of study subjects are required for widespread adoption of smartwatches in healthcare. Piwek et al. [12] also criticizes the absence of generalized regulatory frameworks for wearables that ensure quality standards. The authors see further concerns in the areas of privacy and security [12].

Well-researched application scenarios for smartwatches include sleep monitoring, fall detection and stress detection. In sleep monitoring, various sleep parameters such as total sleep time, sleep efficiency and wake after sleep onset are recorded by wearing a smartwatch or smart ring [14]. This creates the possibility of long-term recording of sleep in order to address poor sleep quality and better understand changes. Asgari Mehrabadi et al. [14] compared the tracking quality of a commercially available smartwatch and a smart ring with medical-grade actigraphy and found that the two wearables already generate acceptable mean differences and indicated significant correlations with the actigraphy for sleep tracking. Chang et al. [15] have presented SLEEPGUARD, a concrete application that uses a smartwatch to monitor sleep, determine sleep quality and provide information about sleep events. The application analyzes body movements, acoustic events in connection with sleep disorders and ambient lighting.

When older people fall, this often leads to serious injuries or, in the worst case, even death [16]. However, many of these falls remain unnoticed, especially in the home environment. Smartwatches can help to solve this problem by developing automated fall detection applications that process data from accelerometer sensors, for example. Mauldin et al. [17] and Kraft, Srinivasan, and Bieber [16] present various approaches to fall detection in their work.

Stress detection is discussed as the third application example in detail in section 2.1.3 by presenting various implementations and later in section 5.5 using a specific case study.

2.1.2. Recording Health Data with Empatica E4

A variety of smartwatches from different manufacturers can be found on the market today, but the usability of these smartwatches for research purposes is usually very limited. Only a few manufacturers allow researchers to gain direct access to the sensors of the smartwatch in order to use the signals from the sensors unprocessed and, if possible, in real time. One exception is the Empatica E4[1], which was developed specifically for the use in clinical trials, research studies and remote patient monitoring. The Empatica E4 is a further development of the Empatica E3 presented by Garbarino et al. [18] in 2014. The performance of the Empatica E4 in accurately measuring physiological stress indicators has been evaluated in several studies [19, 20, 21].

Empatica provides researchers with two modes for data acquisition. The first mode is the *recording mode* [18]. For this purpose, the Empatica E4 is equipped with an internal memory that allows up

[1] `https://www.empatica.com/research/e4/`

to 60 hours of data to be recorded. Once the data has been recorded, it can simply be transferred to the Empatica desktop platform via a USB connection, which then stores the data in the Empatica Cloud. Analysis can then be carried out via a dedicated dashboard, which has access to the Empatica Cloud. The second available mode is the *streaming mode* [18]. With this mode, the sensor data can be recorded and processed in real time using a Bluetooth connection. Empatica offers a desktop integration development tool and a mobile API for Android and iOS to process the real-time data. The Empatica E4 is equipped with numerous different sensors described below:

- *Electrodermal Activity (EDA):* The Empatica E4 is equipped with an EDA sensor, which measures the electrical conductance of the skin surface. The EDA measurements can be used as an indication of physiological arousal, as the sweat glands are controlled by the sympathetic nervous system [22]. High electrical conductance values can for example be used as an indicator for stress detection [10]. The Empatica E4 measures EDA using two electrodes through which a low alternating current is applied to the skin [18]. EDA measurements can vary considerably between different measurement locations [23]. However, in most cases, wrist measurements show strong correlation with traditional measurement locations [24, 25]. The Empatica E4 can measure conductance in the range of [0.01, 100] microsiemens with a standard sampling rate of 4 Hz [18].

- *Photoplethysmography (PPG):* The PPG sensor can be used to measure *Blood Volume Pulse (BVP)*, *Heart Rate (HR)* and *Heart Rate Variability (HRV)*. The explanations of the measurements and the sensor specifications are taken from the work of Garbarino et al. [18]. The PPG sensor illuminates the skin with green and red light and measures the reflected light. The human heart pumps blood into the periphery with every heartbeat. The heartbeat generates a pressure wave, which leads to a change in volume. This change in volume correlates with the change in the concentration of oxyhaemoglobin. Oxyhaemoglobin absorbs light at certain wavelengths, which the PPG sensor can detect by measuring the reflected light. BVP is measured by the Empatica E4 with a fixed sampling rate of 64 Hz. The PPG sensor is equipped with 4 LEDs (2 green, 2 red) and 2 photodiodes for this purpose. The firmware dynamically compensates for differences in skin color and external light intensity. HR and HRV can be calculated with the aid of BVP. Changes in HR and HRV can be processed by corresponding stress detection applications as an increasing HR and decreasing HRV can be indicators for stress [26, 27].

- *Skin Temperature (TEMP):* The Empatica E4 uses an optical infrared thermometer for temperature measurement, which monitors the infrared radiation emitted by the skin without contact [18]. It can measure both the skin temperature and the ambient temperature, which is essentially a low-pass version of the body temperature. The TEMP sensor can generate a voltage from the temperature differences of the sensor components, which is used to determine the skin temperature [28]. It should be noted, however, that skin temperature is not representative of core body temperature. The Empatica E4 measures TEMP with a sampling rate of 4 Hz and is calibrated for a temperature range of -40 to 115°C. The measurement accuracy is ±0.2°C [18]. The temperature measurements can be used for stress detection. During stress the sympathetic nervous system constricts the blood vessels, which can lead to a drop in skin temperature [29].

- *Accelerometer (ACC):* The ACC sensors measure motion based activity of a person in a 3-dimensional space. This results in the three signals ACC_X, ACC_Y and ACC_Z for the different axes X, Y and Z, which are all recorded with a sample frequency of 32Hz [18].

2.1.3. Stress Detection using Smartwatch Health Data

A wide range of health problems such as an increased risk of cardiovascular disease [30, 31, 32], a weakened immune system [30], a reduced mental performance [33], cancer [30, 31], depression [31, 34], diabetes [30, 35, 36] and substances addiction [37] can all be consequences of long-term stress. Stress is a physiological reaction of the body to stressful situations, such as sport competitions or exams. During stressful situations, the sympathetic nervous system is activated in order to cope with the changed metabolic requirements [38, 39]. As a result, the heart rate, blood pressure and sweat rate increase [39, 40]. In addition to situational stress, chronic stress occurs when the stress-related sympathetic response is constantly elevated. Preventive measures are particularly important in the case of chronic stress. In this respect, stress detection applications for wearables can be used very easily, as they are only minimally invasive for the individual, but at the same time provide long-term analyses and can suggest recommendations for individual stress situations.

The foundation of the first stress detection methods can be found in emotion recognition. The aim of emotion recognition is to automatically recognize the emotion of an individual at a certain point in time. Examples of emotions are happy, sad, neutral, and angry. Mirsamadi, Barsoum, and Zhang [41] presented various methods that use recurrent neural networks to automatically classify speech signals. Analyzing video content with deep neural networks, as described by Tzirakis et al. [42], can also be used for emotion recognition. Most stress detection methods work with physiological signals, as described in section 2.1.2. Santos Sierra et al. [43] have presented a method that uses a fuzzy decision algorithm for stress detection with HR and galvanic skin response data. Healey and Picard [44] also demonstrated various methods of stress detection using physiological sensors. To this end, they generated a data set that measured the relative stress level of test subjects in various driving situations in road traffic. The measured signal data was used to train a linear discriminant analysis (LDA) model for binary classification into stress and non-stress.

In 2018, Schmidt et al. [10] presented the WESAD data set, which contains the sensor data of an Empatica E4 covering 15 different subjects. The WESAD data set is presented in more detail in section 5.1.1 for the experiments in this thesis. Various classic machine learning models were trained with this physiological signal data. These include LDA, decision tree, random forest, AdaBoost and k-nearest neighbour. A 3-class classification using the classes baseline, amusement and stress, as well as a 2-class classification using the classes stress and non-stress were considered. For the 3-class stress detection, the AdaBoost classifier achieved the best result with an accuracy of 80.34%. An accuracy of 93.12% was achieved with the LDA classifier for the 2-class stress detection. Figure 2.1 illustrates the 3-class stress detection for two physiological sensor signals of an Empatica E4.

The WESAD data set was subsequently used by various other researchers to evaluate their own methods. Siirtola [45], for example, analyzed the performance of the three classifiers LDA, random forest and quadratic discriminant analysis for different combinations of sensors in the WESAD data set. Li and Liu [46] developed stress detection methods that use a 1-dimensional convolutional

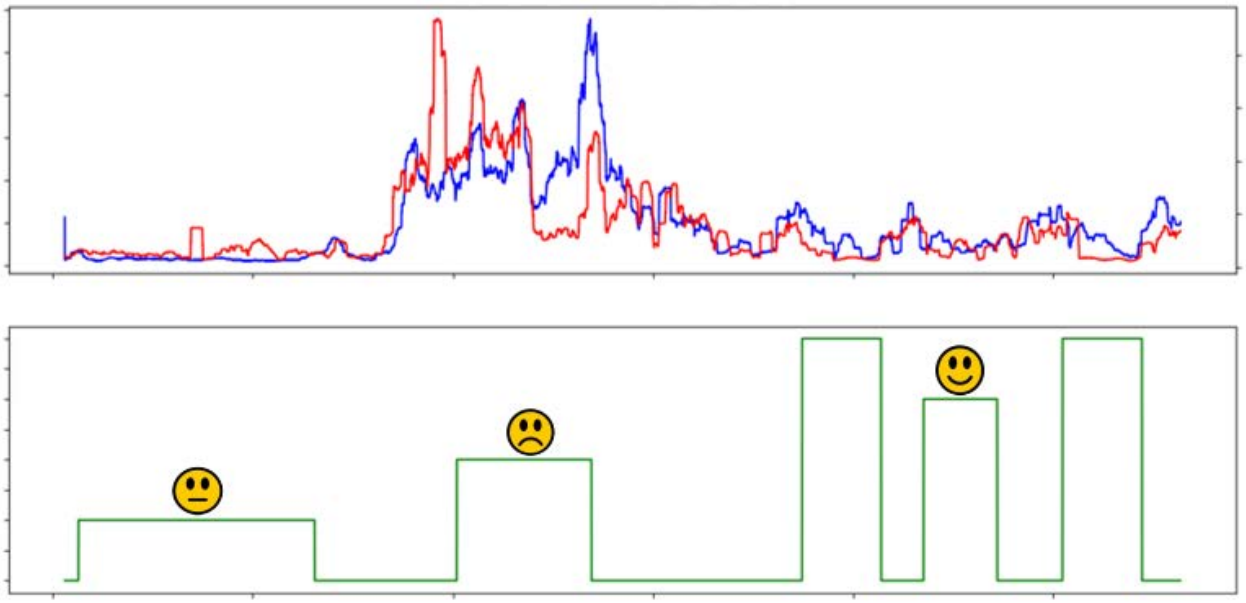


Figure 2.1.: Illustration of 3-class stress detection with multimodal sensor data from Schmidt et al. [10]. The upper half of the figure shows two measured health signals from an Empatica E4. The lower part of the figure illustrates the classification results of a stress detection application. Certain time periods of the signals are assigned to one of the three classes baseline (neutral smiley), stress (sad smiley) and amusement (happy smiley).

neural network (CNN) or a multilayer perceptron neural network (MLP). An accuracy of 99.55% was achieved with the CNN and an accuracy of 98.38% with the MLP for a 3-class classification. Gil-Martin et al. [47] have similarly presented stress detection methods that utilize CNNs. They also focussed on testing different signal processing steps. In addition to a 2-class and a 3-class classification, the authors also carried out a 5-class classification and compared all their results with those of existing literature. Souza et al. [48] and Eren and Navruz [49] were also able to achieve similar classification results with a recurrent neural network (RNN) and a feed forward deep learning artificial neural network using five hidden layers, respectively. Lange, Degenkolb, and Rahm [50] presented the first differentially-private stress detection approach considering a transformer architecture. The classification results for different privacy guarantees were analyzed.

2.2. Generating Synthetic Data Sets with Generative Adversarial Networks

In the field of machine learning, large training and test data sets are of great importance in order to achieve a good performance and generalizability of the models. However, especially in the area of health data, the creation of such data sets is often not possible or only possible with a disproportionate amount of effort. The reasons for this are, for example, a very limited number of people who suffer from specific diseases or have certain other sought-after characteristics. Ethical concerns and a lack of financial resources can also make it difficult to create large data sets. One of the main reasons, however, are data privacy concerns, as it must be ensured that the privacy of the persons in the data set is guaranteed and that these persons cannot be identified by possible

re-identification attacks. For this reason, Imtiaz et al. [51] used a generative adversarial network (GAN) to generate a medical health data set.

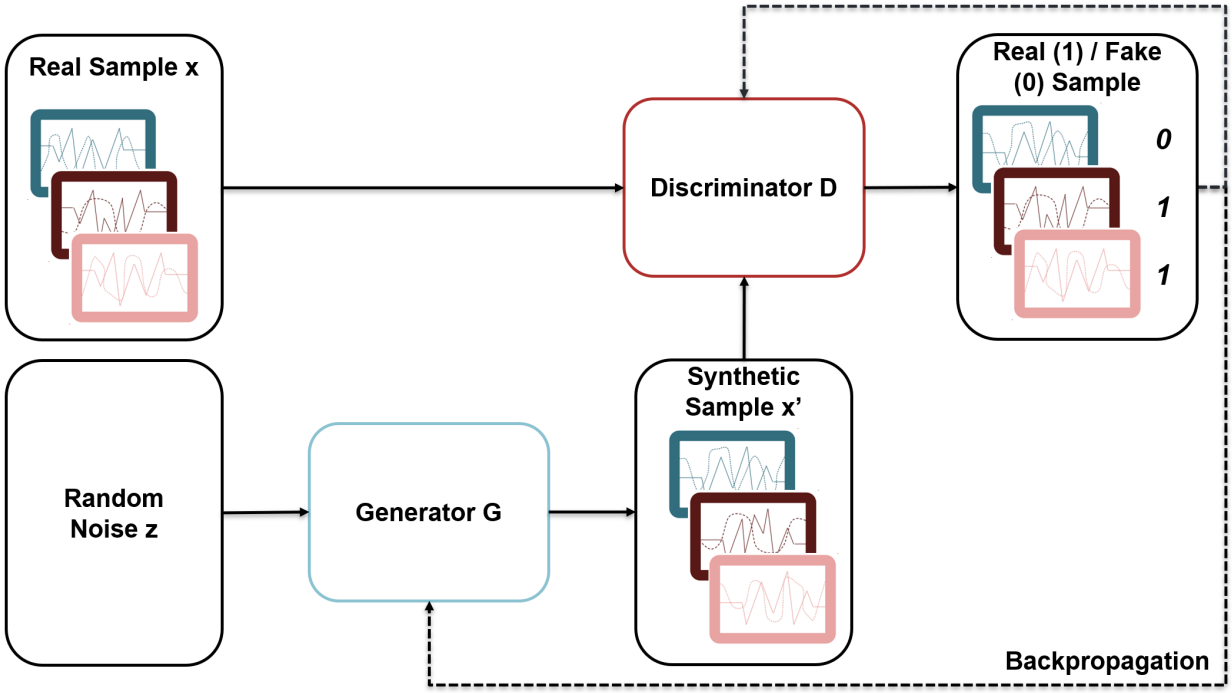


Figure 2.2.: General architecture of a generative adversarial network. The figure shows the two simultaneously trained neural networks, the generator G and the discriminator D. G uses a random noise z to generate a synthetic sample x'. D receives both x' and a real sample x in order to decide within a binary classification task whether the sample is real or synthetic. Only D has access to the ground truth in order to calculate error signals, which are passed on to G for learning during backpropagation.

The GAN was introduced by Goodfellow et al. [52, 53] in 2014 as a machine learning framework consisting of two neural networks trained simultaneously against each other. Figure 2.2 illustrates the basic architecture of a GAN. The two papers by Creswell et al. [54] and Imtiaz et al. [51] describe the functionality of a GAN as follows: The first neural network is the generator G, which aims to learn the data distribution in order to generate new data samples [51]. For this purpose, G receives a random noise z as input, with which it generates a synthetic data sample x'. Subsequently, x' is passed on to the second neural network, the discriminator D [51]. D is a binary classifier that attempts to recognize whether the data is a sample from the original data set x or a synthetic data set x'. G is trained to generate samples that are as realistic as possible without having direct access to the real samples. Learning takes place through the interaction of G with D during backpropagation. As D can determine the error signals of the classification by accessing the ground truth, these error signals can be forwarded to G for further training [54].

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2.1}$$

Imtiaz et al. [51] describe the two-player minmax game value function as shown in equation 2.1. G and D are two differentiable functions. G receives the random noise z as input, which allows G generating the data set $G(z)$ and simultaneously learn the distribution pg over the sample x with a

prior on the input noise sample $p_z(z)$. After a sample x' from $G(z)$ has been forwarded to D, which also receives the real sample x from the real data distribution $p_{data}(x)$, it checks the authenticity of the data. D now predicts probabilities $D(x)$ for the data, where 1 means that the sample originates from the real data set and 0 means that it is a synthetic sample. D can then be trained to maximize the probability that the classification is correct for real and synthetic samples. On the other hand, G is trained in such a way that $log(1 - D(G(z)))$ is minimized.

In addition to the original architecture by Goodfellow et al. [52], various papers can be found in literature that extend this architecture or present their own GAN architectures. These include, for example, the conditional GAN (cGAN) [55] and the DoppelGANger (DGAN) [56], both of which are used in section 5.1.2 to generate synthetic data sets. The implementation of the cGAN and DGAN presented by Wenzlitschke [11] is employed here.

2.3. Re-Identification and Identity Inference

When de-identification is used as an anonymization technique, individual entries in a data set are stripped from their identifying personal information. This might include (user-)names, locations, affiliations, or other relevant metadata. In this way they are protected from harm, when their possibly sensitive data is released to the public, or at least this is what should be achieved. However, there are many cases in which such privacy measures are broken and individuals are re-identified, especially in the context of health data [8]. In these cases, we see adversaries aiming to infer the identity behind a record from a data set or finding records related to a target individual, which opposes the concept of de-identification. The general goal of these attacks can be summarized under the term identity inference. In 2016, Henriksen-Bulmer and Jeary [57] published a systematic literature review on successful re-identification attacks. A total of 55 papers were selected for the final review. The authors found that 72.7% of the successful re-identification attacks took place from 2009 onwards. In addition, most attacks were tested on multiple data sets. The authors pointed out that new and improved anonymization techniques are needed for open source publishing. Various re-identification attacks, which are relevant for this work, are presented in chapter 3.*

Another type of attacks are membership inference attacks introduced by Shokri et al. [58]. These attacks only aim to resolve whether a target is present in the data set, while identity inference goes one step further and tries to recognize the actual samples in the data set that belong to a target.*

2.4. Dynamic Time Warping

DTW is a set of algorithms used to measure the similarity between temporal sequences based on alignment [59]. It aligns given time series samples by minimizing the difference between corresponding elements, accommodating temporal distortions. To achieve this, the technique considers local temporal dependencies and enables flexible matching by warping or stretching one sequence to fit the other as closely as possible. The constructed alignment matrix quantifies the distance between each pair of elements to determine the overall correspondence.

2.4.1. Historical Context of DTW

In literature, the first versions of DTW can be found in the context of automatic speech recognition. The aim of automatic speech recognition is to automatically capture spoken language and convert it into textual data. In this context, Vintsyuk [60] presented a method in 1968 that can recognize words in speech signals without first having to perform a time normalization between the unknown speech signal and the respective class standard. During time normalization, the two signals are first brought to the same length so that they can then be compared with common distance measures such as a Euclidean distance. Based on the calculated distance scores, the unknown signal can then be classified as the word with the shortest distance of the respective class standard. Vintsyuk [60] identifies the problem with this approach that the classification quality is largely dependent on the quality of the time normalization method. To circumvent this, Vintsyuk [60] proposes to classify the unknown signal by a complete search across all possible class standards by distinguishing word components using the dynamic programming method. The proposed algorithm for element-wise word recognition provides the best agreement between the elements of the class standard and the unknown signal by searching the shortest path in the resulting graph.

In 1970 and 1978, Velichko and Zagoruyko [61] and Sakoe and Chiba [62] independently presented the first concrete approaches of DTW for automatic speech recognition, which replaced classical methods of time normalization with dynamic programming-based methods. Velichko and Zagoruyko [61] carried out experiments to recognize 203 Russian words in spoken language. The work of Sakoe and Chiba [62] introduced the term "time-warping", which today gives its name to the concept of dynamic time warping.

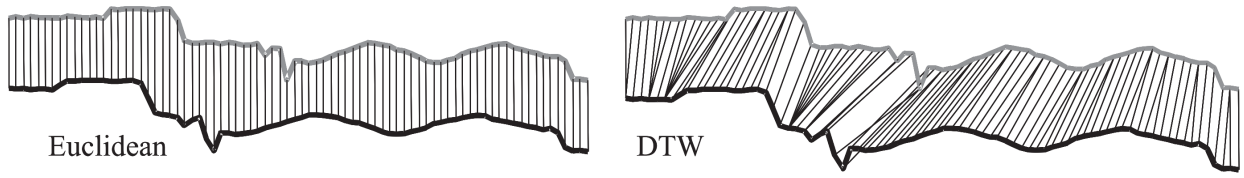


Figure 2.3.: Comparison of DTW and Euclidean distance from Keogh and Ratanamahatana [63]. The figure compares two time series by calculating their distance using either DTW or Euclidean distance. In both cases, the distance obtained is the sum of all distances between the matching features, but DTW matches the characteristic patterns of the time series, while the Euclidean distance matches the timestamps independently of the feature values [64]. The time series were shifted vertically for easier visualization.

Figure 2.3 illustrates the differences between Euclidean distance and DTW. With Euclidean distance, only the data points of both time series at the same index are compared with each other, whereas with DTW the best alignment between the time series is sought [63]. This has the advantage that the distance scores between similar time series do not become extremely large even with time series of different lengths, distortions, compressions or noisy time series.

2.4.2. Functionality and Optimization Strategies

DTW calculates a distance between two time series Q(q_1, q_2, q_3, q_n) and C(c_1, c_2, c_3, c_m), of length n and m [59, 63]. For this purpose, each index in Q is matched with one or more indices in C in a

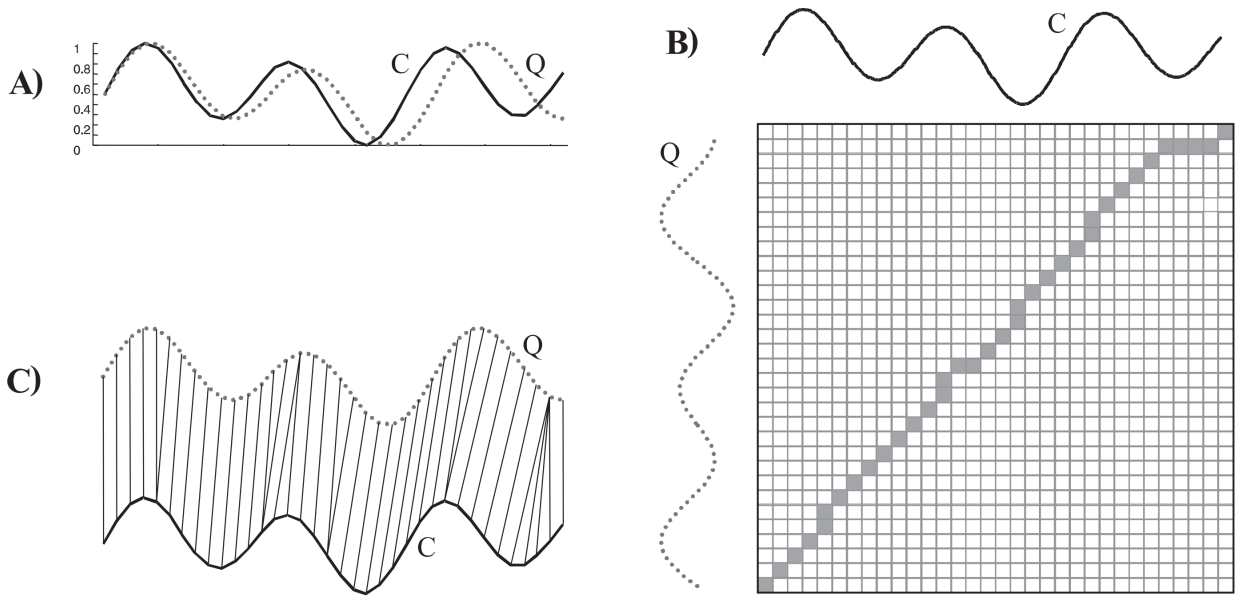


Figure 2.4.: Visualization of the functionality of DTW from Keogh and Ratanamahatana [63]. Subfigure A) illustrates the two time series Q and C, which are similar but out of phase. To calculate DTW, a warping matrix is now created in B) to search for the optimum warping path. This warping path is characterized by filled squares. Subfigure C) shows the resulting alignment between Q and C.

way that minimizes the difference between the two time series [65]. Specifically, this implies that an n-by-m matrix is constructed which contains the distances $d(q_i, c_j)$ between the data points q_i and c_j [63]. Figure 2.4 illustrates this graphically. For the two time series Q and C, a matrix is first constructed with the distance scores for all data points. Subsequently, the best alignment of the two time series is then sought using a warping path for the optimization problem (shown in Figure 2.4 in section B by the diagonal highlighted in gray), which minimizes the distance of Q and C [63]. Section C shows this alignment graphically once more by displaying alignment paths between the corresponding data points. Various distance measures $d(q_i, c_j)$ can be selected. A common choice is the Euclidean distance [59]. Formally, DTW can be represented as a minimization problem with the equation 2.2 [59, 63, 64]. A distance between the time series Q and C is determined by minimizing the warping path ϕ of an alignment over the sum of all distance scores of ϕ.

$$DTW(Q, C) = \min_{\phi}\left(\sum_{i,j \in \phi} d(q_i, c_j)\right) \tag{2.2}$$

In the basic version of DTW, n x m distances must be calculated to construct the distance matrix, resulting in a time and space complexity of O(nm) [63]. In order to avoid this, various optimization strategies can be found in literature. For example, a global constraint can be used to limit the matrix calculations [63, 66]. Common global constraints are, for example, the *Sakoe-Chiba band* [62] or the *Itakura parallelogram* [67]. In these cases, the area in which the warping path may lie along the optimal diagonal is limited by a simple band or a parallelogram [63]. Distances are therefore only calculated for elements of the matrix within the constraint. Rakthanmanon et al. [66] propose a total of 9 further optimization strategies in their work, 5 of which originate from previous literature and 4 of which were first presented in their paper. By implementing the optimization strategies

suggested, the amortized costs of DTW with an average complexity of less than O(n) are comparable to those of a Euclidean distance [66].

2.4.3. Comparison of Implementations

There are many different Python implementations for DTW, which offer a different range of functions on the one hand and require different runtimes on the other. Well-known implementations include DTW-Python, FastDTW, TSLearn, PyTS and DTAIDistance [68]. Runtime experiments for time series of lengths of 1000 and 2000 with 20 runs have shown that the distance_fast implementation of DTAIDistance with 0.0097s can be run in a significantly reduced runtime than the second-placed package TSLearn with 0.0164s [68]. In later experiments, the Python package DTAIDistance[2] with the distance_fast method presented by Meert et al. [9] will be used due to the good runtime and the large scope of functions.

[2] `https://dtaidistance.readthedocs.io/en/latest/`

3. Related Work

Following the background chapter presenting fundamental concepts for this thesis, in this chapter existing literature on already published re-identification attacks will be presented. Due to the research field of re-identification attacks on smartwatch data, which has so far received little attention, publications covering re-identification attacks in different neighboring research domains will be shown first. Therefore, literature on re-identification attacks on health data will be examined in section 3.1. The focus will then be shifted in section 3.2 to literature on similarity-based re-identification attacks in various research domains. Finally, section 3.3 presents methods for the specific area of similarity-based re-identification attacks on health data.

3.1. Re-Identification Attacks on Health Data

In 2016, El Emam et al. [8] found that many countries' data privacy laws allow the sharing of health data for secondary purposes without the consent of the patient concerned, under the premise that the data is de-identified. However, it was found that de-identification methods do not provide sufficient protection due to their ease of reversibility. This has significant implications for the way in which health data is disclosed. These include a potential reduction in the availability of health data for secondary purposes such as research and an increase in the disclosure of identifiable health information. El Emam et al. [8] conducted a meta-study to systematically review successful re-identification attacks on data sets that have undergone transformations to conceal the true identity of individuals. Fourteen relevant studies were identified that met the authors' inclusion criteria, six of which concerned health data. Two of the most successful attacks in terms of the proportion of re-identified records are briefly discussed below.

The first of these two re-identification attacks, which was presented by Brownstein, Cassa, and Mandl [69], concerns the publication of health maps. These mappings are needed, to determine the risk of diseases such as yellow fever and cholera, as the risk of disease is strongly influenced by location. For this purpose, the addresses of patients are mapped in order to publish patterns, correlates and predictors of diseases. The maps generated are then usually made available to the public. The authors found various published articles in which more than 19,000 patient addresses were plotted as individual data points on maps. The authors then tested whether these maps could be used to re-identify patients. For this purpose, a simulated map with 550 geographically coded addresses in Boston was created. The authors succeeded in accurately identifying 432 of the 550 addresses using this method, which corresponds to 79%.

The second re-identification attack was publicized by a court case between a daily newspaper in Illinois and the Illinois Department of Public Health [70]. The daily newspaper demanded the release of data from the Illinois Health and Hazardous Substances Registry (cancer registry). The data should include the type of cancer, the postcode and the date of diagnosis. The department commissioned an expert who succeeded in re-identifying 18 of the 20 individuals in the data set in a multi-stage process. To do this, she used several publicly available data sets and analyzed them for common factors. The results were verified by the Illinois Department of Public Health.

