KU LEUVEN

FACULTY OF ENGINEERING TECHNOLOGY

TECHNOLOGY CAMPUS DE NAYER

Continuous unobtrusive user authentication using gait for wearable devices

utilising machine learning algorithms

Jelle MEEUS

Promotor(en): Toon Goedemé

Co-promotor(en): Pradip Mainali

Master Thesis submitted to obtain the degree of
Master of Science in Engineering Technology:
Electronics-ICT ICT Engineering

Academic Year 2017 - 2018

Acknowledgements

First, I would like to thank my promoter Prof. Dr. Ir. Toon Goedemé and co-promoter Dr. Pradip Mainali for their guidance and support during my thesis.

I would also like to thank the people at Vasco Data Security for giving me the opportunity to work on this project and a special thanks to the volunteers that helped during the data collection process.

Jelle Meeus
May 2018

Abstract (English)

In recent years, much research has been done to find new authentication methods that try to avoid explicit input from a user. This techniques use patterns and biometrics from a user to recognize machine learning models. One of these biometrics is the way a person walks. It can be captured by sensors on a smartwatch or smartphone, easily and unobtrusively.

The aim of this thesis is to develop a method that is based on an existing barebone implementation. This application consists of a wearable application to record data and a server application to process this data offline. With a study of state-of-the-art recognition of human activity and gait recognition, this implementation was studied, expanded and improved.

In this project, a human activity recognition system was placed in cascade with a gait recognition system to design a continuous gait-based authentication model. These systems are traditional machine learning models and use a new feature-extraction technique that is fast and accurate. The new implementation allows data to be captured, offline to be trained offline on the server and for new data to be evaluated on the wearable. We have explored deep learning, but the traditional approach with manually designed functions performs better.

Abstract (Nederlands)

In de voorbije jaren is er veel onderzoek gebeurd naar nieuwe authenticatie methoden die expliciete input van een gebruiker proberen te vermijden. Deze methoden gebruiken patronen en biometrie van een gebruiker om met machine learning modellen deze te herkennen. Een van deze biometrische eigenschappen is de manier waarop een persoon wandelt. Deze kan met wearable sensors op een smartphone of smartwatch, gemakkelijk en wrijvingsloos worden opgenomen.

Het doel van deze thesis is om een continue authenticatie methode op basis van gait te ontwikkelen die beter presteerd dan een reeds bestaande barebone implementatie. Deze implementatie bestaat uit een wearable applicatie om data op te nemen en een server applicatie om deze data offline te verwerken. Met een studie van state-of-the-art human activity recognition en gait recognition werd deze implementatie bestudeerd, uitgebreid en verbeterd.

In dit project werd een human activity recognition systeem in cascade met een gait recognition systeem geplaatst om een continu gait-gebaseerd authenticatie model te ontwerpen. Deze beide systemen gebruiken traditionele machine learning modellen en maken gebruik van een nieuwe feature-extraction techniek die snel en accuraat is. De nieuwe implementatie laat toe om data op te nemen, offline te trainen op de server en nadien voor nieuwe data op de wearable te evalueren. Bovendien werd onderzoek gedaan om dit probleem met deep learning aan te pakken, maar de traditionele aanpak met manueel ontworpen features presteert beter.

Nederlandse samenvatting

In de voorbije jaren is er veel onderzoek gebeurd naar nieuwe authenticatie methoden die expliciete input van een gebruiker proberen te vermijden. Deze methoden gebruiken patronen en biometrie van een gebruiker om met machine learning modellen deze te herkennen. Een van deze biometrische eigenschappen is de manier waarop een persoon wandelt. Deze kan met wearable sensoren op een smartphone of smartwatch, gemakkelijk en wrijvingsloos worden opgenomen.

Doel

Het doel van deze thesis is om een continue authenticatie methode op basis van gait te ontwikkelen die beter presteerd dan een reeds bestaande barebone implementatie. Deze implementatie bestaat uit een wearable applicatie om data op te nemen en een server applicatie om deze data offline te verwerken. Met een studie van state-of-the-art human activity recognition en gait recognition werd deze implementatie bestudeerd, uitgebreid en verbeterd.

State-of-the-art

Een eerste stap is te detecteren wanneer een persoon wandelt. Dit is een probleem uit Human Activity Recognition (HAR). Uit de literatuur leren we dat specifieke bewegingen van het lichaam transleren naar karasteristieke patronen in de sensor data. Met de juiste gekozen technieken kunnen we deze patronen extraheren uit de data en met behulp van machine learning een classifier trainen. Een volgende en moeilijkere stap is om eenmaal wanneer we zeker zijn dat een persoon wandelt beslissen over welke persoon het gaat. Deze kan idem met machine learning worden opgelost.

Experimenten

Om deze systemen te evalueren is er een dataset van 5 vrijwilligers opgenomen die wandelen en andere activiteiten uitoefenen. In dit project werd een human activity recognition systeem in cascade met een gait recognition systeem geplaatst (zie figuur 1) om een continu gait-gebaseerd authenticatie model te ontwerpen. Deze beide systemen gebruiken traditionele logistic regression machine learning modellen. Een eerste manier om beide systemen te verbeteren is om een nieuwe feature extractie techniek te bouwen. Deze feature is gebaseerd op de derivative filter uit het veld van de computer visie. Na experimenten waarbij verschillende versies van deze feature werden uitgeprobeerd werd een beste techniek geselecteerd. Een volgende verbeteringstechniek is om de

Figure 1: Overzicht van gait detection (wandel detectie) model en gait recognition (authenticatie model)

preprocessing aan te passen. Na implementatie van een Gaussiaanse filter presteren te modellen beter. Bovendien werd onderzoek gedaan om dit probleem met deep learning aan te pakken, maar de traditionele aanpak met manueel ontworpen features presteert beter.

Deze twee systemen worden vervolgens getest op gepaste publieke datasets. We testen de gait recognition methode op IDNet [1]. De gait detection worden getest op de USC-HAD [2] en PAMAP2 [3] datasets.

Resultaten

De nieuwe implementatie laat toe om data op te nemen, offline te trainen op de server en nadien nieuwe data op de wearable te evalueren. Deze twee modellen maken gebruik van deze nieuwe feature-extraction techniek die snel en accuraat is. Er kan verder onderzocht worden om het geheugen- en CPU-gebruik te verminderen. Ook kan de toevoeging van nieuwe gebruikers worden vergemakkelijkt door een betere manier te ontwikkelen om data van de wearable naar de server te verplaatsen.

