Reidentifying Anonymised Data Using Machine Learning

Martin Kristoffer Svensen

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
(Information Security)
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2020

Reidentifying Anonymised Data Using Machine Learning

Martin Kristoffer Svensen

Reidentifying Anonymised Data Using Machine Learning

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

In a digital world dominated by data, where vast amounts of data is collected about individuals each day, privacy is becoming an ever growing concern. With the introduction of wearable fitness trackers and smartwatches and their growing popularity, potentially sensitive health-related data is now also being collected about individuals, which further emphasises the need for privacy.

Common privacy-preserving actions today usually involves anonymisation of personal data, to prevent the person from being identifiable in a dataset, should it be sold, shared or otherwise published. Privacy regulations are in place, and in the process of improving, such as with the introduction of the EU General Data Protection Regulation (GDPR) being recently introduced in 2018 [1]. However, as we will see in this thesis, these efforts to protect privacy may not be sufficient.

With deep learning technology blurring the lines between identifiable and non-identifiable personal data due to its tremendous learning capabilities and ability to recognise patterns and correlations between data, there are concerns that current data anonymisation processes are no longer sufficient to protect a person's privacy. To test this theory, we will apply deep learning to an anonymised Fitbit dataset in an attempt to reidentify the participants within the dataset. In this thesis, we will be implementing a Siamese neural network architecture as a solution to this problem.

Acknowledgements

I would like to extend my thanks to those who have made this thesis possible.

I want to thank my supervisors, Michael A. Riegler, Pål Halvorsen, Steven A. Hicks and Vajira Thambawita, for their continued support, guidance and advice throughout the entire period of working on this thesis. They have also provided technical assistance on the subject of deep learning, and been good partners for discussing progress and results.

Last but not least, I want to thank my family and friends for providing me with support and encouragement to help me stay motivated and see this project through until the end.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are currently living in a digital age dominated by data. Data plays a huge role in business these days. With data comes valuable insight into your business, allowing you to understand recent trends and find ways to improve.

Data has always been a big part of business, even before the digital age we now live in. There are several ways data was collected and stored before the digitalisation. Businesses would store logs of shipments and sales etc. When businesses moved into the digital world, data collection and storage became much easier and more efficient. It also became possible to collect new types of data such as website traffic, both for the website as a whole and for specific pages on the site. With this data, you might better understand what areas of the website are more popular, which in turn tells you what is more worth investing resources into. It was not long until businesses also realised they could collect data about the individual customer as well. Moreover, this data proved to be particularly valuable.

Take, for example, a marketplace like Amazon. They can collect data on what items a customer purchases, what they items they browse, what reviews they give etc. With this data they can use algorithms to predict what items the customer might be interested in, and present these to the customer, increasing the likelihood that the customer will spend money and that they will have a more pleasant experience with the website. The more data you had on a particular customer, the more accurately your algorithms could predict their interests, and it is for this reason that data itself soon became attractive merchandise to be bought and sold between different companies, in order to learn more about their customers.

Whenever you browse online, you leave behind a digital trail of data, commonly referred to as a "digital footprint". This data is collected and processed either by the website itself or by third-party data analytics companies in order to learn more about you so they can tailor their services to better suit your needs. These companies would also sell this data for profit, as well as purchase your data from other companies in order to increase the quality and quantity of the data they possess. The collection and processing of customer data is nothing new, however. That is something we have seen since the early days of the digital age.

In more recent years, wrist-worn sensors such as smartwatches, Fitbits etc. have become increasingly popular. This is yet another source of valuable data to be profited from. These devices collect data about a person's physical fitness and health data, which will allow you to learn even more about them and thus allow for more accurate predictions. Fitness and health data could be beneficial to an insurance company, for example. If they had access to such data, they could use it to adjust insurance premiums depending on the customer's health conditions. This thesis will focus on these devices and their related applications. Specifically, Fitbit will be the main focus.

1.1 Motivation

Some of the data collected by these wearable activity trackers can be considered especially sensitive since they are related to the wearer's physical health. Having sensitive data about you collected and potentially sold or shared naturally raises some concerns regarding privacy.

Is the data anonymised in the event of being sold or shared? How well is it anonymised, and is it possible for the data to retain its anonymity when aggregated with other data? If the data is not properly anonymised, it could infringe upon a person's right to privacy. If sensitive personal information such as the data collected by fitness trackers ends up in the wrong hands could be used for profiling an individual, which can be considered invasive to their privacy.

How well is the data protected? Fitness applications are notorious for having data breaches or leaks. In 2018 alone there were several data breaches in fitness applications which compromised the data of millions of users. There was the global activity heatmap released by Strava which revealed secret military bases [2]. A similar incident happened later the same year when data leaked from the Polar fitness application compromised the location and activity of soldiers and secret agents [3]. 150 million users were compromised in the rather large data breach involving MyFitnessPal [4], whose leaked data is on the dark web for sale today [5]. PumpUp also had a data breach which even compromised user private messages [6]. Garmin has also had some privacy and security issues fairly recently. They had a major privacy breach in September 2019 where hackers stole payment data from the Garmin South African shopping portal [7]. They were also hit by a ransomware attack in July 2020 which forced them to shut down all their services worldwide, which took them several days to recover from [8]. The fact that we so frequently see significant data breaches involving fitness applications implies the security and/or data management of these applications is insufficient.

Deep learning is often applied in data analysis due to its ability to learn about data and make accurate predictions based off of what it has learnt. Deep learning consists of artificial neural networks, which are capable of learning similarly to the way the human brain learns. This, combined with the ever-increasing computational power from new technology, allows

for much greater efficiency in learning about specific problems such as analysing a specific type of data than we humans are capable of. In other terms, by applying such technology, we are able to learn more about data, than we would have been able to otherwise. This also leads to better predictions in for example health risks [9], traffic flow [10], stock markets [11] or even human behavior [12]. As we can learn more about individuals by applying deep learning technology in data analysis, the need for privacy is becoming more apparent.

The recent trend and rising awareness surrounding digital privacy have resulted in improved regulations regarding the processing of personal data, such as the EU General Data Protection Regulation (GDPR) [1], in an effort to preserve our privacy. The GDPR grants rights to the data subject [13] and restricts the processing of special categories of personal data such as religion, racial or ethnic origin, biometric identification data, health data etc. [14]. To comply with these regulations, personal data will usually be anonymised before being sold or shared. However, the ever-evolving deep learning technology raises some concerns regarding personal data and anonymisation. Is it possible that anonymised data can be reidentified with deep learning technology? Consider, for example, a fitness dataset from Fitbit, Garmin, Polar or similar services. These datasets will usually include physical data for physical activity, sleep, heart rate and perhaps even food consumption or GPS coordinates. Deep learning could be used to learn a lot about a person, including their activity pattern, sleep pattern and more by analysing this type of data. This information could also be used for profiling, which once again emphasises the need for proper anonymisation.

1.2 Problem Statement

When data is anonymised, this is usually done by removing 'identifiers' from the dataset. These identifiers are data directly connected to a person's identity, such as name, country, date of birth, phone number, email etc. As we covered in the problem statement, with deep learning being more widely used, especially for analysing data (both personal and otherwise), it raises the concern that this anonymisation process may no longer be sufficient to protect privacy.

In light of this, we have chosen to look into the possibility of using deep learning to reidentify individuals in an anonymised fitness dataset. We have two main research questions we aim to answer, and they are as follows:

Question 1 Could a chunk (for example a day or a week) of deidentified physical activity data be considered a biometric identifier?

If it is possible to find patterns in physical activity data, which is unique enough for a deep learning algorithm to distinguish between persons based solely on this data, it would be considered a biometric identifier.

Question 2 Is it possible to use deep learning in order to reidentify individuals in a dataset that has been previously anonymised?

This would involve being able to distinguish between participants in an anonymous dataset based on pattens in their data, for example, by treating the data as a biometric identifier, as stated above.

We aim to answer these questions through our main objective:

Objective Implement a deep learning neural network architecture to reidentify individuals in an anonymised fitness dataset.

1.3 Scope and Limitations

Based on the objective mentioned in our problem statement, the scope for this thesis will be implementing a Siamese Neural Network architecture to reidentify participants of an anonymised Fitbit dataset.

We will be limiting ourselves to a single dataset, called PMData [15], which we will describe in chapter 3. Thus, we will also be limiting ourselves to data collected by the Fitbit Versa 2 [16], as this was the device used for data collection in PMData [15]. Furthermore, although reidentification of data can be achieved through other architectures as well, we will be limiting ourselves to a Siamese architecture, as this has proven effective in comparison-based identification problems which we will explore in chapter 2. With this, we will be able to observe whether fitness data can be used as a sort of biometric identifier as we will be implementing the Siamese architecture in a similar fashion to how it would be used for facial recognition. Within this dataset, we will also be limiting ourselves to only testing data for steps and heart rate, as we had to prioritise due to time constraints.

1.4 Research Methods

In this thesis, we use an iterative experiment-based approach to designing and implementing a solution for the problem of reidentifying individuals in an anonymised dataset. We implementing an existing architecture for solving similar problems and modify this to suit our purpose. Then, we will go through several iterations where we conduct reidentification experiments, assess the results and implement improvements accordingly.

1.5 Main Contributions

Over the course of this thesis, we have researched and implemented a Siamese neural network architecture which is capable of reidentifying participants in an anonymised dataset. It works similarly to facial recognition by taking two chunks of data and computing a similarity score

between them. This similarity score is then used to determine whether the two chunks of data comes from the same person or not.

Our final implementation uses heart rate data for comparison and achieves reidentification of participants in the dataset we used with an accuracy of up to 77.4%. We can, therefore, conclude that there is a good potential for reidentifying anonymised physical activity data by using deep learning to compare chunks of data. This approach is limited, however, in that it requires you to have one or more, already identified, chunks of data from the person you wish to identify in a dataset, which the Siamese neural network can use for comparison.

With our main objective achieved, we can answer the research questions from our initial problem statement:

Question 1 *Could a chunk (for example a day or a week) of deidentified physical activity data be considered a biometric identifier?*

Our implemented Siamese neural network is capable of distinguishing between anonymous participants in a dataset by computing the similarities between their data. It does this by taking two chunks of heart rate data as input, where a chunk is a day of beats-per-minute (bpm) values. These chunks are sent through identical neural networks to generate feature vectors for them. The absolute distance between these feature vectors is then used to calculate a similarity score, which can be used to determine how similar these chunks of data are. The hypothesis is that if two chunks of data compute similar features in the neural networks, the chunks are likely to belong to the same person due to having similar patterns.

This gives reason to believe that physical activity data can be used as a biometric identifier, as it can be used to identify a person in an anonymous dataset. An obvious limitation to this, however, is that humans change over time, and so does their activity patterns, meaning the Siamese neural network may not be able to reidentify individuals if the chunks of data to be compared are far apart in terms of when it was collected. Another limitation is that you need to already possess one or more chunks of data from the person you wish to identify, as you need something to compare to the chunks in the anonymised dataset.

Question 2 *Is it possible to use deep learning in order to reidentify individuals in a dataset that has been previously anonymised?*

As mentioned, our implementation was able to reidentify participants of a dataset with an accuracy of 77.4%, which proves that this is indeed possible. It is based on the theory that since humans are habitual creatures, their behaviour will likely have some sort of pattern, which would be reflected in their physical activity data. In such a case, one should be able to distinguish between individuals based on patterns found in their physical activity data. If these patterns are unique enough, they could be used as a form of biometric identifier which could allow a deep learning algorithm to identify an individual based on their physical activity data.

Objective *Implement a deep learning neural network architecture to reidentify individuals in an anonymised fitness dataset.*

Our implementation of a Siamese neural network takes two chunks of data as input, compares these and computes a similarity score based on the absolute distance between their features. A high similarity score means the patterns in each chunk is similar and is therefore interpreted as the two chunks belonging to the same person. With reaching accuracies up to 77.4% in testing, we can conclude that the implementation is capable of reasonably reidentifying anonymised data, as long as you have available data from the person you wish to identify, for comparison. The source code for this implementation can be found on GitHub here: https://github.com/Zeykes/Fitbit_Reidentification

1.6 Thesis Outline

This thesis is split into four chapters. The first is an introduction, followed by background research on the topics of fitness trackers, privacy and deep learning. The third chapter details our own research and experimentation process in chronological order, and the final chapter concludes the thesis, where we summarise our experiments and present future work. Below is a summary of what each chapter entails, excluding chapter 1.

Chapter 2: Background In this chapter, we present background research and works relevant to this thesis. This chapter is split into four parts; fitness trackers, relevant works using Fitbit, research on privacy and fitness data, and a summary at the end.

The first part briefly goes into the history of fitness trackers, including their origin and rise to popularity, as well as the state of fitness trackers today. As our thesis explores privacy concerns in fitness data through reidentification of anonymous datasets, this section provides context to means in which fitness data is collected.

The second part explores related works on Fitbit devices and data, which has relevance for our thesis because we will be using Fitbit in the data collection process for the thesis itself. This section gives context regarding the accuracy of the data collected through Fitbit devices, as well as security and privacy surrounding these devices.

The third part will explore privacy, where we look into how data is generally anonymised, and why this process is important. Furthermore, we go into detail on what potentially sensitive information can be indirectly obtained through analysing fitness data, and how this may be a cause to view fitness data itself as sensitive, further emphasising the necessity of privacy regarding such data.

Finally, we will summarise the chapter, highlighting the important takeaways and their relevance for our thesis.

Chapter 3: Experiments and Results In this chapter, we detail our experimentation process in chronological order. We test and improve

our Siamese neural network implementation iteratively, going through the following steps, in order:

Experiments. Here we conduct experiments using the current implementation, observe and assess the results, leading into the next experiment.

Reflections. Here, we summarise a batch of experiments, discuss the results, and lead into which improvements should be made, either to the model implementation or to the experimentation method itself.

Improvements. Here, we make improvements based on our reflection of the previous batch of experiments. These improvements may be for the implementation, or for the experimentation method if we find it necessary. The next batch of experiments will be conducted after these improvements have been implemented.