3.2. Similarity-Based Re-Identification Attacks

Several works propose similarity-based attacks based on encoded data. The methods utilize the preservation of similarity in the original data space and the encoded data space. Following, various similarity-based re-identification attacks are presented in the three domains of authentication, privacy-preserving record linkage (PPRL) and acceleration.*

Authentication. In the domain of authentication, biometric images such as fingerprints are used as keys to log into systems or applications. The original images are encoded to templates using Bloom filter, neural networks, etc. [71]. The resulting templates are utilized in the authorization process to determine if a user is privileged to get access to the system. Due to the preservation of the similarity between original images and templates, similarity-based attacks aim to construct an image where the encoded template is similar to the target template. Therefore, similarity-based attack methods [72, 73] compare a fake template with a target template and iteratively optimize the construction process to obtain a new image being used to generate a new template.*

Dong, Jin, and Jin [72] propose a genetic algorithm enabled similarity-based attack framework that uses cancellable biometrics schemes to show that these schemes have the property of preserving similarity and are therefore very vulnerable to similarity-based attacks. Cancellable biometrics are means to ensure the security and privacy of a biometric template. For this purpose, an irreversible but similarity-preserving transformation of the original template is used. The deployment of a genetic algorithm requires no training data, and the execution of the attack is very time-effective.

In their work, Yang et al. [73] developed two attacks to reconstruct palm prints. Nowadays, palm prints are used for personal authentication in various applications, which is why the security of such systems is of great importance. The authors have demonstrated two cross-database online palm print reconstruction attacks with style-transfer, both of which are based on a CNN. In the first method, the optimization object is the input image that can be reconstructed from the binary template image. The second method trains the CNN for style transfer with a template data set and only one style image to reduce style loss between the source and target domains. Both methods show a high success rate, making defense strategies necessary.

Privacy-Preserving Record Linkage. Similarity-based attacks also exist in the context of PPRL. PPRL aims to identify duplicate records between two or more databases containing sensitive information. Therefore, the data owners encode the plain text data to encodings that are compared to determine duplicates [74]. Due to the preservation of similarities, the proposed attacks [75, 76] construct a graph consisting of records as nodes and similarities as edges using a publicly available plaintext database and the encoded one. The attack utilizes the similarity graphs to determine a mapping between nodes representing encoded and plaintext records based on similar graph features such as indegree/outdegree, PageRank, etc.*

The graph matching attack on PPRL presented by Vidanage et al. [75] can be applied to any PPRL method that calculates similarities between encoded values. The authors have successfully tested their attack on various common encoding methods. These include Bloom filters, tabulation min-hashing and two-step hashing. Due to the good re-identification performance, the authors conclude that data privacy limitations in PPRL applications need to be investigated in more detail.

Culnane, Rubinstein, and Teague [76] reviewed the PPRL approach followed by the UK Office for National Statistics and found several problems that pose a serious risk to privacy. On the one hand, the authors found that incorrect cryptographic assumptions were made in combination with incorrect information on the necessary entropy for hash-based message authentication code (HMAC) keys, which are used to encrypt names. On the other hand, the provision of similarity tables with HMAC names poses a risk of frequency attacks. The biggest risk identified by the authors, however, is that using a similarity graph to attack plaintext similarity scores provides an index of HMAC-encrypted names that allow the recovery of names in plaintext.

Acceleration. Due to the increasing relevance of sensors in manufacturing processes, mobility and life sciences, a tremendous amount of sensor data is collected and analyzed. As described in section 2.1.2, mobile devices such as smartphones and smartwatches in particular are equipped with various types of sensors for different helpful use cases. However, the collected data also bears the risk of endangering the privacy of users. For instance, accelerometer data can be used to predict the location of metro riders [77].*

For this purpose, Hua, Shen, and Zhong [77] propose two attacks using accelerometers based on a side-channel attack, which allows an attacker to gain access to the accelerometers of a smartphone or smartwatch. The authors created their own data set, which records more than 120 hours of data from six underground railway lines in three major cities. The aim of the two attacks is to secretly track underground riders. The first attack, an ensemble interval classifier based on supervised learning, requires collecting training data for each station interval. For this reason, the authors propose a second attack based on semi-supervised learning, which only requires data for a very small number of labelled station intervals. The real-world experiments showed that the recognition accuracy reached 94% when a user travelled on the metro for 6 stations.

3.3. Similarity-Based Re-Identification Attacks on Health Data

Recent work from Saleheen et al. [78] proposed a re-identification attack called *WristPrint* using accelerometry data from a total of 353 participants being recorded for 190,078 hours (70 days with at least 8 hours per day) resulting in 51.3 billion data points. The attack aims to determine the trace from an anonymized database regarding an available trace where the user is known. The attack computes similarities between the anonymized and known time series. Therefore, the traces are split into smaller segments to build meaningful features using a neural network. The network consists of convolutional layers and gated recurrent units to address the time aspect. Moreover, the base model classifies resulting features if the segment from the known user corresponds to the anonymized one. The authors suggest various aggregation strategies to determine the similarity between traces based on the segment similarities. Through various experiments, the authors were able to show that their attack model can achieve a re-identification accuracy of 0.96 when 20% of each participant's data is used in the test set. The re-identification risk, which occurs when the data of various individual activities is accessible to the attacker, was also considered. The activities examined include stationary, walking, sports and exercise. By far, the highest re-identification risk was found for the sports activity followed by the exercise activity, as shown in figure 3.1.

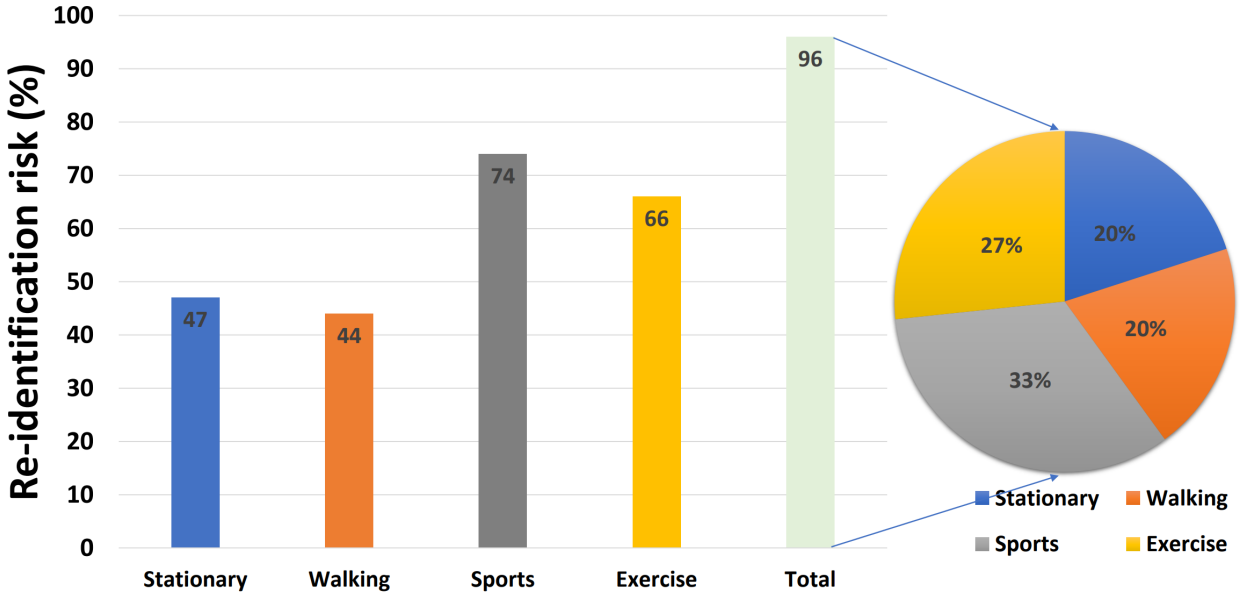


Figure 3.1.: Re-identification risk of the WristPrint attack model for different activities from Saleheen et al. [78]. The figure shows the accuracy that the attack model can achieve for the four activities stationary, walking, sports and exercise if a user shares one day of wrist-worn accelerometry data publicly or the attacker has access to this data. The total column shows the re-identification risk if no distinction was made between the activities in the data.

The following approaches of this work do not use a supervised feature extraction and classification model, since the performance depends on readily available training data. All methods proposed in section 4.5 can be used for each available individual, because the data is not split into training and test data sets. Moreover, various sensor data types will be considered and thus the work will not only focus on accelerometry data. In contrast to the evaluation in chapter 6, where a rank-based evaluation pipeline is used, the work of Saleheen et al. [78] only considers the true matching rate and the false acceptance rate, which does not allow a more differentiated view for the attacker. A rank-based evaluation makes it possible to additionally consider whether an individual being searched for is for example among the top 3 ranked ones. This strong reduction in the number of individuals also creates a risk for the privacy of the individuals in the data set. Since different activities carried out by an individual have a significant influence on the risk of re-identification, this work examines whether this can also be determined similarly for the stress level of an individual.

4. Methodology

In this chapter, the concepts presented in chapter 2 are now used to develop a total of four different re-identification attacks for smartwatch health data. For this purpose, the existing literature presented in chapter 3 is also used to adopt and further develop suitable concepts. Section 4.1 first describes the general attack scenario. The various actors involved and the different ways in which they can obtain the required data records are shown. In section 4.2 an overview of the entire attack framework is presented, the sub-aspects of which are analyzed in more detail in the following sections. Thereafter, section 4.3 presents the various signal preprocessing steps required to carry out the re-identification attacks. Section 4.4 then outlines three methods of complexity reduction to enable the attacks to be carried out more efficiently. Finally, section 4.5 introduces the four different DTW-based similarity attacks. These include the Single-DTW-Attack, the Multi-DTW-Attack, the Slicing-DTW-Attack and the Multi-Slicing-DTW-Attack.

4.1. Attack Scenario

The DTW re-identification attacks aim to identify possible threats based on a small sample of information-rich sensor data of a target person. This sample is compared with other but similar samples stored in a database on the basis of their similarity in order to identify the target in the data set. The attack scenario is illustrated in figure 4.1 and comprises the three different actors - device owner, data owner and attacker - which are explained below:

- *Device Owner:* The device owner is usually a private individual who wears a smartwatch and uses its sensors to record various health data such as BVP, EDA, ACC and TEMP over a longer period of time. The data generated is usually not stored on the smartwatch itself, but transferred to various fitness apps, which then store the data in aggregated form on the user's smartphone or in a cloud storage.
- *Data Owner:* The data owner here is different from the smartwatch or device owner and could be a company, institution, or person. The data owner wants to use this user data for improving their product or providing smart health features like stress detection through training machine learning models. To ensure privacy for device owners, their incoming data is anonymized by removing any identifying information like name or location.*
- *Attacker:* The attacker in this scenario has access to a small sample of data from the device owner's smartwatch. On the one hand, this data may also be contained in the device owner's data set or, on the other hand, it may have been generated at a completely different time. The attacker can either be an insider on the part of the data owner or a user who wants to find out information about the device owner for personal reasons. A third possibility would be that the data owner himself is the attacker in order to recover lost information from the given data privacy promise.

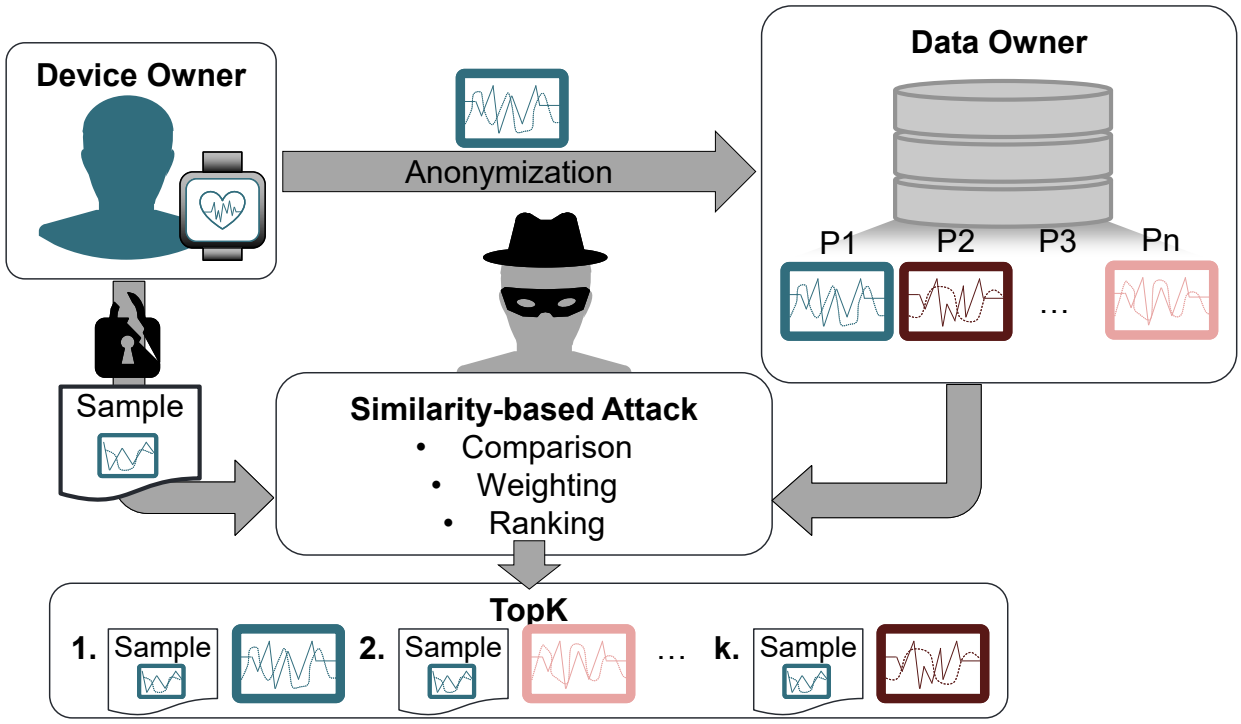


Figure 4.1.: The attack scenario consists of a device owner (target), a data owner and an attacker. The device owner sent the collected data via his device to the data owner, who anonymizes and stores the data for analysis capabilities. The attacker aims to determine the corresponding time series data maintained by the data owner utilizing a sample from a known device owner.*

This involves the device owner generating health data with a smartwatch. This health data is forwarded to the data owner in anonymized form. All directly identifying information is removed for this purpose. However, the data itself is passed on unchanged so as not to reduce the usability of the data for the purposes of the data owner. The DTW attacks now enable an attacker to find out the original identity of such data samples. As a result, any promised data privacy guarantees are negated. The only prerequisite is that the attacker has access to a short data sample (attack data) of the device owner, which was recorded at any time. With the two data sources, the attacker can carry out a similarity-based attack by calculating the similarity of the attack data to the anonymized data in the database. The attacker can now use the similarity scores to perform a ranking, whereby the person in the database with the highest similarity or lowest distance to the attack data is most likely to be the device owner.

By successfully performing one of the similarity attacks on the time series signals recorded from the target's device, the attacker can correlate the data samples back to them. One objective could be to collect more data stemming from the target. In any case, user privacy would be broken irreparably, making the anonymization useless in terms of real provided security. In the described scenario, it does not matter if user data is stored or only processed before deletion, since the attack can also be executed directly on any arriving data sample.*

There are various conceivable scenarios in which the attacker could gain access to the device owner's attack data sample. These include publication by the device owner themselves, data leak by the smartwatch manufacturer and security vulnerabilities allowing attacks.

Publication by the device owner themselves. In the most trivial scenario, the device owner publishes the attack data themselves. For example, as described in the work by Saleheen et al. [78], this could be done through a share function via a fitness app to which the smartwatch is connected. In this case, the device owner wears a smartwatch to record various activities, such as running, cycling or swimming. The smartwatch synchronizes the recorded data from the smartwatch with a smartphone fitness app. The device owner can now use the fitness app's share function to share their data with friends or all users of the fitness app. In this scenario, the attacker only has to retrieve the published data.

Data leak by the smartwatch manufacturer. Many smartwatch manufacturers offer cloud services for smartwatch owners to store smartwatch data securely and access it conveniently from multiple devices. However, the security of these cloud services depends heavily on the manufacturer and is usually not transparent to the user. Smartwatch health data that is synchronized with a cloud can be exposed to various risks, such as DDoS attacks, SQL injections or back door attacks [79]. However, these attacks are usually carried out by highly skilled cyber criminals. An insecure authentication management system, where a large number of people have access to user data, can lead to data being accessed unnoticed. Smartwatch health data can also be circulated due to a possible data leak.

Security vulnerabilities allowing attacks. Possible security vulnerabilities in the software or hardware of the smartwatch or connected fitness apps can enable attackers to carry out various attacks to gain access to an attack data sample. A few examples are outlined in the following. A major problem is the lack of physical security controls offered by many smartwatches, such as secure user authentication mechanisms, PIN systems and data encryption [79, 80, 81]. If the attacker succeeds in stealing or replacing the device owner's smartwatch, they can access the data without much effort. Many smartwatches use Bluetooth to transfer sensor data to a smartphone, as they cannot communicate directly with the internet. An attacker could carry out a man-in-the-middle attack by using a sniffer to steal unauthorized data. To do this, the attacker tries to detect transmission signals while the smartwatch is communicating with the smartphone. [79, 81, 82]. Malware and phishing methods for smartwatches and smartphones can also be used by the attacker to acquire the required attack data sample.

4.2. Overview of the Attack Framework

A novel attack framework with eight stages was developed for the DTW re-identification attacks, which is illustrated in figure 4.2. The various stages are summarized below for a brief overview. In the following sections, the corresponding stages are presented in more detail with reference to the respective number. For stages (1), (3), (5) and (6), additional data sets, methods and attacks can be integrated into the framework in further research work on the corresponding interfaces.

A user is initially able to integrate and select different data sets (1), which are processed consistently in a signal preprocessing pipeline (2). In addition, various complexity reduction methods can be applied in advance to reduce the runtime of the attacks (3). These methods include downsampling, which attempts to represent the data points of the signals with fewer data points. Independently of

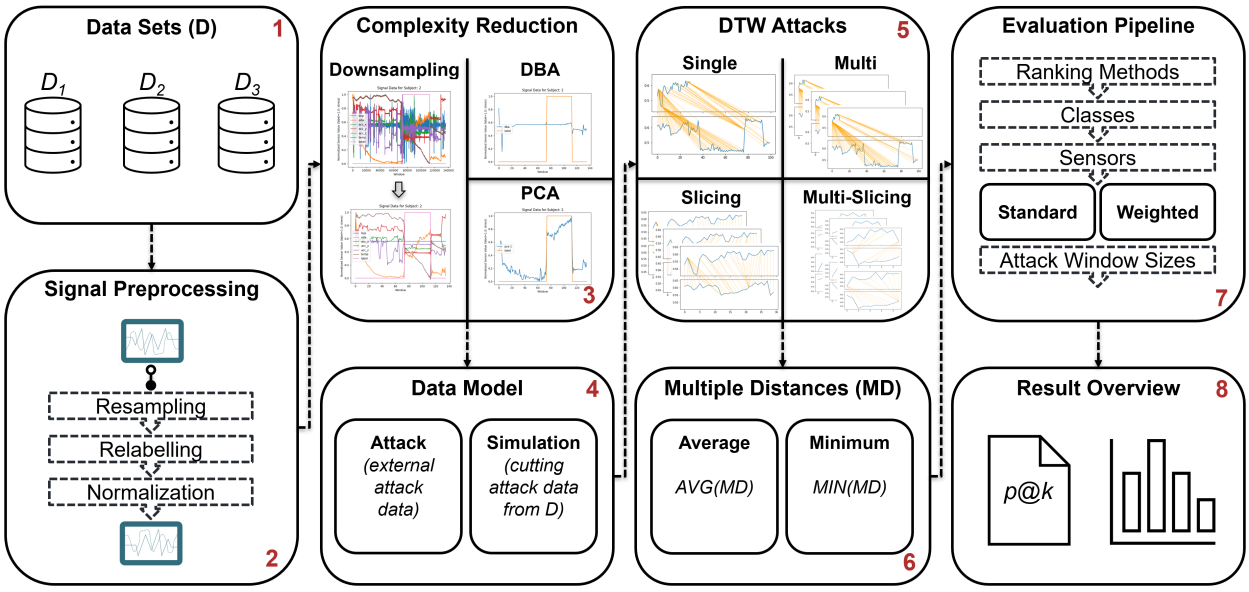


Figure 4.2.: Overview of the attack framework. 1) A user selects a data set D on which the re-identification attack is to be carried out. 2) The data set is passed through a signal preprocessing pipeline to ensure data consistency. 3) Various complexity reduction methods can optionally be applied to reduce the runtime of the attacks. 4) The data model takes care of splitting the data, whereby the attack data can come from an external source or be cut out of D. 5) Four different DTW attacks are available to the user, which calculate distance scores between the attack data and D. 6) For attacks which generate multiple distances, one of the two methods for handling multiple distances must be selected. 7) The distance scores of the attacks are evaluated using a standardized four-stage rank-based evaluation pipeline. 8) A result overview with the various results is available to the user.

downsampling, dynamic time warping barycenter averaging (DBA) or a principal component analysis (PCA) can also be used to represent several sensor signals with a single signal. Subsequently, one of the four attacks Single-DTW-Attack, Multi-DTW-Attack, Slicing-DTW-Attack or Multi-Slicing-DTW-Attack can be selected to calculate distance scores between the attack data and the data set (5). For the Multi-DTW-Attack, the Slicing-DTW-Attack and the Multi-Slicing-DTW-Attack, the resulting multiple distances must be reduced to one distance for further processing (6). An average or minimum method is available for this purpose. In stage (7), the distances are evaluated rank-based using a four-stage evaluation pipeline. This involves first determining the optimum ranking method, then considering the different stress and non-stress classes, evaluating the various sensors using sensor combinations (standard) or weighted sensor values (weighted) and finally determining the effect of different attack set sizes. This evaluation can now be used to create a result overview with the different precision@k scores and various result visualizations (8).

4.3. Signal Preprocessing

Various signal preprocessing steps are necessary in order to be able to further process the raw signal data of the Empatica E4 for the re-identification attacks (see stage (2) of the attack framework). In this work, the signal preprocessing of Gil-Martin et al. [47] was largely adopted, which consists of

the three steps signal resampling, relabelling and normalization. As described in section 2.1.2, the sensors of the Empatica E4 record signals at different sampling rates. For this reason, the signals are resampled to a standardized sampling rate to ensure that a data point is available for each signal at every point in time. BVP at 64 Hz is selected as the standardized sampling rate. All other signals are upsampled to this standardized sampling rate by applying a fast Fourier transform. For this purpose, the resample method from the Python package SciPy is used.[3]

In the second step, the labels of the signal data are adjusted. These are initially available in the WESAD data set, which is described in more detail in section 5.1.1, in the five characteristics *baseline*, *amusement*, *stress*, *meditation* and *recovery*. Initially, all data points labelled *meditation* and *recovery* are removed from the signal. This is possible because the signals for all subjects were recorded sequentially in different phases of a laboratory session. In order to be able to use the methods of Wenzlitschke [11] to generate synthetic data sets with GANs in section 5.1.2 without prior adaptation and to be able to perform a binary stress detection task in section 5.5.2, the remaining three classes are converted into the two classes *non-stress* and *stress*. For this purpose, the labels *baseline* and *amusement* are combined to *non-stress*. The result of the relabelling leads to an average of 70% of the data points of each subject being assigned the label *non-stress* and 30% of the data points being assigned the label *stress*.

Third, a min-max normalization is now performed to remove the scaling difference between the signals. The result of the normalization are signals that lie in the value range of [0, 1], whereby the relationships within the data are still mapped. The uniform range of values shortens the training time for the GAN models, as the GAN converges faster during training, which reduces the time required to learn the optimal weights [11]. DTW alignments for the similarity-based re-identification attacks can also be calculated much more efficiently for smaller values. In addition, the comparability of the signals of different subjects in the rank-based evaluation is increased.

The three signal preprocessing steps are a basic requirement for all integrated data sets. When importing each data set, the preprocessing must be adapted to the conditions of the data set, such as different sampling rates or different labels. With the now standardized data sets, further signal processing steps for complexity reduction can then be carried out in the next section.

4.4. Complexity Reduction

Despite many possibilities to increase efficiency for the calculation of DTW alignments, as described in section 2.4.2 and section 2.4.3, high hardware requirements and a long computation time are necessary to perform the DTW attacks for very large data sets. For this reason, various methods for complexity reduction have been integrated into the attack framework, as shown in stage (3) of figure 4.2. The three methods downsampling, DBA and PCA are discussed in detail below. In addition to an individual application, downsampling and DBA or downsampling and PCA can also be applied to the signal data in combination.

[3] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample.html`

4.4.1. Downsampling

A first possibility to increase the runtime efficiency of DTW is to reduce the high sampling rate of 64 Hz after signal preprocessing. For this purpose, the same resample method based on a fast Fourier transform from the Python package SciPy is used, as described in section 4.3. However, in this case, the signals are downsampled. For this purpose, the user can specify a downsampling factor (DSF), which determines how strongly the signal is to be downsampled. The following rule applies: $length(signal)/DSF$. If a user specifies a DSF = 1 for a signal with 100,000 data points (hereinafter referred to as windows), the signal is not downsampled. With a DSF = 10, however, the previously 100,000 windows are represented by 10,000 windows. In other words, 10 windows of the original signal are represented by 1 single value in the downsampled signal.

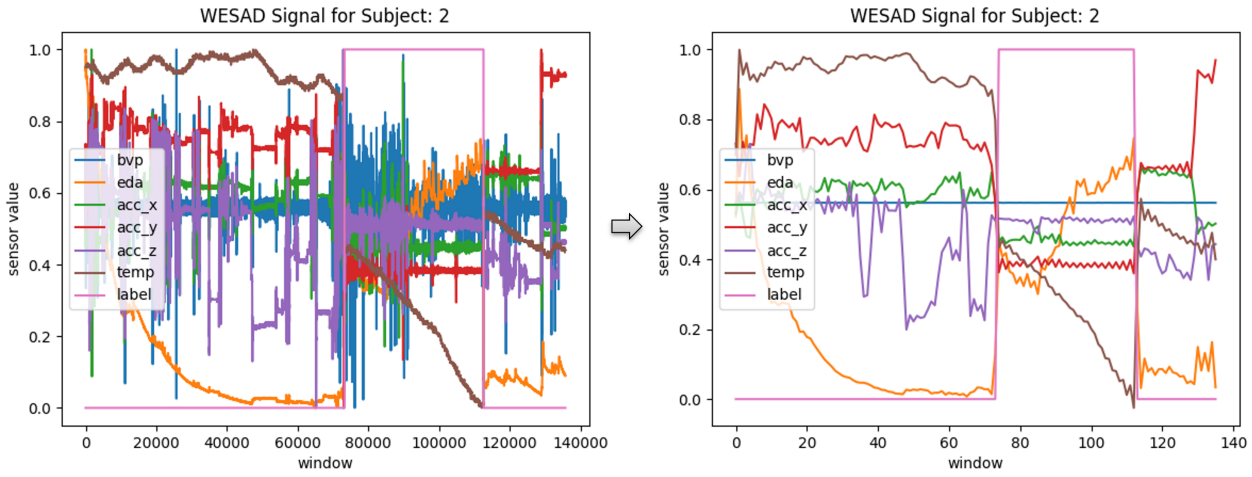


Figure 4.3.: Illustration of downsampling. The left plot shows the time series data for six signals of subject 2 from the WESAD data set after signal preprocessing. The x-axis shows the different windows of the time series, where one window corresponds to one data point. The normalized values of the signals are shown on the y-axis. The two labels stress and non-stress are also shown, whereby the subject is stressed when the label assumes the value 1.0. The plot on the right shows the signals downsampled by a factor of 1000. The signal therefore only consists of 136 windows, instead of the previous 136,000.

The result of the downsampling is shown in Figure 4.3. The left-hand plot shows the data for a subject of the data set after signal preprocessing. The six different signals BVP, EDA, ACC_X, ACC_Y, ACC_Z and TEMP are available for the subject. The label (stress = 1 or non-stress = 0) is also displayed in the plot. The signal consists of approx. 136,000 windows. If a downsampling with a DSF = 1000 is applied, 6 signals with only 136 windows are obtained on the right-hand plot. If DTW alignments are then calculated between signals of length 136,000 or signals of length 136, the calculation between the signals with fewer windows is significantly more time-efficient. In addition to the improvement in runtime efficiency, downsampling has a second effect, which can also be seen in the plot on the right. A signal smoothing effect occurs, which eliminates very short and strong signal fluctuations. These fluctuations can be caused, for example, by measurement inaccuracies of the sensors or by recorded noise. Downsampling reduces these effects, which is reflected in smaller DTW distance scores and can therefore have a positive influence on re-identification performance.

4.4.2. Dynamic Time Warping Barycenter Averaging

In addition to the downsampling method, the runtime performance of DTW attacks can also be improved by reducing the number of sensor signals that are taken into account. This can be achieved by averaging methods, which form an average over n sequences. A common averaging method is *dynamic time warping barycenter averaging* (DBA) [83, 84, 85]. DBA aims to address the problems of the two previous averaging methods *nonlinear alignment and averaging filters* (NLAAF) and *prioritized shape averaging* (PSA). NLAFF [86, 87] systematically combines non-linear alignment operations such as DTW and averaging operations for signal data. The application of NLAFF leads to the problem of a significant increase in the length of the average sequence. Over the entire NLAAF process, an average frequency of length $n \times l$ could thus be generated, where n stands for the number of sequences to be averaged and l reflects the length of the sequences [83]. PSA [88], on the other hand, is based on hierarchical clustering to dynamically create an averaging sequence. Due to the implementation, in which coordinates can be repeated, PSA also has the problem that the number of coordinates of each average sequence can double [83].