Contents

1 Introduction **3**

1.1 Background and purpose 3

1.2 Goal 3

1.3 Research questions 4

2 Literature Study **5**

2.1 Introduction to Security and Authentication 5

2.2 Biometric recognition 6

2.3 Gait 8

2.4 Traditional Machine Learning (ML) 9

2.4.1 Support Vector Machine (SVM) 11

2.4.2 Logistic Regression (LR) 11

2.5 Deep Learning (DL) 13

2.5.1 Architecture 14

2.5.2 Stochastic gradient descent (SGD) 15

2.5.3 Convolutional neural networks (CNNs) 15

2.5.4 Long short term memory (LSTM) 15

2.5.5 Overfitting and underfitting 16

2.5.6 Activation functions 16

2.5.7 Pooling layers 16

2.5.8 Dropout layers 16

2.5.9 Fully connected layer 17

2.5.10 Transfer learning 17

2.6 State-of-the-art Gait Authentication 17

2.6.1 Wearable Sensor Based approach 18

2.7 State-of-the-art Activity Recognition 20

2.8 Statistically Significant Validation Population Size 22

2.9 Datasets 23

2.10 Conclusion 24

3 Approaches to improvement **25**

3.1 Goal 25
3.2 Beginning 25
3.2.1 Preprocessing 27
3.2.2 Feature Extraction Method 27
3.2.3 Initial gait detection 27
3.2.4 Initial gait recognition 28
3.3 Data collection 28
3.4 Improving with Traditional machine learning 28
3.4.1 Preprocessing 28
3.4.2 Derivatives 28
3.5 Deep learning 29
3.6 Conclusion 29

4 Experiments **31**
4.1 Gait recognition 31
4.1.1 Derivatives 31
4.1.2 Filter 31
4.1.3 Result 36
4.2 Gait detection 36
4.3 Android implementation 36
4.4 Computation time 40
4.5 Conclusion 40

5 Validation **41**
5.1 Gait recognition 41
5.2 Gait detection 41
5.3 Conclusion 41

6 Discussion and Future Work **45**
6.1 Future Work 45

List of Figures

1 Overzicht van gait detection (wandel detectie) model en gait recognition (authenticatie model) . . . xii

1.1 Person walking with SenseID wearable . . . 4

2.1 Block diagram of enrollment, verification and identification [4] . . . 7
2.2 False Matching Rate (FMR) and False Non-Matching Rate (FNMR) (left) for a given treshold t. The Receiver Operating Characteristic curve (ROC) curve relating FMR to FNMR (right) at different thresholds. [4] . . . 8
2.3 Events in a gait cycle [5] . . . 9
2.4 A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. [6] . . . 10
2.5 Machine learning components overview . . . 11
2.6 Two separable sets and two of the infinitely many possible planes that separate the sets [7] . . . 12
2.7 The two closest points of the convex hulls determine the separating plan [7] . . . 12
2.8 Feed-forward neural network with multiple layers [8] . . . 14
2.9 LeNet-5 network architecture [9] . . . 15
2.10 Three nonlinear activation functions used by NNs: the sigmoid function, the rectified linear unit (ReLu) and the Leaky ReLu) [10] . . . 16
2.11 Diagram of a CNN. [11] . . . 17
2.12 Comparison between wearable sensors (WS) and non-wearable sensors (NWS) systems. [12] . . . 18
2.13 Methodological layout of existing Wearable Sensor Based approaches. [13] . . . 19
2.14 Neural network for gait recognition using Wearable Sensor Based approach. [14] . . . 20
2.15 An example of activity data from the x-axis of the 3-axis accelerometer [2] . . . 21

3.1 Initial Android User Interface (UI) . . . 26
3.2 Overview of walk detection model and the authentication model . . . 26

4.1 ROC curve given for collected dataset of 5 users . . . 32
4.2 ROC curve given for collected dataset of 5 users using different feature extraction techniques . . . 33

4.3 ROC curve given for collected dataset of 5 users using different filters and derivative feature . . . 35
4.4 Comparison of using a Moving Average (MA) filter with width 20 and a Gaussian filter with size 13. In blue is shown useful information of gait . . . 35
4.5 ROC curve given for collected dataset of 5 users using ED, FFT and derivative features 36
4.6 ROC curve given for collected dataset of 5 users using FFT and derivative features . 37
4.7 ROC curve given for collected dataset of 5 users using FFT and derivative features . 37
4.8 Android User Interface (UI) . . . 38
4.9 SenseID wearable research platform . . . 39

5.1 ROC curve Authentication IDNet dataset using FFT and derivatives features . . . 42
5.2 ROC curve Gait detection USC-HAD dataset using FFT and derivatives features . . 42
5.3 ROC curve Gait detection PAMAP2 dataset using FFT and derivatives features . . . 43

List of Tables

2.1 z-score for different confidence level 22
2.2 sample size for different confidence level and margin of error for p=0.5 23
2.3 sample size for different confidence level and margin of error for p=0.95 23
2.4 Requirements of datasets for gait detection and gait recognition 23
2.5 Comparison of publicly available datasets 24

4.1 Computation time of features ED, FFT and derivatives 40

List of symbols and acronyms

Glossary

accelerometer an electromechanical device that will measure acceleration forces. 17, 18, 25, 31

gait cycle The period of time from the occurrence of an event in one foot until the recurrence of the same event in the same foot. xv, 8, 20

gyroscope a device used for measuring or maintaining orientation and angular velocity. 17, 18, 25

multimodal biometric system a biometric system using multiple biometric modalities i.g. face and fingerprint of a person or multiple fingers of a person). 17

ROC Receiver Operating Characteristic curve (ROC) is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. xv,

supervised learning the training data consists of input vectors with labelled target values. 11

Acronyms

AI artificial intelligence. 9

CNN convolutional neural network. xiii, xv, 14, 15, 17, 20

DL Deep Learning. xiii, 13, 20

ED Euclidian Distances. 27, 29, 36

FFT Fast Fourier transform. 27, 29, 36

FMR False Matching Rate. xv, 8

FNMR False Non-Matching Rate. xv, 8

FSB Floor Sensor Based. 17, 18

HAR Human Activity Recognition. xi, 20

LR Logistic Regression. xiii, 11, 13

LSTM long short term memory. xiii, 15, 16

MA Moving Average. xvi, 31

ML Machine Learning. xiii, 9, 20

MVB Machine Vision Based. 17, 18

NN neural network. xv, 13, 14, 15, 16, 17

NWS non-wearable sensors. xv, 18

PCA Principal component analysis. *Glossary:* PCA

PIN Personal Identication Number. 3

ReLu rectified linear unit. xv, 16

RNN recurrent neural network. 14, 15

ROC Receiver Operating Characteristic. *Glossary:* ROC, xv, xvi, 8, 31, 36

SGD stochastic gradient descent. xiii, 14

SVM Support Vector Machine. xiii, 11, 20, 27

WS wearable sensors. xv, 18

WSB Wearable Sensor Based. xiii, xv, 17, 18, 20, 18

Symbols

Chapter 1

Introduction

This master degree project was proposed within a research project called SenseID conducted by the company Vasco Data Security in collaboration with the university KULeuven. The company project aims to research new methods to further improve user authentication based on capturing contextual data and biometrics. The goal of this project is to develop a gait authentication algorithm that can be used on computationally limited devices such as smart phones and smart watches.