Chapter 4: Conclusion In this chapter, we summarise our experimentation process and present our results and conclusions. We also reiterate our main contributions once more, and present future work we believe to be the natural next step for the work done in this thesis.

Chapter 2

Background

2.1 History of Fitness Trackers

In this section, we will study how fitness trackers have evolved to be as we know them today.

2.1.1 Origin and Evolution

To find the origin of fitness trackers, one must first determine what the definition of a fitness tracker is. When people talk about fitness trackers today, they are most likely referring to wearable digital devices that include features such as a clock, step counter, heart rate monitor and more. However, this is only the modern definition. These features have been around for much longer as stand-alone devices.

The first pedometer was created as early as 1780 by Abraham-Louis Perrelet [17]. The first wearable pedometer was created in 1965 and was called "Manpo-Kei" which translates from Japanese to "10,000-step meter" [18]. This came along with the notion that 10,000 steps daily was the optimal amount, although this did not have much scientific basis at the time [19].

The first wearable heart rate monitor was invented in 1977 by the Finnish professor Seppo Säynäjäkangas, who later went on to found Polar Electro. Polar Electro began selling wireless personal heart rate monitors in retail in 1983 [20].

Polar would go on to improve wearable fitness technology and eventually create fitness trackers that included both pedometer, heart rate monitor functionality and more in the late 1990s / early 2000s. These were the early models of the fitness trackers we know today. Garmin later joined them in 2003 [21], and Fitbit in 2007 [22], which further pushed the tracker technology forward.

Smartwatches were yet another type of wrist-worn technology that emerged around 2012 with the introduction of the Sony SmartWatch [23]. Large mobile/electronics companies such as Samsung, Apple and many others would also make their own smartwatches shortly after, contributing to further growth in this market [24]. These smartwatches would also

incorporate fitness tracking functions such as step count and heart rate monitoring, effectively combining with the fitness tracker technology. Naturally, fitness tracker manufacturers like Polar, Fitbit and Garmin would soon also develop their own smartwatches to join this trend [25–27]. Today, smartwatch and fitness tracker technology have merged to the point where the line between them has become blurred.

2.1.2 Rise to Popularity

It was not long after smartphones became commonplace that people gained interest in moving smart-technology to wearable devices. Although wearable fitness trackers have been around for quite some time, as we saw in section 2.1.1, they did not gain much popularity amongst consumers until around 2013-2014. Fitbit was the company dominating the market of fitness wearables in 2014, with companies such as Samsung, Garmin and Jawbone also making their presence known [28]. In the following years, Apple, Xiaomi and later Huawei have also come on board, with Apple dominating the market in recent years [29]. The overall market for wearables has also grown significantly during this time, is estimated to reach as much as USD 91.98 billion by 2027 [30].

2.1.3 Manufacturers Today

The largest manufacturers of fitness trackers today, going by their market share, are Apple, Xiaomi, Huawei, Samsung and Fitbit [28]. Other manufacturers such as Garmin, Polar, Lifesense and others are still in the market, although to a lesser extent. Fitbit has especially dropped in recent years, from being the largest manufacturer in the past to holding just a fraction of the market share today. Jawbone was very relevant in the earlier years of fitness trackers but has since gone out of business in 2017 [31].

2.1.4 Fitbit Acquired by Google

Google announced its intent to buy Fitbit for USD 2.1 billion in November 2019 [32]. Shareholders have since approved the deal in January 2020 [33]. This effectively means Google gets to expand its already vast data empire with the addition of Fitbit data.

It has already been said how much Google knows about you through the data collected via their services, and the privacy risks involved. Now they can potentially gain access to your fitness data as well through Fitbit.

2.2 Related Work - Fitbit

For this thesis, we have chosen Fitbit as our main focus. We chose Fitbit mainly due to convenience, as we would be participating in a data collection experiment by Simula, using Fitbit, during the time period of working on the thesis, and we could use this data for our own experiments.

In this section, we will look at some relevant research, that is specifically related to Fitbit.

2.2.1 Accuracy of Data Collected Through Fitbit

Several studies have conducted on whether or not Fitbit devices are capable of providing accurate and reliable step counts. The first study we looked at was conducted by having participants wear trackers while walking and running on a treadmill in set intervals, and comparing the tracker results with manually observed results [34]. This study concluded that the trackers are capable of reliably and accurately recording the wearer's step count both when running and walking. Another study conducted in similar conditions yielded similar results. However, this study also measured the accuracy of the distance measured by the tracker and found this to be very inaccurate and unreliable [35]. We also looked at a study that tests how the tracker attributes steps if the wearer is doing activities other than simply running or walking. These activities include vertical jumping, hopping, squatting and standing up from a sitting position. This study found that although the devices are fairly accurate recording steps when running or walking, they will also incorrectly record steps while performing other activities [36].

When it comes to the accuracy of sleep patterns recorded by Fitbit devices, a study published by the Journal of Medical Internet Research shows that although they might be convenient and good enough for the average person to gain insight into their sleep quality and patterns, they are lacking in terms of more specific and in-depth tracking, making them unfit to be used as a substitute for polysomnographies in medical care [37].

To summarize, while the devices can measure steps while walking or running accurately, they will incorrectly add extra steps to the counter during any other activities, making it somewhat unreliable for regular use. The distance measurement, which is based on step count and stride length (a set value in settings), is rather inaccurate and should not be trusted [38]. The sleep pattern function is good enough for a consumer to use as gross estimates; however, it is not accurate enough to be particularly valuable to medical personnel.

2.2.2 Security and Privacy

Although the security and encryption protocols of the Fitbit ecosystem has improved over the years, there is still room for improvement, as shown in [39]. This study explores potential security vulnerabilities in the device itself, the application, the Fitbit server and the connections between these. One of their findings is that the device actually logs and sends step count data of finer granularity than Fitbit provides its user. This suggests the device collects more data than it informs the user of, which could potentially be a breach of privacy. Another interesting discovery they made is how the Fitbit device actually responds to any nearby device with the same Bluetooth protocol (BTLE), rather than just the device it is paired

with. By capturing the Bluetooth traffic and analysing the data, you can figure out the private addresses of nearby devices. Since Fitbit for some reason opts not to use the privacy-enhancing feature of the BTLE protocol, which allows developers to change these private addresses frequently to avoid tracking [40], the private addresses of Fitbit devices remain static. Furthermore, the Fitbit application collects these addresses discovered through Bluetooth and reports it to the Fitbit servers which means they could potentially map out a user's surroundings based on nearby devices, which would be a breach of privacy.

2.3 Related Work - Privacy

Having secure devices are important by itself, but the focus of this thesis is the privacy issues raised by the information in the data itself. Thus, in this section, we will look at research on privacy that relevant to this thesis.

2.3.1 Anonymisation

Before being sold or shared, user data is often anonymised in some way. This is done per privacy policies and regulations in order to prevent the user from being identified by their data. The exact anonymisation process may vary depending on the data and policies in question. The most common approach to anonymisation is to remove any identifiers from the data. This approach is referred to as deidentification. Identifiers are data linked directly to a person's identity such as name, email, telephone number, address, birthday etc. One commonly used standard for deidentification is the HIPAA (Health Insurance Portability and Accountability Act) deidentification standard [41]. This standard lists all identifying data that must be removed in order to deidentify the data.

It has been known for a long time that you only need a few identifiers in order to directly identify a person. This was proven as early as the year 2000 when researcher Latanya Sweeney managed to identify 87% of Americans using only their birth date, zip code and sex [42]. However as technology develops over the years, especially in the field of data science and machine learning, there have been more frequent cases of users being reidentified through 'anonymised' data as well [43]. An example of such a case was when researchers Arvind Narayanan and Vitaly Shmatikov managed to reidentify some of the users in Netflix' dataset of anonymous movie ratings (released in 2006), by comparing the data with another non-anonymised dataset found elsewhere [44]. They showed once again in 2009 how cross-referencing data sets can expose users' identities, using Flickr and Twitter [45]. With these examples, we have seen how easily users can be identified by cross-referencing different data sets. It is only natural to believe the same can be achieved for anonymous fitness data as well. In fact, there was a study conducted on exactly this in 2018 [46]. For this study, they used a scenario where one organisation shared a deidentified data set containing physical activity data, health data and demographic

information. They were then able to reidentify this data by using machine learning to cross-reference the data with a different data set containing names, demographic information and physical data. This was proven to be very effective, with a success rate of 67-94% (varying by which algorithm was used).

There is most likely a significant amount of information that can be inferred about a person through their fitness data. As Christovich said, "If computer scientists can predict political leanings, sexual orientation, or emotional stability based on an individual user's Facebook likes [47], it is highly probable that similar inferences can be made from data gathered by fitness trackers, which measure users' lives in granular detail" [43, p. 111].

As we have seen, there have already been many cases of anonymised data being reidentified. It is therefore natural to assume that it should be possible to reidentify anonymised Fitbit data similarly. If this is indeed possible, then it may suggest Fitbit's anonymisation process is insufficient. As mentioned in the problem statement (section 1.2), we will study the possibility of reidentifying anonymised Fitbit data, which will allow us to learn whether the current anonymisation process is sufficient.

2.3.2 Fitness Data Reveals Health Information

Data collected by a fitness tracker can reveal much information about the wearer. Given how closely fitness and health are tied together, it would make sense that fitness data could, in addition to providing information about the wearer's fitness level, also provide information about the wearer's health. This health information may be inferred by analysing the collected fitness data.

An example of this would be heart rate. There have been several studies on the usefulness of heart rate as a predictor for various diseases. These have shown that an elevated heart rate can potentially predict diseases such as cardiovascular disease and metabolic syndrome, as well as being associated with insulin resistance, insulin precursor presence and the acute insulin response [48–52]. With that said, heart rate without context does not necessarily tell us much on its own. A low resting heart rate could, for example, imply the wearer is either very athletic or is taking a drug such as a beta-blocker [53]. If the wearer's heart rate is very high, it could either imply they are exercising, or they are suffering from some heart condition. It is difficult to tell which is actually the case without looking at the context. Therefore, by looking at the correlation between heart rate and the wearer's activity data, one gains further insight as to which case is more likely.

A study conducted in 2017 revealed a strong association between high heart rate and temperature measurements and elevated hs-CRP (high-sensitivity C-reactive protein) levels which suggest wearable technology may contribute to detecting inflammatory responses even earlier than the wearer/patient by themselves [54]. The same study also suggests that heart rate recorded by wearable devices may be useful in assisting in the discovery of other diseases such as cardiovascular disease and metabolic syndrome as well [54]. Another study conducted earlier this year (2020)

has also shown potential for predicting influenza-like illnesses by using machine learning to analyse heart rate and sleep data collected from wearable fitness trackers [55].

Given how much health information we can obtain by analysing a wearer's collected fitness data already, and how this will only increase as both wearable technology and analytics technology continues to develop, we may need to start considering fitness data to be as sensitive as health data in terms of privacy. Furthermore, seeing how many of the aforementioned disease predictions rely on multiple types of data being analysed together, it may be a worthwhile privacy measure to separate this data in the database and obscure the connection between them. We believe the matter of fitness data sensitivity, and obscuring the connection between data in the database to be topics worth investing.

2.4 Related Work - Siamese Neural Network

A Siamese neural network is a type of deep learning architecture that uses two identical convolutional neural networks connected at their output to compute the similarity between two inputs [56].

Siamese neural networks date back as far as the 1990s where it was introduced by Bromley et al. in order to verify signatures by looking at their similarities [56]. It is a type of architecture used to distinguish whether two entities are similar or not. The name is likely inspired by the fact that the architecture consists of two identical networks (i.e., twins) that are joined at their outputs. A Siamese neural network takes two inputs, passes each through identical neural networks, and finally uses the output from these to calculate the similarity between the features extracted, by the convolutional neural networks, from the original inputs.

Siamese neural networks have been applied in a variety of research and experiments, including facial recognition [57], human reidentification [58, 59], object tracking [60, 61] and symbol recognition [62, 63]. These works show that Siamese neural networks have been proven quite effective when it comes to computing similarities between entities, which is why we believe it is a good choice for our problem of reidentifying fitness data.

2.5 Summary

In this chapter, we have seen how fitness trackers have evolved throughout the years and gained immense popularity in the last decade. It is a market that has grown significantly, to the point where it is rather common to see people around you wearing fitness trackers or smartwatches in your daily life. This naturally also means there are vast amounts of fitness data collected from people every day, which is why our problem statement is relevant, as insufficient anonymisation of data could affect a significant number of people.

We have also looked into Fitbit in more detail, as this is what we will use for data collection purposes in our thesis. We looked at

research on the accuracy of the data collected via Fitbit devices, as well as research regarding the security and privacy of these devices and the Fitbit ecosystem. From this, we have learned that Fitbit is reliable when it comes to step count, but loses accuracy if the wearer is performing physical activities outside of running or walking, such as jumping, hopping or squatting. As for the security of the devices, we found a study that highlighted potential vulnerabilities in Fitbit's Bluetooth protocol. Although there are a few issues with Fitbit devices, we will be using a newer device than what was tested in the aforementioned research. It is therefore unclear whether these issues still exist, but we believe they will have little impact on our work.

We covered topics in privacy such as anonymisation, where we looked at what an anonymisation process entails. We also referred to research showing that you do not need many identifiers in order to uncover the identity of individuals in a dataset, which shows us that anonymisation processes must be thorough in order to be effective. We would be taking it one step further by using data itself as an identifier, in a dataset that has been anonymised.

We have covered how there is much more to fitness data than just the raw data itself, as there is much information to be gained by analysing such data. This further emphasises the need for privacy in such data and helps us understand why the possibility of reidentification through deep learning could be a cause for concern.

Lastly, we looked at Siamese neural networks, briefly how they work, and what their usage areas are. We have seen that it is quite effective at solving problems related to identification by recognising similarities in input data. In the next chapter, we will go through our own implementation of a Siamese neural network, and the experiments we have conducted with it in an attempt to reidentify individuals in a fitness dataset.