DBA is a heuristic strategy that consists of iteratively refining an initial arbitrary average sequence to minimize the sum of squared DTW distances from the average sequence to the set of sequences [83]. Here, the sum of the respective distances between each coordinate of the average sequence and the coordinates of the sequences connected to it is computed. This results in the fact that during the DTW computation, the contribution of a coordinate of the average sequence to the total sum of squared distances is equal to the sum of Euclidean distances [83]. The way DBA works is that each coordinate of the average sequence is calculated as the barycenter of its associated coordinates of the sequence set [83]. This ensures that each coordinate minimizes its share of the within group sum of squares, which also minimizes the total within group sum of squares. Once all barycenters are calculated, the updated average sequence is defined [83].

Petitjean, Ketterlin, and Gançarski [83] describe the procedure of DBA in two steps, which are performed for each refinement iteration:

- DTW is computed between each sequence and the temporal average sequence to be refined to determine the associations between the coordinates of the average sequence and the coordinates of the sequence set.
- Each coordinate of the average sequence is updated as a barycenter of the coordinates assigned to it in step 1.

The Python package DTAIDistance, which is used to calculate DTW alignments, also offers methods for calculating DBA.[4] For this, only the set of sequences and an initial average sequence must be provided. Since no informed starting point is available for the sequences, the BVP signal is used as the initial average frequency in the following experiments. The advantage of the BVP signal is that after downsampling, a straight line with a single value is usually produced, which generally lies between the curves of the other signals. This effect can be seen in figure 4.3. During the experiments, the best precision@1 scores for the DTW attacks were also achieved using BVP as the initial average sequence.

[4] `https://dtaidistance.readthedocs.io/en/latest/modules/dtw_barycenter.html`

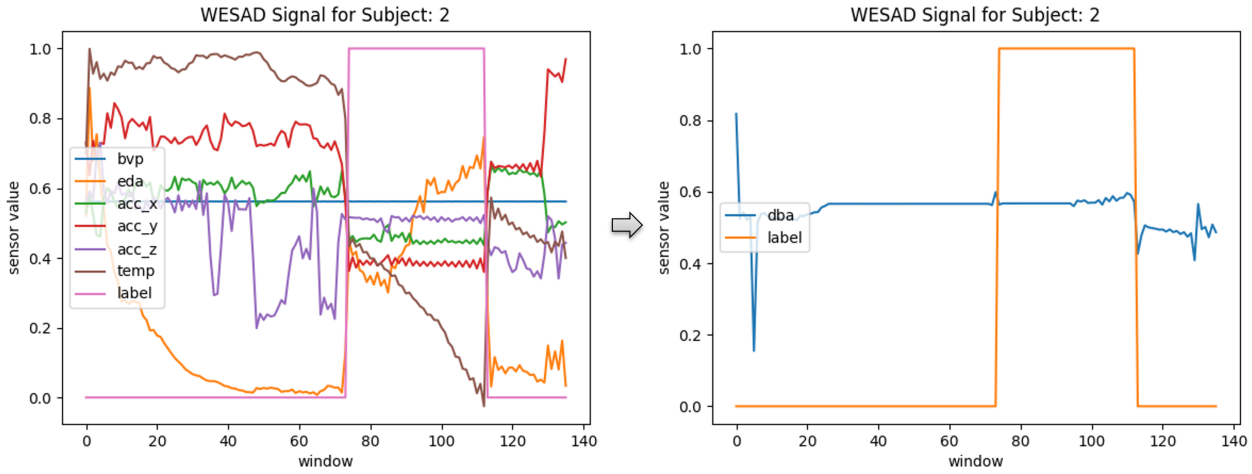


Figure 4.4.: Illustration of dynamic time warping barycenter averaging. The left plot shows the downsampled signals from figure 4.3. The x-axis shows the different windows of the time series, where one window corresponds to one data point. The normalized values of the signals are shown on the y-axis. The two labels stress and non-stress are also shown, whereby the subject is stressed when the label assumes the value 1.0. The six signals are now reduced to one single signal (dba) using the DBA method.

Figure 4.4 shows the effects of DBA for the set of frequencies BVP, EDA, ACC_X, ACC_Y, ACC_Z and TEMP with the initial average sequence BVP. It can be seen in the right-hand plot that the six signals are averaged into a single signal. By choosing BVP as the initial average, the BVP straight line can still be recognized, whereby strong deviations of the other signals are usually recognizable by smaller peaks. An attacker can now use DBA to average the six signals in the data set and the six signals in the attack set into one signal each, which means that the subsequent runtime for calculating the DTW attacks is only $\frac{1}{6}$ of the previous runtime.

4.4.3. Principal Component Analysis

Another way to reduce the number of different sensor signals is the *principal component analysis* (PCA), which goes back to Pearson [89] and Hotelling [90]. The aim of PCA is to project data points from a high-dimensional data space, such as that provided by six different sensors, into a data space with the lowest possible dimensions so that the information present in the original data set is reproduced as well as possible. The dimensional reduction is achieved by combining several variables using principal components, which are linear combinations of the centered or standardized output variables. A principal axis transformation can be used to determine the coefficients of a principal component, which represent the majority of the total variance of the data.

The mathematical calculation of the PCA is carried out using an eigenanalysis on the centered data of the covariance matrix or on the standardized data of the correlation matrix. This allows the eigenvalues, which are the zeros of the covariance matrix or the correlation matrix, to be determined [91]. For each eigenvalue, there are p-dimensional eigenvectors that solve the homogeneous linear system of equations. The eigenvectors can be used to decompose the covariance matrix into n (number of eigenvectors) matrices of rank 1 by means of a spectral decomposition [91]. The resulting n linear combinations of the centered characteristic variables are called principal components.

This procedure allows the potential principal components to be calculated iteratively, with the first principal component representing the largest proportion of the explained total variance. This proportion decreases for each additional principal component. To determine the most suited number of principal components, various heuristic criteria such as the *cutoff criterion* [92] or the *Kaiser criterion* [93] can be found in literature. In the following experiments, only the first principal component is calculated in order to compare the method with DBA in terms of runtime. For the subjects of the WESAD data set used later, this first principal component explains an average proportion of 41.5% of the total variance. Further experiments using more than one principal component for the DTW attacks would also be conceivable. The PCA is integrated into the attack framework by the Python package scikit-learn.[5]

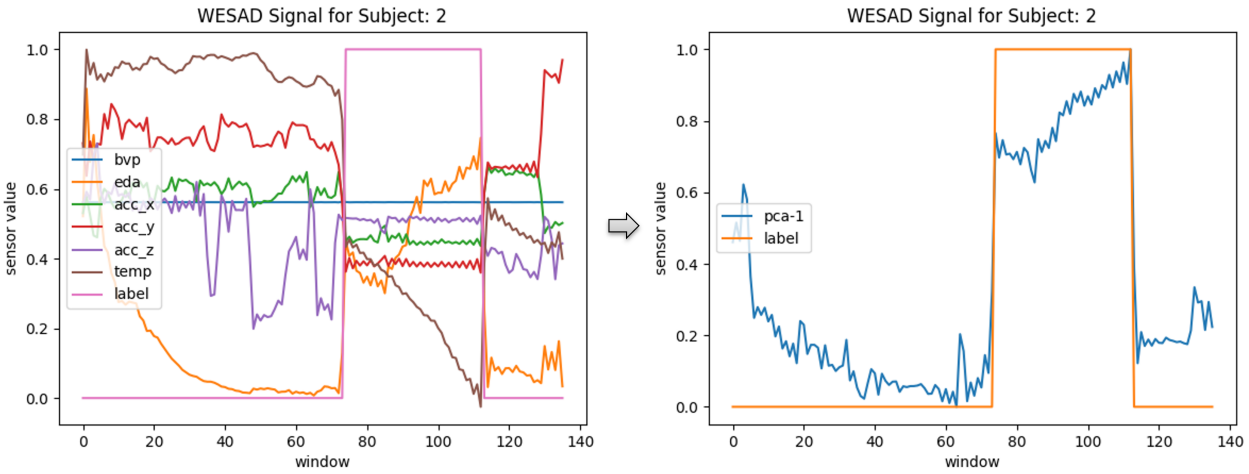


Figure 4.5.: Illustration of principal component analysis. The left plot shows the downsampled signals from figure 4.3. The x-axis shows the different windows of the time series, where one window corresponds to one data point. The normalized values of the signals are shown on the y-axis. The two labels stress and non-stress are also shown, whereby the subject is stressed when the label assumes the value 1.0. The six signals are now reduced to one single signal (pca) using the PCA method.

Figure 4.5 shows the application of PCA to the six downsampled signals in the left plot. Note that the values of the first principal component were subsequently min-max normalized in order to keep the DTW distance values comparable overall. The first principal component is calculated and displayed in the right plot. It can be seen that the first principal component is clearly different from the DBA signal in figure 4.4. Instead of a straight line with peaks, which is primarily influenced by the BVP signal, the curve of the signal is very similar to the EDA signal. The EDA coefficient thus appears to be very influential for this principal component. It can also be seen that during the stress phase, the signal assumes significantly larger normalized values than before the application of PCA. The signals thus become very distinct for the stress and non-stress phase. It is important to note that the strong influence of EDA cannot be generalized to the signals of all subjects, as the PCA is calculated independently for each subject and the attack set. This can reduce the performance of the DTW attacks in case of strong deviations between the attack set and the subject data set.

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

4.5. Re-Identification Attacks based on Dynamic Time Warping

The preprocessed and possibly reduced signal data can now be used to perform the similarity-based re-identification attacks. Four different attacks are presented below, all of which use DTW as a distance measure. In all attacks, the attack set is compared to the subject signals in the data set and one or more distance scores are calculated.

4.5.1. Single-DTW-Attack

The simplest and most trivial DTW attack is the *Single-DTW-Attack*, which is illustrated in figure 4.6. After the signal data of each person (P) in the data set and the attack set have been processed in stages (1-4) by preprocessing them as described in section 4.3 and performing a possible complexity reduction as described in section 4.4, the signals can now be used for the attack. To do this, a DTW alignment between S_1 and the corresponding signal of the attack set is first calculated and stored for each sensor signal (S) contained in P_1. This procedure is first carried out for all sensors S_1 to S_m of a person. The procedure is then repeated for P_2 to P_n. The result of the attack is $n \times m$ many DTW distances, which can be analyzed rank-based in the evaluation pipeline in stage (7) (see figure 4.2). The smaller the distance between S and the attack set, the more similar these signals are and, accordingly, the more likely it is that the person being searched for was re-identified in the data set. As the DTW alignments are calculated individually for each person, the calculations can easily be performed in parallel to significantly increase runtime performance. With the *Single-DTW-Attack*, the length of the attack set is irrelevant. It can be very small and only contain a few seconds of data, or it can be longer than the data set signal. The attack can be carried out in any case, only the performance of the attack can vary significantly for different attack set sizes. This effect is analyzed in more detail in section 6.2.4.

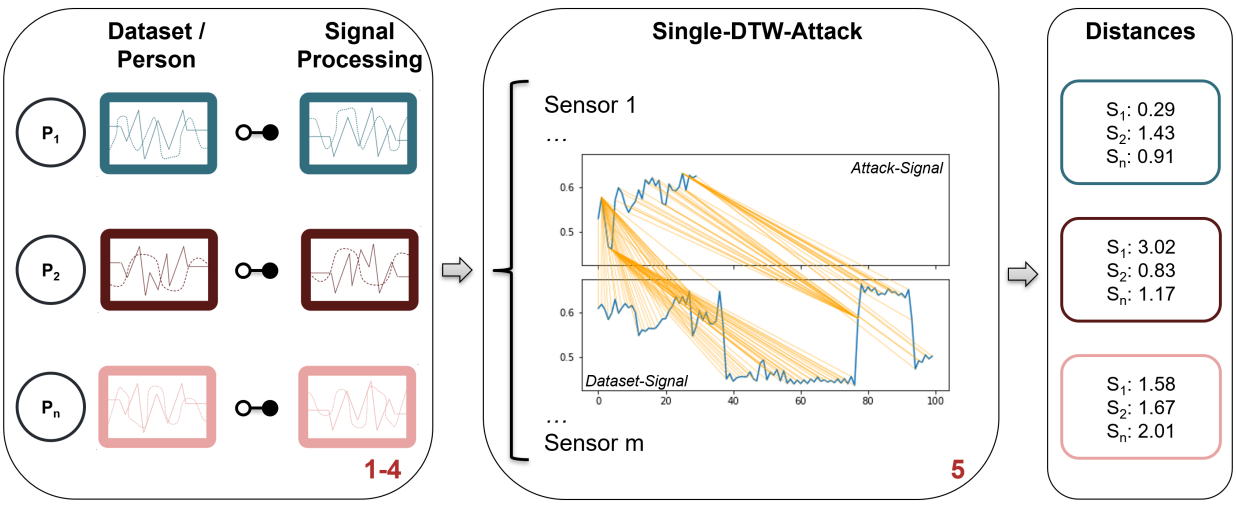


Figure 4.6.: Overview of Single-DTW-Attack. Stages (1-4) show the first four stages from figure 4.2 in a simplified form. The signal data is processed for each person (P) in the data set, with reference to both signal preprocessing and possible complexity reduction. The attack can now be carried out in stage (5). For each sensor signal (S) of a person, a DTW distance to the corresponding signal of the attack set is calculated and stored. The attack set and S are not processed any further for the Single-DTW-Attack.

4.5.2. Multi-DTW-Attack

The second attack, the *Multi-DTW-Attack*, is strongly based on the *Single-DTW-Attack*. The main difference lies in the division of the attack set into multiple attack sets of equal length. Figure 4.7 illustrates the workflow of the Multi-DTW-Attack. Stages (1-4) show the signal processing of the data in a simplified form. Before the attack is carried out, the attacker first specifies an integer multifactor $multi \geq 1$. This multifactor determines the length of the individual attack sets. With a $multi = 2$, the attack set is divided into two multi attack sets of equal length. This results in: $length(multi\ attack\ set) = length(attack\ set)/multi$. When determining the multifactor, a trade-off must be made between a very small multifactor, which can hardly bring out the possible effect of the *Multi-DTW-Attack*, and a very large multifactor, which leads to very small attack sets and therefore contains very little information in the individual multi attack sets. For the following experiments in chapter 5, a $multifactor = 3$ is used, which has proven to be the most suitable for the data sets used there.

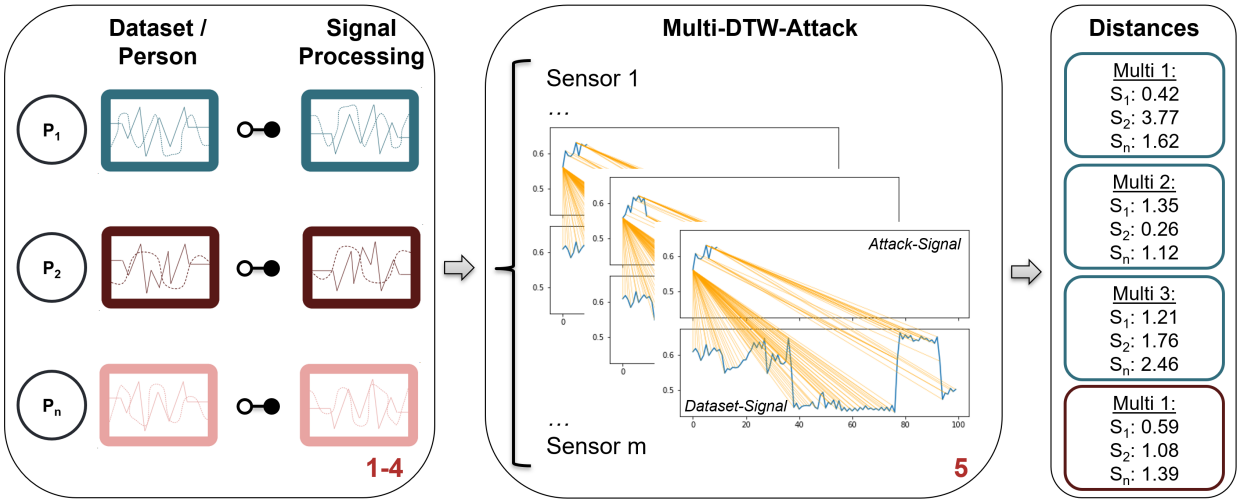


Figure 4.7.: Overview of Multi-DTW-Attack. Stages (1-4) show the first four stages from figure 4.2 in a simplified form. The signal data is processed for each person (P) in the data set, with reference to both signal preprocessing and possible complexity reduction. Before the attack is carried out in stage (5), the attack set is divided into multiple attack sets of equal length. For each sensor signal (S) of a person, a DTW distance is calculated for each corresponding signal of each multi attack set and then stored. S is not processed any further for the Multi-DTW-Attack.

After dividing the attack set into multi attack sets, the attack procedure in stage (5) is equivalent to the *Single-DTW-Attack*, whereby a DTW alignment is now calculated for each S_m of P_n for each corresponding signal of each multi attack set, and the distance scores are stored. This results in $n \times m \times multi$ distances from the attack. For a uniform evaluation, also in comparison to the *Single-DTW-Attack*, the multiple distance scores must be summarized afterward. Possible approaches for this are presented in section 5.3.1. The basic idea behind the *Multi-DTW-Attack* is that small attack window sets can possibly be better aligned with the signals in the data set, as there are fewer deviations within the signal. Deviations can occur, for example, when a person performs activities such as sleeping, sitting, walking in different sequences or in combination. By splitting the attack window sets, the influence of these effects is reduced.

4.5.3. Slicing-DTW-Attack

The third attack is the *Slicing-DTW-Attack*, which, in contrast to the *Multi-DTW-Attack*, does not divide the attack set but the signals (S) of the data set. The attack is illustrated in figure 4.8. Stages (1-4) do not differ in the *Slicing-DTW-Attack*. The basic idea is to adapt each S_m to the length of the corresponding signal in the attack set. In a realistic scenario, each P_n has a varying length of signal data. However, these differences in length can have an influence on the performance of the DTW attacks, which is shown in detail in section 5.2. By matching the data set signals to the length of the attack signal, the influence of length differences between the individual data set signals is minimized. Each S_m of P_n is now divided into overlapping slices of the length of the attack signal. A sliding window approach is used, with each new slice starting at half the length of the previous slice. For example, if the attack signal has a length of 20 windows and S has a length of 100 windows, slice 1 contains the windows 1-20, slice 2 the windows 11-30, slice 3 the windows 21-40, and so on. After the division, there are therefore 9 slices of equal length. If the division of the windows into slices does not work out exactly, windows at the end of the signal that cannot be allocated are ignored. It should be noted that a different number of slices can be produced for each P_n due to signals of different lengths.

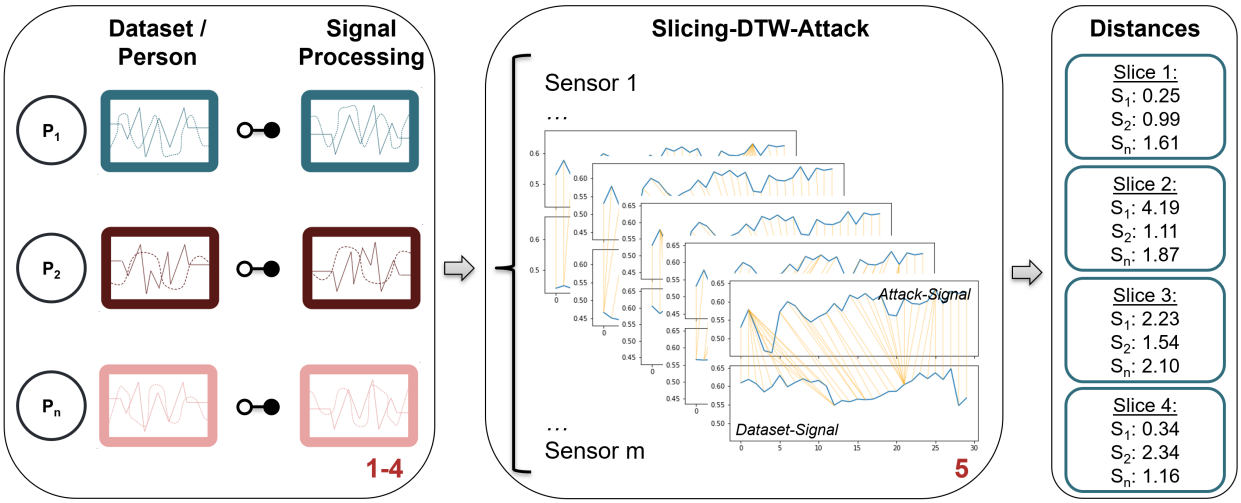


Figure 4.8.: Overview of Slicing-DTW-Attack. Stages (1-4) show the first four stages from figure 4.2 in a simplified form. The signal data is processed for each person (P) in the data set, with reference to both signal preprocessing and possible complexity reduction. The attack can now be carried out in stage (5). The sensor signal (S) of a person is divided into various overlapping slices of the same length of the attack set before the attack is carried out. For each slice of S of a person, a DTW distance is now calculated for each corresponding signal of the attack set and then stored. The attack set is not processed any further for the Slicing-DTW-Attack.

Once the data set has been divided into slices, the attack takes place in stage (5). For this, a DTW alignment is calculated between each signal of the attack set and each slice of S_m of the respective P_n. Afterward, the distance scores are saved. The attack results in $n \times m \times AVG(slices(P_n))$ distances, where $AVG(slices(P_n))$ stands for the average number of slices across all P_n. It is important to note that with very small attack signals, a very large number of slices can result. This can have a negative effect on both the runtime of the attack and the memory requirements. In the worst case, with an attack signal size of one window, the number of slices corresponds to the

length of S. It should also be noted that the following is mandatory for the *Slicing-DTW-Attack*: $length(attack\ signal) \leq length(data\ set\ signal)$.

4.5.4. Multi-Slicing-DTW-Attack

The fourth and final attack is the *Multi-Slicing-DTW-Attack*, which results from a combination of the *Multi-DTW-Attack* and the *Slicing-DTW-Attack*. The idea is to combine the advantages of both attacks, i.e. the reduction of the influence of deviations within the signals of one person and the reduction of the influence of length differences between the signals of different persons, in one attack. The *Multi-Slicing-DTW-Attack* is illustrated in figure 4.9. It is noticeable that the attack set is first divided into multi attack sets. For the subsequent experiments, a multifactor $multi = 3$ is selected, as for the *Multi-DTW-Attack*. The data set signals are then divided into slices of the length of the shorter multi attack set according to the sliding window approach.

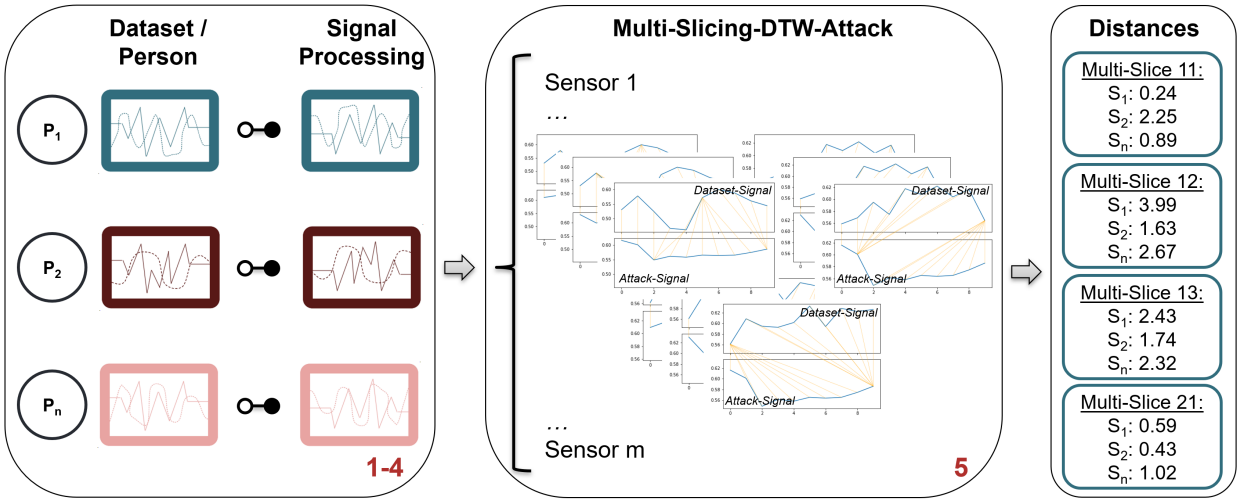


Figure 4.9.: Overview of Multi-Slicing-DTW-Attack. Stages (1-4) show the first four stages from figure 4.2 in a simplified form. The signal data is processed for each person (P) in the data set, with reference to both signal preprocessing and possible complexity reduction. The attack can now be carried out in stage (5). By combining the Multi-DTW-Attack and the Slicing-DTW-Attack, the attack set is divided into multiple attack sets before the attack is carried out and the sensor signal (S) of a person is divided into various slices of the same length of the attack set. For each slice of S of a person, a DTW distance is now calculated and stored for each corresponding signal of each multi attack set. This attack leads to numerous distance scores per person and sensor, which must then be summarized to form a single distance.

The attack calculates a DTW alignment for each S_m of P_n between all multi attack sets and all slices. The calculated DTW distances can subsequently be stored. By utilizing the multi as well as the slicing effect, this attack generates the most distance scores compared to the others, which again has an impact on runtime performance and memory requirements. This results in $n \times m \times multi \times AVG(slices(P_n))$ distance scores. Especially in comparison to the *Slicing-DTW-Attack*, the smaller multi attack sets also generate more slices, which should be taken into account when executing the attack. To carry out the attack, the following is mandatory for the *Multi-Slicing-DTW-Attack*: $length(attack\ signal)/multi \leq length(data\ set\ signal)$.

5. Experimental Setup

Following the presented methodology, this chapter presents the various experiments in this thesis. In section 5.1, the smartwatch data sets used to conduct the experiments are first presented and exploratory analyzed. The first data set is the WESAD data set, which was recorded with an Empatica E4. Subsequently, two synthetic cGAN and DGAN data sets, which are necessary to verify the scalability of the DTW attacks, are introduced. Section 5.2 deals with the underlying data model, which describes the division of the data into the data set of the data owner and the attack set. The attacks are evaluated rank-based, for which section 5.3 presents possibilities for handling multiple distances, such as those produced by the Multi-DTW-Attack, the Slicing-DTW-Attack and the Multi-Slicing-DTW-Attack. The metrics used for rank-based evaluation and rank selection methods for identical ranks are shown, afterward. A four-stage evaluation pipeline is presented in section 5.4 for this purpose. Various ranking methods, the stress and non-stress classes, the different sensors and sensor combinations, as well as the attack window sizes are evaluated one after the other in order to determine the optimal parameters. Finally, section 5.5 describes privacy experiments that aim to determine an optimal trade-off between the usability of the data and the re-identification risk. For this purpose, noise generated according to a Laplace distribution is added to the sensor data and a stress detection task is performed to measure the usability of the data.

5.1. Smartwatch Health Data Sets

There are hardly any data sets to be found in literature that cover a large number of subjects and at the same time contain the variety of sensors that for example the Empatica E4 offers. A very well-known data set is the WESAD data set, which was recorded on a small number of individuals in a clinical study with an Empatica E4. However, in order to be able to evaluate re-identification attacks in depth, data sets with numerous individuals are required. Since these cannot be found for, among others, privacy reasons, two GANs are used in the following experiments in addition to the WESAD data set. These GANs take the WESAD data set as input and are able to generate an unlimited number of synthetic subjects from it. The GAN implementation by Wenzlitschke [11] is used with some adaptations, which has already been trained on the WESAD data set for privacy-preserving stress detection. A detailed exploratory data analysis is carried out for the WESAD data set as well as for both the cGAN and the DGAN data sets.