1.1 Background and purpose

Over the last decade there has been an increase in the amount of mobile devices such as mobile phones, tablets etc. They are no longer only used for calling or sending text messages. These devices are used in mobile applications such as web browsing, emailing, e-banking and e-commerce. As a result, they contain personal valuable information. Thus, there is a need for security.

Common methods to provide security to these devices are the use of a Personal Identication Number (PIN) code or the use of a fingerprint scanner. These however, require explicit user interaction from the user. According to a survey [15] users choose to disable the PIN method. The most common reason to turn off this authentication method is that entering a PIN takes too much time. Furthermore the authentication only applies to the start of a transaction. The user will not be authenticated before, throughout and after the duration of a transaction.

The SenseID project searches to provide answers to these two problems. The goal of the SenseID project is to research new ways to uniquely, reliably, and continuously identify people. Wearable devices can be used to record biometric data and other contextual data. Such data can be used to continuously and transparently authenticate users in an unobtrusive manner.

1.2 Goal

One of these biometrics that can be easily collected is the gait of a person. In this project, we define gait as the manner in which a person walks. We can use the accelerometer and gyroscope sensors to collect and develop an unobtrusive gait recognition method. In previous work, this has been done before with some promising results. In this thesis project we take a look at existing methods and attempt to improve the state-of-the-art algorithms to provide a more computationally efficient and accurate method for gait authentication for computationally limited devices.

Figure 1.1: Person walking with SenseID wearable

In this thesis we will discuss the state-of-art and go over some of the basic concepts in chapter 2. Followed by the achieved successes in 3

1.3 Research questions

The one research question that encapsulates this work is

Can we develop a computationally inexpensive and accurate gait-based continuous authentication model for mobile devices?

One of the approaches of this project is to use traditional machine learning, in which we hand-craft a feature extraction technique. A follow-up on the previous approach is to further improve the learning and classification process with deep learning.

More specific research questions can be proposed:

1. Can we develop a feature extraction technique that is both computationally inexpensive and accurate to authenticate a person based on his walking pattern?
2. Can we improve the accuracy of models based on existing hand-crafted feature extraction techniques for gait recognition by crafting a suitable deep learning network architecture?
3. How much walking data is needed to enroll a new user in an authentication model?

Chapter 2

Literature Study

In this chapter we give a brief overview of some of the basic concepts regarding security and authentication. We will then further discuss biometrics authentication based on gait. We will then give a study of the state-of-the-art techniques regarding gait authentication using traditional machine learning, followed by a study of methods using deep learning.

2.1 Introduction to Security and Authentication

A computer system compromised of hardware, software, and processes is often an abstraction of an actual business model that exists in the real world outside the computer system [16]. Every element of a transaction has to be projected onto a computer model. Actual users are humans, which exist outside the system. A user ID can be used as an abstract object the projection of a human into the computer system. This object is referred to by the system when user access to information assets is defined, and the system will also trace user actions and record an audit trail referring to the actual user by his abstract object ID.

In a security system there are three main security processes [16](Authentication, Authorization and Identification) working together to provide access to resources in a controlled manner. Authentication is the process of validating user identity. Authorization means providing users with the access to resources that they are allowed to have and preventing users from accessing resources that they are not allowed to access. Accounting providing an audit trail of user actions. This is sometimes referred to as auditing.

In this thesis we'll focus on the process of authentication. In this process we want to validate the user identify to the security system. A user may claim to be a represented object in the computer system. It is up to the the authentication process to ascertain the claimed user identity by verifying user-provided evidence. This evidence is based on characteristics or unique information. Below are listed three types of credentials, sometimes known as 'the three factors of authentication'. [16]

1. Something you know: Based on a secret shared between the user and the authenticator.
2. Something you are: Requires the authenticator to authenticate the user based on biometric information, such as fingerprints, retina scan, facial scan, etc.
3. Something you have: Requires the possession of an authentication token, which is an actual object.

The use of multiple authentication factors considerably increases the security of a system from a user authentication perspective. If one authentication credential might fail, then a second one is taken into consideration. Using multiple types of credentials will likely increase the time it takes for users to log in.

2.2 Biometric recognition

One of the credential types as part of the 'three factors of authentication' is biometric information. According to [4] any human physiological and/or behavioral characteristic can be used as an authentication factor when it satisfies the following requirements:

- Universality: each person should have the characteristic.
- Distinciveness: any two persons should be sufficiently different in terms of the characteristic.
- Permancence: the characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time.
- Collectability: the characteristic can be measured quantitatively.

However, in a practical biometric system there are a number of issues that must be considered [4]:

- Performance: which refers to the achievable recognition accuracy and speed, the resources required to achieve the desired recognition accuracy and speed, as well as the operational and environmental factors that affect the accuracy and speed;
- Acceptability: acceptability, which indicates the extent to which people are willing to accept the use of a particular biometric identifier (characteristic) in their daily lives.
- Permanence: the characteristic should be sufficiently invariant over a period of time.
- Circumvention: which reflects how easily the system can be fooled using fraudulent methods.

A practical biometric system should meet the specified recognition accuracy, speed, and resource requirements, be harmless to the users, be accepted by the intended population, and be sufficiently robust to various fraudulent methods and attacks to the system. One of the types of biometrics which satisfies the above conditions is the gait, which we'll discuss in section 2.3.

In general a biometric system is a pattern recognition system that works by collecting biometric data from an individual, extracting a feature set, and comparing this feature against a template set in a database. Depending on the application, a system may operate in two modes, either in verification or identification mode: [4]

1. Verification mode: The system validates a person's claimed identify by comparing the collected biometric data against the person's biometric template stored in the systems database. The system conducts a one-to-one comparison to determine whether the claim is true or not.
2. Identification mode: The system recognizes an individual by comparing the collected biometric data against all of the users in the database. In this mode a system conducts a one-to-many comparison.