Chapter 3

Experiments and Results

The source code for our implementation of a Siamese Neural Network can be found in our GitHub repository here: https://github.com/Zeykes/Fitbit_Reidentification

3.1 Planning

Before we begin implementing a Siamese Neural Network architecture to tackle the problem statement from section 1.2, we will be studying existing implementations of Siamese Architectures and use one, or several, of these as a baseline to build off of. If necessary, we will convert an architecture to be able to work with our data for our purposes. We will then improve this implementation through several iterations where we make modifications, run tests and compare results to find ways of improving until we achieve a satisfying result.

As mentioned in section 1.3, we will also be testing with two different sets of data, steps and heart rate. This is to gain insight into how the choice of data affects the possibilities for reidentifying a participant. Ideally, we would test a broader range of datasets and types to gain an even better insight into what data is most suitable for reidentification, but we had to limit ourselves due to time constraints.

To summarise, in this chapter, we will:

- give a general explanation of Fitbit data, and the dataset we have used.
- describe the process of preparing the data for machine learning experiments.
- describe how we implemented a Siamese Neural Network architecture, and converting it to work with Fitbit data.
- iterate over, and improve the implementation.

3.2 Fitbit Data

3.2.1 General

Fitbit is a company that aims to promote health and fitness. They achieve this by developing wearable devices and applications to assist people in living healthy and getting fit [64]. They also have a means of engaging with the community in order to help motivate people to a healthier lifestyle. These include their fitness-related blog [65], a collection of success stories [66] and their own forum where the community can interact with each other [67].

3.2.2 Device Used

For testing purposes related to this thesis, we have used the Fitbit Versa 2, which is one of the newest devices from Fitbit at the time of writing. The Fitbit Versa 2 is a smartwatch meaning it includes more features than the regular trackers. These features include showing notifications from a paired smartphone, controlling Spotify on the paired device, as well as Alexa support with its built-in microphone. Alongside these features, it also collects various amounts of fitness data, which we will go into detail on in Section 3.2.3.

3.2.3 Extracting Data

When logged in on the Fitbit website, one can visit their account page and export their data. This can be done in either of two ways. The first method is to export a smaller dataset of their activities in the past week or month, while the other involves requesting all data they have collected for your account since its creation. Seeing as the former only gives a summary of activities, weight, food and sleep for in a limited time-frame, we will largely focus on data obtained via the other method which, assuming Fitbit abides by article 12 of the GDPR [68], should include all stored data pertaining to the user. Table 3.2.4 gives a short description of all the files received when requesting your account data.

3.2.4 PMData: A Sports Logging Dataset

During the period 15 November 2019 - 31 March 2020, we took part in a data logging experiment named PMData [15] conducted by Simula Research Laboratory. This logging experiment had 16 participants wear a Fitbit Versa 2 [16] to monitor their activity and collect data, as well as responding daily to Google Forms and a questionnaire in the pmSys [69] app [70]. The goal of PMData was to create a high-quality dataset for use in research. The dataset is publicly available here: https://datasets.simula.no/pmdata/.

We decided to use this dataset for our research in this thesis. This is due to the relevance for the thesis, as well as due to practicality since we also took part in the data collection process. In table 3.2.4, we briefly explained

the various data files you can obtain from exporting your Fitbit data. Not all of these are relevant for our project, as we are only interested in fitness data since this would provide information we could potentially use to identify an individual. We chose to focus on steps and heart rate for our project. Steps can give us information about an individual's daily activity patterns. Heart rate, on the other hand, is affected by multiple factors such as fitness level, age, sex, potential cardiovascular diseases etc. [71–73]. Due to heart rate being related to such a multitude of factors, we hypothesize that an individual's heart rate pattern could be used as an identifier within a Fitbit dataset. Although we will mainly be using steps and heart rate from this dataset, we will discuss other relevant data later in section 3.13.

File	Rate of Entries	Description
calories.json	Per minute.	Time series data of calories burnt.
steps.json	Per minute.	Time series data of steps taken.
distance.json	Per minute.	Time series data of distance travelled.
sleep.json	When it happens (usually daily).	Time spent in each sleep stage (light, deep, REM and awake), per sleep session.
lightly_ active_minutes.json	Per day.	Number of lightly active minutes per day.
moderately_ active_minutes.json	Per day.	Number of moderately active minutes per day.
very_ active_minutes.json	Per day.	Number of very active minutes per day.
sedemtary_ minutes.json	Per day.	Number of sedentary minutes per day.
heart_rate.json	Per 5 seconds.	Heartbeats per minute (bpm) at a given timestamp.
time_in_heart_rate_ zones.json	Per day.	Number of minutes spent in different heart rate zones (fat burn, cardio, peak).
resting_ heart_rate.json	Per day.	Resting heart rate for the given day.
exercise.json	When it happens. 100 entries per file.	Detailed data per activity. Data includes timestamps for start and stop, a breakdown of time spent in different activity levels, type of activity, and metrics such as time, steps, calories etc.
sleep_score.json	When it happens (usually daily).	Contains a score between 0-100 for each sleep session, calculated based on composition (e.g. data from sleep.json), duration, revitalisation, number of deep sleep minutes, resting heart rate and restlessness.

Table 3.1: The Fitbit data files from the PMData dataset.

3.3 Technology

The technologies used for experimentation in this thesis are listed in table 3.2. TensorFlow will be used for all machine learning related functionality such as models, layers as well as the functions for training and using models. NumPy and pandas will be used for data processing and organisation, while random is used for shuffling input arrays before training. Lastly, we use pyplot from the matplotlib library to plot learning curves.

Technology	**Version**	**Description**
Python [74]	3.8	Programming language used.
pip [75]	20.2.3	Package installer for Python. Handles installing and updating packages.
CUDA Toolkit [76]	10.1	Toolkit by NVIDIA for using the GPU for compute-heavy work such as machine learning. Also required by Tensorflow to run its machine learning functions on the GPU.
cuDNN [77]	7.6	CUDA Deep Neural Network library, for optimising and tuning the GPU specifically for using deep neural networks.
Tensorflow [78]	2.3.1	An open source platform for machine learning, with the easy-to-use Keras API to allow for easily building machine learning solutions.
Numpy [79]	1.18.5	Open source Python package for using arrays, vectors, matrixes as well as useful numerical computing tools.
Pandas [80]	1.0.5	Open source data analysis and manipulation tool. Allows for using dataframes to easily manipulate and organise large datasets.
random [81]	-	Part of the Python Standard Library. A package containing various randomisation functions. We will mainly use this to shuffle inputs to a machine learning model.
matplotlib [82]	3.2.2	A library for visualisation and plotting plotting. We will mainly be using matplotlib.pyplot to plot learning curves.

Table 3.2: Technologies used in this thesis.

3.4 Siamese Neural Network

For our project, we have implemented and used two different siamese convolutional neural network architectures [57, 63]. In figure 3.1, we have illustrated the high-level architecture of a Siamese neural network.

As seen in figure 3.1, a siamese neural network starts by taking two inputs and passes each through identical neural networks. These convolutional neural networks will each produce a feature vector, called h1 and h2 in figure 3.1. It will then compute the absolute distance between these feature vectors, $|h1 - h2|$, and calculate a similarity score through a Sigmoid function using this distance. This similarity score is the final output of the Siamese neural network, and this value is interpreted as the similarity between the two inputs. In the context of our problem, we interpret a score of 0.5 or higher as meaning the two data chunks used as inputs originate from the same person. If the score is lower than 0.5, the data chunks belong to different persons.

When Siamese neural networks are used in, for example, facial recognition, you will usually have a database of identified images, which you will use for comparison with an unidentified image A, in order to find a match. A match would then identify image A since it matches with an already identified image B in the database. A match is considered as a comparison that achieves a similarity score of above 0.5.

In our case, the order of comparison is reversed. We will have a deidentified dataset of fitness data, and an identified data chunk A for comparison. Comparing the identified chunk A to the deidentified dataset will allow us to identify which anonymous participant in the dataset corresponds to the same person as chunk A, by assessing which chunks achieve a similarity score of above 0.5 with chunk A.

This also means we are limited in who we can reidentify in a dataset, as we require at least one identified chunk of data from that person, on the same format as those in the dataset, which the Siamese neural network can use for comparison.

We have implemented two different siamese neural networks over the course of this project, and we will go into the details of their specific implementations in sections 3.5.2 and 3.11.

3.5 Steps

3.5.1 Data Processing

Initially, we focused on steps as this dataset would require the least amount of effort to clean and use. The Fitbit dataset for steps is a time series with an integer step-value recorded every minute. This data is recorded in a file called `steps.json`, as listed in table 3.2.4.

The dataset is somewhat inconsistent as minutes are missing here and there in the dataset. We made the assumption that if a minute is missing, there were no steps taken in that minute and thus we would fill in these missing minutes with 0 as the step-value.

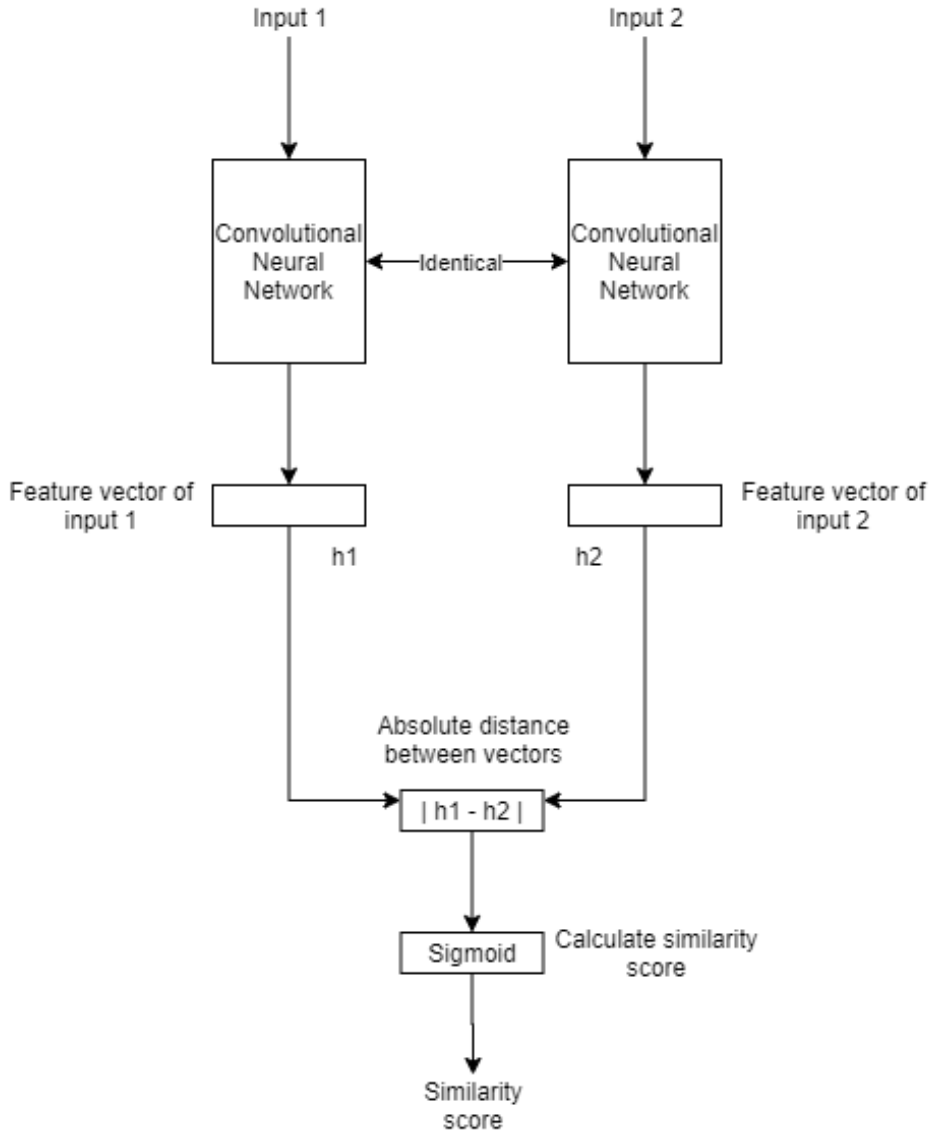

Figure 3.1: General architecture of a Siamese Neural Network.

After we had filled in the missing data, we created a function to separate the data into smaller chunks of data, that would be used as inputs for the Siamese model. Each chunk would contain a week of data for a single participant. A week is considered all values from Monday 00:00:00 to, and including Sunday 23:59:00. At 1 value per 1 minute, this amounts to 10080 values per week.

When the data was organized into chunks, we finally removed the timestamp column, leaving only the step-values themselves as a single array in chronological order. A single chunk had a length of 10080, which is the number of minutes in a single week.

3.5.2 Initial Model

We used a siamese architecture inspired by the facial recognition experiment by Shubham Panchal [57] for our initial experimentation. The inner architecture for the convolutional neural network part of the Siamese architecture (as shown in figure 3.1) is illustrated in figure 3.2 under "Before". This architecture consists of convolutional 2D layers, max-pooling layers, a dense layer, and a flatten layer. The convolutional layers create feature maps from the inputs which are then down-sampled by the max-pooling layers by highlighting the features that are most present. The result from this is then flattened (transformed from a feature map to a single dimension feature vector) via the flatten layer before being sent through a dense layer. This composition of layers make up the sequential model which corresponds to the "Convolutional Neural Network" part of figure 3.1, and the output of this sequential model corresponds to "feature vector" h1 and h2 of figure 3.1.

The original architecture was designed to take images as input. We would be using one-dimensional arrays with Fitbit data as input, which means the architecture in its original state was incompatible with our purpose. In other words, we would have to modify the model before we could use it. This was achieved by switching from 2D layers to 1D layers, and subsequently changing the defined input shape, as well as changing the kernel size and pool size of the layers to one dimension. Figure 3.2 shows the sequential model before and after modification. With that, we were able to make the siamese model compile and train on Fitbit data successfully.