5.1.1. An Overview of the WESAD Data Set

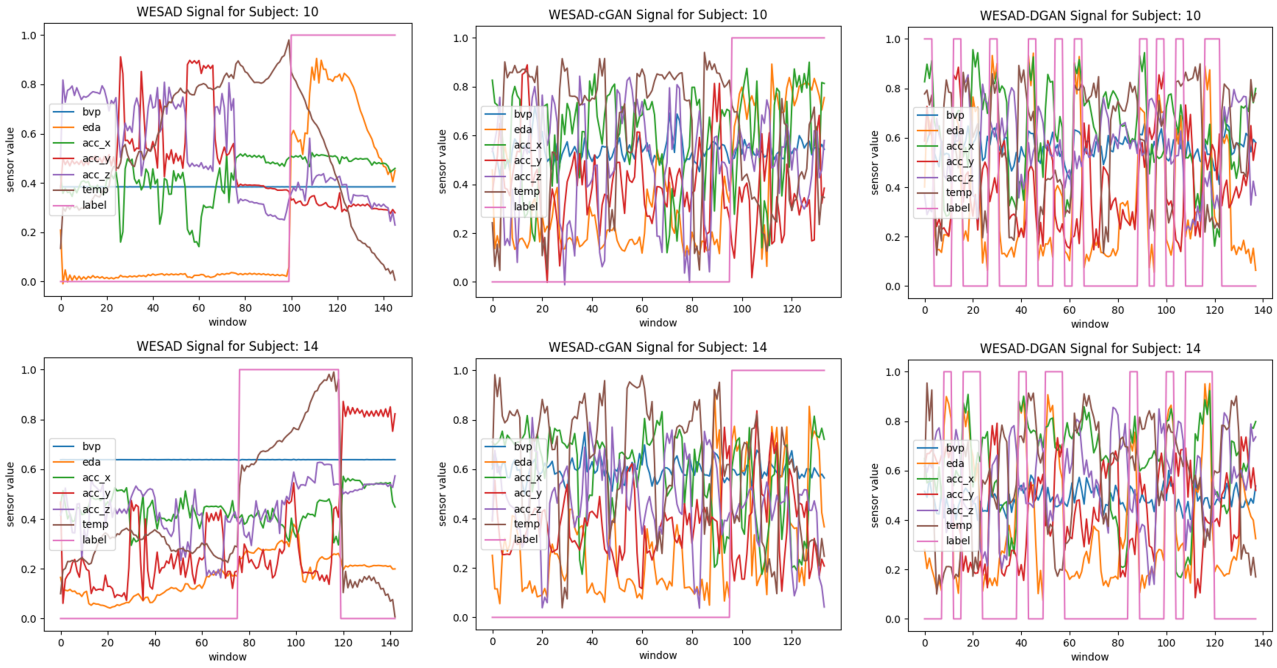
The multimodal data set for **WE**arable **S**tress and **A**ffect **D**etection (WESAD) was presented by Schmidt et al. [10] in 2018 and made publicly available to researchers. Since then, WESAD has been used as a data basis in numerous research projects, particularly in the field of stress detection [45, 46, 47, 48, 49, 50]. The data set is based on two different sensor modalities. On the one hand, data was recorded with a chest-worn RespiBan ACC, ECG, EDA, TEMP, electromyogram and respiration with a sampling rate of 700 Hz, and on the other hand, with a wrist-worn Empatica

E4 with the sensors BVP, EDA, TEMP and ACC. The sensor modalities of the Empatica E4 have already been described in detail in section 2.1.2. In addition, WESAD contains self-reports for each subject, which describe the subjective experience during an affective stimulus. For the experiments in this thesis, only the sensor data of the Empatica E4 is used in order to focus the re-identification attacks on the currently widespread smartwatches, which accordingly pose a great risk with regard to the privacy of the device owners.

The data set was generated in a laboratory study with a total of 15 subjects, 12 of whom were male and 3 female. The average age is 27.5 years. A total of approx. 36 minutes of data are available for each subject, covering the affective states *baseline*, *amusement*, *stress*, *meditation* and *recovery*. As already described in section 4.3, the five classes are reduced to the binary classes *stress* and *non-stress*. To record the data, all subjects took part in a 90-minute laboratory protocol with five different phases:

- *Baseline:* In this phase, the subject's data was recorded for 20 minutes in a neutral affective state. The subjects were sitting or standing at a table while neutral reading material, such as magazines, was available to them.
- *Amusement:* During the second phase, eleven funny video clips were played to the subjects, each separated by a short sequence of five seconds. The amusement phase lasted about 6.5 minutes in total.
- *Stress:* In order to record realistic stress data, the subjects were subjected to the *Trier Social Stress Test*. First, the subjects had to give a five-minute speech in front of a three-person panel about their personal strengths and weaknesses. Subsequently, the subjects were asked to count from 2023 to zero, with an increment of 17, starting from the beginning with each mistake. The subjects had 5 minutes for each of the two tasks.
- *Meditation:* After the amusement and stress phase, the subjects were to be de-excited again in order to bring them back close to the neutral affective state. For this purpose, the subjects received 7 minutes of controlled breathing exercise, which was instructed by an audio track.
- *Recovery:* Finally, the sensors were synchronized and removed from the subjects and the subjects were informed that the panel members were only normal researchers.

Figure 5.1a shows an example of the sensor data from two selected subjects of the WESAD data set. The data was processed using the signal preprocessing pipeline described in section 4.3. In addition, the signal was downsampled with a $DSF = 1000$ for a clear illustration. It can be seen that the signals can vary greatly between the two subjects and that some trends, such as the temperature during the two classes stress and non-stress, can also run in opposite directions. In this example, the temperature decreases during the stress phase for subject 10, while it increases for subject 14. The order in which stress and non-stress data occur is also not always consistent.



(a) (b) (c)

Figure 5.1.: Exploratory data analysis for the various data sets in this work. The six signals BVP, EDA, ACC_X, ACC_Y, ACC_Z and TEMP are each shown in normalized form. In addition, the label (stress or non-stress) is shown, whereby the subject is stressed if label = 1.0. The signals were each downsampled with a DSF = 1000. (a) shows the data of subjects 10 and 14 from the WESAD data set, (b) shows the data of subjects 1010 and 1014 from the WESAD-cGAN data set and (c) shows subjects 1010 and 1014 from the WESAD-DGAN data set.

5.1.2. Introducing conditional GAN and DoppelGANger

With 15 subjects, the WESAD data set is only a very small data set that contains a very specific (biased) group of people. However, no data sets with significantly more subjects can be found in literature. Therefore, the WESAD data set is only of limited use for the evaluation of re-identification attacks. Because of these two reasons, two additional synthetic GAN data sets are used in this work, which were trained on the WESAD data set. GANs can be used to generate an arbitrarily high number of subjects that contain realistic sensor data. In this thesis, a cGAN and DGAN data set are used, which are based on the work of Wenzlitschke [11].[6]

Conditional GAN. Mirza and Osindero [55] presented the conditional GAN (cGAN) in 2014, which extends the GAN architecture of Goodfellow et al. [52] in certain parts. The main difference is that in addition to the random noise sample z (see figure 2.2), the generator also receives an auxiliary information label y, such as a class label, as input. With this auxiliary information, a cGAN architecture can achieve better results in the generation of multimodal data sets [54]. For the generator, z and y are combined into a common hidden representation so that it can generate a synthetic sample. The discriminator receives y and the sample to be classified. The cGAN architecture of Wenzlitschke [11] is based on the work of Ehrhart et al. [94] who use a long short-term memory neural network (LSTM) for the generator and a fully convolutional network

[6] https://github.com/geheim01/Privacy-Preserving-Smartwatch-Health-Data-Generation-Using-DP-GANs

for the discriminator instead of two LSTM for the generator and discriminator. Wenzlitschke [11] likewise, achieved better classification results in stress detection with the synthetic data of a GAN with the fully convolutional network discriminators than with an LSTM discriminator. To address the challenges of unstable cGAN training, the cGAN architecture is coupled with a diversity term derived from the distance between two generated samples. The diversity term was introduced into the loss calculation with the aim of training the generator's ability to generate different samples with different random noise inputs and to avoid mode collapse. The respective stress labels of the time series signals are used as auxiliary information y when training the cGAN.

DoppelGANger. The DoppelGANger (DGAN) was presented by Lin et al. [56] in 2020. The architecture of the DGAN differs from the GAN presented in section 2.2 in three aspects [11]. First, the DGAN uses a decoupled generation of metadata and measurements using an auxiliary discriminator, which makes it possible to condition the generated measurements based on the generated metadata. The generation task is decoupled into two tasks, the generation of metadata and the generation of synthetic samples, by using several generators. Second, the mode collapse problem is addressed by adding fake metadata to capture the minimum and maximum values of each generated sample. Third, a batch RNN generator is used to capture the temporal correlations and to generate representative time series samples. To capture long-term relationships and correlations between data points in a time series, DGAN uses RNN, specifically LSTM. Lin et al. [56] have shown in their experiments that a DGAN can achieve a better performance compared to the baseline GAN models, with significantly less training time.

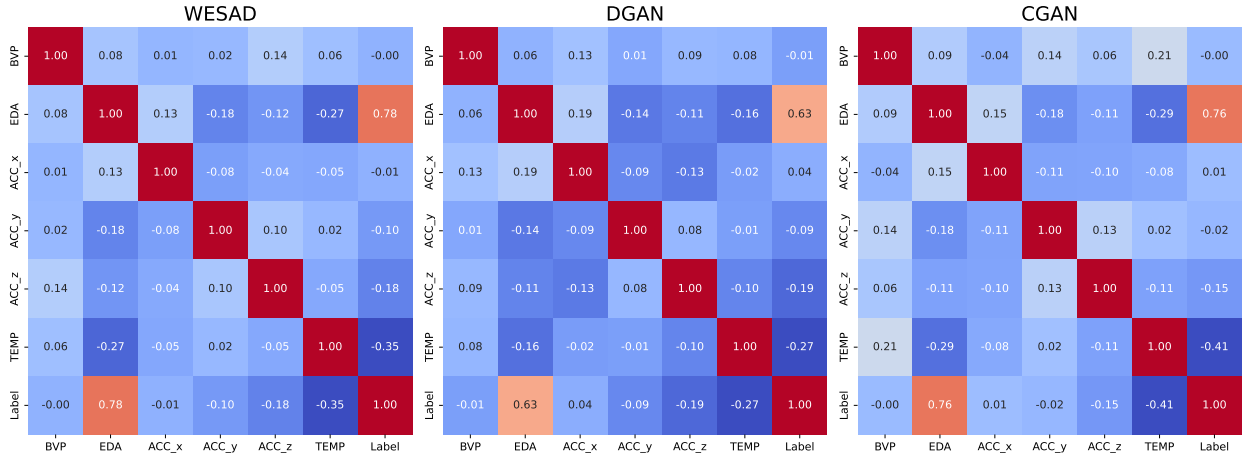


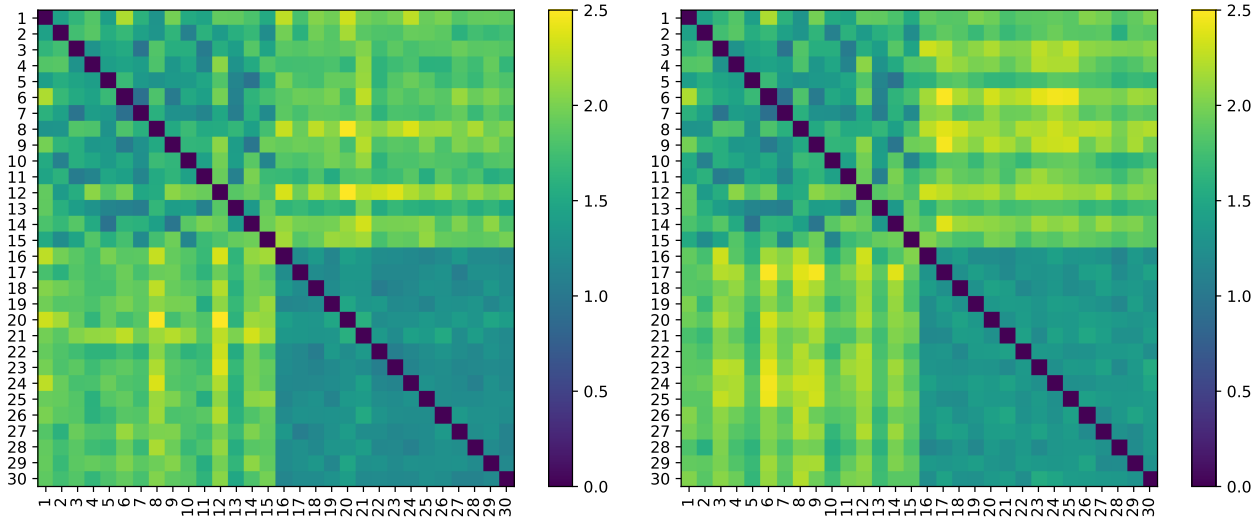
Figure 5.2.: Correlation analysis for the WESAD, WESAD-cGAN and WESAD-DGAN data set from Lange, Wenzlitschke, and Rahm [95]. The correlations between the six sensors in the data set, inclusive of the label, were calculated for all three data sets. A value of 1.0 or -1.0 reflects the strongest positive or negative correlation. This can also be seen in the very dark colors. Values close to 0.0, on the other hand, show that the two sensors in question have little or no correlation with each other. In the correlation analysis between the data sets, the correlation structure of the WESAD data set should now also be recognizable in the WESAD-cGAN and WESAD-DGAN data sets.

Figure 5.1b shows an example of the sensor data of two selected subjects from the WESAD-cGAN data set. It can be seen that there is more variance in the data compared to the WESAD data set (see figure 5.1a), whereby significant changes between the stress and non-stress phase are also visible here. The non-stress phase of the WESAD-cGAN data set is always located at the end of the

data. In the WESAD-DGAN data set, which is shown as an example in figure 5.1c, stress phases of different lengths are distributed over the entire time series of a subject. From a purely visual perspective, the two GAN data sets can hardly be distinguished.

Determining the quality of GAN data sets is not trivial. Metrics have already been introduced for the evaluation of GANs that involve computer vision, but not for time series GANs. Wenzlitschke [11] therefore, use the three desiderata proposed by Yoon, Jarrett, and Schaar [96] in their work. These include *diversity* (the distribution of samples and real data points should be equal), *fidelity* (samples should be indistinguishable from the real data points) and *usefulness* (samples should be as useful in the same application context as the real data points). The detailed evaluation of the GAN data sets can be found in the work by Wenzlitschke [11].

In the following, a special focus is placed on a correlation analysis that describes the fidelity of the GAN data sets in comparison to the WESAD data set. Figure 5.2 shows the correlation matrices of the three data sets. The correlation between the respective sensors of a data set is calculated and displayed. The correlation between the same sensors (e.g. BVP and BVP) is always 1.0, while the other sensors correlate positively or negatively with each other. In the WESAD data set, the strongest positive correlation can be found between the label and EDA, while the strongest negative correlation occurs between the label and TEMP. Ideally, the GAN data sets now exactly replicate the correlation structure of the WESAD data set. For both the WESAD-cGAN and the WESAD-DGAN data set, the previously described strong correlations between label and EDA as well as label and TEMP are recognizable, as are slight differences in the correlation strength. The correlation structure between the three data sets is also essentially similar between the other values.



(a) (b)

Figure 5.3.: DTW distance heatmaps for WESAD-cGAN (a) and WESAD-DGAN (b) data set. The figure shows the distance scores between the subjects of the WESAD data set (1-15) and the WESAD-cGAN or WESAD-DGAN data set (16-30). The distance value averaged over all sensors is displayed in the matrix, where the distance between the same subjects (e.g. 1 and 1) is equal to 0.0. The distance scores are represented by colors, with blue indicating a very small distance and yellow a very large distance. When looking at the distance scores between the data sets, the spread of the distance scores between the subjects should ideally be similar.

For the re-identification attacks, the two GAN data sets in this work are additionally analyzed using a DTW distance heatmap. For this purpose, DTW alignments are calculated between all subjects of the WESAD data set and all subjects of the WESAD-cGAN or WESAD-DGAN data set, with the distances averaged across all sensors. Figure 5.3a shows the DTW distances between the subjects of the WESAD data set (1-15) and the WESAD-cGAN data set (16-30). It can be seen that both very large and very small distance scores occur between the subjects of the WESAD data set. The data set thus contains subjects that are very similar and those that differ strongly. The distance scores between the subjects in the WESAD-cGAN data set are always very similar and are close to 1.0. In other words, although all subjects differ from each other, there are no very similar or dissimilar subjects. Furthermore, it can be seen that the distance scores for the subjects between the data sets are very large. Figure 5.3b shows the same behavioral patterns for the WESAD-DGAN data set. Due to the high distance scores between the subjects of the WESAD and the WESAD-cGAN or WESAD-DGAN data set, it is not advisable to combine the subjects, which is why the data sets are considered separately in the experiments. In all analyzes, no significant differences between the two synthetic data sets are discernible, which is why both GAN data sets are used in addition to the WESAD data set for the evaluation of the previous presented DTW attacks.

5.2. Data Model

The data model, which is illustrated in stage (4) in figure 4.2, has two different modes, the *attack mode* and the *simulation mode*. These two modes determine which data is used as the attack set for the re-identification attacks. The modes are briefly differentiated below.

Attack mode. The simplest mode is the attack mode. If this is selected, a realistic DTW attack can be carried out. The attack set is known to the attacker and is used as external input for the attacks. DTW distance scores between the signals of the attack set and the signals of the subjects of the data set can then be calculated using the various attacks described in section 4.5.

Simulation mode. By contrast, in simulation mode, the attack set is cut out of the data set. This mode is used to carry out the experiments in this work, in particular to determine the performance of the DTW attacks. Hyperparameters can also be optimized in advance using the simulation mode. The attacker specifies three parameters for this. The first parameter is the *attack window size*, which reflects the length of the attack set or the number of windows that are to be cut out of the data set. The second parameter, the *additional windows*, determines how many additional windows are to be cut out of the data set signals at the two edges of the attack set in order to prevent direct alignment at the edges. By default, 1000 additional windows are used for the following experiments. The number changes with the DSF. With a DSF = 1000, one additional window is removed. The *label* (stress or non-stress) is specified as the third parameter, as the attacks are carried out separately, either with an attack set with stress or non-stress data.

If all parameters have been transferred with valid values, a data set is split into attack set and remaining data set. To do this, all stress data for the specified label, for example, is selected and transferred to a new time series. From this new time series, the attack set with the specified attack window size is now cut out exactly from the middle. From the remaining data set signals, not only

the attack window is cut out but also the specified number of additional windows at each edge. Cutting out the data set results in two time series, which are simply merged again. This can lead to jumps in the signals, which, however, can already exist at the point by ignoring the meditation and recovery label of the WESAD data set. Especially in realistic data sets, such jumps are common due to measurement errors or short-term non-wearing of the smartwatch. For similarity-based attacks, jumps in the signals potentially make re-identification more difficult.

In simulation mode, a split into attack set and data set is performed for each subject of the data set and not just for the subject for which the simulation is currently being performed. In other words, if an attack is carried out with subject 1 as the searched subject, for example, the attack set is cut out of the data of subject 1. In addition, an attack set of the same length is cut out of all other subjects, which is ignored for the simulation. The reason for this is the influence of the length of the data set on the DTW distance. If only an attack set were to be cut out of the searched subject, the remaining data set signals for this subject would be significantly shorter than those of the other subjects. Systematically shorter data set signals would overestimate the performance of the DTW attacks, as slightly smaller distance scores would result for the shortened subjects.

This can be demonstrated by the following experiment. Three signals with random values in the range [0, 1] are created. The first signal has a length of 10 windows and is used as the attack set. The second signal has a length of 90 windows and corresponds to the shortened signal by cutting out the attack set. The third signal has a length of 100 windows and represents one of the unshortened signals of the unsearched subjects. If a DTW distance is now calculated between signals 1 and 2 and signals 1 and 3 and this procedure is repeated a total of 10,000 times, an average distance of 2.359 is obtained for signals 1 and 2 and an average distance of 2.518 for signals 1 and 3. If these calculated DTW distances are now transferred to the actual attack scenario, the person being searched for always receives a 0.159 smaller distance value. This would result in the person being searched for is ranked significantly higher in the ranking process due to the often small overall differences between the subjects. The actual performance of a DTW attack would thus be systematically overestimated. For this reason, attack sets are cut out of all subjects in simulation mode.

5.3. Rank-Based Evaluation

In order to evaluate the DTW attacks in a meaningful way, a representative metric must first be selected that is relevant to the objectives of this work. In the attack scenario, the goal is to evaluate a number of attack sets in terms of their similarity to the existing target data. Saleheen et al. [78] evaluated their attacks with a true matching rate and false acceptance rate. However, a rank-based evaluation is particularly useful for the evaluation of calculated distance scores, as this enables a variety of additional evaluations. For example, a statement can also be made whether a person cannot be directly re-identified but can always be found in the top 3 distance scores. This reduction of the subjects to a small group also creates a risk for the privacy of the subjects in the data set.

In a rank-based evaluation, all calculated distance scores are first converted into ranks. In the simplest scenario, exactly one distance score D was calculated between each person P in the data set and the attack set. All $P_n : D_n$ pairs are now first sorted in ascending order according to the

distance scores. According to the standard (1224) ranking, the pair with the smallest distance is assigned rank 1, the pair with the second-smallest distance is assigned rank 2, and so on. This means that the person with rank 1 is most likely the person being searched for. However, by looking at several sensors, multiple distance scores occur. For a basic understanding, only the ranking for one distance score per person is shown in this section. The handling of the various sensors is described in detail in section 5.4.3. The Multi-DTW-Attack, the Slicing-DTW-Attack and the Multi-Slicing-DTW-Attack also lead to several distance scores per person. Section 5.3.1 below first shows how these multiple distance scores are handled so that they can then be evaluated uniformly with the Single-DTW-Attack. Finally, section 5.3.2 describes the evaluation metrics used and the rank selection methods for identical distance scores between several persons.

5.3.1. Handling of Multiple Distances

Figure 5.4 shows the three blocks of distance scores per person that result from the Multi-DTW-Attack. Each of these distance blocks contains the distance scores for the included sensors. It should be possible to evaluate all attacks uniformly. For this purpose, the multiple distance blocks are combined into one block. Two methods for handling multiple distances are presented below.

Average method. With the average method, the arithmetic mean is calculated for each sensor distance score over the distance blocks of a person.

Minimum method. With the minimum method, on the other hand, the minimum distance over all distance blocks of a person is selected for each sensor.

As the Multi-Slicing-DTW-Attack has both a multi and a slicing component and therefore generates double multiple results, two methods are required to handle the distance blocks. First, the multiple distance blocks of the Slicing-DTW-Attack are reduced to one distance block. As the slicing attacks are carried out three times due to the multi-component, three reduced slicing distance blocks are created. These can then be reduced in the second step. With the two methods average and minimum, a total of four combinations of methods can be derived: average-average, minimum-minimum, average-minimum, minimum-average, whereby the first method summarizes the slices and the second one summarizes the multi distance blocks.

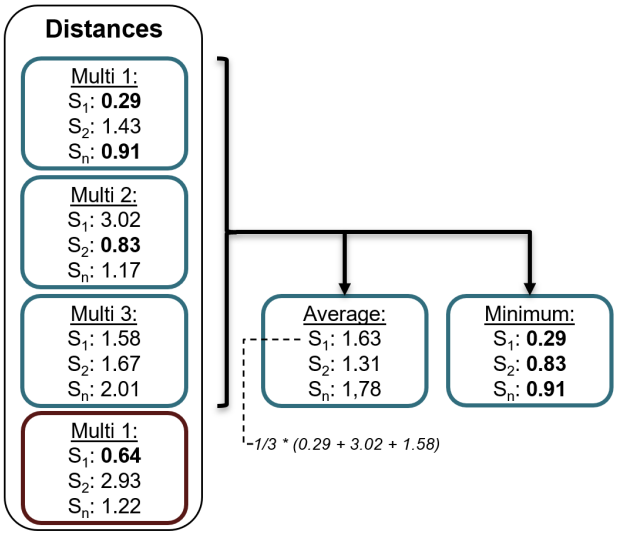


Figure 5.4.: Handling multiple results for rank-based evaluation. For attacks with a multi- or slicing effect, it is necessary to specify whether the average or minimum method should be used.

The evaluation of the average and minimum methods with the three data sets WESAD,

WESAD-cGAN and WESAD-DGAN showed that the average method achieved the best re-identification performance for the Multi-DTW-Attack for 2/3 of the data sets. For the Slicing DTW-Attack, on the other hand, the minimum method achieved the best performance for all three data sets. The exact results can be found in table 1 in the appendix. The experimental setup follows the evaluation pipeline presented in section 5.4 below. For the Multi-Slicing-DTW-Attack, it has been shown that the best re-identification results can be achieved with the minimum-minimum method. The results can be found in table 2 in the appendix. For all further experiments, the average method is therefore used for the Multi-DTW-Attack, the minimum method for the Slicing-DTW-Attack and the minimum-minimum method for the Multi-Slicing-DTW-Attack.

5.3.2. Evaluation Metrics and Rank Selection Methods

In their work, Berrendorf et al. [97] provided an overview of metrics for rank-based evaluation for link prediction and entity alignment methods. They also compared different rank selection methods, which are necessary for evaluation when several persons have received the same rank. The evaluation of this work is strongly based on the recommendations of Berrendorf et al. [97].

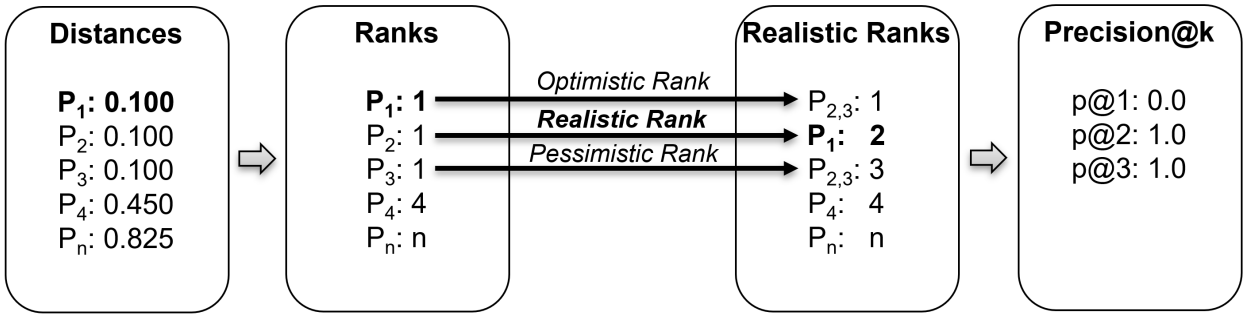


Figure 5.5.: Overview of ranking and rank selection methods. First, the distance scores calculated between the persons (P) in the data set, and the attack set are sorted in ascending order. The person with the lowest distance score is assigned rank 1, the person with the second-lowest distance score rank 2, and so on. If the rank of the person being searched for (marked in bold) exists more than once, a realistic rank selection method is used. Finally, the precision@k can be calculated.

The procedure is illustrated in figure 5.5. In a simplified form, the DTW attack was used to calculate one distance score per person between the attack set, and the data set signals. The distance scores were sorted in ascending order and converted into ranks. The searched person P_1 was ranked best by the procedure, and would therefore most likely be the same person as the attack set. As there are 2 other people in this example (P_2 and P_3), who have also been given rank 1, and these cannot be differentiated further, a rank selection procedure must be used to determine the exact rank of P_1. Possible ranks for P_1 are rank 1, rank 2 and rank 3. Berrendorf et al. [97] demonstrate three rank selection methods for this: the optimistic rank, the pessimistic rank and the realistic rank.

$$rank^{+}(S, \alpha) = |\{\beta \in S \mid \beta > \alpha\}| + 1 \tag{5.1}$$

An initial naive selection criterion is the *optimistic rank*. This is based on the assumption that the person being searched for is in first place among those with the same score. Equation 5.1 shows how

the optimistic rank is determined. S represents the sorted list of distances for each test instance with $S = [\beta_1, ..., \beta_n]$ and α corresponds to the score of the true person [97].

$$rank^{-}(S, \alpha) = |\{\beta \in S \mid \beta \geq \alpha\}| \tag{5.2}$$

In contrast, the *pessimistic rank* (see equation 5.2) is based on the hypothesis that the searched person is in last place among those with the same score [97].

$$rank(S, \alpha) = \frac{1}{2}(rank^{+}(S, \alpha) + rank^{-}(S, \alpha)) \tag{5.3}$$

However, Berrendorf et al. [97] suggest using the *realistic rank* as a ranking selection method. This can be easily reproduced and neither overestimates nor underestimates the actual performance of a procedure. The realistic rank is the arithmetic mean of the optimistic rank and the pessimistic rank, which can be seen in equation 5.3. If no unique rank can be selected because, for example, four people have the same distance scores, a pessimistic rank is selected at the corresponding point, whereby the person being searched for is assigned rank 3 among the four people. By applying a realistic rank selection method, rank 2 is assigned to the searched P_1 in the previous example in figure 5.5. Precision@k (p@k) scores can now be calculated from the realistic ranks.

In general, p@k indicates the proportion of relevant items among the top-k retrieved items. A higher p@k score therefore indicates more relevant results in the top-k list. In the given attack scenario, the goal is to rank the correct target sample as high as possible. Therefore, only its inclusion or absence in the top k list is measured. The most relevant k value is k=1, as it represents the likelihood that the target data is ranked first and allows for direct re-identification. To obtain a better picture, k=3, k=5 are also considered. When deciding on the most powerful version of an attack, the most threatening case of k=1 is mainly considered. If the results are the same, the k value is gradually increased until a clear decision is possible.

5.4. Evaluation Pipeline

All DTW attacks with all possible configurations and parameters are evaluated according to a standardized evaluation pipeline. Results and interim results are saved in a folder structure at the appropriate location. As can be seen in stage (7) in figure 4.2, this evaluation pipeline consists of four stages: ranking methods, classes, sensor ranking and attack window sizes. Section 5.4.1 describes the ranking methods, which are used to reduce several distance scores per person, which occur due to the presence of several sensors per person, to one score and thus to be able to derive a unique rank per person. The influence of the classes (stress and non-stress) is then evaluated separately and averaged. The procedure for this is discussed in section 5.4.2. Section 5.4.3 deals with the sensor ranking. First, the standard sensor ranking is described, which tests all possible combinations of sensors, and then the weighted sensor ranking is explained, in which the ideal weighting of a sensor for the ranking is determined. Finally, section 5.4.4 describes how different attack window sizes are tested. Each phase can be regarded as a separate evaluation that analyses a specific aspect in detail.

The phases build on each other so that the results of the previous evaluation phases are also taken into account. Following the four partial results, an overall result can be calculated. The previously calculated best sensor combination and the best attack window size are used for the overall result. However, an average value is always used for the classes and the best ranking method is tested in advance with a smaller test setup to increase runtime performance.