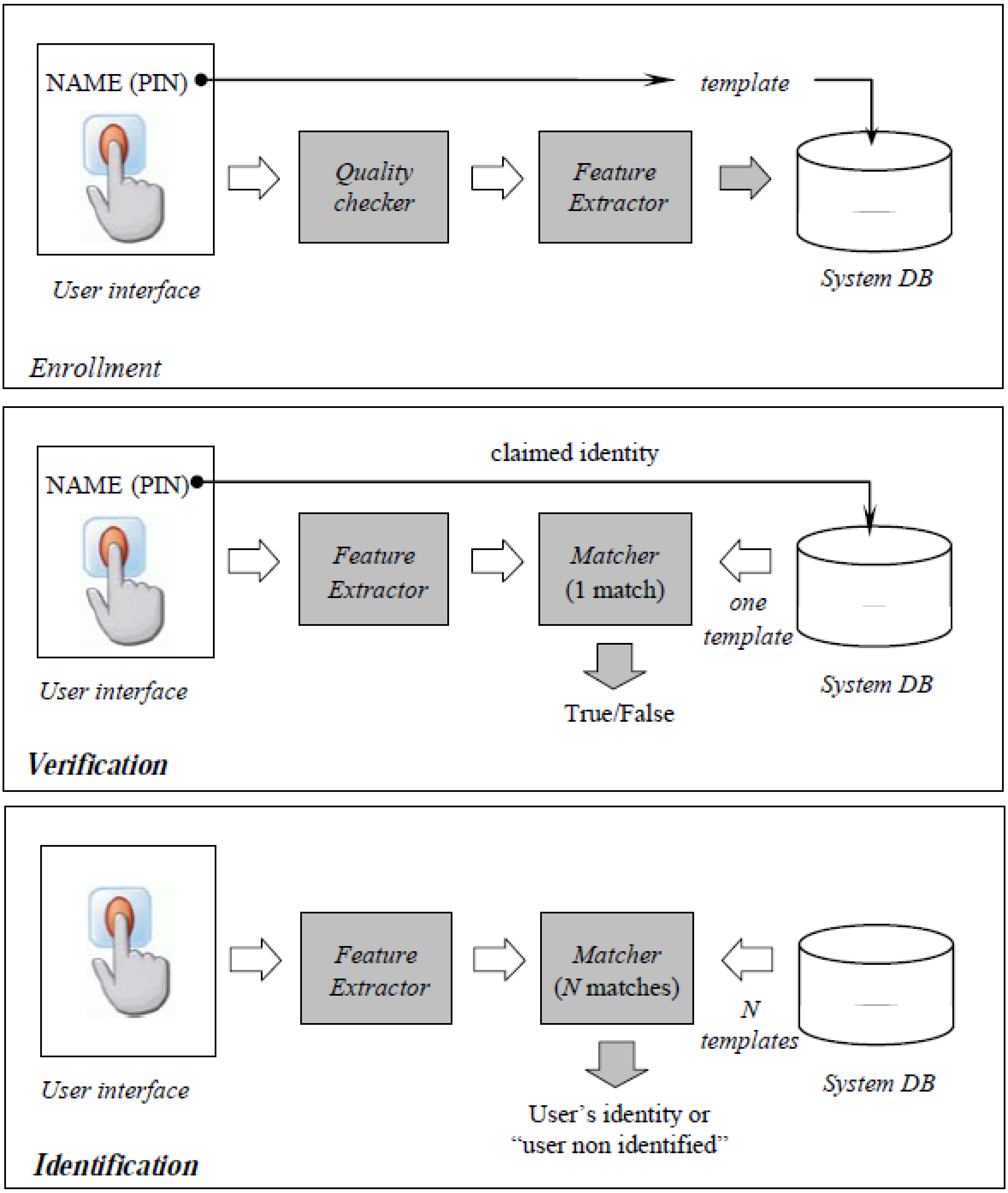


Figure 2.1: Block diagram of enrollment, verification and identification [4]

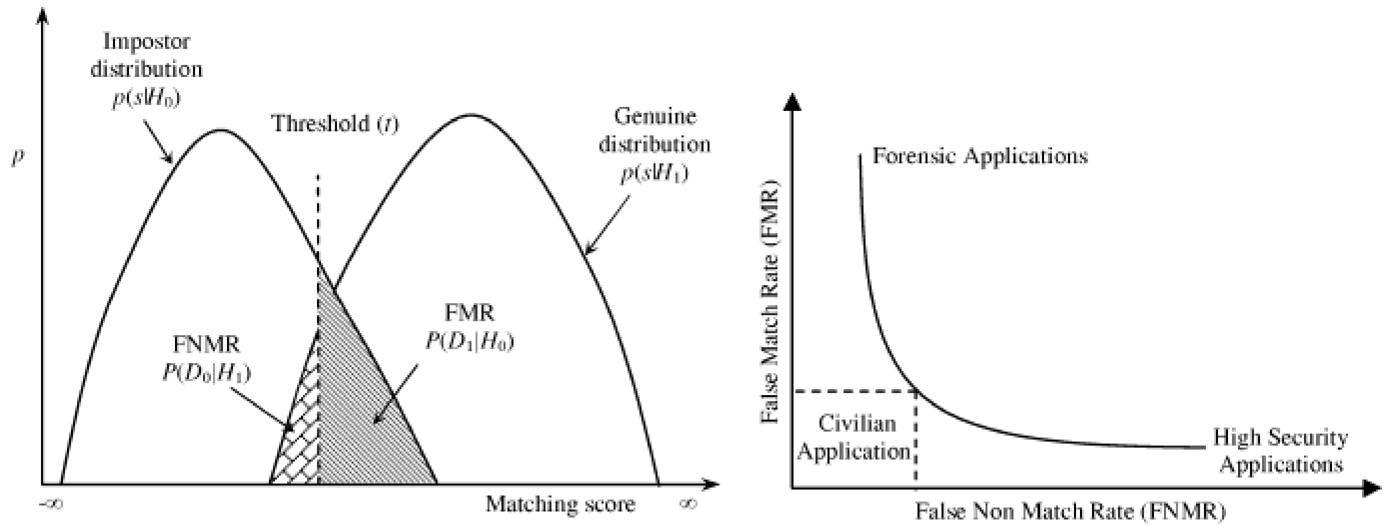


Figure 2.2: False Matching Rate (FMR) and False Non-Matching Rate (FNMR) (left) for a given treshold t. The ROC curve relating FMR to FNMR (right) at different thresholds. [4]

A general overview of the workings of a biometric system is given in figure 2.1. A biometric system can make two types of errors [4]:

1. mistaking biometrics from two different users to be the same is called a false match
2. mistaking two biometric from the same person to be from two different persons is called a false nonmatch.

$$FMR = \frac{\text{number of nonmate pairs whose matching score exceeds the threshold}}{\text{all nonmate pairs}} \tag{2.1}$$

$$FNMR = \frac{\text{number of mate pairs whose matching scores are less than the threshold}}{\text{all mate pairs}} \tag{2.2}$$

In figure 2.2 can be seen how a threshold t can have an impact on the amount of FMR and FNMR. The value of which threshold t to pick is application dependent. In applications such as criminal identification, FNMR is much more important than FMR. It is preferable to falsely identify people than to not miss identifying a criminal. Whereas for highly security applications FMR is critical.

2.3 Gait

We can go over some of the terminology used in gait analysis. Gait and walking are often used interchangeably, however there is a slight difference. We can define walking as 'a method of locomotion involving the use of the two legs, alternately to provide both support and propulsion with at least one foot being in contact with the ground at all times.' [17] Gait is defined as the manner in which a person walks. Gait speaks more about the style of walking, rather than the process itself.