3.6 Experimentation - Phase One

3.6.1 Arranging the Input

Before we could train the model with data, we had to organize the inputs for the model's `fit()` function. We did so by creating three arrays: `input_1`, `input_2` and `labels`. The first two arrays would be the chunks of step-data coming from the function mentioned in section 3.5.1, arranged in such a way that each chunk gets compared with all chunks in the set. In other words, if we have N chunks of data in total (for all participants combined), the two inputs would each be of size N^2. As an example, if we had three chunks in total (N=3), from the function mentioned in section 3.5.1 and name them `c1`, `c2` and `c3`, the input arrays would be:

input_1	c1	c1	c1	c2	c2	c2	c3	c3	c3
input_2	c1	c2	c3	c1	c2	c3	c1	c2	c3
labels	1	0	0	0	1	0	0	0	1

Table 3.3: Example of what the input arrays looks like.

The `labels` array contained ground truths (0 or 1) on whether each of the comparisons between `input_1` and `input_2` are true negatives (0) or

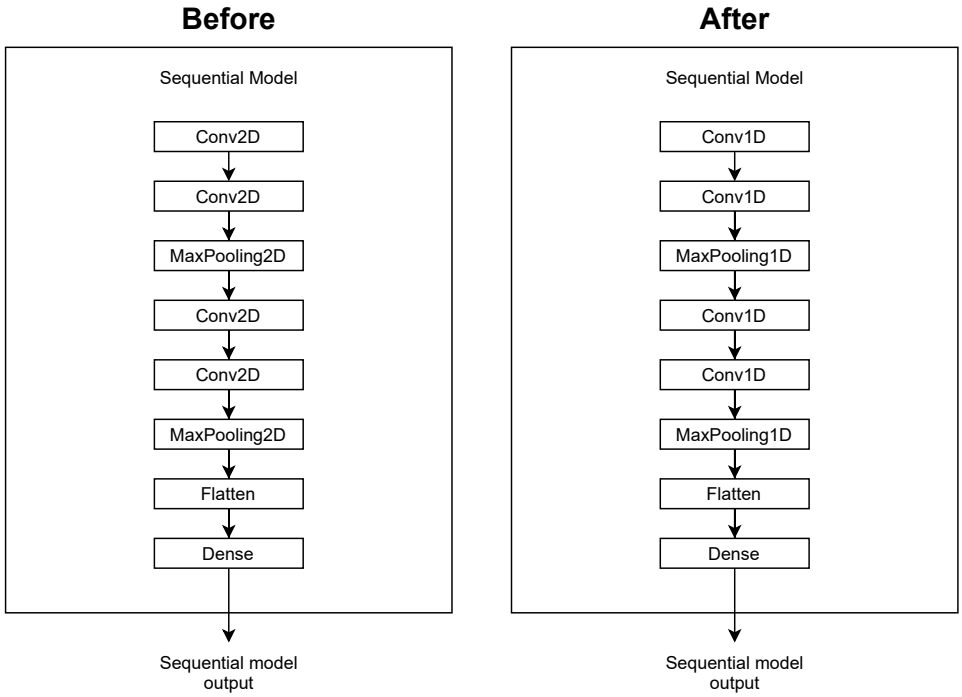

Figure 3.2: Comparison of the Convolutional Neural Network of initial model before and after our modification. This is the inner architecture of the box labeled "Convolutional Neural Network" in the Siamese architecture in figure 3.1.

true positives (1). For example, if the first chunk in `input_1` comes from participant A, and the chunk in the same position of `input_2` also comes from participant A, then the ground truth at that position in the `labels` array would be 1; otherwise, it would be 0.

3.6.2 Initial Test

With the inputs organized and ready, we began the first experiment. It was conducted with a minimal part of the dataset as its main purpose was to confirm that the model would compile and train successfully with the given inputs. This experiment was conducted using 3 weeks of data from 2 participants. Although the code was able to run without issues, the results were not so good with the model's loss value remaining unchanged during the training process. This would suggest that the model did not learn much.

We tested the resulting model with another 3 weeks of data from each participant. These were 3 weeks of data not used in training. The test was conducted by using the model's `predict()` function in order to predict whether two chunks of data originated from the same participant or not. Every single test resulted in the same similarity score of around ~0.50. This confirms the aforementioned suspicion that the model did not learn much,

as these scores suggest the model is guessing based on probability. Since there were only two participants involved, there was a 50% chance that either is the correct answer, which is why the model would produce around 0.50 similarity score between any two chunks.

3.6.3 Improving Testing Conditions

At this point, in order to make the experimentation process more efficient, we improved the code for processing data into chunks and arranging the inputs, by creating functions for these. The functions would take as parameters the number of weeks and participants to be used, thus minimising the work required to perform experiments with varying sizes of data.

3.6.4 Experimenting with Different Data Sizes

We then used this new function to run multiple experiments with different parameters for the number of weeks and parameters. These experiments showed positive improvements in the sense that the model was now actually learning something when we increased the amount of data used for training. However, the predictions remained highly inaccurate.

3.6.5 A New Approach to Input Arrangement

A different approach to inputs was also tried. The old approach arranged chunks in the arrays `input_1` and `input_2` such that each chunk was compared with every chunk in the set, thus meaning `input_1` and `input_2` were each of length N^2. The new approach would instead create two identical input lists, then randomly shuffle them, keeping the lists as length N. This means every chunk would be compared to a single, random other chunk. This method would make `input_1` and `input_2` each be of length N instead. Before shuffling, each chunk in the arrays would be turned into a tuple which included a participant ID in addition to the chunk to ensure the ground truths in the `labels` array would be correctly attributed. After the inputs were shuffled and the `labels` array created, the participant IDs were removed again, and all inputs were converted to arrays for training. This approach did not see much improvement over the old one besides a slightly improved execution time, as arrays of length N (new version) naturally compute faster than arrays of length N^2 (old version).

3.6.6 Balancing the Dataset

We hypothesised that one of the causes for the model's poor performance might be due to the data used for training being imbalanced, as this has been an issue in other machine learning applications such as [83]. If we consider the following:

- N is the total number of chunks (for all participants combined).

- P is the number of participants from which we have data-chunks.
- X is the number of chunks per participant, in other words $\frac{N}{P}$.

With our original input method, there would be N comparisons made for every single chunk. Out of these, only X comparisons would be a true positive as there are only X chunks from the same participant. In such a case, if there were 16 participants there would be 15 true negatives for each true positive comparison, leading to an imbalanced training dataset. When the data is imbalanced in such a way, it significantly lowers the probability that a random prediction between two chunks is a true positive due to it being perceived as statistically unlikely. This would, in turn, incentivise the model to predict low scores (i.e., predict comparisons as non-similar) simply because it has a high probability of being correct.

To solve the issue of data imbalance, we implemented yet another approach to creating the inputs for the model. In this approach, we split up the chunk-matching process into several steps. For each chunk C prepared by the function in section 3.5.1, the following actions are performed:

1. Create temporary lists `same` and `not_same`.
2. Iterate through the list of chunks, storing those having the same participant as C in `same`, and those from other participants in `not_same`.
3. Shuffle same and `not_same`, then use slicing to cut `not_same` into being the same length as `same`.
4. Concatenate `same` and `not_same` into a single list `to_add`, and shuffle it.
5. Append the `to_add` list to `input_2` and append the value C to `input_1` n times, where n is the length of `to_add`.

This approach ensures we have an equal amount of true positives and true negatives in the training dataset. In other words, the training dataset will now be balanced. Using the same example as previously, if we have three chunks in total (N=3), from the function mentioned in section 3.5.1 and name them `c1`, `c2` and `c3`, the input arrays would now be:

input_1	c1	c1	c2	c2	c3	c3
input_2	c1	c2	c2	c3	c3	c1
labels	1	0	1	0	1	0

Table 3.4: Example of what the balanced inputs look like.

3.6.7 Implementing `EarlyStopping`

We also implemented a callback function for the model, called `EarlyStopping`. This is a callback function gotten from the Tensorflow API which allows the model to stop training by itself after X epochs without improvement.

X is the patience value parameter for `EarlyStopping`, while the monitor parameter determines what metric the model should use to check for improvement between epochs. We used a patience value of 15, and `val_loss` (validation loss) as the monitored value.

3.6.8 Discovering and Fixing Bugs

When we tried to conduct an experiment using the entire dataset, we encountered some issues. Some of the chunks were smaller than they should be, suggesting there was missing data. We had originally assumed that all participants had data from 1. November - 31. March. However, upon closer inspection of the original `steps.json` data files, we realised that there were actually some discrepancies in the datasets between the different participants. Some either began at a later date, ended at an earlier date or both. This naturally leads to issues in the data processing since we originally assumed they were all sharing the same time frame.

To solve this issue, we envisioned three possible solutions:

1. Use only data from the participants with complete datasets (1. November - 31. March, amounting to 21 whole weeks). Only 6 participants had complete datasets.

2. Use only the data from a period that is included in every dataset (e.g. take the entire period of the smallest dataset, and the equivalent period from each other dataset). This would amount to 14 whole weeks of data for each participant.

3. Exclude the participants with the least amount of data. Participants 10 and 11 had just over 14 weeks, whereas every other participant had at least 18 weeks worth. This option would result in 14 participants with 18 weeks of data from each.

Optimally we would run tests using each of the proposed solutions in order to determine what was the best approach. However, due to time constraints, we would instead choose one out of the three, so we could prioritise testing and improving the model. In the end, we chose option 2, as this would allow us to experiment with all participants at once without any issues. We prioritised more participants over more data as this would give us more variety in the data, and we believed this would provide the best results.

3.6.9 Summary - Phase One

In this experimentation phase, we have mainly been improving the method in which we conduct experiments. We have improved efficiency in how to modify the quantity of data to be used with each experiment, as well as found what we believe to be the most optimal (balanced) method of arranging inputs for training the model. The actual model itself has not seen much progress. However, the improvements made

to the experimentation process itself will likely result in higher quality experiments going forward.

3.7 Heart Rate

At this point, due to the lack of results thus far, we decided to switch from using step data to using heart rate data instead.

3.7.1 Data Processing

The Fitbit dataset for heart rate is found in a file called `heart_rate.json`, as seen in table 3.2.4. Heart rate is also a time series with values recorded roughly every 5 seconds. The recorded values are a timestamp, beats-per-minute (bpm) and a confidence value representing the error margin for the bpm value. The timestamp is on the format "yyyy-mm-dd HH:MM:SS", whereas bpm and confidence values are integers.

Again, like just like we saw with steps in section 3.5.1, heart rate data is also inconsistent. However, with heart rate, it is not just an occasional missing value. Instead, the timestamps would shift by 1 second occasionally. For example if the previous timestamp was "2019-11-01 00:00:05", the next might be "2019-11-01 00:00:11" rather than the expected "2019-11-01 00:00:10".

To solve this and make the data consistent, we processed the data with the following steps:

1. Loaded the data into a pandas dataframe.
2. Resampled the dataset to a 1-second frequency using the pandas `resample()` function. By doing this, there was no longer an issue of inconsistency in timestamps.
3. Used linear interpolation to fill in bpm values for the newly created timestamps.
4. Round off the bpm values and cast them to int (the pandas `interpolate()` function creates float-values).
5. Resample the dataset back to its original 5-second frequency. The dataset will now be consistent with a bpm value every 5 seconds.
6. Remove the confidence and timestamp columns, leaving only bpm.

We decided to fill values by linear interpolation as we believe this will result in a reasonable and realistic dataset. Filling with 0 as we did for steps would naturally not work here as a living human's heart rate should never be 0.

Finally, due to the heart rate dataset being much larger than the steps dataset, we decided to have each chunk contain one day of data, rather than one week as we did with steps. This is so the input shape for the model would not be too large.

3.7.2 Additional Features

Initially, beats-per-minute (bpm) was the only feature used. We decided to expand this by adding day-of-the-week features, encoded as one-hot vectors [84]. To implement this, we added a few steps to our data processing. These steps would take place between step 5 and 6 from the process detailed in section 3.7.1:

1. Create a column with "weekday" values, which are integers between 1-7 (where 1 is Monday and 7 is Sunday), corresponding to the day of the week for a given timestamp.
2. Create a column for each day: mon, tue, wed, thu, fri, sat, sun.
3. Fill values in the newly created columns. These values will be 1 if that is the day corresponding to the weekday value created in step 1, or 0 otherwise.
4. Remove the weekday column.

Table 3.5 is an example of how each row would look like after being processed with these new steps included.

bpm	**mon**	**tue**	**wed**	**thu**	**fri**	**sat**	**sun**
68	0	1	0	0	0	0	0

Table 3.5: Example row from a processed heart rate dataset.

With these one-hot encoded day-of-the-week features, the model would be able to see correlations in heart rate patterns between different days of the week as well, hopefully allowing it to learn more about the dataset.

3.7.3 A New Environment

Due to the heart rate dataset being much larger than steps with having one value every 5 seconds as opposed to one value every minute, we encountered some challenges. The first was the model not being compatible due to the larger dataset, which was solved by simply changing the input shape to match the new size. The other challenge was hardware. Up until this point, we had been working locally on a private computer which did not have the resources to handle heavier computations. For that reason, we transitioned to working remotely on Simula's supercomputer "ex3" instead [85].

3.8 Experimentation - Phase Two

The following experiments were conducted using 16 participants, 7 weeks of training data and 3 weeks of testing data unless stated otherwise. The experiments also made use of the `EarlyStopping` function with a patience

value of 10, monitoring loss. This function will cause the model to stop its training if it goes on for 10 epochs without improvements in the loss value. All experiments used the same layers as shown in figure 3.2 unless otherwise stated.

3.8.1 Experiment 1: `kernel_size=64`.

Using the base model with the kernel size parameter to the `Conv1D` layers set to 64.

Training Loss	Training Accuracy	Test Score
0.6932	0.4770	0.5000

Table 3.6: Results for experiment 1. Test score is an example output value, as all tests output the same, or an extremely similar score.