The WESAD data set includes 15 subjects for the evaluation. To perform a full sweep, each subject is used once as a target and a full attack evaluation is performed. A full attack involves taking the attack set from the data and calculating the similarity scores of all subjects before ranking them. Then the p@k scores for the 15 given ranks are derived. Finally, all 15 results of these individual attacks are averaged based on their p@k scores to determine the overall precision scores. The experiments for the WESAD-cGAN and WESAD-DGAN data sets are set up identically. In the experiments with the GAN data sets with 1000 subjects, each subject is also attempted to be re-identified once as a target.

5.4.1. Ranking Methods

In phase 1 of the evaluation pipeline, the most appropriate ranking method is determined. The ranking method is responsible for combining the individual distance scores of a person at sensor level to determine a unique and final rank for the overall similarity between the attack set and a person's data in the data set. The data set contains the six signals BVP, EDA, ACC_X, ACC_Y, ACC_Z and TEMP without complexity reduction. In order to prevent an excessive influence of ACC on the ranking results in the evaluation, an arithmetic mean is first calculated for ACC_X, ACC_Y and ACC_Z. The average ACC distance score is then processed further using the ranking methods alongside BVP, EDA and TEMP. Two different ranking methods are presented below:

Score method. The simplest form of aggregation is the score method. Here, the average distance score of a person's sensors is calculated. The ranking is then performed using the calculated mean value of the sensor distance scores.

Rank method. The rank method, on the other hand, recognizes that some sensors may match very well, while others may not match at all. For this reason, each sensor is ranked individually, assigning a rank to the distance score of each sensor based on the other samples tested. The individual ranks are then averaged and re-ranked, resulting in an overall rank.

If a sensor score is low for all subjects, but other results are better, the lower overall score could significantly lower the averaged result for the score method. However, the rank method may be more forgiving and instead focus on the superior distance scores provided by the other sensors.

In the evaluation pipeline, both ranking methods are tested in the standard configuration, whereby the method with which the best p@k score can be achieved is selected. However, in order to increase the runtime performance of the evaluation pipeline, the ranking method to be used can also be defined in advance, as it is the case for the evaluation of the experiments in chapter 6.

5.4.2. Classes

Based on the selection of the ranking method, the classes (labels) are considered next. As described in the data model in section 5.2, DTW attacks are carried out with both exclusively stress and non-stress data in the experiment setup. In the evaluation, p@k scores are therefore calculated separately for the stress and non-stress data. On average, there is a class distribution of 70% non-stress and 30% stress data. The classes can be of central importance for the re-identification of a person, as the recorded modalities of a person can differ depending on their affective state [27]. The attacker's attack set can contain both types of data, which is why these are evaluated individually in order to find the most vulnerable in terms of privacy. As described in section 2.1.3, the stress detection task has been well researched in literature. For example Li and Liu [46] were able to achieve a classification accuracy of over 98% on the WESAD data set. It is therefore possible for the attacker to use existing stress detection applications to classify the data to carry out the DTW attack on the potentially more dangerous data. In order to finally derive an overall result of a DTW attack, a weighted p@k score is additionally calculated. The use of a standard mean would neglect the prevalence of neutral data, which is why a weighted mean is considered to be the most representative and effective for the present data model. The weights are calculated on the basis of the average class distribution in the data set.

5.4.3. Sensor Ranking

Until now, the distance scores of the various sensors were simply summarized into one distance score using the ranking methods. Now the influence of the different sensors on the ranking is to be examined. Two different types of sensor ranking are considered for this purpose. In *standard sensor ranking*, all possible combinations of sensors are analyzed, while in *weighted sensor ranking*, the optimum weighting is calculated for each sensor.

5.4.3.1. Standard Sensor Ranking

With standard sensor ranking, each sensor is initially considered individually. Consequently, the ranking is primarily carried out using only the distance values of one sensor. Then every possible combination of 2, 3 and 4 sensors is analyzed. If the combination contains at least 2 sensors, the specified ranking method is used to summarize the distance scores into one score. A similar approach was used by Siirtola [45] in the evaluation of their stress detection methods to determine the influence of the sensors on the results.

Figure 5.6 illustrates the standard sensor ranking process. First, a distance score per sensor (S) was calculated for each person (P) in the data set using one of the four DTW attacks. A ranking is then carried out for each possible sensor combination (SC). For example, if sensors S_1, S_3 and S_4 are in SC_l, a separate ranking is carried out for each S_m. The realistic rank is determined during the ranking. The calculated realistic ranks are then averaged and re-ranked. This corresponds to the ranking method *rank*, while the *score* method can also be used analogously. With the *score* method, the distance scores are first averaged before an overall realistic rank is calculated. This

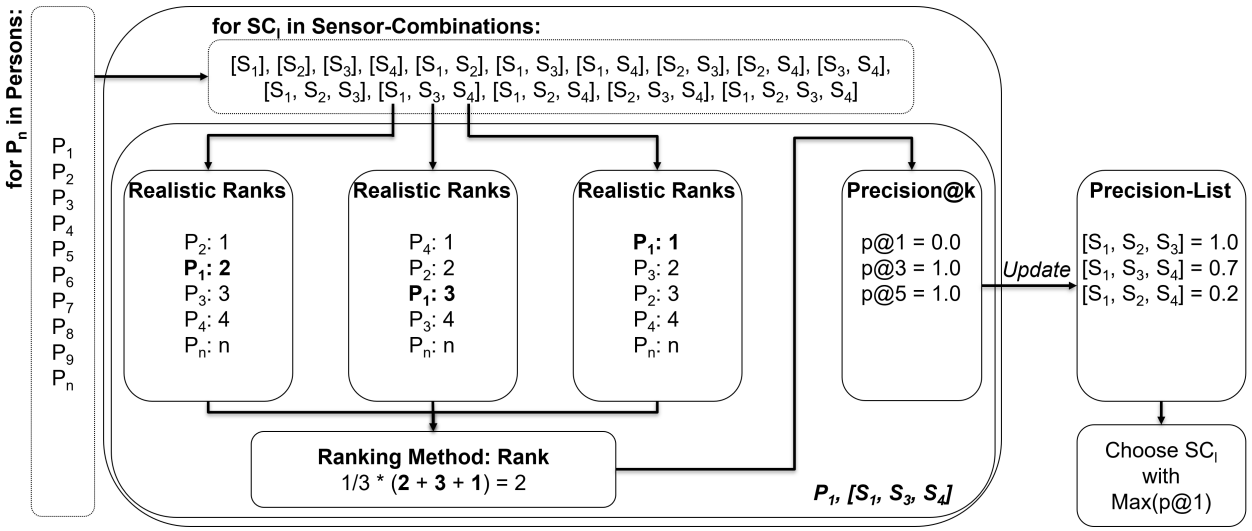


Figure 5.6.: Overview of standard ranking. For each person (P) in the data set, each sensor combination (SC) is tested. Ranking is performed for each sensor (S) of the SC_l and the realistic rank is determined. The realistic ranks can be summarized using one of the two ranking methods rank or score to calculate p@k scores. The p@k scores for each SC_l are stored in a precision list to finally select the SC_l with maximum p@1 from the average p@k scores per SC.

results in a final rank for each S of SC, which is used to determine the p@k scores. The procedure is first repeated for the respective P_n for all SC_l, and the results are saved in an external precision list. The procedure is then repeated iteratively for each P_n of the data set. The p@k scores of the precision list are subsequently averaged for each SC_l so that the sensor combination that generates the best p@k scores can be selected. When selecting the sensor combination, the maximum p@1 is selected first. If no clear decision is possible, the procedure is repeated for the SC_l with maximum p@1 for k=3 and k=5 until a clear decision is possible. If no decision is possible with p@5 either, the choice is made at random from the remaining SC_l.

5.4.3.2. Weighted Sensor Ranking

With appropriate knowledge of the underlying data, DTW attacks can be further improved by adapting the ranking to the available sensor modalities. A sophisticated attacker can determine the optimal sensor weightings in advance based on the anonymized data set available to him. This sophisticated attacker, which comes very close to a worst-case scenario, is simulated using a grid-search approach to determine the optimal sensor weighting. The weighted sensor ranking breaks through the previous evaluation pipeline structure in such a way that the ranking methods become irrelevant. They are replaced by a weighted calculation of a final rank across all four sensors BVP, EDA, ACC and TEMP. A weight W_{S_m} is calculated for each S_m, which is multiplied by the distance of the sensor. Figure 5.7 illustrates the weighted sensor ranking procedure. Here, a DTW distance per S_m to the attack set is first calculated for each P_n of the data set. Now all weights in the range [0, 1] are tested with a step size of 0.2. It is important to note that $W_{S_1} + W_{S_2} + W_{S_3} + W_{S_4} = 1.0$. The calculated DTW distances are multiplied by the respective W_{S_m} and the results for all S_m are added together. A ranking is now carried out using the weighted and summed distances per person.

After ranking, the p@k scores can be calculated using the final ranks and saved in the external precision list. After repeating the procedure across all P_n, the average p@k can be calculated for each permissible weight combination (WC). Finally, the WC_i with the maximum p@1 is selected, or the procedure is repeated for k=3 and k=5 until the best WC_i is determined. This difference in weighted sensor ranking makes it possible to consider less relevant sensors with lower impact compared to standard sensor ranking, so that they can provide some useful insights without compromising the more important sensor alignments.

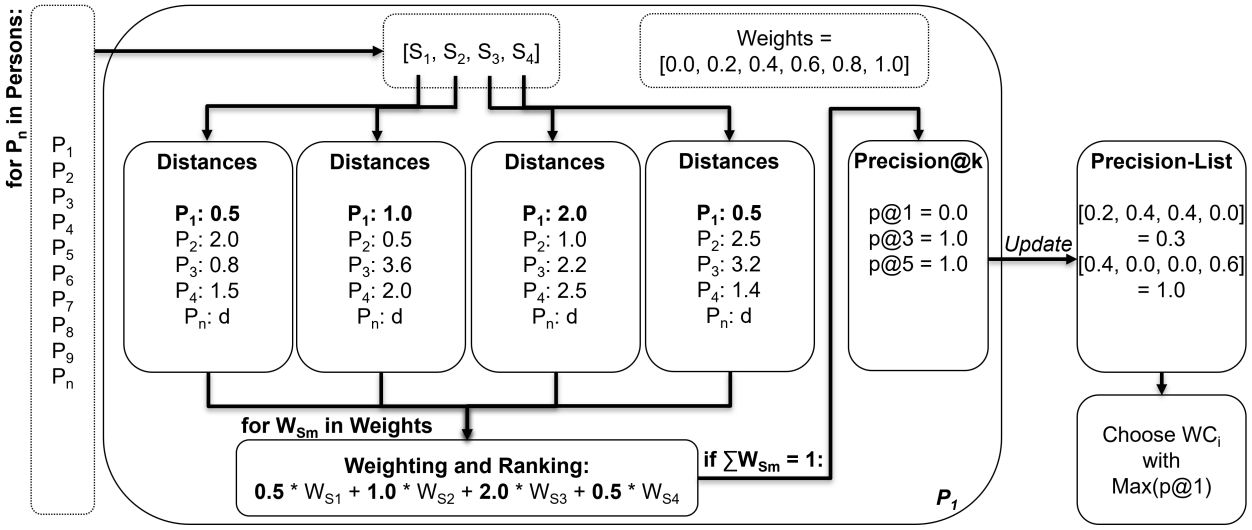


Figure 5.7.: Overview of weighted ranking. The weighted sensor ranking replaces the ranking methods. The best sensor weighting (W) is calculated for each person (P) in the data set. To do this, the DTW distances for each sensor (S) and P are multiplied by the corresponding weights, and the results for each S are added together and ranked. The procedure is similar to the ranking method score with the inclusion of weights. The p@k scores can then be calculated similarly to the standard sensor ranking and saved in an external precision list. The best weight combination (WC) can be derived from the precision list using the maximum p@1.

5.4.4. Attack Window Sizes

An attacker may have different amounts of target data available for the comparison of the DTW attacks. For this reason, the influence of the attack window size should also be analyzed. In a naive scenario, it can be assumed that better p@k scores for the re-identification attacks can be achieved with more data. However, particularly with similarity-based distance measures, slight deviations due to measurement inaccuracies, for example, or because the person performs activities in a different order, can lead to very large distance scores. It therefore makes sense to test different attack window sizes for all attacks in order to be able to evaluate different data requirements and corresponding threat levels. The sizes are measured in windows (number of data points), whereby the attack window size is scaled with the DSF in order to maintain the proportion between the attack window size and the windows in the data set.

For the original signals with a sampling rate of 64 Hz, the attack window sizes 1000–36,000 are tested in a step size of 1000 for the Single-DTW-Attack and the Slicing-DTW-Attack. For the

Multi-DTW-Attack and the Multi-Slicing-DTW-Attack, the attack window sizes 1000-12,000 are tested for each of the three multi attack sets, resulting in a total amount of data equal to the other two attacks. This means that approx. 16 seconds of data are used as attack set in the minimum scenario and approx. 9 minutes of data in the maximum scenario out of a total of 36 minutes. In the following experiments, it is assumed that the attacker has an attack set with a length of 36,000 windows available. The evaluation pipeline tests with which attack window size the best re-identification performance can be achieved, and then chooses this size for the overall evaluation.

5.5. Privacy vs. Usability

Up to this point in the thesis, only the threat of DTW attacks and ways to improve the attacks have been discussed. In this section, however, the focus will be on the privacy of the individuals in the data set. Advanced anonymization techniques are presented, which go far beyond the simple removal of all directly identifying characteristics of the data set. The aim is now to bring the privacy of the individuals to the highest possible level, at which the re-identification risk can be significantly reduced. At the same time, the usability of the anonymized data should be restricted as little as possible. The ideal trade-off between privacy and usability must therefore be found, which allows the data to be used in a targeted manner for the corresponding application purposes without threatening the privacy of the individuals in the data set.

There are various anonymity measures that disclose data in such a way that no inferences can be drawn about a single individual. These go much further than de-identification, as the data itself is changed in addition to the removal of the directly identifying metadata. The best-known anonymity measure is *k-anonymity*, which was introduced by Sweeney [98]. With k-anonymity, equivalence classes are formed in which each sequence of values occurs at least k times, so that each entry in the data set is indistinguishable from k-1 other entries excluding the sensitive attributes. Thus, k-anonymity protects against a correct re-identification of a person with their sensitive attributes with a confidence of 1/k. Building on this, Machanavajjhala et al. [99] first presented the anonymity measure *l-diversity*, which is intended to eliminate the weaknesses of k-anonymity, and then Li, Li, and Venkatasubramanian [100] published *t-closeness* based on the findings of l-diversity. In order to establish anonymity, each of the three anonymity measures requires anonymization techniques that change the data in such a way that it corresponds to the specified degree of anonymity. Possible anonymization techniques are adding noise to the data, adding dummy data, generalizing the data and suppressing unique data. In their work, Shou et al. [101] investigated k-anonymity for the anonymization of time series data. Complex queries, such as range queries and pattern matching, should still be possible on the published data. However, the authors note that k-anonymity cannot effectively address this problem, as serious pattern loss can occur.

Another approach, in which the data itself is not published, but only the query results, is *differential privacy* as presented by Dwork [102]. The goal of differential privacy is that an attacker learns approximately the same information about each individual record, regardless of its presence or absence in the original database. This is achieved by adding independent and identically distributed noise from a Laplace distribution to the query result. A privacy budget ϵ can be set in advance, which specifies the degree of privacy provided, whereby a smaller value of ϵ implies stronger privacy

guarantee and larger perturbation noise. However, especially for complex queries, the proof of differential privacy is not trivial and privacy is only considered on a query-related basis, so that no anonymized data set can be published.

Hereafter, the noise of the data will be tested in a simplified form according to a Laplace distribution. However, due to the difficulties mentioned and the fact that the raw data itself would not be allowed to be published, the algorithm is not constructed in a differential private manner. Section 5.5.1 first briefly outlines the process and the effects of noise injection. Subsequently, section 5.5.2 shows the test setup of a stress detection application as used by Lange, Wenzlitschke, and Rahm [95] to classify the noisy time series data into stress and non-stress. The stress detection should show the usability of the data for each degree of noise. Finally, in section 5.5.3, the DTW attacks will be tested with the noisy WESAD data set. By comparing the results of the stress detection and the re-identification performance, the ideal trade-off between usability and privacy can be derived.

5.5.1. Noise Injection using Laplace Distribution

A common anonymization technique is to add noise to the data. With noise, random values are drawn from a previously specified distribution and added to the original signal. This procedure is very suitable for time series data, as a random value is drawn for each data point and added to the signal. In this work, as with differential privacy, a Laplace distribution is chosen. The probability density function of the Laplace distribution can be seen in equation 5.4.

$$f(x) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}} \tag{5.4}$$

A continuous random number x is subject to the Laplace distribution if it has the probability density $f(x)$ with the location parameter μ, which represents the position of the distribution peak, and the scale parameter σ, which represent the non-negative exponential decay [103]. For the experiments in this work, the WESAD data set is selected and noise is added. The parameter μ is set to 0 by default. The level of noise is specified by σ, which is called noise multiplier (NP) in the following. The larger the NP, the more the data is changed, and the more privacy can potentially be guaranteed. The curve of the Laplace density function with different NPs is shown in figure 5.8. If, for example, an $NP = 1$ is specified, a random value x is drawn and $f(x)$ is added to the corresponding data point in the signal.

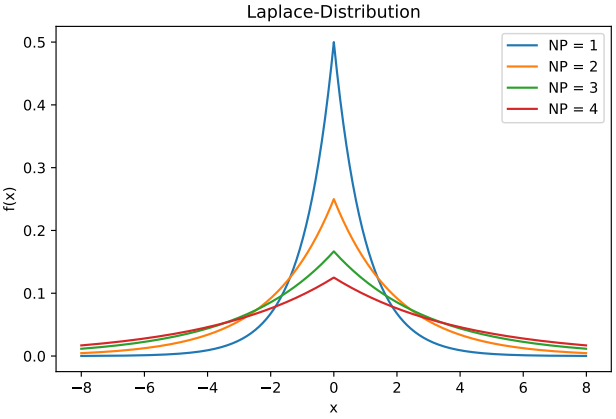


Figure 5.8.: Illustrating the density function of Laplace distribution. The noise multiplier (NP) reflects the scale parameter of the Laplace function to determine the non-negative exponential decay. The larger the NP, the larger the noise added to x.

5.5.2. Noisy Stress Detection

A stress detection is carried out to test the usability of an application with noisy data. As described in section 2.1.3, the signal data is classified into stress and non-stress in a binary task. The aim is now to determine the influence of noise on the classification results. The concrete implementation of stress detection was taken over from Lange, Wenzlitschke, and Rahm [95].[7] The approach uses a CNN for the classification task, which was trained and tested on the WESAD data set with 10 epochs. Lange, Wenzlitschke, and Rahm [95] found that the model presented by Gil-Martin et al. [47] achieved the best results in the non-private setting. For the experiments, different NPs are tested for adding noise to the WESAD data set in the value range [0, 15], whereby a step size of 0.1 is used up to an $NP = 1$ and from then on a step size of 1 is used. The f1-score, which was calculated using a leave-one-out cross-validation with 10 repetitions, is applied as the evaluation measure. The exact experimental setup, with the corresponding preprocessing, the model architecture and the defined hyperparameters is described in detail in the corresponding papers [47, 95].

5.5.3. Reducing the Risk of Re-Identification Attacks

In order to determine the ideal trade-off between privacy and usability, the four different DTW attacks are carried out for the noisy WESAD data set, with the same NPs as for stress detection. It is important to note that only the data owner's signal data is noisy. The attacker's attack set remains unchanged, as this comes closest to the realistic scenario of smartwatches, which usually record and store the sensor data unchanged.

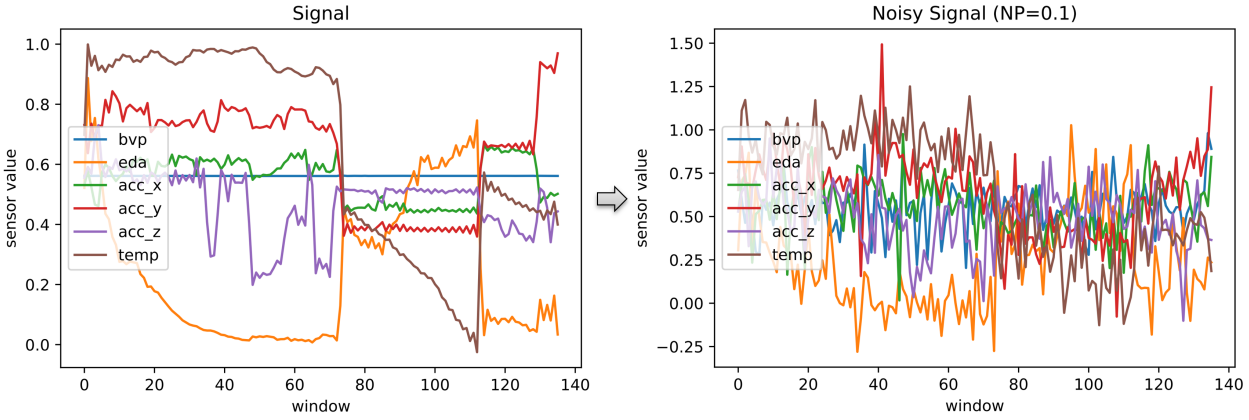


Figure 5.9.: Noisy WESAD data set. The figure shows the effects of adding noise using the Laplace distribution. The left plot shows the original signals of subject 2 from the WESAD data set downsampled with a DSF=1000. The right plot presents the signals downsampled with a small noise multiplier (NP) = 0.1. Due to the noise, the values of the signals can shift to positive and negative values outside the original value range of [0, 1].

The experimental setup corresponds to the procedure described in the previous sections. The *simulation mode* of the data model is used to carry out the attack once for each subject. The data is downsampled with a $DSF = 1000$ and the DTW attacks are performed using *standard sensor*

[7] https://github.com/luckyos-code/Privacy-Preserving-Smartwatch-Health-Data-Generation-Using-DP-GANs

ranking without applying further complexity reduction methods. To optimize the runtime, the attack is only performed for the best attack window size per DTW attack, which is calculated on the original WESAD data set in advance. The evaluation follows the evaluation pipeline presented in section 5.4, whereby only the most threatening p@1 is used for the comparison with the stress detection f1 scores. Since adding noise using a Laplace distribution by drawing random values can have random influences on the p@1 scores, each attack is repeated a total of 10 times and the p@1 scores are averaged across all runs.

Figure 5.9 shows the effects of adding noise to the WESAD data set using the Laplace distribution. A very small $NP = 0.1$ was selected. A strong change in the signal is already recognizable here, which is made clear by larger and smaller peaks. This noisy data is then used to train and test the stress detection and to carry out the DTW attacks. The ideal case would now be that only minimal losses of the classification performance occur, with a simultaneous strong reduction of the re-identification risk due to the DTW attacks.

6. Evaluation

This chapter presents the results of the experiments described in chapter 5. The four different DTW re-identification attacks are evaluated rank-based, as described in section 5.3. The measure p@k is used to determine the re-identification risk for the most threatening scenario of k=1, as well as the two scenarios k=3 and k=5. The evaluation of the DTW attacks follows the evaluation pipeline described in section 5.4, whereby the *score* method is always used for the ranking methods in a simplified form and a weighted mean is calculated for the two classes as default. All DTW attacks are deterministic and therefore produce the same results with identical parameters. For this reason, each attack is carried out exactly once. Only the addition of noise to the data for the privacy vs. usability experiments produces slightly varying results due to the random values drawn. For this purpose, both the re-identification attacks and the stress detection are each repeated 10 times in order to form a meaningful average of the results.

Section 6.1 begins by considering the various methods of complexity reduction in order to use the results obtained there for a more efficient calculation of subsequent experiments. For this purpose, the effect of downsampling and the effects of DBA and PCA on the WESAD data set as well as on the WESAD-cGAN and WESAD-DGAN data set with 15 subjects each are shown. Based on these results, section 6.2 evaluates the DTW attacks for the WESAD data set according to the evaluation pipeline, focusing on the ranking methods, the classes, the sensor ranking and the attack window sizes. This is followed by section 6.3, which shows the results of the weighted sensor ranking for the WESAD data set, with the aim of improving the previous re-identification results. The scalability of the attacks is now tested in section 6.4 with the WESAD-cGAN and WESAD-DGAN data set with a total of 1000 subjects. The aim is to determine whether the re-identification performance can also be achieved with significantly larger data sets. In section 6.5, the runtime of the DTW attacks is examined in a realistic scenario in order to determine the threat risk of the attacks, also with regard to limited hardware resources of the attacker. Finally, section 6.6 evaluates the privacy vs. usability experiments. For this purpose, the influence of noise according to a Laplace distribution on the DTW attacks is compared with the results of a noisy stress detection task.

6.1. Evaluating Complexity Reduction

Complexity reduction methods aim to significantly reduce the runtime of DTW attacks, while ideally maintaining a similarly high level of re-identification performance. Two different approaches are used here. With downsampling, an attempt is made to reduce the sampling rate of the sensors retrospectively. For example, 10 data points of the signal are represented by a single data point. This corresponds to a $DSF = 10$. At the same time, two methods of sensor reduction are used, DBA and PCA, which reduce the number of sensors. DBA follows an averaging approach, which averages the sensors, while PCA calculates the factor loadings of the sensors for the first principal component to determine the influence of a sensor on the reduced signal. All methods are tested on the original WESAD data set and the two synthetic GAN data sets with 15 subjects each.

6.1.1. Evaluation of Downsampling

WESAD data set. Downsampling was evaluated with the four downsampling factors $DSF = \{1, 10, 100, 1000\}$, with the largest reduction occurring at a $DSF = 1000$. Figure 6.1 shows the p@k results of downsampling for the four DTW attacks using the WESAD data set. The p@1 score is the most threatening scenario, as it measures the probability that the person being searched for could be re-identified. A $k = 3$, on the other hand, determines the percentage of cases in which the person being searched for is included in the top 3 ranked persons. It can be seen that the results of the Single-DTW-Attack and the Multi-DTW-Attack generally hardly differ. For example, the p@1 for a $DSF = 1$, which corresponds to no downsampling, is 0.182 for both attacks. It can also be seen that the p@k scores for both attacks increase the stronger the downsampling is. With a $DSF = 1000$, a p@1 of 0.301 is achieved for the Single-DTW-Attack and a p@1 of 0.309 for the Multi-DTW-Attack. As expected, the precision scores increase for higher k values. With the $DSF = 1000$ which reaches the best results when the Single-DTW-Attack achieves a p@3 of 0.540 and a p@5 of 0.692. It can be concluded from this that the two attacks can only rarely correctly identify the person being searched for, but a reduction to a limited number of relevant persons in which the person being searched for is most likely included is possible.

The Slicing-DTW-Attack clearly exceeds the p@k scores of the two previous attacks. For example, a p@1 score of 0.989 can already be achieved with a $DSF = 1$, which also increases with rising DSF and reaches a p@1 of 1.0 for a $DSF = 1000$, whereby 100% re-identification can be achieved for the WESAD data set. The Multi-Slicing-DTW-Attack reverses the trend of the three previous attacks. The p@k scores decrease with increasing DSF. While a p@1 of 0.661 could still be achieved for a $DSF = 1$, this decreases by 5.5% to a p@1 of 0.606 for a $DSF = 1000$. The results for the Multi-Slicing-DTW-Attack also suggest that the combination of the multi and the slicing effect cannot improve the p@k scores.

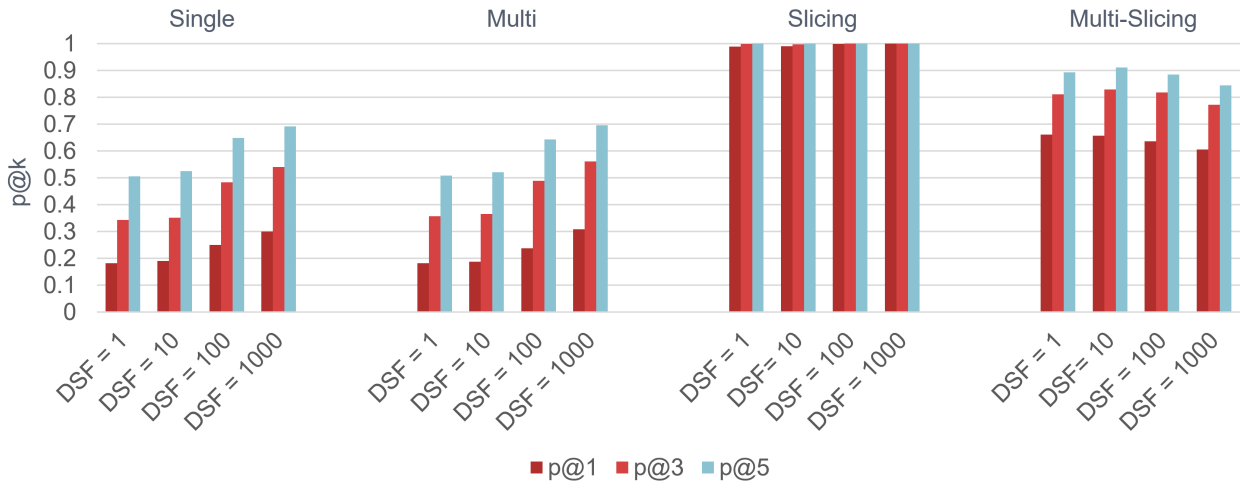


Figure 6.1.: Downsampling precision@k results for WESAD data set. The figure shows the precision@k (p@k) results of the downsampling factors $DSF = \{1, 10, 100, 1000\}$ for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) in a bar chart. The p@k scores are shown for the most threatening scenario of $k = 1$ (re-identification) and for the two scenarios $k = 3$ and $k = 5$, in which the person being searched for must be included in the top 3 or top 5 ranked persons.