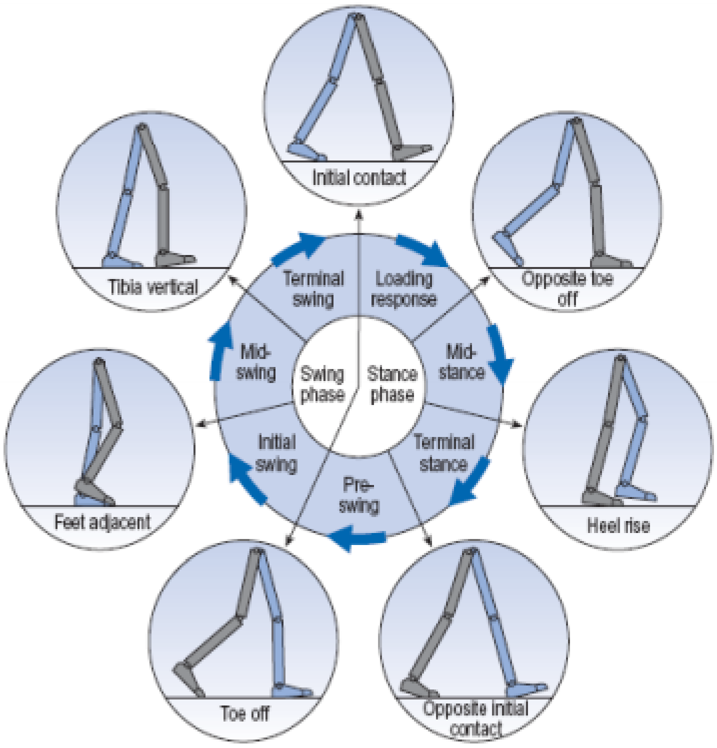


Figure 2.3: Events in a gait cycle [5]

Figure 2.3 shows an illustration of gait cycle. We will not go in depth about each of these events. Gait is a behavioral biometric and can be variant over time, due to several factors [18] like e.g. walking in a straight line, walking down- or uphill, fast or slow,... The same study shows that gait changes over a longer period of time i.e. greater than 2 months.

When we compare gait against other types of biometric data, gait is less distinctive, but it is sufficiently discriminatory to allow verification [4]. The main advantage of preferring gait is that it can provide an unobtrusive authentication method compared to other biometric systems like fingerprint or face recognition which require explicit user interaction. [19]

2.4 Traditional Machine Learning (ML)

In this section we cover general machine learning concepts. The literature about this topic is quite large so we only discuss a small part of this discipline. Figure 2.4 gives an overview of AI, Machine learning and deep learning.

In short, machine learning is a subfield of artificial intelligence (AI) which allows computers to learn. [20] Usually a machine learning algorithm is given a set of data and information about properties of the data. In this manner a computer can make predictions about unseen data in the future. This is possible because all non-random data contains patterns that allow machines to generalize. In order to do this, it trains a model with what it determines are the important aspects of the data. One of the weaknesses is that when an algorithm encounters a pattern previously unseen, it is likely to be misinterpreted [20]. There are many different machine learning algorithms, with their own strengths and suited to different types of problems. In 2.4.1 and 2.4.2 we discuss

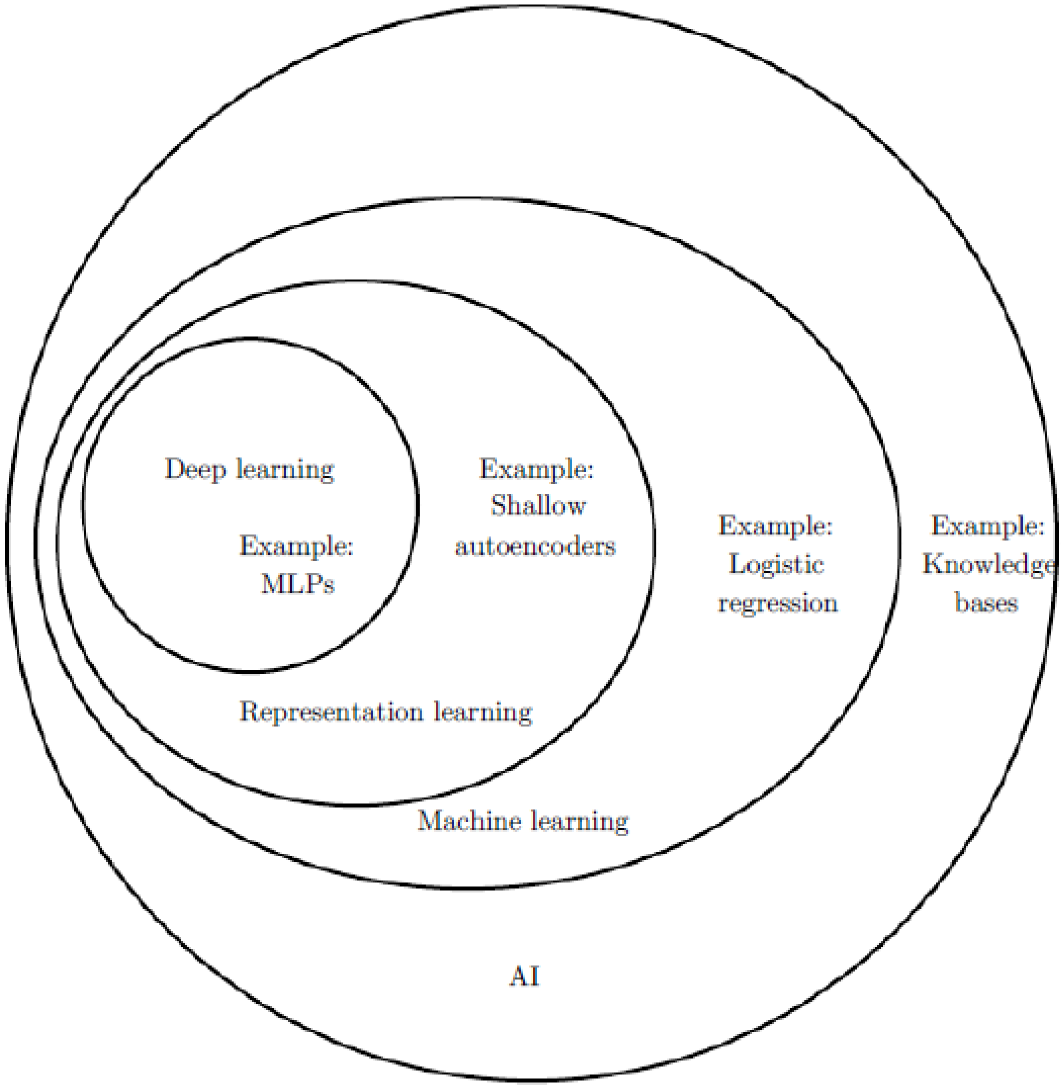


Figure 2.4: A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. [6]