Both the loss value and accuracy stabilised within the first epoch and remained unchanged until the training finished. The final epoch had a loss of 0.6932 and accuracy of 0.4770. All tests resulted in the exact same similarity score of 0.5000136. We concluded that the model is learning very little, if anything.

3.8.2 Experiment 2: `kernel_size=128`.

Mostly the same as experiment 1, with the difference being a larger kernel size parameter for the `Conv1D` layers. The results for this experiment can be seen in table 3.7.

Training Loss	Training Accuracy	Test Score
0.6932	0.4923	0.5001

Table 3.7: Results for experiment 2. Test score is an example output value, as all tests output the same, or an extremely similar score.

Just like with the previous experiment, both loss and accuracy stabilised in the first epoch and barely changed until completion. The resulting loss, accuracy and test score was also similar. The similarity score produced when testing was identical for every test case here as well, so we once again conclude the model is not learning. We also conclude that kernel size seemingly has little to no impact on the model's performance.

3.8.3 Experiment 3: No `Flatten` and `MaxPooling1D` Layers.

This experiments has both `MaxPooling1D` layers and the `Flatten` layer disabled.

Epoch	Training Loss	Training Accuracy	Test Score
46	**0.6552**	**0.6431**	**N/A**
56	0.6980	0.5000	0.5502*

Table 3.8: Results for experiment 3. Best result is indicated in bold. Test score has been marked with * because the output had an unexpected format of a list, rather than a single value. The displayed value in the table is the average of the values in the list.

Here we actually saw a development in both loss and accuracy, where they gradually improved over the first 46 epochs before getting significantly worse again in the last 10 epochs. The epoch that had the best results, before the decline, can be seen in table 3.8 marked in bold. We were not able to test this epoch's model as it would be overwritten by subsequent epochs in the current implementation. Because of this, we realised we would need a method to test earlier iterations of a model, considering the last epoch was far from the best one in this particular case.

Since there was a development in the training process, we concluded that the model is actually learning something. It would be difficult to determine how well this model performs as we were unable to test the best iteration. However, one strange occurrence in this experiment is that the output format of the model has changed. When obtaining a similarity score between two chunks in testing, we would now get a list of scores rather than a single score like we expected. We assume this is due to disabling the `Flatten` layer, because `Flatten` changes the output shape. `Flatten` works by taking in a shape of N dimensions and transforming this to a single dimension output [86]. For example:

```
[[[ 1,  2,  3],
  [ 4,  5,  6]],
 [[ 7,  8,  9],
  [10, 11, 12]]]
```

will be transformed to

```
[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]]
```

when sent through a `Flatten` layer. In the next experiment, we will verify our assumption that removing the `Flatten` layer was the cause for the strange output format.

3.8.4 Experiment 4: No `Flatten` Layer.

This experiment uses the base model with the `Flatten` layer disabled, in order to verify the assumption made in the previous experiment: that the strange outputs are caused by the `Flatten` layer being disabled.

Epoch	Training Loss	Training Accuracy	Test Score
28	**0.6556**	**0.6568**	**N/A**
38	0.6938	0.5000	0.5182*

Table 3.9: Results for experiment 4. Best result is indicated in bold. Test score has been marked with * because the output had an unexpected format of a list, rather than a single value. The displayed value in the table is the average of the values in the list.

The results in this experiment are very similar to the previous one, and the outputs is still a list of scores rather than a single value. This confirms our assumption that the lack of a `Flatten` layer is responsible for the strange output format. By comparing the results in table 3.8 (experiment 3) to those in table 3.9 (experiment 4), we also see that turning the `MaxPooling1D` layers back on had little effect on the overall results for this model.

3.8.5 Summary - Phase Two

In this experimentation phase, we have established the baseline results for the base (i.e. unchanged) model, before modifying it and assessing what effect these modifications had. We have explored how flattening the feature maps from the convolutional layers changes the behaviour the `Dense` layer, seeing as the model actually started learning when we removed the `Flatten` layer. Unfortunately, this modification also changes the shape of the final output for the model, making it difficult to assess the actual final similarity score, so we will have to reinsert the `Flatten` layer to solve this. However, we may have gotten a clue as to why the model was performing poorly since the `Dense` layer seemingly performed much better with a non-flattened input.

3.9 Improvements

In order to gain better insight into the model's performance, we added validation data to the inputs when training the model. The training dataset was split in two, training and validation, where 70% was training data and 30% was validation data. We also implemented plotting of learning curves to make it easier to see how well the model performed during training, as well as how it developed throughout the training process. Lastly, we implemented another callback function from the Tensorflow API called `SaveCheckpoint`. This function saves the best iteration of the model from the training process to a file, so you can load and use it later. A value to be monitored (loss or accuracy for example) is supplied as an argument to the callback function, and this value is used as the criteria by the function to assess which epoch produced the best model. We have chosen to use `val_loss` (validation loss) as the criteria for what is considered the best

model. The epoch that achieves the lowest value for `val_loss` will be considered the best result, and therefore saved by the `SaveCheckpoint` function.

We believe it would be a good idea to test both the model produced by the best epoch (as saved by the `SaveCheckpoint` function), and the final epoch before training ends. This would allow us to compare the results between the two, and confirm whether the presumed "best" version actually achieved the best results.

3.10 Experimentation - Phase Three

These experiments were conducted after the improvements above, and will therefore include plotted learning curves for additional insight. The experiments themselves will be mainly focused on testing different compositions of layers in order to see if the model can be improved.

3.10.1 Experiment 5: Base Model.

This experiment uses the model in its initial state after being converted to 1D as seen in figure 3.2. The results can be seen in table 3.10, and the learning curve in figure 3.3.

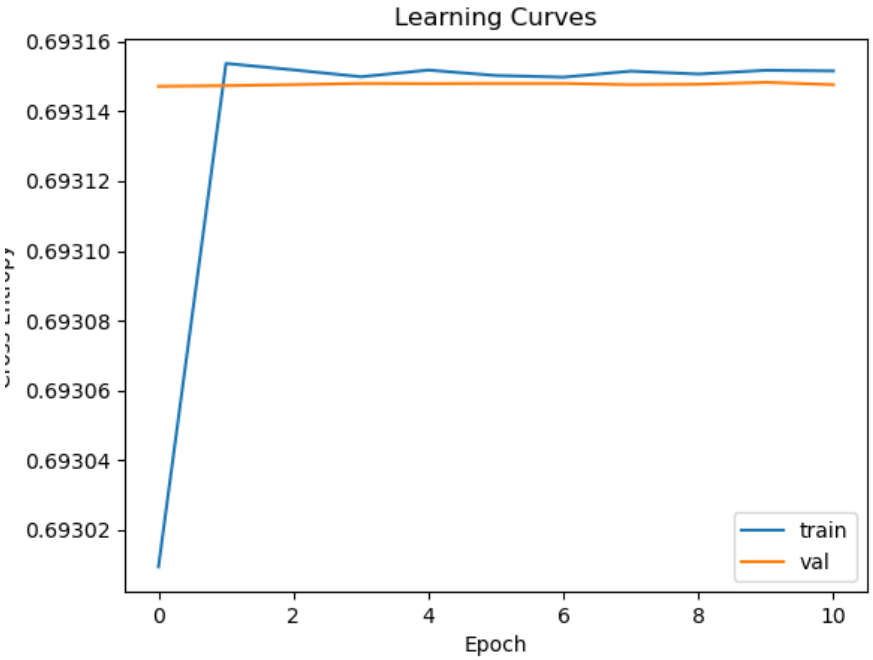

Figure 3.3: Learning curve for experiment 5.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Accuracy
1	0.6930	0.4913	0.6931	0.5011	N/A

Table 3.10: Results for experiment 5. Only shows 1 epoch as there was very minimal difference between epochs overall.

As seen in figure 3.3, the base model is not learning at all. The curve stabilises within the first epoch and barely changes afterwards. The "Test Accuracy" column in table 3.10 has been marked "N/A" because every single prediction using the model on test data, returns the exact same similarity score of 0.50000334. Thus, we can not determine accuracy as any tests passed would have been due to sheer luck. Given the fact that the training dataset was balanced with an equal number of true and false comparisons, it is reasonable to believe that the model is guessing purely based on probability rather than actually predicting similarity. We are not interested in test results on whether the model guessed correctly by pure chance. We will therefore have to experiment with the layers in the model to see if we can get it to learn.

3.10.2 Experiment 6: Replication of Experiment 4.

This is experiment 4 replicated after the improvements mentioned on page 33 in order to see its learning curve, as well as be able to test the best epoch. The results can be seen in table 3.11 and the learning curve in figure 3.4. We believe there was value in replicating this experiment after improvements in order to see the learning curve as well as being able to test the best epoch since there was a large discrepancy between the performance of the best epoch and the last epoch for this experiment. As with experiment 3, four `Conv1D` layers, two `MaxPooling1D` layers and one `Dense` layer is used, in the same order as in figure 3.2. The `Flatten` layer is not used.

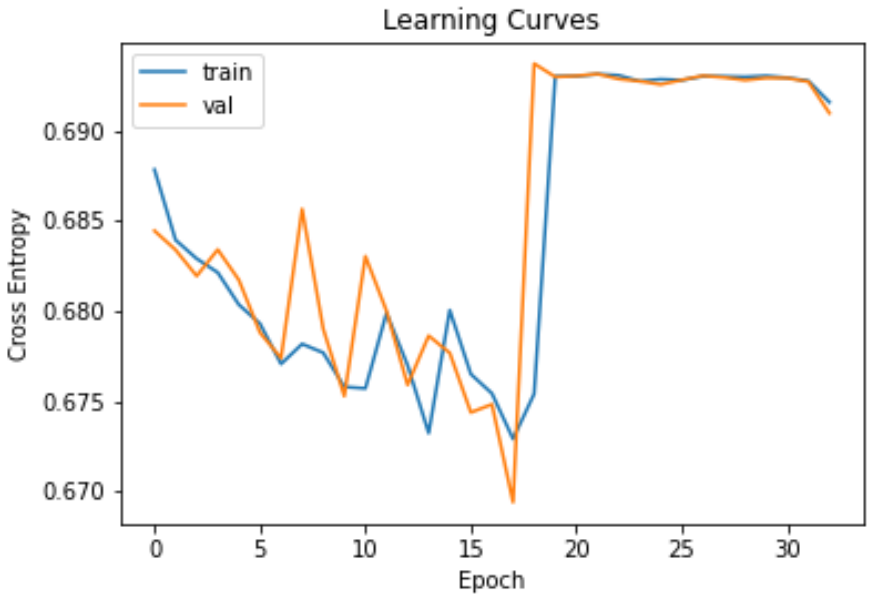

Figure 3.4: Learning curve for experiment 6.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Accuracy
18	**0.6729**	**0.6319**	**0.6694**	**0.6422**	**N/A**
33	0.6916	0.6011	0.6910	0.6480	N/A

Table 3.11: Results for experiment 6. Best result is indicated in bold.

Test accuracy has not been computed as it is difficult to interpret the output when the `Flatten` layer was removed, as noted in experiments 3 and 4. It is therefore marked as "N/A" in table 3.11. If we compare the learning curves in figure 3.4 (this experiment) and 3.3 (experiment 5: base model), we see that by removing the `Flatten` layer, the model is actually learning something. This suggests that flattening the output from the `Conv1D` layers somehow interferes with the `Dense` layer's ability to learn. We will therefore attempt to move the `Flatten` layer, so it comes after `Dense` for our next experiment, to see if this affects the results.

3.10.3 Experiment 7: `Flatten` After `Dense`.

In this experiment, we tried a different approach where we rearranged the order of the `Flatten` and `Dense` layers. The `Flatten` layer would now come last, after the `Dense` layer. The rest of the layers are as they appear in figure 3.2. The results can be seen in table 3.12 and the learning curve in figure 3.5.

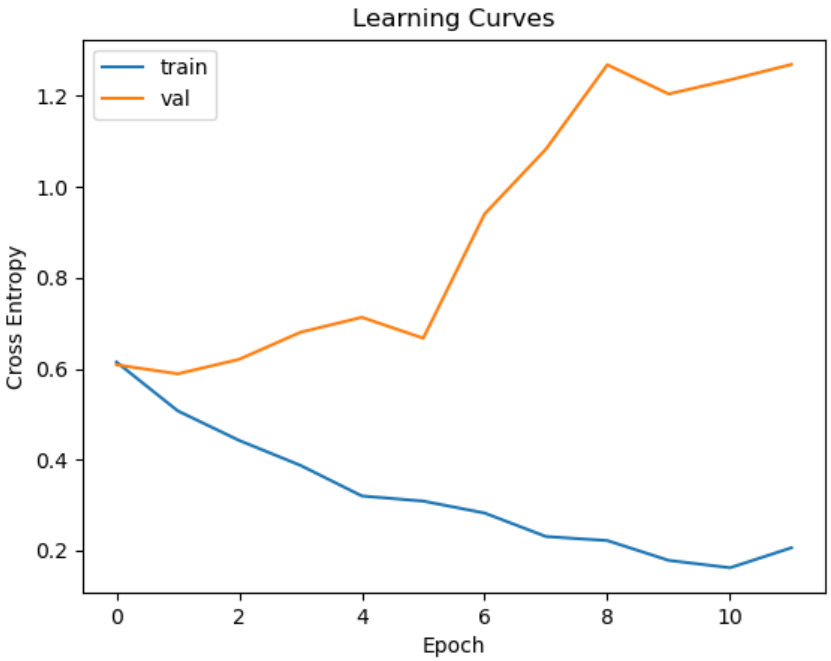

Figure 3.5: Learning curve for experiment 7.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Accuracy
2	**0.5073**	**0.7767**	**0.5889**	**0.6943**	**0.7743 (223/288)**
12	0.2062	0.9471	1.2694	0.6476	0.7674 (221/288)

Table 3.12: Results for experiment 7. Best result is indicated in bold. Test Accuracy shows the number of tests passed in parenthesis.