GAN data sets. Table 6.1 compares the p@1 scores for the WESAD data set with those of the two GAN data sets. The results reveal that the DTW attacks for the synthetic data sets can achieve significantly worse results. For the Single-DTW-Attack with a $DSF = 1000$, a p@1 of 0.251 (-5%) was achieved for the WESAD-cGAN data set and a p@1 of 0.127 (-17.4%) for the WESAD-DGAN data set compared to the WESAD data set. These losses are even greater for the Multi-DTW-Attack, with 15.2% for the WESAD-cGAN data set and 20.8% for the WESAD-DGAN data set. In contrast, very high p@1 scores of 0.963 (-3.7%) for the WESAD-cGAN data set and 0.959 (-4.1%) for the WESAD-DGAN data set can still be achieved for the Slicing-DTW-Attack. The losses are most significant for the Multi-Slicing-DTW-Attack, which is slightly below the best p@1 of the Single-DTW-Attack for the $DSF = 1$ with 0.250 (-41.1%) on the WESAD-cGAN data set. With a p@1 of 0.196 (-46.5%), the WESAD-DGAN data set continues to outperform the results of the Single-DTW-Attack and the Multi-DTW-Attack. In general, the results of the GAN data sets also confirm the findings on downsampling on the WESAD data set. Thus, with a few exceptions, the p@1 increases with stronger downsampling for the Single-DTW-Attack, the Multi-DTW-Attack and the Slicing-DTW-Attack, while it decreases with increasing DSF for the Multi-Slicing-DTW-Attack, except for the WESAD-DGAN data set. Since the best p@k results were achieved in the majority of cases with a $DSF = 1000$ and this also reduces the runtime the most, a $DSF = 1000$ is used as standard for all further experiments.

DSF	Single			Multi			Slicing			Multi-Slicing		
	W	C	D	W	C	D	W	C	D	W	C	D
1	0.182	0.126	0.139	0.182	0.115	0.093	0.989	0.960	0.951	**0.661**	**0.250**	0.175
10	0.191	0.126	**0.141**	0.188	0.115	**0.101**	0.990	0.961	0.948	0.657	0.246	0.173
100	0.251	0.175	0.099	0.238	0.123	0.096	0.999	0.959	0.953	0.636	0.233	0.129
1000	**0.301**	**0.251**	0.127	**0.309**	**0.157**	**0.101**	**1.000**	**0.963**	**0.959**	0.606	0.207	**0.196**

Table 6.1.: Downsampling results for WESAD, WESAD-cGAN and WESAD-DGAN data set. The table shows the p@1 scores for the WESAD (W), the WESAD-cGAN (C) and the WESAD-DGAN data set (D). The downsampling factors $DSF = \{1, 10, 100, 1000\}$ for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) are taken into account. For each attack and data set, the DSF that achieves the best p@1 is marked in bold.

6.1.2. Evaluation of Sensor Reduction

WESAD data set. The p@k results of the sensor reduction methods for the WESAD data set can be found in figure 6.2. There, the DBA and PCA results are compared with those of the standard sensor ranking. For the Single-DTW-Attack, the Multi-DTW-Attack and the Slicing-DTW-Attack, it can be seen that the DBA and PCA results are lower than those of the standard sensor ranking. While a p@1 of 0.301 was achieved for the Single-DTW-Attack with standard sensor ranking, this drops to 0.297 (-0.4%) when using DBA and to 0.160 (-14.1%) with PCA. The results of the Multi-DTW-Attack are equivalent to those of the Single-DTW-Attack. The losses are significantly greater for the Slicing-DTW-Attack. Here, the p@1 scores are 0.780 (-22%) for DBA and 0.360 (-64%) for PCA. However, the differences in the p@3 scores are significantly smaller for the Slicing-DTW-

Attack for DBA and PCA. Otherwise, with the Multi-Slicing-DTW-Attack, the p@1 results can be increased to 0.713 (+10.7%) using DBA. With a p@1 of 0.227 (-37.9%), the PCA results are significantly lower than those of the standard sensor ranking. In summary, it can be stated that the runtime can be reduced to 1/6 with DBA and PCA, but on the other hand the re-identification performance is significantly reduced except for the Multi-Slicing-DTW-Attack. The lowest p@k losses can be achieved using DBA.

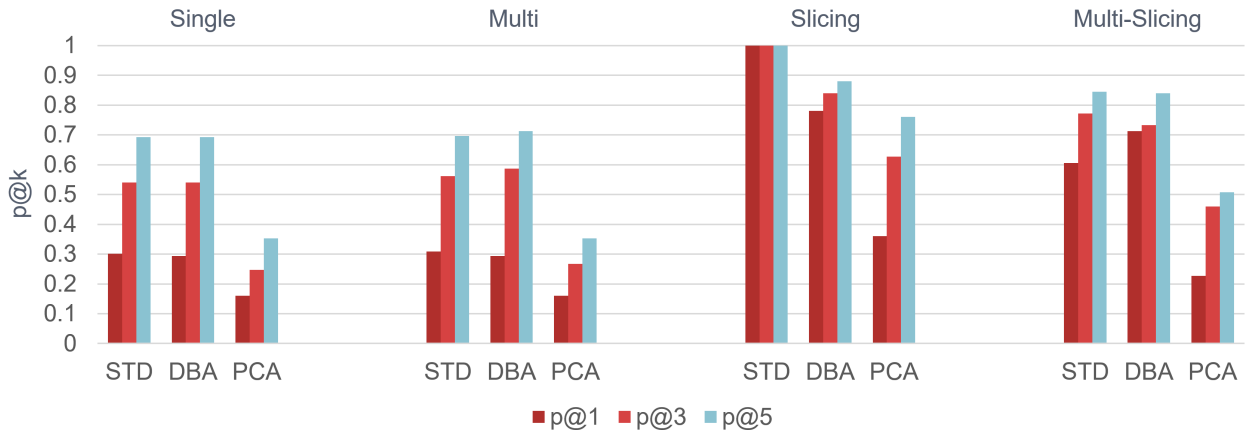


Figure 6.2.: DBA and PCA precision@k results for WESAD data set. The figure shows the precision@k (p@k) results of the two sensor reduction methods DBA and PCA compared to the standard sensor ranking results (STD) for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) in a bar chart. The p@k scores for a $DSF = 1000$ are shown for the most threatening scenario of $k = 1$ (re-identification) and for the two scenarios $k = 3$ and $k = 5$, in which the person being searched for must be included in the top 3 or top 5 ranked persons.

GAN data sets. When comparing the sensor reduction results of the WESAD data set with the GAN data sets, it is noticeable for the Single-DTW-Attack and the Multi-DTW-Attack that the findings of the WESAD data set cannot be transferred. The results can be found in table 6.2. For example, the best p@1 score of 0.267 can be achieved with DBA for the WESAD-cGAN data set, while the best p@1 score of 0.159 can be achieved with PCA for the WESAD-DGAN data set. In contrast, for the Multi-DTW-Attack, the best re-identification results were generated with DBA for both GAN data sets. However, the results of the WESAD data set can be confirmed for the Slicing-DTW-Attack and the Multi-Slicing-DTW-Attack. The best p@1 scores for the Slicing-DTW-Attack can be generated with the standard sensor ranking. If DBA is used, there are significant performance losses. For example, the p@1 of 0.963 with standard sensor ranking is reduced to 0.533 (-43%) for the WESAD-cGAN data set with DBA and to 0.229 (-73.4%) with PCA. With the WESAD-DGAN data set, the losses are even greater, at 70.8% for DBA and 80% for PCA.

For the Multi-Slicing-DTW-Attack, the best p@1 scores can be achieved using DBA, as with the WESAD data set. However, at 0.333 for the WESAD-cGAN data set and 0.225 for the WESAD-DGAN data set, these are significantly lower than the results of the WESAD data set at 0.713. Overall, no generally valid conclusion can be drawn from the results. The results for the Single-DTW-Attack and Multi-DTW-Attack in particular vary greatly depending on the method and data set. For the Slicing-DTW-Attack, exclusively the standard sensor ranking leads to very high

p@1 scores. Only for the Multi-Slicing-DTW-Attack can it be determined across all data sets that higher p@1 scores can be achieved using DBA. For the further evaluations, the standard sensor ranking is used exclusively to show the worst-case scenario of re-identification performance, which occurs primarily with the Slicing-DTW-Attack. DBA and PCA can be seen as a way of reducing the runtime for DTW attacks, but in some cases this can lead to significant performance losses.

Method	Single			Multi			Slicing			Multi-Slicing		
	W	C	D	W	C	D	W	C	D	W	C	D
STD	**0.301**	0.175	0.127	**0.309**	0.157	0.101	**1.000**	**0.963**	**0.959**	0.606	0.207	0.196
DBA	0.293	**0.267**	0.133	0.293	**0.285**	**0.159**	0.780	0.533	0.251	**0.713**	**0.333**	**0.225**
PCA	0.160	0.163	**0.159**	0.160	0.133	0.087	0.360	0.229	0.159	0.227	0.211	0.113

Table 6.2.: DBA and PCA results for WESAD, WESAD-cGAN and WESAD-DGAN data set. The table shows the p@1 scores for the WESAD (W), the WESAD-cGAN (C) and the WESAD-DGAN data set (D) with a $DSF = 1000$. The sensor reduction methods DBA and PCA compared to the standard sensor ranking for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) are taken into account. For each attack and data set, the complexity reduction method that achieves the best p@1 is marked in bold.

6.2. Evaluating Re-Identification Attacks on WESAD Data Set

With the findings from the evaluation of the complexity reduction, the DTW attacks on the original WESAD data set are now evaluated according to the evaluation pipeline. A $DSF = 1000$ and the standard sensor ranking without the application of DBA or PCA are used. The evaluation pipeline consists of the four stages ranking methods, classes, (standard) sensor ranking and attack window sizes, which are evaluated separately in the following subsections.

6.2.1. Evaluation of Ranking Methods

In the first stage of the evaluation pipeline, the two ranking methods *rank* and *score*, which were presented in section 5.4.1, are evaluated in order to subsequently select the method that generates the best p@k scores. The evaluation is carried out as follows: an average of all the individual results calculated for the various sensor combinations and attack window sizes is calculated. By using the simplified evaluation pipeline, a weighted mean is already used here for the two classes stress and non-stress. This results in a p@k score for the *rank* method and one for the *score* method for each DTW attack and k value. To select the best ranking method for each attack, the p@1 score is now considered first. If one of the scores is greater than the other, the ranking method with the higher score is selected. If the scores are identical, an attempt is made to make a clear decision for the following k value ($k = 3$ or $k = 5$). However, in the case that no decision can be made for $k = 5$, a ranking method is chosen at random. Note that the calculated p@k scores for the four stages of the evaluation pipeline are used exclusively to determine the best parameters and not as an indicator

of the performance of the DTW attacks. The performance of the attacks is then calculated using the best parameters of each stage.

k	Single		Multi		Slicing		Multi-Slicing	
	rank	score	rank	score	rank	score	rank	score
k=1	**0.277**	0.272	0.264	**0.264**	0.999	0.999	0.350	**0.473**
k=3	0.505	0.476	**0.485**	0.470	1.000	1.000	0.597	0.648
k=5	0.631	0.626	0.619	0.609	**1.000**	**1.000**	0.724	0.724

Table 6.3.: Evaluation of ranking methods. The table shows the p@k scores of the ranking methods *rank* and *score* for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing). The attacks were performed on the WESAD data set with a $DSF = 1000$ and with the standard sensor ranking. The best method per attack according to the selection procedure was marked in bold.

Table 6.3 provides the results for the ranking methods. For the Single-DTW-Attack, the decision could already be made for a k=1. Here, better p@1 scores could be achieved with the *rank* method. For the Multi-DTW-Attack, the *rank* method could only be selected as the best method for a k=3. However, no clear decision can be made for the Slicing-DTW-Attack. In this case, the results do not differ between the two methods. The procedure would now choose a method at random. The *score* method delivers the best p@1 scores for the Multi-Slicing-DTW-Attack. Overall, the *rank* method seems to lead to better results for a slight majority of attacks for the WESAD data set. However, for the two most threatening attacks, the Slicing-DTW-Attack and the Multi-Slicing-DTW-Attack, no clear decision is possible, or the *score* method is more suitable. The evaluation for the two GAN data sets showed that better results were achieved for all attacks with the *score* method (see table 3 in the appendix). For this reason, the *score* method is used in the simplified evaluation pipeline.

6.2.2. Evaluation of Classes

Using the score method, the two classes *stress* and *non-stress* are now analyzed. The results for the sensor combinations and the attack window sizes are again averaged. Table 6.4 shows the results for the classes. It can be clearly observed that with *non-stress* data the Single-DTW-Attack, the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack achieve significantly better p@1 scores. For the Slicing-DTW-Attack, the same results are generated for both classes, which means that no clear decision can be made. If a person is not stressed, which usually means that the signal data is also calmer, better re-identification results can be achieved. An attacker can exploit this knowledge and apply a stress detection, as described in section 2.1.3, to classify the attack set to obtain better results. These findings are particularly interesting in contrast to the work of Saleheen et al. [78] whose results are presented in figure 3.1. The re-identification risk of the WristPrint attack illustrated there was significantly higher for the activities *sports* and *exercise*, where a relatively high stress level can be assumed, than for the potentially less stressful *stationary* and *walking* activities. The DTW attacks can therefore generate the best re-identification results for the *non-stress* data that occurs much more frequently in reality.

k	Single			Multi			Slicing			Multi-Slicing		
	non	stress	mean	non	stress	mean	non	stress	mean	non	stress	mean
k=1	**0.297**	0.247	0.282	**0.304**	0.224	0.280	0.999	0.999	0.999	**0.571**	0.374	0.512
k=3	0.548	0.404	0.505	0.553	0.387	0.503	1.000	1.000	1.000	0.758	0.537	0.692
k=5	0.720	0.533	0.664	0.700	0.519	0.646	**1.000**	**1.000**	1.000	0.842	0.642	0.782

Table 6.4.: Evaluation of classes. The table shows the p@k scores of the classes *non-stress* (non) and *stress* (stress) for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing). The attacks were performed on the WESAD data set with a $DSF = 1000$ and with the standard sensor ranking. The best class per attack according to the selection procedure was marked in bold. For further evaluations, the weighted mean (mean) is used.

6.2.3. Evaluation of Sensor Ranking

The influence of the different sensors on the re-identification performance will now be shown. The four sensors BVP, EDA, TEMP and ACC are evaluated separately and in each combination of two, three and four sensors. The exact procedure for the standard sensor ranking is described in detail in section 5.4.3.1. The evaluation of the sensor ranking also follows the previous procedure. The ranking method *score* and the weighted mean are used for the classes and an average score is calculated from the results of the various attack window sizes.

k	Single			Multi			Slicing			Multi-Slicing		
	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5
B	**1.000**	1.000	1.000	**1.000**	1.000	1.000	0.997	1.000	1.000	**0.991**	1.000	1.000
E	0.067	0.217	0.387	0.067	0.215	0.387	0.993	1.000	1.000	0.174	0.351	0.471
T	0.084	0.250	0.433	0.104	0.233	0.383	0.996	1.000	1.000	0.045	0.172	0.307
A	0.276	0.465	0.603	0.268	0.485	0.634	1.000	1.000	**1.000**	0.547	0.793	0.903
B+E	0.176	0.539	0.757	0.176	0.555	0.757	1.000	1.000	**1.000**	0.701	0.894	0.947
B+T	0.338	0.648	0.811	0.335	0.656	0.762	1.000	1.000	**1.000**	0.508	0.758	0.874
B+A	0.609	0.887	0.963	0.637	0.878	0.958	1.000	1.000	**1.000**	0.923	0.975	0.987
E+T	0.100	0.186	0.404	0.065	0.194	0.359	1.000	1.000	**1.000**	0.077	0.197	0.358
E+A	0.104	0.259	0.522	0.094	0.260	0.482	1.000	1.000	**1.000**	0.415	0.692	0.783
T+A	0.176	0.429	0.542	0.189	0.424	0.538	1.000	1.000	**1.000**	0.297	0.524	0.707
B+E+T	0.260	0.540	0.706	0.233	0.528	0.698	1.000	1.000	**1.000**	0.506	0.799	0.902
B+E+A	0.217	0.603	0.823	0.213	0.614	0.810	1.000	1.000	**1.000**	0.822	0.910	0.960
B+T+A	0.411	0.690	0.807	0.440	0.655	0.770	1.000	1.000	**1.000**	0.715	0.873	0.920
E+T+A	0.120	0.285	0.444	0.083	0.280	0.412	1.000	1.000	**1.000**	0.300	0.562	0.676
B+E+T+A	0.289	0.569	0.755	0.290	0.574	0.737	1.000	1.000	**1.000**	0.660	0.875	0.936

Table 6.5.: Evaluation of sensor ranking. The table shows the p@k scores of the sensors BVP, EDA, TEMP and ACC and each combination of two, three and four sensors for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing). The attacks were performed on the WESAD data set with a $DSF = 1000$ and with the standard sensor ranking. The best sensor combination per attack according to the selection procedure was marked in bold.

The corresponding results for all possible sensor combinations can be found in table 6.5. It can be very clearly deduced that for the Single-DTW-Attack, the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack the sole use of BVP leads to the best p@1 scores. The sensor combinations that contain BVP also generate better p@1 scores than those without BVP. The second-best p@1 scores across those three DTW attacks can be generated for the BVP+ACC sensor combination. For the Slicing-DTW-Attack, different sensor combinations have hardly any influence on the re-identification performance. In this case, all results achieve p@1 scores of over 0.99. However, the best sensor is ACC or all combinations of two, three and four sensors. The selection procedure would randomly select a combination from the sensor combinations marked in bold in the table. In contrast to the other three attacks, the use of multiple sensors appears to be advantageous for the Slicing-DTW-Attack. Overall, it can be concluded from the evaluation that the BVP and ACC sensors are particularly sensitive for data privacy, as there is the highest re-identification risk.

6.2.4. Evaluation of Attack Window Sizes

The last stage of the evaluation pipeline focuses on the attack window sizes. The aim of this evaluation is to find out what amount of data is required to achieve the optimum re-identification performance. As described in section 5.4.4, the attack window sizes $1-36$ are tested for the Single-DTW-Attack and for the Slicing-DTW-Attack. With a $DSF = 1000$, 36 windows correspond to the maximum number of stress windows available in the data set. For the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack, only the attack window sizes $1-12$ were tested for each multi attack set due to the tripartite division of the attack set in order to maintain comparability with the other two attacks. The curve of the p@k scores for the four DTW attacks with different attack window sizes is shown in figure 6.3. The exact p@k scores can also be found in table 4 in the appendix. The best sensor combination for each attack is used to determine the p@k scores, which is the BVP sensor for the Single-DTW-Attack, the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack. The sensor combination $BVP+EDA+TEMP+ACC$ was randomly selected for the Slicing-DTW-Attack.

Figure 6.3 illustrates the phenomenon that the best p@1 scores can be achieved with the Single-DTW-Attack and the Multi-DTW-Attack with the smallest attack window size of 1 (approx. 16, resp. 48 seconds). This means that the fewer data is used, the more clearly it can be assigned to an individual and, accordingly, the better the re-identification performance. The performance decreases continuously as the amount of data increases. In principle, the same phenomenon can be seen for the Multi-Slicing-DTW-Attack, where the best attack window size is 3. This is only different for the Slicing-DTW-Attack. Here, the maximum p@1 score of 1.0 can only be achieved from an attack window size of 2, although this remains the same as the attack window size increases. The basic assumption that more data leads to better re-identification results can therefore not be confirmed. However, the use of more data is not disadvantageous for the overall best-performing Slicing-DTW-Attack. The overall performance of a DTW attack on the WESAD data set after selecting the optimal parameters according to the evaluation pipeline can be found in table 4 in the appendix. The corresponding overall results have also been analyzed for the discussion of table 6.1 in the row $DSF = 1000$.

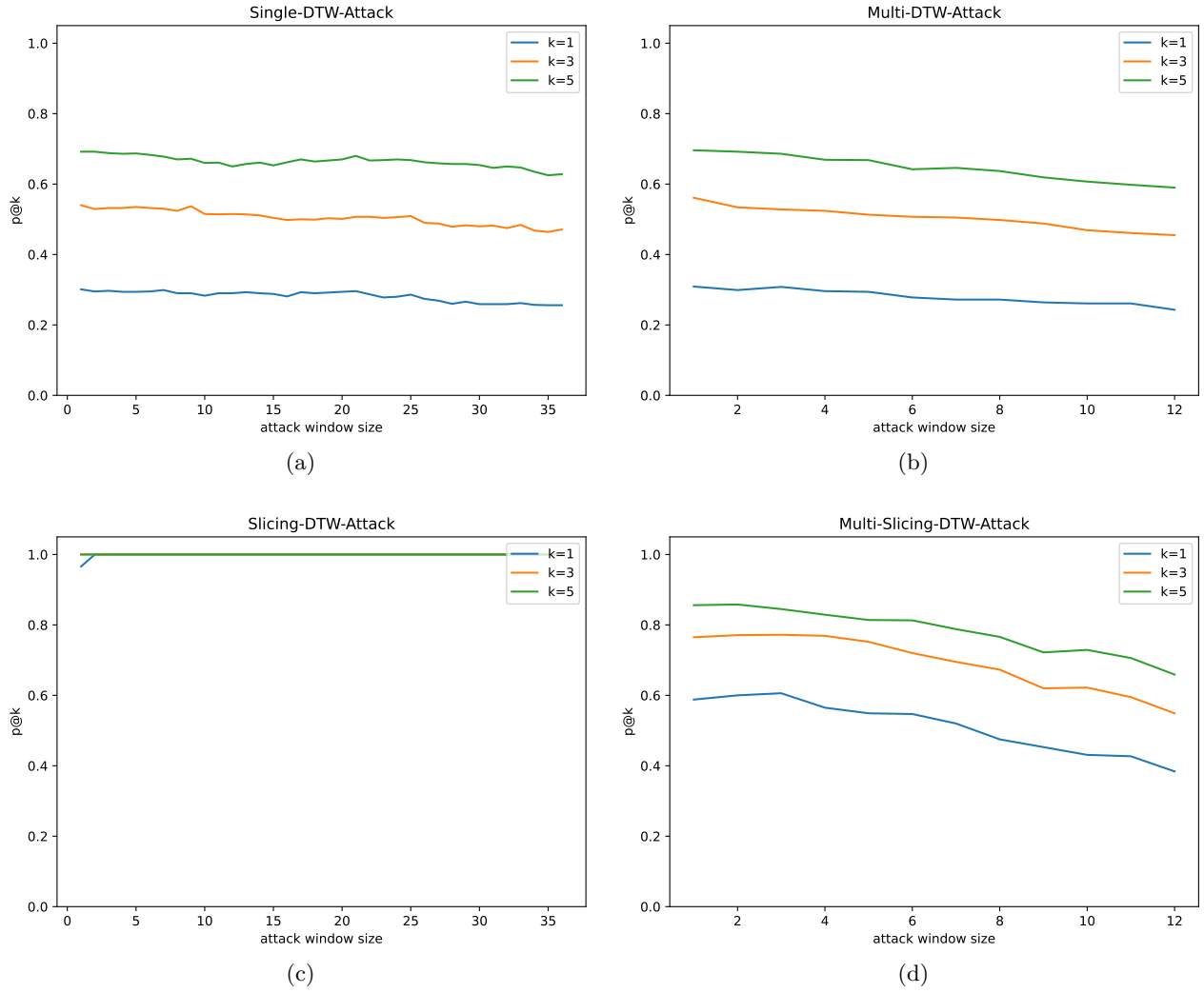
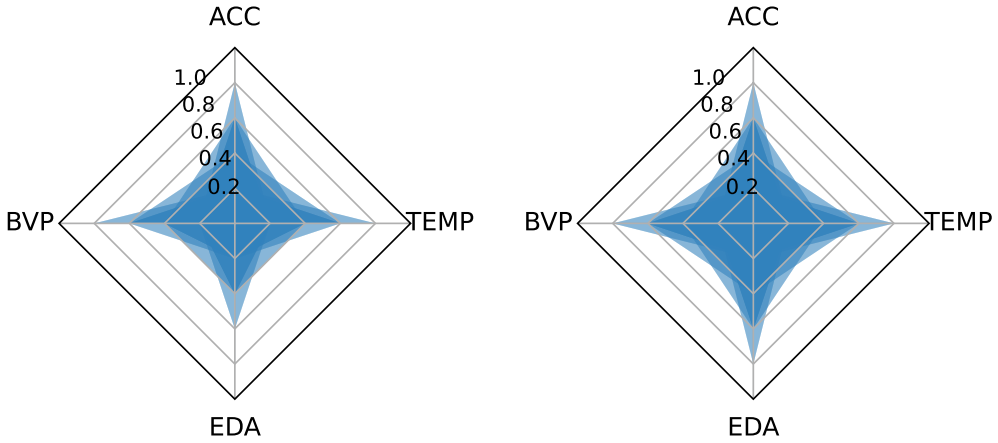


Figure 6.3.: Evaluation of attack window sizes. The figure shows the curve of the p@k scores of the different tested attack window sizes for the four re-identification attacks Single-DTW-Attack (a), Multi-DTW-Attack (b), Slicing-DTW-Attack (c) and Multi-Slicing-DTW-Attack (d). The attack window sizes 1-36 were tested for the Single-DTW-Attack and the Slicing-DTW-Attack, while the attack window sizes 1-12 were tested for the Multi-DTW-Attack and Multi-Slicing-DTW-Attack due to the tripartite division of the attack set. The attacks were performed on the WESAD data set with a $DSF = 1000$ and with the standard sensor ranking.

6.3. Evaluating Weighted Sensor Ranking

In the case of weighted sensor ranking, an attempt is made to further improve the re-identification performance by incorporating the characteristics of the data set. For this purpose, an optimal weighting is calculated for each sensor by multiplying the DTW distance (D) by the corresponding weight (W), so that the following applies: $D_W = D_{S_1} * W_{S_1} + D_{S_2} * W_{S_2} + D_{S_3} * W_{S_3} + D_{S_4} * W_{S_4}$. The resulting weighted and summed distances are then used for the ranking. These weights can be calculated in advance in a realistic attack scenario by simulating attacks on the data set of the data owner. In this experiment setup, the weight combinations $\{W_{S_1}, W_{S_2}, W_{S_3}, W_{S_4}\}$ that generate the maximum p@k are considered, as illustrated in figure 5.7. Due to the fact that several weight

combinations can lead to the maximum p@k, the strength of the weighting per sensor is shown in figure 6.4 over all the best sensor weightings found. Figure 6.4a shows the weights for the Single-DTW-Attack, the Multi-DTW-Attack and the Slicing-DTW-Attack, where the weights for stress and non-stress are identical. It is noticeable that the sensors BVP, TEMP and ACC lead at the same weight strength of up to 0.8 to the maximum p@k scores. The sensor EDA, on the other hand, is only considered with a maximum weighting of 0.6. EDA therefore appears to be of slightly less importance for achieving the maximum re-identification performance. For the Slicing-DTW-Attack, for which the weights are shown in figure 6.4b, no differences in the weighting of the sensors can be detected. There, each sensor can be considered with a maximum weighting of 0.8 in order to achieve the best p@k scores.



(a) (b)

Figure 6.4.: Evaluation of weighted sensor ranking. The radar charts illustrate the best found sensor weightings for each class at $k = 1, 3, 5$ for (a) the Single-DTW-Attack, Multi-DTW-Attack and Multi-Slicing-DTW-Attack and (b) the Slicing-DTW-Attack. At each k-value, all combinations that delivered the same optimal results were included.

For the WESAD data set, each DTW attack achieves a p@1 of 1.0 using the best weights calculated on the same data set. To test the weighted sensor ranking in a realistic scenario, the best weights were first calculated on the WESAD-cGAN and WESAD-DGAN data set with 15 subjects. From this set of weights, the weight combination $\{W_{BVP} : 0.4, W_{EDA} : 0.2, W_{TEMP} : 0.2, W_{ACC} : 0.2\}$ was randomly selected, which is one of the best weights for each DTW attack. This weight combination was used to calculate the p@k scores for the standard sensor ranking compared to the weighted sensor ranking on 15 additional but different subjects of the WESAD-cGAN and WESAD-DGAN data set. Note that since the DTW attacks were performed on 15 different subjects from the previous evaluation, the p@k scores also differ slightly compared to the previous evaluation of the GAN data sets. The results are shown in table 6.6. It is noticeable that the weighted sensor ranking achieves better p@k scores than the standard sensor ranking, even for unknown but structurally similar subjects. For a $k = 1$, the p@k with standard sensor weighting increases by approximately 0.1 for the Single-DTW-Attack, the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack. The maximum p@1 of 1.0 can be achieved with the Slicing-DTW-Attack using weighted sensor ranking. From this it can be concluded that as long as the data structure is similar, an attacker can calculate sensor weights in advance and use them to improve re-identification attacks on unseen attack sets.

Data Set	k	Single		Multi		Slicing		Multi-Slicing	
		STD	WGT	STD	WGT	STD	WGT	STD	WGT
cGAN	k=1	0.135	0.237	0.121	0.219	0.956	1.000	0.218	0.315
cGAN	k=3	0.262	0.467	0.261	0.419	0.964	1.000	0.381	0.611
cGAN	k=5	0.443	0.600	0.405	0.571	0.978	1.000	0.548	0.792
DGAN	k=1	0.132	0.241	0.112	0.179	0.953	1.000	0.185	0.292
DGAN	k=3	0.304	0.421	0.282	0.421	0.972	1.000	0.344	0.513
DGAN	k=5	0.437	0.554	0.441	0.667	0.976	1.000	0.524	0.713

Table 6.6.: Evaluation of weighted sensor ranking on GAN data sets. The table compares the p@k results of the Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) on the WESAD-cGAN (cGAN) and the WESAD-DGAN (DGAN) data set. First, the best weight combinations were calculated on 15 subjects of the GAN data sets. From these, $\{W_{BVP} : 0.4, W_{EDA} : 0.2, W_{TEMP} : 0.2, W_{ACC} : 0.2\}$ was randomly selected. With this weight combination, the evaluation was carried out on 15 different subjects of the corresponding data set.