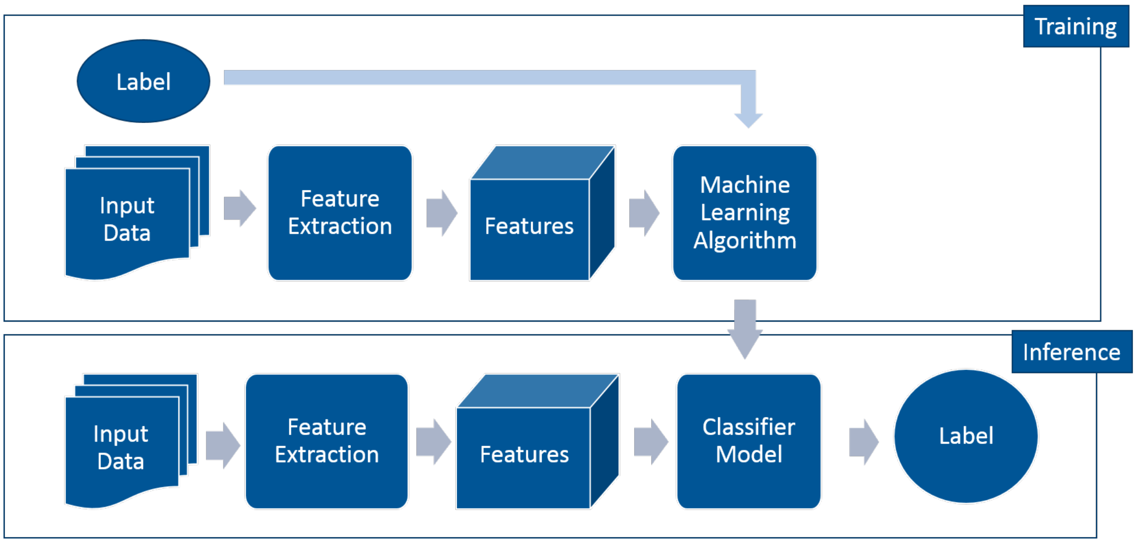


Figure 2.5: Machine learning components overview

two common supervised learning algorithms.

Figure 2.5 shows a general view of components used in supervised learning. Two phases be distinguished:

- Training phase: choosing model parameters based on labelled training data
- Inference phase: making some sort of prediction on new unseen data

After we have selected a model that has been fitted on the training dataset. We can use a test dataset of unseen data to estimate how well the model performs. If we are satisfied with its performance, we can now use this model to predict new future data.

2.4.1 Support Vector Machine (SVM)

A SVM model is based on the concept of decision planes that define decision boundaries which are planes that separate a set of data in different classes. A SVM model constructs hyperplanes in a multidimensional space that separates data in different classes.

A paper [7] shows how SVM work using an intuitive geometric. The figure 2.6 shows two linearly separable classes and a decision plane.

One way to construct the plane (see figure 2.7) as far as possible from both sets is to construct the smallest convex sets that contain all the data in each class (i.e. the convex hull) and find the closest points in those sets. [7] Then, construct the line segment between the two points. The plane γ, orthogonal to the line segment w, that bisects the line segment is chosen to be the separating plane. [7]

2.4.2 Logistic Regression (LR)

A LR model is known as a binary classifier and is part of supervised learning. It uses the sigmoid function shown in equation 2.4. A LR model uses one or more independent variables $X = [X_1, X_2, \ldots, X_k]^T$

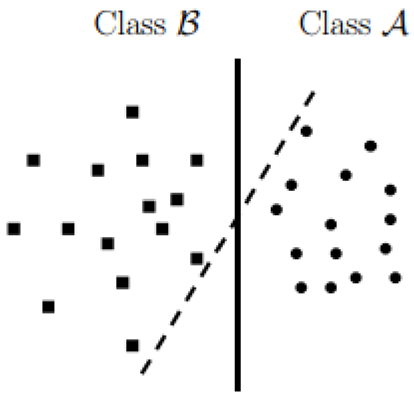


Figure 2.6: Two separable sets and two of the infinitely many possible planes that separate the sets [7]

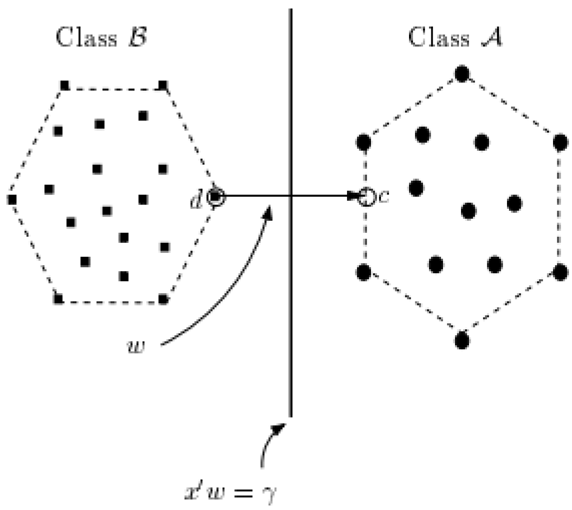


Figure 2.7: The two closest points of the convex hulls determine the separating plan [7]

and regression coefficients $\beta = [\beta_0, \beta_1, \ldots, \beta_k]^T$ to predict the probability of a binary outcome.

A logistic function is

$$p(\theta) = \frac{L}{1 + e^{-m(x - x_0)}} \tag{2.3}$$

where L is the curve's maximum value, m steepness of the curve, x_0 x-value of the function's midpoint. A standard logistic function with $k = 1$, $x_0 = 0$, $L = 1$ is called the sigmoid function and can be written as

$$p(\theta) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

Using equation 2.4 we can predict our probability as

$$p(\theta) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k)}} \tag{2.5}$$

One requirement to use a LR is that the training data has to be linearly separable. Which means that for k-dimensional training data we must be able to define a $k-1$-dimensional decision boundary to separate the data into two separate classes.

2.5 Deep Learning (DL)

Deep learning is part of machine learning. Since 2010s three reasons can be given why deep learning has become successful:

- more computation power, due to Moore's law
- more training data available
- novel and better algorithms

The main strength of neural networks is that they can handle complex nonlinear functions and discover dependencies between different inputs. The major downside of neural networks is that they require very large datasets to recognize objects in a realistic setting. [20] Neural networks also are a black box method. A network might have hundreds of nodes and thousands of synapses, making it impossible to determine how the network came up with the answer that it did. Another downside is that there are no definitive rules for choosing some of the training parameters and network size for a particular problem. These decisions usually require a good amount of experimentation. Choosing the wrong parameters can cause the network to overgeneralize on noisy data, or otherwise it might never learn, given the training data you have. [20].