The model in this experiment has produced the best results so far, with the lowest loss values, highest accuracy and as many as 77.4% of tests passed for the "best" version in table 3.12. However, as the learning curve in figure 3.5 shows, the loss and validation loss diverge almost immediately with validation loss continuously rising. This suggests the model is overfitting significantly, which is a potential for improvement. We believe the great increase in performance can be attributed to the `Dense` layer now being able to learn from the convolutions produced by the preceding `Conv1D` layer without them being flattened first. We have to keep in mind that this experiment was conducted on a relatively small part of the available dataset, so the natural next step will be to experiment using this same model with more data. If this were to be equally successful, it would be worthwhile looking into ways of solving the issue of overfitting.

3.10.4 Experiment 8: `Flatten` After `Dense` - More Data.

This experiment is a replication of experiment 7 with additional data being used both for training and testing. For this experiment, we used 50 days of heart rate data for training, and 20 days for testing.

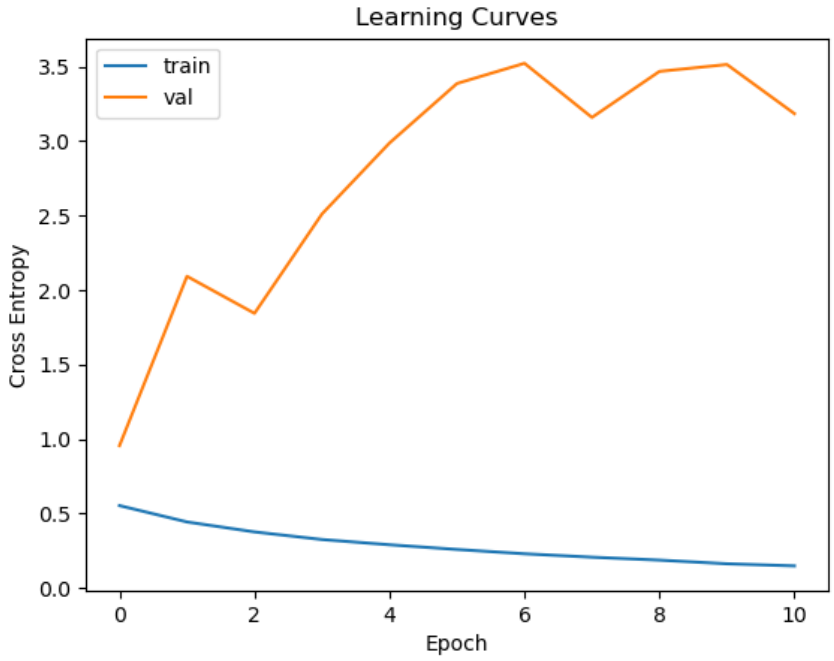

Figure 3.6: Learning curve for experiment 8.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Accuracy
1	**0.5519**	**0.7209**	**0.9548**	**0.5288**	**0.7052 (9027/12800)**
11	0.1611	0.9431	3.5154	0.5372	0.6651 (8513/12800)

Table 3.13: Results for experiment 8. Best result is indicated in bold. Test Accuracy shows the number of tests passed in parenthesis.

As we can see from the learning curve in figure 3.6, the issue with overfitting has increased significantly from the previous experiment, with validation loss trending upwards immediately after the first epoch. Despite this, it still performs relatively well in testing, with 70.5% of tests passing ("best" version from table 3.13. From this, we can infer that overfitting gets worse as the dataset increases in size, and the performance has likely been

impacted because of this. However, it still remains as the best model we have produced thus far, considering the test results.

It would be a good idea for a future experiment to try methods for solving the overfitting problem such as adding `Dropout` layers, and test if this improves performance.

3.10.5 Summary - Phase Three

In this experimentation phase, we retested some of the experiments from the previous phase in order to get their learning curves. As we mentioned in the summary for phase two (Section 3.8.5), the `Flatten` layer seems to cause problems for the following `Dense` layer, resulting in the model not learning. When removing the `Flatten` layer, the output of the model was no longer on the expected format, so this did not solve the problem. In phase three we built upon these discoveries by trying to place the `Flatten` layer *after* the `Dense` layer instead, which resulted in a significant improvement in the model's performance, with testing accuracy reaching as high as 77.4%. This iteration did, however, have some significant overfitting issues as seen in the learning curve in figure 3.5.

3.11 New Model

We decided to get inspiration by implementing and testing a new Siamese architecture. This architecture was first proposed in a paper by Koch et al. [62] and later implemented by Harshall Lamba [63]. This model's inner architecture of the "Convolutional Neural Network" part of the Siamese architecture illustrated in figure 3.1, can be seen in figure 3.7.

The difference between this model and the previous one (section 3.5.2) is the inner architecture (layer composition) of the "Convolutional Neural Network" part of figure 3.1. The higher level Siamese architecture is functionally the same for both. The main difference between the inner architectures is that this version has three `MaxPooling1D` layers which are all positioned between two `Conv1D` layers, whereas the previous architecture had two `MaxPooling1D` layers which were positioned after every other `Conv1D` layer. In addition to this, the new model had a significantly higher value supplied as the parameter for determining the dimensionality of the output shape from the `Dense` layer. The new model also had its own custom functions for the layer arguments: `kernel_initializer`, `bias_initializer` and `kernel_regularizer`.

We will be focusing on the implementation of this model by Harshall Lamba [63]. This architecture was designed to recognise the similarity between images of physical signatures, in an effort to distinguish whether a signature is authentic or not. In other words, just as with the architecture by Shubham Panchal [57], this new architecture was also designed to take images as input, so we would need to modify it before we can use it for Fitbit data. The modification was done in the same way as for the previous architecture we used, in that we changed all `Conv` and `MaxPooling` layers

from 2D to their 1D counterparts, as well as modifying parameters where necessary. We also had to reduce the size of the output dimensionality for each of the `Conv1D` layers and the `Dense` by half in order to reduce the memory usage of the model, because our input shape of `(17280, 8)` was so large it would lead to an "Out of Memory" exception otherwise. After switching the layers to 1D and reducing the output dimensionality of layers, the new model architecture was ready to train with Fitbit data.

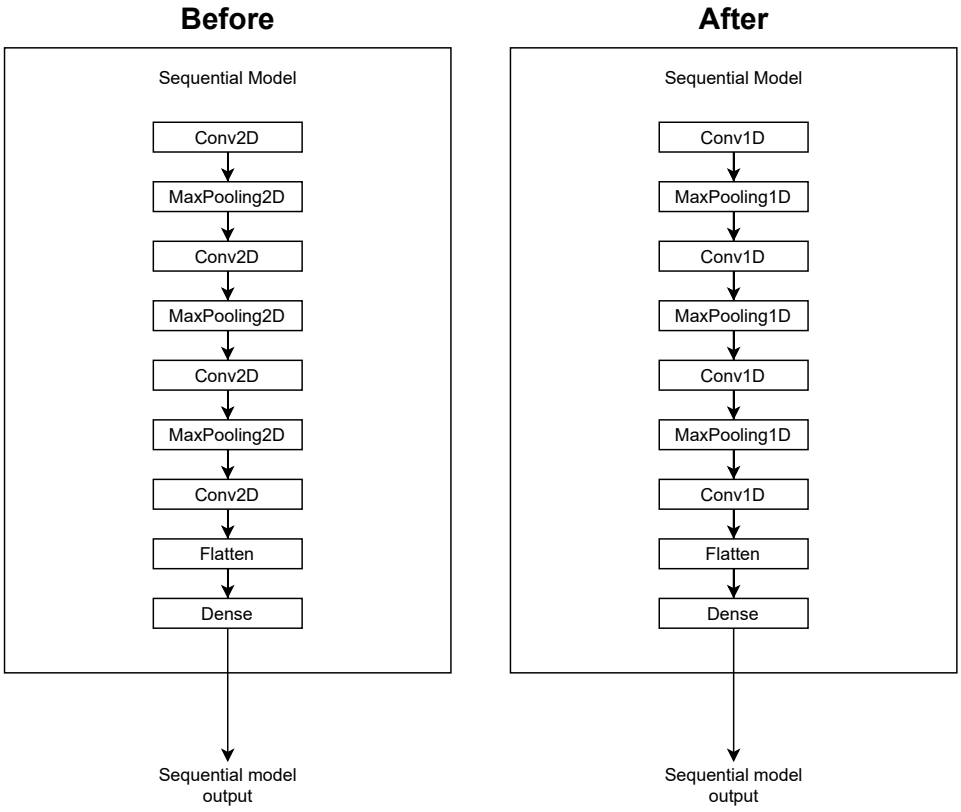

Figure 3.7: Comparison of the Convolutional Neural Network of the new model before and after our modification. This is the inner architecture of the box labeled "Convolutional Neural Network" in the Siamese architecture in figure 3.1.

3.12 Experimentation - Phase Four

The following experiments were conducted using 16 participants, 7 weeks of training data and 3 weeks of testing data unless stated otherwise. The experiments will be making use of the new model architecture described in section 3.11.

3.12.1 Experiment 9: New Base Model.

This experiment uses the new model exactly as it is described in section 3.11. The results can be found in table 3.14 and the learning curve in figure 3.8.

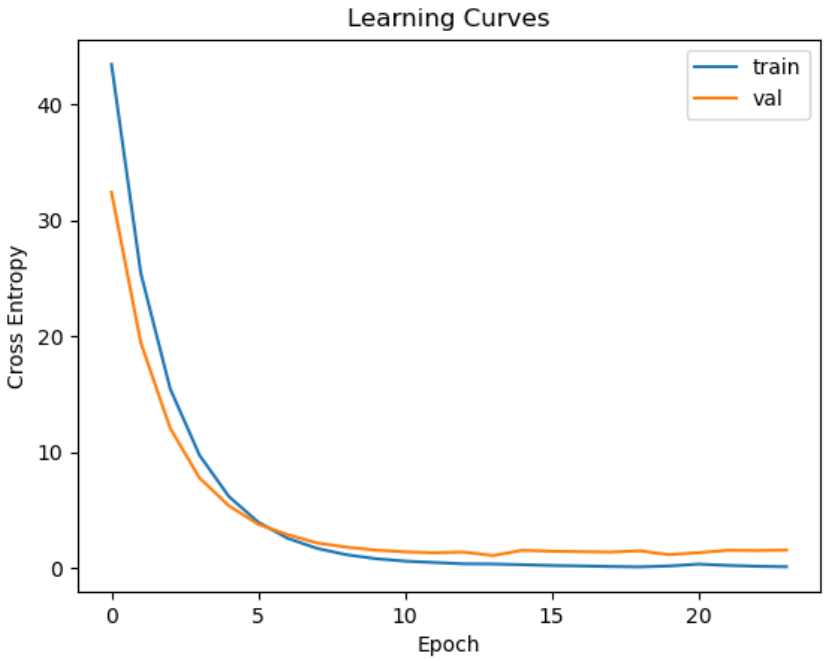

Figure 3.8: Learning curve for experiment 9.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Accuracy
14	**0.3665**	**0.9854**	**1.0976**	**0.6561**	**0.7292 (211/288)**
24	0.1288	1.0	1.5658	0.5732	0.7292 (210/288)

Table 3.14: Results for experiment 9. Best result is indicated in bold. Test Accuracy shows the number of tests passed in parenthesis.

We can see from the results in table 3.14 that this new model, in its initial state, is already performing much better than the previous architecture's base model test in experiment 5. This is also reflected in the learning curve in figure 3.8. In fact, this model is not far behind our current best results in experiment 7.

Another observation to note here is that the loss value began at an incredible 40+, way higher than what we have seen in previous experiments,

before quickly dropping down to normal levels and stabilising. We believe this is likely due to the custom initialiser functions for kernel and bias that are supplied to the layers of this model. We will test this model once more with a larger dataset in the next experiment, to see if it works just as well with more data.

3.12.2 Experiment 10: New Base Model - More Data.

This experiment is the same as experiment 9, but with 50 days of training data and 20 days of test data used instead.

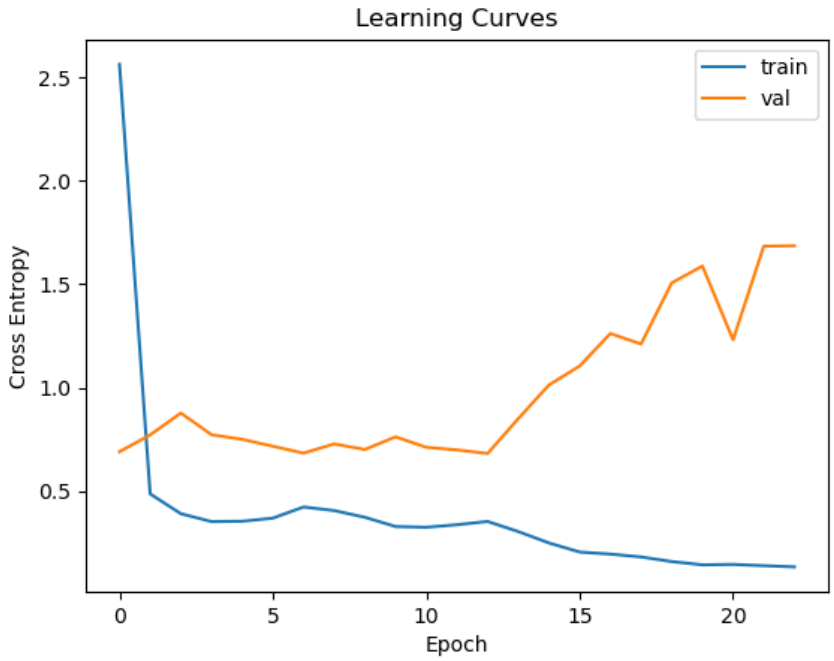

Figure 3.9: Learning curve for experiment 10.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Accuracy
13	**0.3541**	**0.9036**	**0.6825**	**0.6876**	**0.6836 (8750/12800)**
23	0.1346	0.9832	1.6865	0.5965	0.5917 (7574/12800)

Table 3.15: Results for experiment 10. Best result is indicated in bold. Test Accuracy shows the number of tests passed in parenthesis.