6.4. Evaluating Scalability with Synthetic GAN Data Sets

Up to this point, the DTW attacks have only been tested on data sets with 15 subjects, which is due to the limited availability of smartwatch health data sets. In order to test the scalability of the attacks on significantly larger data sets, a WESAD-cGAN and WESAD-DGAN data set with 1000 subjects each was created. Ideally, the p@k scores on the data sets with 1000 subjects should hardly differ from those with 15 subjects. The results can be found in table 6.7. For the attacks, each of the 1000 subjects was used once as a target in accordance with the simulation mode (see section 5.2). The evaluation was carried out following the evaluation pipeline, whereby the attack was only carried out for the best attack window size of the corresponding data set with 15 subjects in order to reduce the runtime. With a p@1 of 0.001 to a maximum of 0.004 for the Single-DTW-Attack, the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack, the results hardly differ from a random guess of the searched person with 0.001. However, the Slicing-DTW-Attack also achieves a p@1 of 0.934 for the WESAD-cGAN data set with 1000 subjects and a p@1 of 0.933 for the WESAD-DGAN data set. Compared to the data sets with 15 subjects, the losses are very low at 0.029 for the WESAD-cGAN and 0.026 for the WESAD-DGAN data set, which means that the Slicing-DTW-Attack can achieve good re-identification results even with large data sets.

Subjects	Single		Multi		Slicing		Multi-Slicing		Random
	C	D	C	D	C	D	C	D	
15	0.175	0.127	0.157	0.101	0.963	0.959	0.207	0.196	0.067
1000	0.002	0.001	0.002	0.002	0.934	0.933	0.004	0.001	0.001

Table 6.7.: Evaluation of scalability using large GAN data sets. The table shows the p@k results of the Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) on the WESAD-cGAN data set (C) and the WESAD-DGAN data set (D) with 15 and 1000 subjects respectively compared to a random guess (Random). The attacks on the larger data sets were performed with a $DSF = 1000$ using the standard sensor, calculated only for the best attack window size per attack of the GAN data set with 15 subjects.

6.5. Runtime Experiments

Following the evaluation of the scalability of the DTW attacks on large-scale data sets, the runtime of the attacks will now be analyzed. For large data sets in particular, the use of distance measures raises the question of whether these DTW attacks can even be carried out in a realistic scenario with the attacker's limited hardware resources. The answer to this question therefore has a strong influence on the threat level posed by the attacks. For this reason, the runtime was measured according to the attack mode (see section 5.2) for the three data set sizes with 15, 100 and 1000 subjects. The attacker has exactly one attack set at his disposal, which is compared with all subjects in the data set. All DTW attacks were calculated on the WESAD-cGAN data set with a $DSF = 1000$ and standard sensor ranking, whereby only the best parameters per attack known from the previous evaluation (best sensor combination and best attack window size) were considered.

Software and hardware environment. On the software side, Python 3.9 is used as the programming language. For the DTW requirements, the open-source implementation of the DTAIDistance library is implemented [9]. The hardware configuration for the experiments includes machines with 64 GB RAM and one core of an AMD EPYC 7551P CPU using two threads, which enables efficient calculations. In order to ensure good comparability and interpretability of the runtime results, the DTW attacks are deliberately performed on only one CPU core. To further reduce the runtime, the implemented parallelization can be used, which makes it possible to perform the DTW distance calculation between the attack set and the data of one subject of the data set to be carried out in parallel on all available threads per subject. Equivalently, the ranking can also be carried out in parallel. The actual runtime therefore depends heavily on the hardware resources of an attacker. Runtime estimates for the DTW attacks on multiple cores can be derived from the results on one core, even if, *runtime/threads* is not 100% achieved in reality.

Scope	Single			Multi			Slicing			Multi-Slicing		
	15	100	1000	15	100	1000	15	100	1000	15	100	1000
Attack	1.073	8.049	66.243	1.024	7.458	62.415	1.117	7.813	71.215	2.399	16.001	154.363
Ranking	0.001	0.009	0.155	0.001	0.014	0.161	0.001	0.006	0.158	0.001	0.006	0.162
Total	1.074	8.058	66.398	1.025	7.472	62.576	1.118	7.819	71.373	2.400	16.007	154.525

Table 6.8.: Runtime results. The table shows the runtime results for the Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) on the WESAD-cGAN data set with 15, 100 and 1000 subjects. The runtime is given in seconds, first separately for the attack and ranking process and then in total. All results were calculated according to the attack mode on one core of an AMD EPYC 7551P CPU.

Runtime results. The runtime results are presented in table 6.8, where these are shown separately for attack and ranking on the one hand and in total on the other. Overall, it is noticeable that the ranking has hardly any influence on the runtime. For the best-performing Slicing-DTW-Attack with 15 subjects it is 0.001 seconds and with 1000 subjects it is 0.158 seconds. As the ranking is identical for all attacks and multiple distances are handled during the attack before the distance scores are stored, the ranking runtime is roughly the same for all four DTW attacks. The calculation of the DTW distances is primarily decisive for the total runtime. The Multi-DTW-Attack requires

the shortest total runtime, with approx. 1 second for 15 subjects and approx. 63 seconds for 1000 subjects. The Single-DTW-Attack requires around 66 seconds for 1000 subjects, followed by the Slicing-DTW-Attack with around 71 seconds. In contrast, the Multi-Slicing-DTW-Attack requires a significantly longer runtime. This takes around 2 seconds for 15 subjects and 155 seconds for 1000 subjects. The results show that the runtime for the attacks themselves increases linearly according to the number of subjects. However, the ranking runtime does not increase linearly. With very large data sets, this runtime could become more significant. Overall, and especially for the Slicing-DTW-Attack, which achieves the best p@k re-identification results, the attacks can be carried out in a few minutes even for the largest data set with 1000 subjects, which can be further greatly reduced by parallelization and the complexity reduction methods shown in section 4.4.

6.6. Evaluating Privacy vs. Usability

The privacy vs. usability case study is intended to show the extent to which the re-identification risk of DTW attacks can be reduced by adding noise according to a Laplace distribution without having a strong negative impact on the usability of the data for other use cases. Usability is reflected in this context by a stress detection application. For this purpose, the re-identification and stress detection results were calculated for different noise levels, which are determined by the noise multiplier (NP). The exact experimental setup was described in detail in section 5.5. Figure 6.5 shows the curve of the p@1 results for the DTW attacks and the f1 score for the stress detection using the NPs 0 (no noise) to 15 (strongest noise). Since even very small NPs are expected to result in significant performance losses, the NPs between 0 and 1 were considered with a step size of 0.1. From 1, the step size is 1. All results can also be found in table 5 in the appendix.

Looking at the stress detection in a first step, it is noticeable that the f1 score starting at 0.88 without adding noise up to 0.865 with a $NP = 0.5$ is only marginally affected. The f1 scores subsequently decrease relatively rapidly up to an $NP = 2$ and then reach a brief plateau with a f1 score of 0.715, until they decrease continuously from an $NP = 3$ and converge to a f1 score of approx. 0.3. In comparison, the p@1 scores of the DTW attacks decrease significantly faster. Due to the very low p@1 scores of the Single-DTW-Attack ($p@1 = 0.301$) and Multi-DTW-Attack ($p@1 = 0.309$) for the unnoised WESAD data set, these attacks only pose a relatively small privacy threat, which already drops to 0.259 and 0.268 respectively with a $NP = 0.5$. With stronger noise, the p@1 scores continuously converge to a random re-identification, which would achieve a p@1 of 0.067. The Multi-Slicing-DTW-Attack starts with a p@1 of 0.606 for the original WESAD data set. Already with a $NP = 2$, the p@1 crosses the curve of the Single-DTW-Attack and Multi-DTW-Attack and then converges slightly faster to the random score. The Slicing-DTW-Attack, which was able to achieve the most threatening results with a p@1 of 1.0 on the WESAD data set, shows a very sharp fall in p@k when small noise values are added. Up to a $NP = 2$ with a p@1 of 0.22, the loss is almost linear. After this, the Slicing-DTW-Attack also continuously converges to the random value. Due to the described plateau, the stress detection f1 scores fall significantly slower with increasing NP than the p@1 scores of the re-identification attacks, which means that in this case study, acceptable stress detection results can still be achieved at a $NP = 2$ with a f1 score of 0.715, while the Slicing-DTW-Attack with a p@1 of 0.22 can hardly be used for re-identification.

However, whether these findings can be transferred to other usability applications must be assessed on a case-by-case basis.

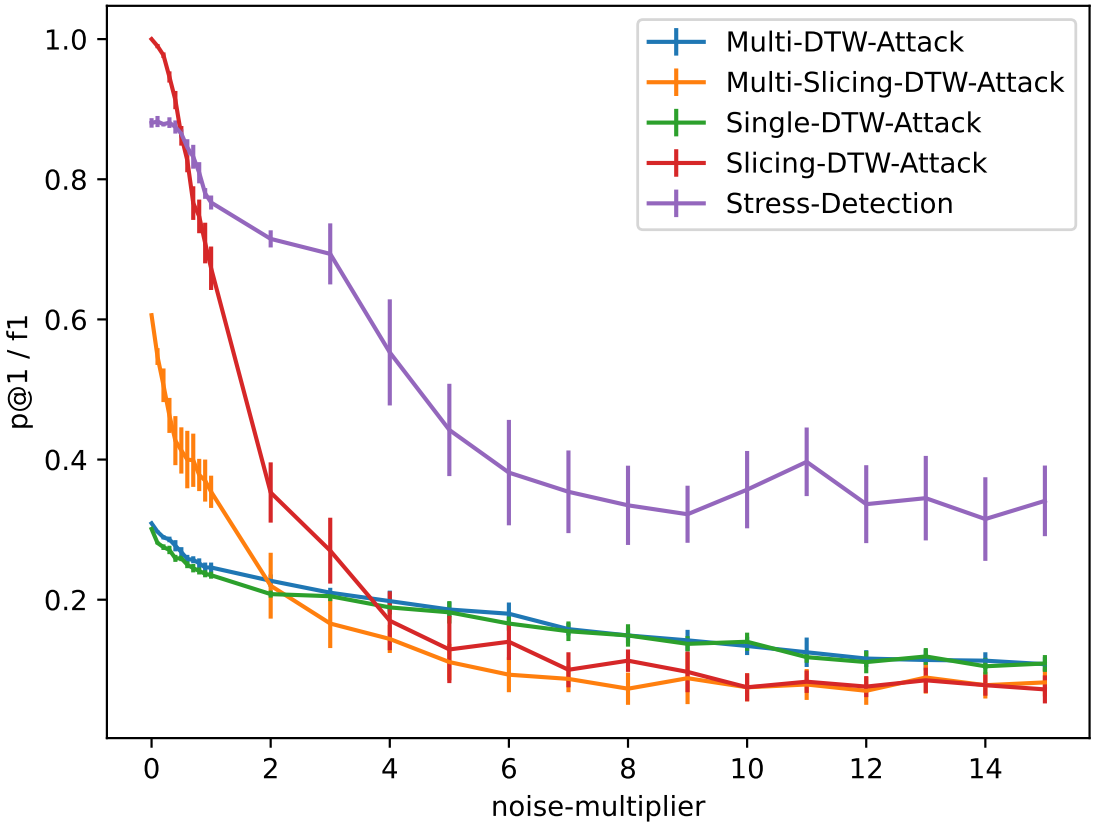


Figure 6.5.: Evaluation privacy vs. usability. The line graph shows the average p@1 results for the Single-DTW-Attack, Multi-DTW-Attack, Slicing-DTW-Attack and Multi-Slicing-DTW-Attack on the noisy WESAD data set for different noise multipliers in comparison to the stress detection f1 scores. Noise multipliers in the range 0 to 15 are considered. The step size between 0 and 1 is 0.1 and from 1 the step size equals 1. In addition, the standard deviation over 10 runs is shown by vertical lines.

7. Discussion

After the results of the different experiments were presented in chapter 6, they are to be contextualized and interpreted in this chapter. To this end, section 7.1 first presents the procedures taken to ensure validity and reliability. Section 7.2 then provides answers to the five research questions presented in chapter 1. Finally, section 7.3 outlines the limitations of this thesis.

7.1. Controlling Validity and Reliability

Validity. In this work, a modular framework for DTW-based re-identification attacks was developed, which enables DTW attacks to be performed on different data sets with a selection of complexity-reducing methods. To ensure the validity of the experiments, a four-stage evaluation pipeline was established, which evaluates all attacks with the corresponding parameter specification in a uniform rank-based manner. The choice of the quality criterion p@k appears to be advantageous compared to others for the interpretation of the results, as p@k also allows a reduction in the number of candidates through the choice of the k parameter. This reduction also represents a privacy threat, but is not taken into account in related work. Since no training and test data is required for the DTW attacks, no train-test leakage can occur and thus no overfitting of the attacks to the corresponding use case can appear. The limited availability of large-scale smartwatch health data sets was addressed by generating two synthetic GAN data sets. Due to the modularity of the attack framework, further attacks and data sets can be integrated very easily, which enables good generalizability for other data and use cases. In principle, the framework can be used for all personal multi-modal time series data with minor adjustments in the respective data preparation, since the findings from the stages of the evaluation framework can generally only be taken into account for other smartwatch health data sets.

Reliability. All DTW attacks presented are deterministic, which means that the results can be exactly reproduced. The corresponding Python code for the attack framework is available in a public repository.[8] The selected data sets can also be downloaded publicly or found in the repository. The results of the privacy vs. usability case study were given as a mean value over 10 runs due to the influence of random noise values. The corresponding results were added to the repository, but can also be recalculated with only a slight deviation due to the relatively small standard deviation.

7.2. A Response to the Research Questions

RQ1: How severe is the actual threat level that is reached in the attack scenario? In order to be able to classify the results correctly, they are first compared with the probability of finding the correct target by chance in the top-k results. This probability can be derived from the formula $p = k/N$, where k is the top-k value for the p@k and N is the number of possible ranks, which here corresponds to the number of possible subjects. The WESAD data set contains health

[8] `https://github.com/tobiasschreieder/smartwatch-dtw-attack-scalable`

data from 15 real subjects, which means that $N = 15$. The probability that the ranking of the subjects is determined by random guessing and that the target searched for is in first place ($k = 1$) would therefore be $p = 1/15 \approx 0.067$. With p@1 scores of 0.301 for the Single-DTW-Attack, 0.309 for the Multi-DTW-Attack, 1.0 for the Slicing-DTW-Attack and 0.661 for the Multi-Slicing-DTW-Attack, all attacks perform significantly better than random guessing for the WESAD data set. The Slicing-DTW-Attack was able to correctly identify all 15 subjects. These considerable advantages over the random chances put the loss of privacy from our attack into perspective, as the results of random guessing would mean perfect privacy instead.

For the Single-DTW-Attack, the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack, the re-identification performance can be further improved by applying the weighted sensor ranking. The experiments were therefore carried out with the two GAN data sets, for which a significant improvement in p@1 was achieved (≈ 0.1) compared to the standard sensor ranking.

Another devastating factor for privacy is the ability to reduce the candidate space by the DTW attacks. If an attack reaches the maximum p@k of 1.0 for a given k, this guarantees the inclusion of the target in these top k results. Therefore, all other candidate samples except those with the highest k-value can be excluded from consideration. This can drastically reduce the search space and thus also the difficulty of re-identification, for instance through additional linked information.

However, the 15 subjects of the WESAD data set only form a very small data basis from which the re-identification performance for large-scale data sets cannot be derived. Therefore, a synthetic cGAN and a DGAN data set with 1000 subjects each were generated. While the Single-DTW-Attack, the Multi-DTW-Attack and the Multi-Slicing-DTW-Attack only achieve p@1 scores, which hardly differ from random guessing, a p@1 score of 0.934 (WESAD-cGAN) and 0.933 (WESAD-DGAN) can still be achieved with the Slicing-DTW-Attack. This shows that the major privacy threat comes from the Slicing-DTW-Attack, which can correctly identify the target even among data sets with 1000 subjects in over 93% of cases, where a single attack can be carried out in a few seconds with a standard modern CPU.

RQ2: Are synthetic data sets generated by generative adversarial networks suitable for determining the scalability of re-identification attacks? The aim of a GAN is to create an arbitrarily large data set by entering a usually relatively small data set, which retains the basic characteristics of the data but does not copy individual subjects from the original data set. In this work, a cGAN and DGAN data set was used, which proved to be the most suitable for the WESAD data set in the evaluation by Wenzlitschke [11]. Even though the basic correlation structure of the WESAD data set is well reproduced by the two GANs, clear differences between the GAN data sets and the WESAD data set were found in section 5.1.2. The differences are primarily due to a recognizably greater variance within a sensor signal and smaller DTW distances between the subjects. For this reason, re-identification with the GAN data sets generally appears to be more difficult, which is also reflected in the significantly lower p@1 scores for the Single-DTW-Attack, Multi-DTW-Attack and Multi-Slicing-DTW-Attack. Nevertheless, the Slicing-DTW-Attack achieves p@1 scores of over 0.93 for both GAN data sets, with 15 as well as with 1000 subjects. Therefore, the GAN data sets seem to underestimate the actual re-identification risk for original large-scale health data sets and can be regarded as a conservative estimate.

RQ3: To what extent are the re-identification attacks tailored to precisely this task? Can they be adapted to other data sets? In general, the four DTW attacks are transferable to other similar cases, based on the description and process provided in this thesis. This is especially true for the naive standard sensor ranking approach, while the findings of the sensor weighted approach may not always work depending on the new data. However, as weights can be recalculated comparatively quickly for other sensors, this does not represent a major obstacle. Nevertheless, there are many tasks where the knowledge of the underlying data characteristics allows a similar exploitation. The obtained results can also be used as a pre-trained attack model with already optimized parameters through the outlined experiments. Overall, the research should convey the idea that similarity attacks can pose a threat to time series health data, and show that some modalities are more threatened than others by evaluating sensor combinations or sensor weights.

RQ4: Are there also beneficial use cases for such similarity searches? Yes, for example, linking similar patients or subjects to quickly improve personalized health applications would be one possible beneficial use case. Furthermore, such tasks could include stress detection, but also the tuning of a person's medication dosage. By finding similar subjects in an existing database, a new subject could benefit from the existing data when adapting these applications to their personal but similar needs and health situation. Consequently, there is also a favorable outcome from the presented similarity-based DTW attacks.

RQ5: What are possible defense mechanisms that are more appropriate than de-identification? How much do these affect the usability of the data for other use cases? The evaluation of DTW attacks has shown that it is not sufficient to remove the identifying meta-data of a data sample to protect against the attacks. The reason for this is the fact that the similarity is calculated in relation to the data itself. Instead, the similarities between the data sample and the captured target sample would have to be hidden by directly breaking or hiding these links. A general possibility would be to only allow a data sample to be collected if there are already enough similar samples of other subjects in the data, making it difficult to distinguish between them. This approach would be comparable to the anonymity measure k-anonymity [98]. As described in detail in section 5.5, the application of k-anonymity to time series data is not trivial and often unsuitable. Depending on the available data, this approach could potentially remove a large amount of threatened but needed samples from such data sets.

For this reason, adding random noises according to a Laplace distribution was considered a more suitable method in this work. Different noise multipliers (NP), which reflect the level of the noise, were tested. The evaluation of the privacy vs. usability case study showed that even with a relatively small $NP = 2$, the p@1 of the best-performing Slicing-DTW-Attack on the noisy WESAD data set drops to 0.353, while a stress detection f1 score of 0.715 can still be achieved. Whether this procedure can be applied to other data sets and use cases must be examined individually for each case, which is why no generally valid privacy guarantee can be derived.

A theoretical privacy guarantee, on the other hand, could be achieved by applying differential privacy, which was presented by Dwork [102]. Using differential privacy, selective amounts of random noise are added to the data so that the individual samples can no longer be distinguished from each other. The difference to the adding of noise in the outlined case study is that a privacy

budget ϵ is additionally specified, which determines the guaranteed level of privacy. Apart from that with differential privacy, the entire data set is never published. An attacker could only receive differential private query aggregates, which would differ slightly from query to query. This approach would change the entire attack scenario, as an attacker could not gain access to the data owner's entire database. Usability could also suffer greatly as certain use cases, such as publishing data sets for research purposes, would not be possible.

7.3. Limitations

The primary limitation of this work is the available data for testing the DTW attacks. In this context, the owners of smart device health data, which are currently mostly the responsible companies themselves, are a decisive factor. Consequently, there is a dependency on study data from the public domain, whose participant numbers are far smaller than those of users of e.g. smartwatches in everyday life. The publicly accessible WESAD data set used also only includes a very small number of 15 test subjects. However, compared to other data sets, it offers the largest number of test subjects with a wide range of sensor modalities, which means that there is no need to limit re-identification attacks to a single signal. With respect to this duality, there are no other larger and less specialized data sets that exhibit the same qualities. To address the size issue, two synthetic GAN data sets were generated to test the scalability of the attacks. The evaluation of the DTW attacks on the different data sets has shown that the GAN data sets can be used for a conservative estimation of the re-identification risk on large-scale data sets. Nevertheless, a verification of these results and a calculation of the actual re-identification risk on a real large-scale data set would be beneficial and preferable to the GAN results.

A second restriction lies in the nature of similarity-based re-identification attacks. The attack scenario in section 4.1 demonstrates that an attacker attempts to identify the device owner in the data owner's data set using the captured attack sample. To do this, a distance between the attack set and the signals of the subject is calculated for each subject. This allows the attacker to determine which subject most likely corresponds to the device owner. However, the attacker cannot determine whether the device owner is included in the data set at all. The attacker therefore needs the information about the inclusion of the device owner in the data set in advance. To change this, an additional stage would have to be built into the attack framework that classifies whether the device owner is present in the data set. This would be possible, for example, by setting a distance threshold. Should the minimum distance score calculated across all subjects exceed the threshold score and the similarity between the most likely subject of the data set and the device owner is therefore not high enough, the device owner is considered not to be included in the data set.

8. Conclusion & Future Work

With the increasing popularity of smartwatches on the market today, the amount of personal health data that is collected and shared with the respective companies is continuously increasing. Although this data collection usually contributes to the improvement and development of smart health services designed to support users, data collection can also threaten users' privacy. The potential privacy risk that exists when working with health data from IoT and smart devices is highlighted by the four proposed similarity-based re-identification attacks in this work. Cases of identity inference are particularly relevant because, as in the example scenario with smartwatch data, the de-identification of data is still common practice.

In this thesis, a novel modular attack framework was presented, in which four different re-identification attacks are currently integrated, all of which use DTW as a distance measure in different ways. The framework offers the possibility of integrating diverse data sets through the standardized data preprocessing pipeline. Furthermore, various measures such as downsampling, DBA and PCA were integrated into the framework to be able to significantly reduce the complexity of the calculation where necessary. In order to evaluate the DTW attacks uniformly, a four-stage rank-based evaluation pipeline was developed, which can evaluate the re-identification risk for each attack parameter specification using the quality criterion p@k. The data basis for the experiments is provided by the WESAD data set, which comprises a total of 15 subjects and the four sensors modalities BVP, EDA, TEMP and ACC. Particularly the Slicing-DTW-Attack, which achieved a p@1 score of 1.0 for the WESAD dataset and was therefore able to correctly identify each of the 15 subjects, proved to be especially threatening. To test the scalability of the attacks for large-scale data sets, a cGAN and DGAN data set with 1000 subjects each were generated from the existing WESAD data set. It was shown that the Slicing-DTW-Attack could also achieve a p@1 of over 0.93 for the two synthetic GAN data sets with 1000 subjects. Considering the fact that the Slicing-DTW-Attack can be performed with only one CPU core within approx. 71 seconds, since parallelization can be applied without any problems, the privacy risk is clearly recognizable.

The results of the Slicing-DTW-Attack make it clear that de-identification alone is not sufficient to prevent re-identification, as the DTW attacks presented use only the signal data itself as a basis. Adding random noise to the data has been proven to be more suitable. For this purpose, a case study was carried out which compares the curve of the p@k scores for the DTW attacks with the f1 scores of a stress detection application for different noise levels. A trade-off was revealed in which the p@1 score of the Slicing-DTW-Attack on the noisy WESAD data set drops to 0.353, while the stress detection still achieves a f1 score of 0.715 (-0.165). However, a general transferability of these findings to other data sets and use cases cannot be derived from this.

To reduce the most important limiting factor, future works should focus primarily on obtaining more data. This would allow the scalability and relevance of DTW attacks not only to be estimated using synthetic data, but also to be evaluated on actual large-scale data sets. Furthermore, it should be investigated to what extent an attacker can recognize whether the target is even contained in the data owner's data set by using thresholds or other suitable methods. This would significantly increase the threat level far beyond the current attack scenario.

Bibliography

[1] Lucas Lange et al. *Privacy at Risk: Exploiting Similarities in Health Data for Identity Inference*. 2023. arXiv: 2308.08310 [cs.CR].

[2] Prerit Datta, Akbar Siami Namin, and Moitrayee Chatterjee. "A Survey of Privacy Concerns in Wearable Devices". In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 4549–4553. DOI: 10.1109/BigData.2018.8622110.

[3] F. Tenzer. *Prognose zum Absatz von Smartwatches weltweit in den Jahren 2022 bis 2027*. Statista. 2004. URL: https://de.statista.com/statistik/daten/studie/500483/umfrage/prognose-zum-weltweiten-absatz-von-smartwatches/ (visited on 02/21/2024).

[4] Amit Kumar Sikder et al. "A Survey on Sensor-Based Threats and Attacks to Smart Devices and Applications". In: *IEEE Communications Surveys & Tutorials* 23.2 (2021), pp. 1125–1159. DOI: 10.1109/COMST.2021.3064507.

[5] Claus-Peter Ernst and Alexander Ernst. "The Influence of Privacy Risk on Smartwatch Usage". In: *AMCIS 2016 Proceedings* (2016).

[6] Fitbit International Limited and Fitbit LLC. *Fitbit Privacy Policy*. 2023. URL: https://www.fitbit.com/global/us/legal/privacy-policy (visited on 07/07/2023).

[7] Fitbit Health Solutions. *Research Pledge*. 2023. URL: https://healthsolutions.fitbit.com/research-pledge/ (visited on 07/07/2023).

[8] Khaled El Emam et al. "A Systematic Review of Re-Identification Attacks on Health Data". In: *PLoS ONE* 6.12 (2011). DOI: 10.1371/journal.pone.0028071.

[9] Wannes Meert et al. *DTAIDistance*. 2020. DOI: 10.5281/zenodo.7158824. URL: https://zenodo.org/records/7158824 (visited on 12/04/2023).

[10] Philip Schmidt et al. "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection". In: *ACM ICMI '18*. 2018, pp. 400–408. DOI: 10.1145/3242969.3242985.

[11] Nils Wenzlitschke. "Privacy-Preserving Smartwatch Health Data Generation For Stress Detection Using GANs". Masterthesis. Leipzig: University Leipzig, 2023. URL: https://dbs.uni-leipzig.de/files/study/theses/2023/pdf/Thesis_Nils_Wenzlitschke.pdf (visited on 12/04/2023).

[12] Lukasz Piwek et al. "The Rise of Consumer Health Wearables: Promises and Barriers". In: *PLOS Medicine* 13 (2016). DOI: 10.1371/journal.pmed.1001953.

[13] Christine King and Majid Sarrafzadeh. "A Survey of Smartwatches in Remote Health Monitoring". In: *Journal of Healthcare Informatics Research* 2 (2018). DOI: 10.1007/s41666-017-0012-7.

[14] Milad Asgari Mehrabadi et al. "Sleep Tracking of a Commercially Available Smart Ring and Smartwatch Against Medical-Grade Actigraphy in Everyday Settings: Instrument Validation Study". In: *JMIR mHealth and uHealth* 8 (2020). DOI: 10.2196/20465.

[15] Liqiong Chang et al. "SleepGuard: Capturing Rich Sleep Information Using Smartwatch Sensing Data". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.3 (2018). DOI: 10.1145/3264908.

[16] Dimitri Kraft, Karthik Srinivasan, and Gerald Bieber. "Deep Learning Based Fall Detection Algorithms for Embedded Systems, Smartwatches, and IoT Devices Using Accelerometers". In: *Technologies* 8.4 (2020). DOI: 10.3390/technologies8040072.

[17] Taylor Mauldin et al. "SmartFall: A Smartwatch-Based Fall Detection System Using Deep Learning". In: *Sensors* 18 (2018), p. 3363. DOI: 10.3390/s18103363.

[18] Maurizio Garbarino et al. "Empatica E3 - A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition". In: (2015), pp. 39–42. DOI: 10.1109/MOBIHEALTH.2014.7015904.

[19] Cameron McCarthy et al. "Validation of the Empatica E4 wristband". In: *2016 IEEE EMBS International Student Conference (ISC)*. 2016, pp. 1–4. DOI: 10.1109/EMBSISC.2016.7508621.

[20] Angela A. T. Schuurmans et al. "Validity of the Empatica E4 Wristband to Measure Heart Rate Variability (HRV) Parameters: A Comparison to Electrocardiography (ECG)". In: *J. Med. Syst.* 44.11 (2020). DOI: 10.1007/s10916-020-01648-w.