Another advantage of neural networks is compared to traditional machine learning methods used in computer vision, the state-of-the-art approach since 2012 was to use hand-crafted features. Neural networks allow us to no longer manually engineer these features, but let the neural network learn which traits are important of the given training data. [20]

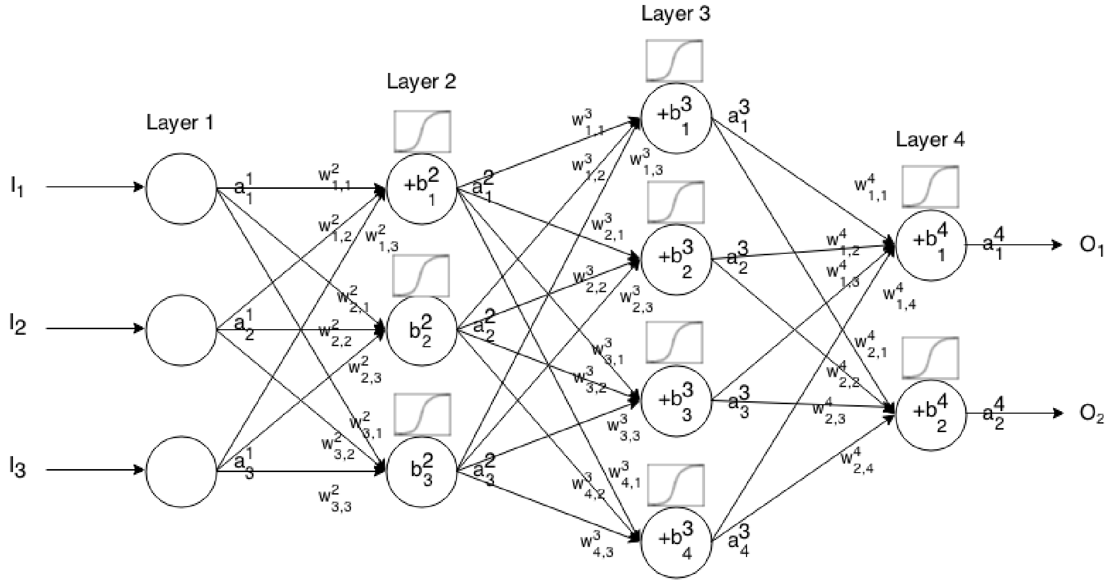


Figure 2.8: Feed-forward neural network with multiple layers [8]

2.5.1 Architecture

Neural networks (NNs) were inspired by the workings of the human brain. A standard NN consists of many simple, connected processors called neurons, each producing a sequence of activations. Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previous active neurons [21].

Figure 2.8 shows a feed-forward neural network

We can write the output of one layer a^i_j as follows

$$a^i_j = \sigma(\sum_k w^i_{jk} \cdot a^{i-1} + b^i_j) \tag{2.6}$$

where:

- σ is an activation function
- w^i_{jk} is the weight from the k^{th} neuron in the $(i-1)^{th}$ layer
- b^i_j is the bias of the j^{th} neuron in the i^{th} layer
- a^i_j represents the activation value of the j^{th} neuron in the i^{th} layer

There are two common neural network architectures: the convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are used to recognize visual patterns directly from pixel images with variability.RNNs are designed to recognize patterns in time series composed by symbols, audio or speech waveforms [10].

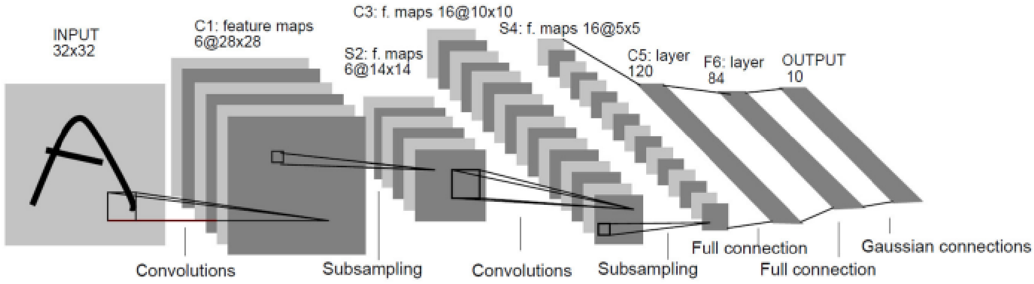


Figure 2.9: LeNet-5 network architecture [9]

2.5.2 Stochastic gradient descent (SGD)

A typical way to minimize a cost function in a NN is using a stochastic gradient descent (SGD). The following equation represents one simultaneous update to the weights θ of a NN:

$$\theta^k = \theta^k - \varepsilon \frac{\partial}{\partial \theta^k} J(\theta) \tag{2.7}$$

where θ is the parameters or weights and ε is the learning rate, the partial derivative $\frac{\partial}{\partial \theta^k} J(\theta)$ measures the rate of increase with respect to the changes in different dimension θ^k. This partial derivative vector is called the gradient.

2.5.3 Convolutional neural networks (CNNs)

Convolutional neural networks (CNNs) are a specialized kind of NNs for processing data that has a known grid-like topology [6]. Examples include time-series data, which can be seen as a 1 dimensional grid taking samples at discrete time intervals, and image data, which can be seen as a 2 dimensional grid of pixels. CNNs have been proven immensely successful in practical applications. CNNs are named after a linear operation that is essential to the network. CNNs were loosely inspired by how the brain processes visual information. The first CNN was proposed in Fukishima's Neocognitron [22]. CNNs were later improved by Yann LeCun [9] in which he applied gradient-based stochastic gradients to very successfully recognize handwritten digits. Historically, a major drawback in 1990s and 200s has been the required computational power to apply CNN to large scale high resolution images.

The figure 2.9 shows LeNet-5 [9], which is applied to document recognition. A typical convolutional neural network consists of several non-linear layers. The input is an image and the output is a vector. The output vector contains the probability of the image belonging to each class. This network has four different types of layers which are named as: convolution, downsampling, full connection, and Gaussian connection. Since the introduction of AlexNet [6] in 2012, networks have more layers and use dropout to reduce overfitting.

2.5.4 Long short term memory (LSTM)

LSTMs are a kind of Recurrent neural networks (RNNs) capable of learning long-term dependencies. The unique about RNNs is that they introduce a memory cell into the architecture of a NN. This

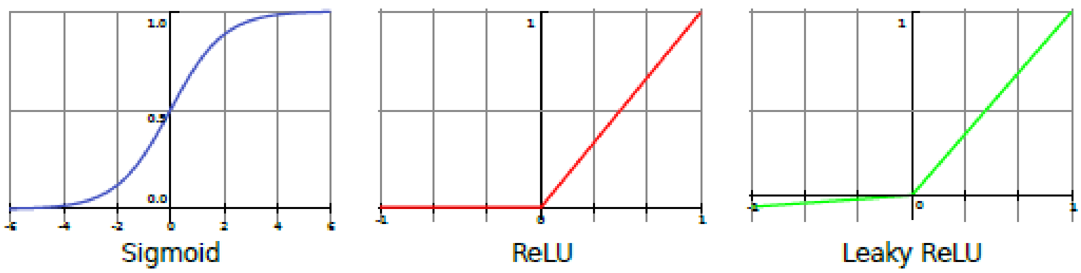


Figure 2.10: Three nonlinear activation functions used by NNs: the sigmoid function, the rectified linear unit (ReLu) and the Leaky ReLu) [10]

allows a RNN compared to a traditional feed-forward neural network to retain a state that can represent information from a longer time window.