This experiment achieved similar results, if not slightly worse results than experiment 9. In loss, validation loss and validation accuracy, this

experiment is superior. However, we believe it is more appropriate to judge its performance in terms of actual accuracy by looking at the number of tests it passed, in which experiment 9 performed better. From this, we can infer that the model starts struggling to differentiate between participants when more data is involved, as experiments 9 and 10 are the same except for the different sizes in data used.

3.12.3 Summary - Phase Four

In this experimentation phase, we have implemented and tested a new model architecture. This new model performed significantly better than the initial version of the previous model, seen in experiment 5 (table 3.10). However, it did not perform quite as well as our improved version of the initial model from experiment 7 and 8. The new model also uses `Flatten` before `Dense`, so it would have been useful to test whether this model could be improved as well by moving the `Flatten`, but we were not able to do this due to time constraints.

3.13 Discussion

When implementing the Siamese neural network, we have been using steps and heart rate data for reidentification. We used steps initially due to being easier to implement. However, as we saw in phase one of our experimentation, step data alone was not enough to be able to distinguish between and reidentify participants in the dataset. We believe the reason for this is due to the nature of step data. As a person will not be moving around all the time, many values in the dataset will be 0, as no steps are taken in a given minute. This means the data from different participants will be less unique as they all have a large number of 0-values.

Heart rate, on the other hand, will realistically never be 0 as long as it is worn by a living human. Thus, the heart rate dataset will not have the same issue as steps. Because of this, heart rate data is likely to be more unique for each participant, making it possible for the Siamese neural network to differentiate between them.

Although we only made use of steps and heart rate, other fitness data could be useful for reidentification as well. Sleep data, for example, could be an interesting area to find patterns to use for distinguishing between people, as this is (generally) a daily occurrence for everyone and may differ in length, quality and composition between individuals. Exercise data could be interesting as well, though this data is highly irregular because people do not necessarily exercise at the same time, for the same amount of time per exercise, and they might not do the same activities in the exercise. Thus, it could prove challenging to make comparisons between participants in a dataset using exercise data only.

As we have discussed, multiple datasets could be useful for reidentification purposes, and we believe there is merit to combining these as well. In theory, the more data (and different kinds of data) you have for each par-

ticipant, the more unique their individual daily patterns will be. In turn, this would make reidentification easier. We would have liked to test this theory, but due to time constraints, we will be leaving this for future work.

One of the limitations for this thesis, as mentioned in scope and limitations (section 1.3), is that the dataset only has 16 participants. Due to the relatively small sample size, there is no guarantee that our implementation will achieve similar results on a larger dataset involving 100+ people. In a larger sample size, there would be more cases of people with similar patterns or heart rate averages, which would make it more challenging for our Siamese neural network to reidentify individuals successfully. In the PMData dataset, we did not have many issues with this as the dataset had much variety in that the participants were of different demographics, having different fitness levels etc. For these reasons, we can not say that our implementation is general enough to be applied to datasets of varying sizes, as we have only tested with PMData. Testing with larger datasets would be a natural course of action in future work.

Another limitation is that we have only tested with data collected from Fitbit devices. Without testing with data from other devices, we can not guarantee that our implementation will achieve similar results. However, we believe it should work similarly with data from other devices, as long as the following conditions are true:

1. The data contents is the same and on a similar format, allowing for the processed result to match the format of the processed Fitbit data.

2. The data collection accuracy of other devices to be used are similar to the accuracy of the Fitbit Versa 2, which we used. Accuracy can affect data quality, and the data quality could affect our model's performance.

3.14 Summary

In this chapter, we have implemented a Siamese neural network architecture that is capable of reidentifying individuals in the PMData dataset with a 77.4% accuracy. We achieved this by first implementing an existing solution and iterating over this implementation, running experiments and making modifications and improvements based on the results. We have also tested this implementation with both step data and heart rate data, with heart rate proving to be more effective for reidentification.

Over the course of the experimentation process, in addition to experimenting with and improving our implementation, we have also improved our experimentation methods themselves by adding elements such as the `SaveCheckpoint` callback function, allowing us to test the model produced by both the best and last epochs and compare them, as well as implementing learning curve plotting for a better overview of results.

We have also discussed our findings and noted some of the limitations for our thesis, which ultimately lead to not being able to conclude that our implementation is general enough to work for larger datasets as we have

not tested this. However, it should be able to work reasonably well with data collected from devices other than Fitbit, given that the data is on a similar format.

Chapter 4

Conclusion

4.1 Summary

We live in a world where enormous amounts of data are collected and stored each day. This data also includes a lot of personal data, which is a cause for concern in regards to privacy rights. To alleviate these concerns, lawmakers worldwide have begun to strengthen privacy laws by restricting what data can be legally collected and stored and putting an emphasis on data subject rights. The GDPR [1], for example, makes it harder to share or sell personal data without each person's explicit consent, unless the data is anonymised.

Deep learning is being widely researched and applied in more and more scenarios with each passing day. Data analysis is an area which has especially seen frequent use of deep learning algorithms, as they allow for increased learning and value obtained from data. This naturally applies to personal data as well, which once again raises concerns regarding privacy. We have seen other works which call standard anonymisation processes into question as deep learning has proven capable of reidentifying individuals in anonymised datasets.

In this thesis, we have looked at how neural networks can be used to reidentify individuals in an anonymous dataset. We explored this by implementing a Siamese Neural Network architecture which takes two chunks of data, obtain their features by using deep learning, and compares these features to compute a similarity score. This score is then used to determine whether these chunks of data come from the same person or not. By comparing data and producing similarity scores in this way, our implementation is capable of identifying which chunks of data in a dataset belong to a given person. We have tested and improved this model over several iterations on the Fitbit dataset available in PMData, ending up with a version that achieved an accuracy of up to 77.4%.

In other words, we can reidentify a person in an anonymised dataset with reasonable accuracy, as long as we possess a small portion of identified data for that person beforehand. The Siamese neural network works by comparing data and determining whether they originate from the same person or not. Therefore, if we are to reidentify someone in an anonymised

dataset, we need to have one or more chunks of data from that person, which we can use with the Siamese neural network to compare to the chunks in the anonymous dataset, which will allow us to find out which of the anonymous participants corresponds to person A. As an example, if we have one or more chunks of data from person A, and we want to identify which of the anonymous participants in a dataset correspond to person A, we can use the chunk(s) of data from person A and compare these with the chunks in the dataset using the Siamese neural network. The resulting similarity scores will allow us to identify which of the participants in the dataset corresponds to person A.

4.2 Contributions

As mentioned in section 1.5, we have implemented a solution to the problem of reidentifying participants in an anonymised dataset. This solution consists of a Siamese neural network which compares chunks of data to produce a similarity score, which determines the likelihood of these chunks of data originating from the same person. Our implementation achieved an accuracy of up to 77.4% by using heart rate data to distinguish between and reidentify participants of the dataset. We have also tested with step data, but this did not achieve any significant results, with only reaching just over 50% accuracy.

With this, we were able to answer the research questions from our initial problem statement:

Question 1 *Could a chunk (for example a day or a week) of deidentified physical activity data be considered a biometric identifier?*

Our implemented Siamese neural network is capable of distinguishing between anonymous participants in a dataset by computing the similarities between their data. It does this by taking two chunks of heart rate data as input, where a chunk is a day of beats-per-minute (bpm) values. These chunks are sent through identical neural networks to generate feature vectors for them. The absolute distance between these feature vectors is then used to calculate a similarity score, which can be used to determine how similar these chunks of data are. The hypothesis is that if two chunks of data compute similar features in the neural networks, the chunks are likely to belong to the same person due to having similar patterns.

This gives reason to believe that physical activity data can be used as a biometric identifier, as it can be used to identify a person in an anonymous dataset. An obvious limitation to this, however, is that humans change over time, and so does their activity patterns, meaning the Siamese neural network may not be able to reidentify individuals if the chunks of data to be compared are far apart in terms of when it was collected. Another limitation is that you need to already possess one or more chunks of data from the person you wish to identify, as you need something to compare to the chunks in the anonymised dataset.

Question 2 *Is it possible to use deep learning in order to reidentify individuals in a dataset that has been previously anonymised?*

As mentioned, our implementation was able to reidentify participants of a dataset with an accuracy of 77.4%, which proves that this is indeed possible. It is based on the theory that since humans are habitual creatures, their behaviour will likely have some sort of pattern, which would be reflected in their physical activity data. In such a case, one should be able to distinguish between individuals based on patterns found in their physical activity data. If these patterns are unique enough, they could be used as a form of biometric identifier which could allow a deep learning algorithm to identify an individual based on their physical activity data.

Objective *Implement a deep learning neural network architecture to reidentify individuals in an anonymised fitness dataset.*

Our implementation of a Siamese neural network takes two chunks of data as input, compares these and computes a similarity score based on the absolute distance between their features. A high similarity score means the patterns in each chunk is similar and is therefore interpreted as the two chunks belonging to the same person. With reaching accuracies up to 77.4% in testing, we can conclude that the implementation is capable of reasonably reidentifying anonymised data, as long as you have available data from the person you wish to identify, for comparison. The source code for this implementation can be found on GitHub here: https://github.com/Zeykes/Fitbit_Reidentification

4.3 Future Work

As of right now, the implementation achieves its goals to a reasonable degree on the PMData [15] dataset. We believe our implementation has the potential to improve even further by experimenting and testing the following:

- Experiment with other data such as sleep activity, distance travelled, exercise, or daily active minutes. We have only tested steps and heart rate, but there are much more data available in a Fitbit data set, or similar, that could potentially be used for reidentification purposes. Data from devices other than Fitbit should be tested as well, although we believe these should achieve similar results as long as the data is on a format similar to Fitbit.

- Combine data. By aggregating activity data such as steps and heart rate, and using them together, it would increase the factors that make up an individual's activity pattern. This would, in turn, make these patterns more unique from one individual to another, which could help the model more accurately distinguish individuals in cases where several participants are in similar heart rate ranges, for example.

- Train and test the model on a larger dataset. The dataset we used, PMData [15], was limited in terms of diversity due to only involving 16 participants. This means our model has not encountered situations with the possibility of several participants having similar activity patterns, which would complicate the reidentification process.
- Test the generality of the implementation. This thesis has only used a single dataset; however, there are methods to test whether the implementation is general enough to distinguish between data from individuals who were not part of the training dataset. One such method is three-cross validation, where one would split the dataset into three parts, train the model with one part and test with the other parts, then rotate. We would recommend splitting the dataset by participants so that each participant is only found in one of the three parts. This would test the model's ability to distinguish between unknown individuals.

In addition to the points above, we also believe there is room for improvement in the model itself. In particular, we observed some overfitting in the later iterations of the model, including the iteration which achieved the best results. One possible method for solving this is to add `Dropout` layers in the model.

Bibliography

[1] *EU General Data Protection Regulation.* European Commission. 25th May 2018. URL: https://gdpr-info.eu/ (visited on 20/11/2020).

[2] Alex Hern. 'Fitness tracking app Strava gives away location of secret US army bases'. In: *The Guardian* (28th Jan. 2018). URL: https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases (visited on 14/10/2019).

[3] Isobel Asher Hamilton. 'A fitness app exposed sensitive location details for thousands of users including soldiers and secret agents'. In: *Business Insider* (9th July 2018). URL: https://www.businessinsider.com/polar-exercise-fitness-app-exposed-soldiers-spies-location-details-2018-7?r=US&IR=T (visited on 14/10/2019).

[4] Agencies. 'Hackers steal data of 150 million MyFitnessPal app users'. In: *The Guardian* (30th Mar. 2018). URL: https://www.theguardian.com/technology/2018/mar/30/hackers-steal-data-150m-myfitnesspal-app-users-under-armour (visited on 14/10/2019).

[5] Alyssa Newcomb. 'Hacked MyFitnessPal Data Goes on Sale on the Dark Web—One Year After the Breach'. In: *Fortune* (14th Feb. 2019). URL: https://fortune.com/2019/02/14/hacked-myfitnesspal-data-sale-dark-web-one-year-breach/ (visited on 14/10/2019).

[6] Zack Whittaker. 'Fitness app PumpUp leaked health data, private messages'. In: *ZDNet* (31st May 2018). URL: https://www.zdnet.com/article/fitness-app-pumpup-leaked-health-data-private-messages/ (visited on 14/10/2019).

[7] *Hackers stole payment data from Garmin South Africa shopping portal*. Security Affairs. URL: https://securityaffairs.co/wordpress/91220/data-breach/garmin-sa-data-breach.html (visited on 01/09/2020).

[8] *[Security Weekly] Garmin Suffers Massive Service Meltdown After Ransomware Attack*. Penta Security Systems Inc. URL: https://www.pentasecurity.com/blog/security-weekly-garmin-ransomware/ (visited on 01/09/2020).

[9] Ryan Poplin et al. 'Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning'. In: *Nature Biomedical Engineering* 2.3 (2018), p. 158.

[10] Yisheng Lv et al. 'Traffic flow prediction with big data: a deep learning approach'. In: *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2014), pp. 865–873.

[11] Xiao Ding et al. 'Deep learning for event-driven stock prediction'. In: *Twenty-fourth international joint conference on artificial intelligence*. 2015.

[12] Sungjoon Choi, Eunwoo Kim and Songhwai Oh. 'Human behavior prediction for smart homes using deep learning'. In: *2013 IEEE RO-MAN*. IEEE. 2013, pp. 173–179.

[13] *EU General Data Protection Regulation, Chapter 3.* European Commission. 25th May 2018. URL: https://gdpr-info.eu/chapter-3/ (visited on 20/11/2020).

[14] *EU General Data Protection Regulation, Article 9.* European Commission. 25th May 2018. URL: https://gdpr-info.eu/art-9-gdpr/ (visited on 20/11/2020).

[15] Vajira Thambawita et al. *PMData: A sports logging dataset*. Feb. 2020. DOI: 10.31219/osf.io/k2apb. URL: osf.io/k2apb.