[21] Luca Menghini et al. "Stressing the accuracy: Wrist-worn wearable sensor validation over different conditions". In: *Psychophysiology* 56.11 (2019). DOI: 10.1111/psyp.13441.

[22] Michael E. Dawson, Anne M. Schell, and Diane L. Filion. "The Electrodermal System". In: *Handbook of Psychophysiology*. Ed. by John T. Cacioppo, Louis G. Tassinary, and Gary G.Editors Berntson. Cambridge University Press, 2007, pp. 159–181. DOI: 10.1017/CBO9780511546396.007.

[23] Angela Scarpa Scerbo et al. "A Major Effect of Recording Site on Measurement of Electrodermal Activity". In: *Psychophysiology* 29.2 (1992), pp. 241–246. DOI: 10.1111/j.1469-8986.1992.tb01693.x.

[24] Ming-Zher Poh, Nicholas C. Swenson, and Rosalind W. Picard. "A Wearable Sensor for Unobtrusive, Long-Term Assessment of Electrodermal Activity". In: *IEEE Transactions on Biomedical Engineering* 57 (2010), pp. 1243–1252. DOI: 10.1109/tbme.2009.2038487.

[25] Marieke van Dooren, J.J.G. (Gert-Jan) de Vries, and Joris H. Janssen. "Emotional sweating across the body: Comparing 16 different skin conductance measurement locations". In: *Physiology & Behavior* 106.2 (2012), pp. 298–304. DOI: 10.1016/j.physbeh.2012.01.020.

[26] Laurie McCorry. "Physiology of the Autonomic Nervous System". In: *American journal of pharmaceutical education* 71 (2007), p. 78. DOI: 10.5688/aj710478.

[27] Giorgos Giannakakis et al. "Review on Psychological Stress Detection Using Biosignals". In: *IEEE Transactions on Affective Computing* 13.01 (2022). DOI: 10.1109/TAFFC.2019.2927337.

[28] Rubén Usamentiaga et al. "Infrared Thermography for Temperature Measurement and Non-Destructive Testing". In: *Sensors* 14.7 (2014), pp. 12305–12348. DOI: 10.3390/s140712305.

[29] Ethan R. Nadel, Robert W. Bullard, and J. A. Stolwijk. "Importance of skin temperature in the regulation of sweating." In: *Journal of Applied Physiology* 31.1 (1971), pp. 80–87. DOI: 10.1152/jappl.1971.31.1.80.

[30] Bruce S. McEwen and Eliot Stellar. "Stress and the Individual: Mechanisms Leading to Disease". In: *Archives of Internal Medicine* 153.18 (1993), pp. 2093–2101. DOI: 10.1001/archinte.1993.00410180039004.

[31] Sheldon Cohen, Denise Janicki-Deverts, and Gregory E. Miller. "Psychological Stress and Disease". In: *JAMA* 298.14 (2007), pp. 1685–1687. DOI: 10.1001/jama.298.14.1685.

[32] Andrew Steptoe and Mika Kivimäki. "Stress and Cardiovascular Disease: An Update on Current Knowledge". In: *Annual review of public health* 34 (2013), pp. 337–354. DOI: 10.1146/annurev-publhealth-031912-114452.

[33] Suzanne C. Segerstrom and E. Miller Gregory. "Psychological stress and the human immune system: a meta-analytic study of 30 years of inquiry". In: *Psychological bulletin* 130.4 (2004), pp. 601–630. DOI: 10.1037/0033-2909.130.4.601.

[34] H.M. van Praag. "Can stress cause depression?" In: *Progress in Neuro-Psychopharmacology and Biological Psychiatry* 28.5 (2004), pp. 891–907. DOI: 10.1016/j.pnpbp.2004.05.031.

[35] Alexandros M. Heraclides et al. "Work Stress, Obesity and the Risk of Type 2 Diabetes: Gender-Specific Bidirectional Effect in the Whitehall II Study". In: *Obesity* 20.2 (2012), pp. 428–433. DOI: 10.1038/oby.2011.95.

[36] Analava Mitra. "Diabetes and Stress: A Review". In: *Ethno-Med* 2282657 (2008), pp. 131–135. DOI: 10.1080/09735070.2008.11886324.

[37] Mustafa al'Absi, Annie T. Ginty, and William R. Lovallo. "Neurobiological mechanisms of early life adversity, blunted stress reactivity and risk for addiction". In: *Neuropharmacology* 188 (2021), p. 108519. DOI: 10.1016/j.neuropharm.2021.108519.

[38] A Angeli et al. "The overtraining syndrome in athletes: A stress-related disorder". In: *Journal of endocrinological investigation* 27 (2004), pp. 603–12. DOI: 10.1007/BF03347487.

[39] Martin Gjoreski et al. "Continuous Stress Detection Using a Wrist Device: In Laboratory and Real Life". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. UbiComp '16. Heidelberg, Germany: Association for Computing Machinery, 2016, pp. 1185–1193. DOI: 10.1145/2968219.2968306.

[40] Walter B. Cannon. "The Wisdom of the Body". In: *Nature* 133.3351 (1934), pp. 82–82. DOI: 10.1038/133082a0.

[41] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. "Automatic speech emotion recognition using recurrent neural networks with local attention". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 2227–2231. DOI: 10.1109/ICASSP.2017.7952552.

[42] Panagiotis Tzirakis et al. "End-to-End Multimodal Emotion Recognition Using Deep Neural Networks". In: *IEEE Journal of Selected Topics in Signal Processing* 11.8 (2017), pp. 1301–1309. DOI: 10.1109/jstsp.2017.2764438.

[43] Alberto de Santos Sierra et al. "A Stress-Detection System Based on Physiological Signals and Fuzzy Logic". In: *IEEE Transactions on Industrial Electronics* 58.10 (2011), pp. 4857–4865. DOI: 10.1109/TIE.2010.2103538.

[44] J.A. Healey and R.W. Picard. "Detecting stress during real-world driving tasks using physiological sensors". In: *IEEE Transactions on Intelligent Transportation Systems* 6.2 (2005), pp. 156–166. DOI: 10.1109/TITS.2005.848368.

[45] Pekka Siirtola. "Continuous Stress Detection Using the Sensors of Commercial Smartwatch". In: *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. UbiComp/ISWC '19 Adjunct. Association for Computing Machinery, 2019, pp. 1198–1201. DOI: 10.1145/3341162.3344831.

[46] Russell Li and Zhandong Liu. "Stress detection using deep neural networks". In: *BMC Medical Informatics and Decision Making* 20 (2020), pp. 285–294. DOI: 10.1186/s12911-020-01299-4.

[47] Manuel Gil-Martin et al. "Human Stress Detection With Wearable Sensors Using Convolutional Neural Networks". In: *IEEE Aerospace and Electronic Systems Magazine* 37.1 (2022), pp. 60–70. DOI: 10.1109/MAES.2021.3115198.

[48] Arturo de Souza et al. "MoStress: a Sequence Model for Stress Classification". In: *2022 International Joint Conference on Neural Networks (IJCNN)*. 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9892953.

[49] Eda Eren and Tuğba Selcen Navruz. "Stress Detection with Deep Learning Using BVP and EDA Signals". In: *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. 2022, pp. 1–7. DOI: 10.1109/HORA55278.2022.9799933.

[50] Lucas Lange, Borislav Degenkolb, and Erhard Rahm. "Privacy-Preserving Stress Detection Using Smartwatch Health Data". In: *4. Interdisciplinary Privacy & Security at Large Workshop*. INFORMATIK 2023. 2023. DOI: 10.18420/inf2023_66.

[51] Sana Imtiaz et al. "Synthetic and Private Smart Health Care Data Generation using GANs". In: *2021 International Conference on Computer Communications and Networks (ICCCN)*. 2021, pp. 1–7. DOI: 10.1109/ICCCN52240.2021.9522203.

[52] Ian Goodfellow et al. *Generative Adversarial Networks*. 2014. DOI: 10.48550/arXiv.1406.2661.

[53] Ian Goodfellow et al. "Generative Adversarial Networks". In: *Commun. ACM* 63.11 (2020), pp. 139–144. DOI: 10.1145/3422622.

[54] Antonia Creswell et al. "Generative Adversarial Networks: An Overview". In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 53–65. DOI: 10.1109/MSP.2017.2765202.

[55] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. 2014. arXiv: 1411.1784 [cs.LG].

[56] Zinan Lin et al. "Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions". In: *Proceedings of the ACM Internet Measurement Conference*. IMC '20. Association for Computing Machinery, 2020, pp. 464–483. DOI: 10.1145/3419394.3423643.

[57] Jane Henriksen-Bulmer and Sheridan Jeary. "Re-identification attacks—A systematic literature review". In: *International Journal of Information Management* 36.6, Part B (2016), pp. 1184–1192. DOI: 10.1016/j.ijinfomgt.2016.08.002.

[58] Reza Shokri et al. "Membership Inference Attacks against Machine Learning Models". In: *S&P 2017*. IEEE. 2017. DOI: 10.1109/SP.2017.41.

[59] Toni Giorgino. "Computing and Visualizing Dynamic Time Warping Alignments in R: The Dtw Package". In: *Journal of Statistical Software* 31.7 (2009), pp. 1–24. DOI: 10.18637/jss.v031.i07.

[60] T. K. Vintsyuk. "Speech discrimination by dynamic programming". In: *Cybernetics* 4.1 (1968), pp. 52–57. DOI: 10.1007/BF01074755.

[61] V. M. Velichko and N. G. Zagoruyko. "Automatic recognition of 200 words". In: *International Journal of Man-Machine Studies* 2.3 (1970), pp. 223–234. DOI: 10.1016/S0020-7373(70)80008-6.

[62] Hiroaki Sakoe and Seibi Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (1978), pp. 43–49. DOI: 10.1109/TASSP.1978.1163055.

[63] Eamonn Keogh and Chotirat Ratanamahatana. "Exact indexing of dynamic time warping". In: *Knowledge and Information Systems* 7 (2005), pp. 358–386. DOI: 10.1007/s10115-004-0154-9.

[64] Romain Tavenard. *An introduction to Dynamic Time Warping*. https://rtavenar.github.io/blog/dtw.html. 2021. (Visited on 12/19/2023).

[65] Mathew J Owens and Jonathan D Nichols. "Using in situ solar-wind observations to generate inner-boundary conditions to outer-heliosphere simulations – I. Dynamic time warping applied to synthetic observations". In: *Monthly Notices of the Royal Astronomical Society* 508.2 (2021), pp. 2575–2582. DOI: 10.1093/mnras/stab2512.

[66] Thanawin Rakthanmanon et al. "Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping". In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. Beijing, China: Association for Computing Machinery, 2012, pp. 262–270. DOI: 10.1145/2339530.2339576.

[67] F. Itakura. "Minimum prediction residual principle applied to speech recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.1 (1975), pp. 67–72. DOI: 10.1109/TASSP.1975.1162641.

[68] *5 Dynamic Time Warping (DTW) Libraries in Python With Examples*. 2023. URL: https://forecastegy.com/posts/dynamic-time-warping-dtw-libraries-python-examples/ (visited on 12/23/2023).

[69] John Brownstein, Christopher Cassa, and Kenneth Mandl. "No Place to Hide — Reverse Identification of Patients from Published Maps". In: *The New England journal of medicine* 355 (2006), pp. 1741–2. DOI: 10.1056/NEJMc061891.

[70] Lisle A. Stalter. "Case Synopsis: Southern Illinoisan v. Illinois Department of Public Health". In: *The Supreme Court of the State of Illinois* 7 (2006), pp. 5–6. URL: https://www.isba.org/sites/default/files/sections/newsletter/%20April%202006_0.pdf (visited on 01/11/2024).

[71] Mulagala Sandhya and Munaga V. N. K. Prasad. "Biometric Template Protection: A Systematic Literature Review of Approaches and Modalities". In: *Biometric Security and Privacy: Opportunities & Challenges in The Big Data Era*. Cham: Springer International Publishing, 2017, pp. 323–370. DOI: 10.1007/978-3-319-47301-7_14.

[72] Xingbo Dong, Zhe Jin, and Andrew Teoh Beng Jin. "A Genetic Algorithm Enabled Similarity-Based Attack on Cancellable Biometrics". In: *10th IEEE International Conference on Biometrics Theory, Applications and Systems, BTAS 2019*. IEEE, 2019. DOI: 10.1109/BTAS46853.2019.9185997.

[73] Ziyuan Yang et al. "Two novel style-transfer palmprint reconstruction attacks". In: *Appl. Intell.* 53.6 (2023), pp. 6354–6371. DOI: 10.1007/s10489-022-03862-0.

[74] Dinusha Vatsalan et al. "Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges". In: *Handbook of Big Data Technologies*. Springer, 2017, pp. 851–895. DOI: 10.1007/978-3-319-49340-4_25.

[75] Anushka Vidanage et al. "A Graph Matching Attack on Privacy-Preserving Record Linkage". In: *International Conference on Information and Knowledge Management*. ACM, 2020, pp. 1485–1494. DOI: 10.1145/3340531.3411931.

[76] Chris Culnane, Benjamin IP Rubinstein, and Vanessa Teague. "Vulnerabilities in the use of similarity tables in combination with pseudonymisation to preserve data privacy in the UK Office for National Statistics' Privacy-Preserving Record Linkage". In: *arXiv preprint arXiv:1712.00871* (2017).

[77] Jingyu Hua, Zhenyu Shen, and Sheng Zhong. "We Can Track You if You Take the Metro: Tracking Metro Riders Using Accelerometers on Smartphones". In: *IEEE Trans. Inf. Forensics Secur.* 12.2 (2017), pp. 286–297. DOI: 10.1109/TIFS.2016.2611489.

[78] Nazir Saleheen et al. "WristPrint: Characterizing User Re-Identification Risks from Wrist-Worn Accelerometry Data". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. CCS '21. 2021, pp. 2807–2823. DOI: 10.1145/3460120.3484799.

[79] Ke Ching and Manmeet (Mandy) Mahinderjit Singh. "Wearable Technology Devices Security and Privacy Vulnerability Analysis". In: *International Journal of Network Security & Its Applications* 8 (2016), pp. 19–30. DOI: 10.5121/ijnsa.2016.8302.

[80] Jiajia Liu and Wen Sun. "Smart Attacks against Intelligent Wearables in People-Centric Internet of Things". In: *IEEE Communications Magazine* 54.12 (2016), pp. 44–49. DOI: 10.1109/MCOM.2016.1600553CM.

[81] Felton Blow, Yen-Hung Hu, and Mary Hoppa. "A study on vulnerabilities and threats to wearable devices". In: *Journal of The Colloquium for Information Systems Security Education*. Vol. 7. 1. 2020, pp. 7–7.

[82] Alejandra Guadalupe Silva-Trujillo et al. "Cybersecurity Analysis of Wearable Devices: Smartwatches Passive Attack". In: *Sensors* 23.12 (2023). DOI: 10.3390/s23125438.

[83] François Petitjean, Alain Ketterlin, and Pierre Gançarski. "A global averaging method for dynamic time warping, with applications to clustering". In: *Pattern Recognition* 44.3 (2011), pp. 678–693. DOI: 10.1016/j.patcog.2010.09.013.

[84] François Petitjean et al. "Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification". In: *2014 IEEE International Conference on Data Mining*. 2014, pp. 470–479. DOI: 10.1109/ICDM.2014.27.

[85] Germain Forestier et al. "Generating Synthetic Time Series to Augment Sparse Datasets". In: *2017 IEEE International Conference on Data Mining (ICDM)*. 2017, pp. 865–870. DOI: 10.1109/ICDM.2017.106.

[86] L. Gupta et al. "Nonlinear alignment and averaging for estimating the evoked potential". In: *IEEE Transactions on Biomedical Engineering* 43.4 (1996), pp. 348–356. DOI: 10.1109/10.486255.

[87] Vit Niennattrakul and Chotirat Ratanamahatana. "On Clustering Multimedia Time Series Data Using K-Means and Dynamic Time Warping". In: 2007, pp. 733–738. DOI: 10.1109/MUE.2007.165.

[88] Vit Niennattrakul and Chotirat Ann Ratanamahatana. "Shape averaging under Time Warping". In: *2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. Vol. 02. 2009, pp. 626–629. DOI: 10.1109/ECTICON.2009.5137128.

[89] Karl Pearson. "On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[90] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of Educational Psychology* 24 (1933), pp. 498–520. DOI: 10.1037/h0071325.

[91] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. "Unsupervised Learning". In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, 2009, pp. 485–585. DOI: 10.1007/978-0-387-84858-7_14.

[92] Li-tze Hu and Peter M. Bentler. "Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives". In: *Structural Equation Modeling: A Multidisciplinary Journal* 6.1 (1999), pp. 1–55. DOI: 10.1080/10705519909540118.

[93] H. F. Kaiser and K. W. Dickmann. "Analytic determination for common factors". In: *American Psychologist* 14 (1959), pp. 425–439.

[94] Maximilian Ehrhart et al. "A Conditional GAN for Generating Time Series Data for Stress Detection in Wearable Physiological Sensor Data". In: *Sensors* 22.16 (2022). DOI: 10.3390/s22165969.

[95] Lucas Lange, Nils Wenzlitschke, and Erhard Rahm. *Generating Synthetic Health Sensor Data for Privacy-Preserving Wearable Stress Detection*. 2024. arXiv: 2401.13327 [cs.LG].

[96] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. "Time-series Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[97] Max Berrendorf et al. "On the ambiguity of rank-based evaluation of entity alignment or link prediction methods". In: *arXiv preprint arXiv:2002.06914* (2020).

[98] Latanya Sweeney. "K-Anonymity: A Model for Protecting Privacy". In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10.5 (2002), pp. 557–570. DOI: 10.1142/S0218488502001648.

[99] Ashwin Machanavajjhala et al. "L-diversity: privacy beyond k-anonymity". In: *22nd International Conference on Data Engineering (ICDE'06)*. 2006. DOI: 10.1109/ICDE.2006.1.

[100] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity". In: *2007 IEEE 23rd International Conference on Data Engineering*. 2007, pp. 106–115. DOI: 10.1109/ICDE.2007.367856.

[101] Lidan Shou et al. "Supporting Pattern-Preserving Anonymization for Time-Series Data". In: *IEEE Transactions on Knowledge and Data Engineering* 25.4 (2013), pp. 877–892. DOI: 10.1109/TKDE.2011.249.

[102] Cynthia Dwork. "Differential Privacy". In: *Automata, Languages and Programming*. Ed. by Michele Bugliesi et al. Springer Berlin Heidelberg, 2006, pp. 1–12. DOI: 10.1007/11787006_1.

[103] Samuel Kotz, Tomasz Kozubowski, and Krzysztof Podgorski. *The Laplace Distribution and Generalizations*. Birkhäuser Boston, 2001. DOI: 10.1007/978-1-4612-0173-1_5.

Declaration of Authorship

Ich versichere, dass ich die vorliegende Arbeit mit dem Thema:

„Re-Identification Attacks on Smartwatch Health Data“

selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Leipzig, den 18.03.2024

Tobias Schreieder

Appendix

Method	Multi-DTW-Attack			Slicing-DTW-Attack		
	WESAD	cGAN	DGAN	WESAD	cGAN	DGAN
average	**0.309**	**0.157**	0.101	0.789	0.903	0.924
minimum	0.295	0.147	**0.114**	**1.000**	**0.963**	**0.959**

Table 1.: Multiple distances: average and minimum method for Multi- and Slicing-DTW-Attack. The table shows a comparison of the results of the average and minimum methods for the Multi-DTW-Attack and the Slicing-DTW -Attack. Both attacks were performed on the WESAD, the WESAD-cGAN and the WESAD-DGAN data set. The evaluation was done according to the evaluation pipeline described in section 5.4. The attack window sizes 1-12 were tested for the Multi-DTW-Attack and the attack window sizes 1-36 for the Slicing-DTW-Attack. The results show that the average method performs best for 2/3 of the data sets for the Multi-DTW-Attack, and the minimum method for all data sets for the Slicing-DTW-Attack. The best method for each attack is used for all further experiments for handling multiple distances.

Method	Multi-Slicing-DTW-Attack		
	WESAD	cGAN	DGAN
average-average	0.363	0.174	0.112
minimum-minimum	**0.606**	**0.207**	**0.196**
average-minimum	0.316	0.163	0.121
minimum-average	0.380	0.144	0.123

Table 2.: Multiple distances: average and minimum method for Multi-Slicing-DTW-Attack. The table shows a comparison of the results of the average and minimum method for the Multi-Slicing-DTW-Attack. The attack was performed on the WESAD, the WESAD-cGAN and the WESAD-DGAN data set. The evaluation was done according to the evaluation pipeline described in section 5.4. The attack window sizes 1-12 were tested. The results show that the minimum-minimum method performs best for all data sets. The best method for the attack is used for all further experiments for handling multiple distances.

Data Set	k	Single		Multi		Slicing		Multi-Slicing	
		rank	score	rank	score	rank	score	rank	score
cGAN	k=1	0.105	**0.129**	0.112	**0.148**	0.873	**0.952**	0.116	**0.152**
cGAN	k=3	0.306	0.323	0.307	0.346	0.968	0.977	0.332	0.359
cGAN	k=5	0.463	0.494	0.453	0.498	0.988	0.990	0.487	0.509
DGAN	k=1	0.085	**0.088**	0.068	**0.078**	0.785	**0.938**	0.075	**0.087**
DGAN	k=3	0.255	0.242	0.232	0.234	0.919	0.958	0.249	0.251
DGAN	k=5	0.410	0.381	0.376	0.361	0.956	0.972	0.397	0.395

Table 3.: Evaluation of ranking methods for GAN data sets. The table shows the p@k scores of the ranking methods *rank* and *score* for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing). The attacks were performed on the WESAD-cGAN and WESAD-DGAN data set with a $DSF = 1000$ and with the standard sensor ranking without using DBA or PCA. The best method per attack according to the selection procedure is marked in bold.

k	Single			Multi			Slicing			Multi-Slicing		
	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5
1	**0.301**	0.540	0.692	**0.309**	0.561	0.696	0.966	1.000	1.000	0.588	0.765	0.856
2	0.295	0.529	0.692	0.299	0.534	0.692	1.000	1.000	**1.000**	0.600	0.771	0.858
3	0.297	0.532	0.688	0.308	0.528	0.686	1.000	1.000	**1.000**	**0.606**	0.772	0.845
4	0.294	0.532	0.686	0.296	0.524	0.669	1.000	1.000	**1.000**	0.565	0.769	0.829
5	0.294	0.535	0.687	0.294	0.513	0.668	1.000	1.000	**1.000**	0.549	0.752	0.814
6	0.295	0.532	0.683	0.278	0.507	0.642	1.000	1.000	**1.000**	0.547	0.720	0.813
7	0.299	0.530	0.678	0.272	0.505	0.646	1.000	1.000	**1.000**	0.520	0.695	0.788
8	0.290	0.524	0.670	0.272	0.498	0.637	1.000	1.000	**1.000**	0.475	0.673	0.766
9	0.290	0.537	0.672	0.264	0.488	0.619	1.000	1.000	**1.000**	0.453	0.620	0.722
10	0.283	0.515	0.660	0.261	0.469	0.607	1.000	1.000	**1.000**	0.431	0.622	0.729
11	0.290	0.514	0.661	0.261	0.461	0.598	1.000	1.000	**1.000**	0.427	0.595	0.706
12	0.290	0.515	0.650	0.243	0.455	0.590	1.000	1.000	**1.000**	0.384	0.549	0.659
13	0.293	0.514	0.657				1.000	1.000	**1.000**			
14	0.290	0.511	0.661				1.000	1.000	**1.000**			
15	0.288	0.504	0.653				1.000	1.000	**1.000**			
16	0.281	0.498	0.662				1.000	1.000	**1.000**			
17	0.293	0.500	0.670				1.000	1.000	**1.000**			
18	0.290	0.499	0.664				1.000	1.000	**1.000**			
19	0.292	0.503	0.667				1.000	1.000	**1.000**			
20	0.294	0.501	0.670				1.000	1.000	**1.000**			
21	0.296	0.507	0.680				1.000	1.000	**1.000**			
22	0.287	0.507	0.667				1.000	1.000	**1.000**			
23	0.278	0.504	0.668				1.000	1.000	**1.000**			
24	0.280	0.506	0.670				1.000	1.000	**1.000**			
25	0.286	0.509	0.668				1.000	1.000	**1.000**			
26	0.274	0.490	0.662				1.000	1.000	**1.000**			
27	0.269	0.488	0.659				1.000	1.000	**1.000**			
28	0.260	0.479	0.657				1.000	1.000	**1.000**			
29	0.266	0.483	0.657				1.000	1.000	**1.000**			
30	0.259	0.480	0.654				1.000	1.000	**1.000**			
31	0.259	0.482	0.646				1.000	1.000	**1.000**			
32	0.259	0.475	0.650				1.000	1.000	**1.000**			
33	0.262	0.484	0.647				1.000	1.000	**1.000**			
34	0.257	0.468	0.635				1.000	1.000	**1.000**			
35	0.256	0.464	0.625				1.000	1.000	**1.000**			
36	0.256	0.471	0.628				1.000	1.000	**1.000**			

Table 4.: Evaluation of attack window sizes. The table shows the p@k scores for the different tested attack window sizes 1-36 resp. 1-12 for the four re-identification attacks Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing). The attacks were performed on the WESAD data set with a $DSF = 1000$ and with the standard sensor ranking without using DBA or PCA. The best sensor combination per attack according to the selection procedure is marked in bold.

NP	Single	Multi	Slicing	Multi-Slicing	Stress Detection
0.0	0.301 ± 0.000	0.309 ± 0.000	1.000 ± 0.000	0.606 ± 0.000	0.880 ± 0.007
0.1	0.281 ± 0.002	0.297 ± 0.002	0.990 ± 0.003	0.547 ± 0.012	0.882 ± 0.008
0.2	0.275 ± 0.004	0.289 ± 0.003	0.977 ± 0.004	0.506 ± 0.024	0.879 ± 0.003
0.3	0.271 ± 0.006	0.286 ± 0.004	0.946 ± 0.008	0.463 ± 0.025	0.881 ± 0.008
0.4	0.259 ± 0.005	0.277 ± 0.008	0.913 ± 0.013	0.427 ± 0.035	0.875 ± 0.009
0.5	0.259 ± 0.004	0.268 ± 0.007	0.862 ± 0.014	0.413 ± 0.033	0.865 ± 0.006
0.6	0.250 ± 0.005	0.258 ± 0.006	0.827 ± 0.017	0.400 ± 0.041	0.846 ± 0.012
0.7	0.245 ± 0.006	0.257 ± 0.005	0.766 ± 0.024	0.399 ± 0.038	0.832 ± 0.017
0.8	0.241 ± 0.005	0.252 ± 0.007	0.747 ± 0.024	0.378 ± 0.023	0.809 ± 0.015
0.9	0.237 ± 0.005	0.246 ± 0.007	0.709 ± 0.029	0.370 ± 0.030	0.780 ± 0.008
1.0	0.235 ± 0.005	0.246 ± 0.007	0.673 ± 0.031	0.354 ± 0.023	0.767 ± 0.010
2.0	0.208 ± 0.005	0.227 ± 0.009	0.353 ± 0.043	0.220 ± 0.047	0.715 ± 0.012
3.0	0.205 ± 0.007	0.210 ± 0.007	0.270 ± 0.047	0.166 ± 0.035	0.694 ± 0.044
4.0	0.189 ± 0.010	0.198 ± 0.015	0.170 ± 0.042	0.144 ± 0.020	0.553 ± 0.076
5.0	0.182 ± 0.016	0.186 ± 0.011	0.129 ± 0.048	0.111 ± 0.026	0.442 ± 0.066
6.0	0.166 ± 0.010	0.180 ± 0.016	0.140 ± 0.026	0.093 ± 0.025	0.381 ± 0.075
7.0	0.155 ± 0.014	0.158 ± 0.009	0.100 ± 0.025	0.087 ± 0.019	0.354 ± 0.059
8.0	0.149 ± 0.016	0.149 ± 0.011	0.113 ± 0.016	0.073 ± 0.023	0.335 ± 0.057
9.0	0.137 ± 0.011	0.142 ± 0.015	0.097 ± 0.029	0.088 ± 0.037	0.322 ± 0.041
10.0	0.140 ± 0.013	0.134 ± 0.013	0.075 ± 0.020	0.075 ± 0.020	0.357 ± 0.055
11.0	0.118 ± 0.008	0.125 ± 0.021	0.083 ± 0.016	0.079 ± 0.022	0.397 ± 0.049
12.0	0.111 ± 0.016	0.116 ± 0.012	0.076 ± 0.015	0.070 ± 0.020	0.336 ± 0.056
13.0	0.119 ± 0.012	0.114 ± 0.015	0.085 ± 0.018	0.089 ± 0.023	0.345 ± 0.060
14.0	0.105 ± 0.012	0.113 ± 0.012	0.078 ± 0.015	0.078 ± 0.019	0.315 ± 0.060
15.0	0.109 ± 0.012	0.108 ± 0.010	0.072 ± 0.020	0.082 ± 0.020	0.341 ± 0.050

Table 5.: Evaluation privacy vs. usability. The table shows the average p@1 results for the Single-DTW-Attack (Single), Multi-DTW-Attack (Multi), Slicing-DTW-Attack (Slicing) and Multi-Slicing-DTW-Attack (Multi-Slicing) on the noisy WESAD data set with 15 subjects for different noise multipliers in comparison to the stress detection f1 scores. Noise multipliers in the range 0 to 15 are considered. The step size between 0 and 1 is 0.1 and from 1 the step size equals 1. In addition, the standard deviation over 10 runs is shown behind the results.