[16] *Fitbit Versa 2*. Fitbit, Inc. URL: https://www.fitbit.com/global/us/products/smartwatches/versa?sku=507BKBK (visited on 28/11/2020).

[17] Richard MacManus. *Health Trackers: How Technology is helping us Monitor and Improve our Health*. Rowman & Littlefield, 2015.

[18] Catrine Tudor-Locke. *Manpo-Kei: The Art and Science of Step Counting: How to Be Naturally Active and Lose Weight*. Trafford Publishing, 2003.

[19] Kazuyuki Kanosue et al. *Physical Activity, Exercise, Sedentary Behavior and Health*. Springer, 2015.

[20] Sally Edwards. *The Heart Rate Monitor Book*. LWW, 1994.

[21] *FORERUNNER*. United States Patent and Trademark Office. URL: http://tmsearch.uspto.gov/bin/showfield?f=doc&state=4803:nejzut.2.28 (visited on 30/08/2020).

[22] *Investor FAQ*. Fitbit. URL: https://investor.fitbit.com/overview/investor-faq/default.aspx (visited on 30/08/2020).

[23] *Sony SmartWatch launches in US for $149.99*. The Verge. URL: https://www.theverge.com/2012/4/12/2943145/sony-smartwatch-release-date-price-usa (visited on 30/08/2020).

[24] *Almost every major consumer electronics manufacturer is now working on a smart watch*. Quartz. URL: https://qz.com/101058/smart-watch-explosion/ (visited on 30/08/2020).

[25] *Designing for Joy With Polar M200 GPS Running Watch*. Polar. URL: https://www.polar.com/blog/introducing-the-new-polar-m200/ (visited on 30/08/2020).

[26] *Meet Fitbit Ionic: A little smartwatch, a lot of fitness tracker*. Fitbit. URL: https://www.macworld.com/article/3219717/meet-fitbit-ionic-a-little-smartwatch-a-lot-of-fitness-tracker.html (visited on 30/08/2020).

[27] *Garmin releases its first luxury smartwatch, the Fenix Chronos*. Garmin. URL: https://techcrunch.com/2016/08/25/garmin-releases-its-first-luxury-smartwatch-the-fenix-chronos/ (visited on 30/08/2020).

[28] *Market share of wearables unit shipments worldwide by vendor from 1Q'14 to 1Q'20*. Statista. URL: https://www.statista.com/statistics/435944/quarterly-wearables-shipments-worldwide-market-share-by-vendor/ (visited on 26/08/2020).

[29] *Apple Dominates Global Smartwatch Sales*. Statista. URL: https://www.statista.com/chart/15035/worldwide-smartwatch-shipments/ (visited on 26/08/2020).

[30] *Fitness Tracker Market to Reach USD 91.98 Billion by 2027; Rising Health Disorders to Brighten Market Prospects, states Fortune Business Insights™*. Intrado GlobeNewswire. URL: https://www.globenewswire.com/news-release/2020/08/03/2071561/0/en/Fitness-Tracker-Market-to-Reach-USD-91-98-Billion-by-2027-Rising-Health-Disorders-to-Brighten-Market-Prospects-states-Fortune-Business-Insights.html (visited on 26/08/2020).

[31] *Jawbone is going out of business*. The Verge. URL: https://www.theverge.com/2017/7/6/15931080/jawbone-going-out-of-business-report (visited on 30/08/2020).

[32] Noor Zainab Hussain Akanksha Rana. 'Google taps fitness tracker market with $2.1 billion bid for Fitbit'. In: *Reuters* (1st Nov. 2019). URL: https://www.reuters.com/article/us-fitbit-m-a-alphabet/alphabets-google-to-buy-fitbit-for-2-1-billion-idUSKBN1XB47G (visited on 10/03/2020).

[33] Jason Aycock. 'Fitbit holders approve Google buyout'. In: *Seeking Alpha* (6th Jan. 2020). URL: https://seekingalpha.com/news/3529970-fitbit-holders-approve-google-buyout (visited on 10/03/2020).

[34] Keith M. Diaz et al. 'Fitbit®: An accurate and reliable device for wireless physical activity tracking'. In: *International Journal of Cardiology* 185 (2015), pp. 138–140. ISSN: 0167-5273. DOI: https://doi.org/10.1016/j.ijcard.2015.03.038. URL: http://www.sciencedirect.com/science/article/pii/S0167527315002764.

[35] Judit Takacs et al. 'Validation of the Fitbit One activity monitor device during treadmill walking'. eng. In: *Journal of Science and Medicine in Sport* 17.5 (2014), pp. 496–500. ISSN: 1440-2440.

[36] Albert Hernandez et al. 'Commercial Activity Trackers Overestimate Step Count: Implications for Ambulatory Activity Monitoring'. In: *HCI International 2019 - Posters*. Ed. by Constantine Stephanidis. Cham: Springer International Publishing, 2019, pp. 446–451. ISBN: 978-3-030-23525-3.

[37] Shahab Haghayegh et al. 'Accuracy of Wristband Fitbit Models in Assessing Sleep: Systematic Review and Meta-Analysis'. In: *J Med Internet Res* 21.11 (Nov. 2019), e16273. ISSN: 1438-8871. DOI: 10.2196/16273. URL: http://www.ncbi.nlm.nih.gov/pubmed/31778122.

[38] Mikel Delgado. 'How fit is that Fitbit?' In: *Berkeley Science Review* (2014).

[39] Britt Cyr et al. 'Security analysis of wearable fitness devices (fitbit)'. In: *Massachusetts Institute of Technology* 1 (2014). URL: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2015/03/20082016/17-cyrbritt-webbhorn-specter-dmiao-hacking-fitbit.pdf.

[40] Carles Gomez, Joaquim Oller and Josep Paradells. 'Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology'. In: *Sensors* 12.9 (2012), pp. 11734–11753.

[41] *HIPAA: The De-identification Standard*. U.S. Department of Health & Human Services. URL: https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html#standard (visited on 03/09/2020).

[42] *"Anonymized" data really isn't—and here's why not*. Ars Technica. URL: https://arstechnica.com/tech-policy/2009/09/your-secrets-live-online-in-databases-of-ruin/ (visited on 04/09/2020).

[43] Michelle M Christovich. 'Why Should We Care What Fitbit Shares-A Proposed Statutroy Solution to Protect Sensative Personal Fitness Information'. In: *Hastings Comm. & Ent. LJ* 38 (2016), p. 91.

[44] *The World's Most Wired Computer Scientist: Arvind Narayanan*. Wired. URL: https://www.wired.com/2012/06/wmw-arvind-narayanan/ (visited on 03/09/2020).

[45] *Pulling back the curtain on "anonymous" Twitterers*. Ars Technica. URL: https://arstechnica.com/tech-policy/2009/03/pulling-back-the-curtain-on-anonymous-twitterers/ (visited on 04/09/2020).

[46] Liangyuan Na et al. 'Feasibility of reidentifying individuals in large national physical activity data sets from which protected health information has been removed with use of machine learning'. In: *JAMA network open* 1.8 (2018), e186040–e186040.

[47] *You Won't Like What Your Facebook 'Likes' Reveal*. Electronic Frontier Foundation. URL: https://www.eff.org/deeplinks/2013/03/facebook-likes-reveal-sensitive-personal-information (visited on 04/09/2020).

[48] D Aune, B ó Hartaigh and LJ3 Vatten. 'Resting heart rate and the risk of type 2 diabetes: a systematic review and dose–response meta-analysis of cohort studies'. In: *Nutrition, Metabolism and Cardiovascular Diseases* 25.6 (2015), pp. 526–534.

[49] Srinivasan Beddhu et al. 'Associations of resting heart rate with insulin resistance, cardiovascular events and mortality in chronic kidney disease'. In: *Nephrology Dialysis Transplantation* 24.8 (2009), pp. 2482–2488.

[50] Andreas Festa et al. 'Heart rate in relation to insulin sensitivity and insulin secretion in nondiabetic subjects.' In: *Diabetes care* 23.5 (2000), pp. 624–628.

[51] Liang Wang et al. 'Resting heart rate and the risk of developing impaired fasting glucose and diabetes: the Kailuan prospective study'. In: *International journal of epidemiology* 44.2 (2015), pp. 689–699.

[52] Dongfeng Zhang, Xiaoli Shen and Xin Qi. 'Resting heart rate and all-cause and cardiovascular mortality in the general population: a meta-analysis'. In: *Cmaj* 188.3 (2016), E53–E63.

[53] *All About Heart Rate (Pulse)*. American Heart Association, Inc. URL: https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse (visited on 08/09/2020).

[54] Xiao Li et al. 'Digital Health: Tracking Physiomes and Activity Using Wearable Biosensors Reveals Useful Health-related Information'. In: *PLoS biology* 15.1 (2017), e2001402.

[55] Jennifer M Radin et al. 'Harnessing wearable device data to improve state-level real-time surveillance of influenza-like illness in the USA: a population-based study'. In: *The Lancet Digital Health* 2.2 (2020), e85–e93.

[56] Jane Bromley et al. 'Signature verification using a "siamese" time delay neural network'. In: *International Journal of Pattern Recognition and Artificial Intelligence* 7.04 (1993), pp. 669–688.

[57] Shubham Panchal. *Face Recognition From Scratch Using Siamese Networks and TensorFlow*. Medium. Apr. 2019. URL: https://medium.com/predict/face-recognition-from-scratch-using-siamese-networks-and-tensorflow-df03e32f8cd0 (visited on 31/10/2020).

[58] Rahul Rama Varior, Mrinal Haloi and Gang Wang. 'Gated siamese convolutional neural network architecture for human re-identification'. In: *European conference on computer vision*. Springer. 2016, pp. 791–808.

[59] Dahjung Chung, Khalid Tahboub and Edward J. Delp. 'A Two Stream Siamese Convolutional Neural Network for Person Re-Identification'. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[60] Ran Tao, Efstratios Gavves and Arnold W.M. Smeulders. 'Siamese Instance Search for Tracking'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[61] Luca Bertinetto et al. 'Fully-Convolutional Siamese Networks for Object Tracking'. In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Gang Hua and Hervé Jégou. Cham: Springer International Publishing, 2016, pp. 850–865. ISBN: 978-3-319-48881-3.

[62] Gregory Koch. 'Siamese neural networks for one-shot image recognition'. In: *ICML deep learning workshop*. Vol. 2. Lille. 2015.

[63] Harshall Lamba. 'One Shot Learning with Siamese Networks using Keras'. In: *Towards Data Science* 5 (2019). URL: https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d.

[64] *Fitbit: About Us*. Fitbit. URL: https://www.fitbit.com/global/us/about-us (visited on 21/08/2020).

[65] *Fitbit: Blog*. Fitbit. URL: https://blog.fitbit.com/ (visited on 21/08/2020).

[66] *Fitbit: Find Your Reason*. Fitbit. URL: https://stories.fitbit.com/ (visited on 21/08/2020).

[67] *Fitbit: Community*. Fitbit. URL: https://community.fitbit.com/ (visited on 21/08/2020).

[68] *EU General Data Protection Regulation, Article 12*. European Commission. 25th May 2018. URL: https://gdpr-info.eu/art-12-gdpr/ (visited on 20/08/2020).

[69] . 'Scalable Infrastructure for Efficient Real-Time Sports Analytics'. In: *ICMI '20: Proceedings of the 2020 International Conference on Multimodal Interaction*. Oct. 2020.

[70] *pmSys*. ForzaSys AS. URL: https://forzasys.com/pmsys.html (visited on 31/10/2020).

[71] Athanase Benetos et al. 'Influence of heart rate on mortality in a French population: role of age, gender, and blood pressure'. In: *Hypertension* 33.1 (1999), pp. 44–52.

[72] G Sundkvist, LO Almer and B Lilja. 'Respiratory influence on heart rate in diabetes mellitus.' In: *Br Med J* 1.6168 (1979), pp. 924–925.

[73] Ivana Antelmi et al. 'Influence of age, gender, body mass index, and functional capacity on heart rate variability in a cohort of subjects without heart disease'. In: *The American journal of cardiology* 93.3 (2004), pp. 381–385.

[74] *Python*. Python Software Foundation. URL: https://www.python.org/ (visited on 11/11/2020).

[75] *Package Installer for Python (pip)*. Python Package Index (PyPI). URL: https://pypi.org/project/pip/ (visited on 11/11/2020).

[76] *CUDA Toolkit*. NVIDIA Corporation. URL: https://developer.nvidia.com/cuda-toolkit (visited on 11/11/2020).

[77] *CUDA Deep Neural Network Library (cuDNN)*. NVIDIA Corporation. URL: https://developer.nvidia.com/cudnn (visited on 11/11/2020).

[78] *TensorFlow*. Google Brain Team. URL: https://www.tensorflow.org/ (visited on 11/11/2020).

[79] *NumPy*. NumPy. URL: https://numpy.org/ (visited on 11/11/2020).

[80] *pandas*. numFOCUS. URL: https://pandas.pydata.org/ (visited on 11/11/2020).

[81] *random – Generate pseudo-random numbers*. Python Software Foundation. URL: https://docs.python.org/3/library/random.html (visited on 11/11/2020).

[82] *Matplotlib: Visualization with Python*. The Matplotlib development team. URL: https://matplotlib.org/ (visited on 11/11/2020).

[83] Vajira Thambawita et al. 'An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification'. In: *ACM Trans. Comput. Healthcare* 1.3 (June 2020). ISSN: 2691-1957. DOI: 10.1145/3386295. URL: https://doi.org/10.1145/3386295.

[84] Michael DelSole. *What is One Hot Encoding and How to Do It*. Medium. Apr. 2018. URL: https://medium.com/@michaeldelsole/what-is-one-hot-encoding-and-how-to-do-it-f0ae272f1179 (visited on 30/11/2020).

[85] *eX3*. Simula. URL: https://www.ex3.simula.no/ (visited on 12/10/2020).

[86] *tf.keras.layers.Flatten*. Google Brain Team. URL: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Flatten (visited on 10/11/2020).