LSTMs were first introduced by Hochreiter and Schmidhuber[23], and were further improved to work immensely well on a large variety of problems.

2.5.5 Overfitting and underfitting

Overfitting occurs when the machine learning model is too complex for the data set, which means that the number of its parameters is too high relative to the number of observations. When a very complex model is overfitting, it achieves high performances on unseen data by memorizing the training data which it was trained on. As a consequence, it is not able to generalize the learned patterns in the data to correctly predict new data. In contrast, underfitting happens when a model is too simple. When a model is too simple, it will not be able capture the underlying structure of the training data and will achieve little performance on new data.

2.5.6 Activation functions

In NNs it is convention to apply a non-linear operation or also known as an activation layer after a convolution layer. In figure 2.10 are three types of activation functions shown. In recent years, researchers have discovered that using the ReLu works better than previously used tanh or sigmoid functions by improving computational efficiency without losing significant accuracy [24].

2.5.7 Pooling layers

Pooling or known as subsampling services to reduce the spatial dimensions drastically. This can be done because the exact location of a specific feature is not as important as its relative location to other features.

2.5.8 Dropout layers

One of the problems of NNs is overfitting as discussed in 2.5.5. A simple method proposed in [25] is to drop out a random set of units along with their connections from the NNs. This significantly improves the performance of NNs.

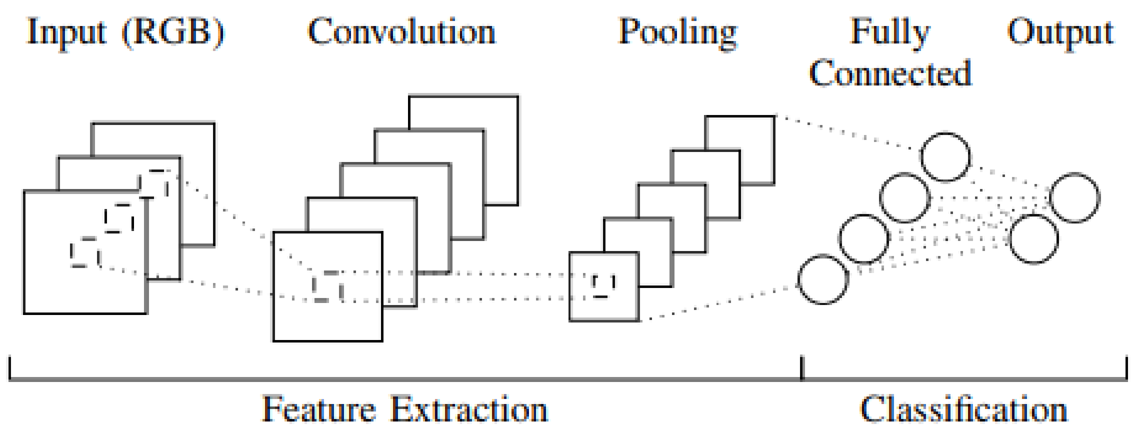


Figure 2.11: Diagram of a CNN. [11]

2.5.9 Fully connected layer

A fully connected layer takes the output of the previous layer and computes which of the activations correspond to a specific class.

2.5.10 Transfer learning

As NNs evolved to contain more layers, they require more training data. Unfortunately for some applications it is unfeasible to collect the necessary amount of data. The idea behind transfer learning is to take a pre-trained model with weights and parameters of a NN that has been trained on a large data set by somebody else, and tuning the model with data specific for that application.

We can freeze the weights of all the other layers except the last layer, as a consequence we can use the pre-trained model as a feature extractor [11]. Figure 2.11 shows the disctinstion of the feature extraction and classification module.

2.6 State-of-the-art Gait Authentication

There are three different approaches in state-of-the-art gait recognition: Machine Vision Based (MVB), Floor Sensor Based (FSB) and Wearable Sensor Based (WSB). There are many studies that demonstrate successfully analyzing the human gait. The different approaches are briefly explained below.

Most of the research based on gait recognition has been using a Machine Vision Based. A MVB system uses gait data recorded from analog and digital cameras. They use machine vision techniques like threshold filtering which converts an image into black and white, pixel count to calculate the number of light or dark pixels, or background segmentation which removes the background of an image to detect gait features.

A different approach is Floor Sensor Based. In this approach sensors are integrated into the floor or are installed in a floor mat. In addition, typically FSB are not used as a standalone system, mainly they are used in combination with other systems as part of a multimodal biometric system.

The most recent approach of these is the Wearable Sensor Based approach. The WSB uses

System	Advantages	Disadvantages
NWS	- Allows simultaneous analysis of multiple gait parameters captured from different approaches - Non restricted by power consumption - Some systems are totally non-intrusive in terms of attaching sensors to the body - Complex analysis systems allow more precision and have more measurement capacity - Better repeatability, reproducibility and less external factor interference due to controlled environment. - Measurement process controlled in real time by the specialist.	- Normal subject gait can be altered due to walking space restrictions required by the measurement system - Expensive equipment and tests - Impossible to monitor real life gait outside the instrumented environment
WS	- Transparent analysis and monitoring of gait during daily activities and on the long term - Cheaper systems -Allows the possibility of deployment in any place, not needing controlled environments - Increasing availability of varied miniaturized sensors - Wireless systems enhance usability - In clinical gait analysis, promotes autonomy and active role of patients	- Power consumption restrictions due to limited battery duration - Complex algorithms needed to estimate parameters from inertial sensors - Allows analysis of limited number of gait parameters - Susceptible to noise and interference of external factors not controlled by specialist

Figure 2.12: Comparison between wearable sensors (WS) and non-wearable sensors (NWS) systems. [12]

motion recording sensors placed on different places on the body such as the hips, wrists or waist to measure characteristics of a person's gait. The most common sensors used are accelerometers and gyroscopes.

An article [12] from 2014 presents a comparison of the above described methods. Figure 2.12 shows a comparison between wearable sensors (WS) and non-wearable sensors (NWS) systems. MVB and FSB systems allow characteristics of the human to be captured much more in-depth. However, these methods require a controlled laboratory environment. Recent developments in WS allow more cost-effective methods in realistic conditions.

A review paper [13] from 2015 compares state-of-the-art using the WSB approach at the time. The basic overview of existing inertial sensor-based gait recognition approaches is shown in figure 2.13. We will discuss more in depth in 2.6.1 In general, all approaches operate according to the following principle:

1. based on the appropriate sensor set-up, inertial data is acquired during a user's walk
2. pre-processing and segmentation step, acquired inertial data is transformed to gait patterns
3. incoming gait patterns are compared with enrolled patterns by appropriate recognition procedure

2.6.1 Wearable Sensor Based approach

In a first step we take a look at the raw data which are collected from inertia based sensors such as an accelerometer and in other works both the accelerometer and gyroscope are used. At first,

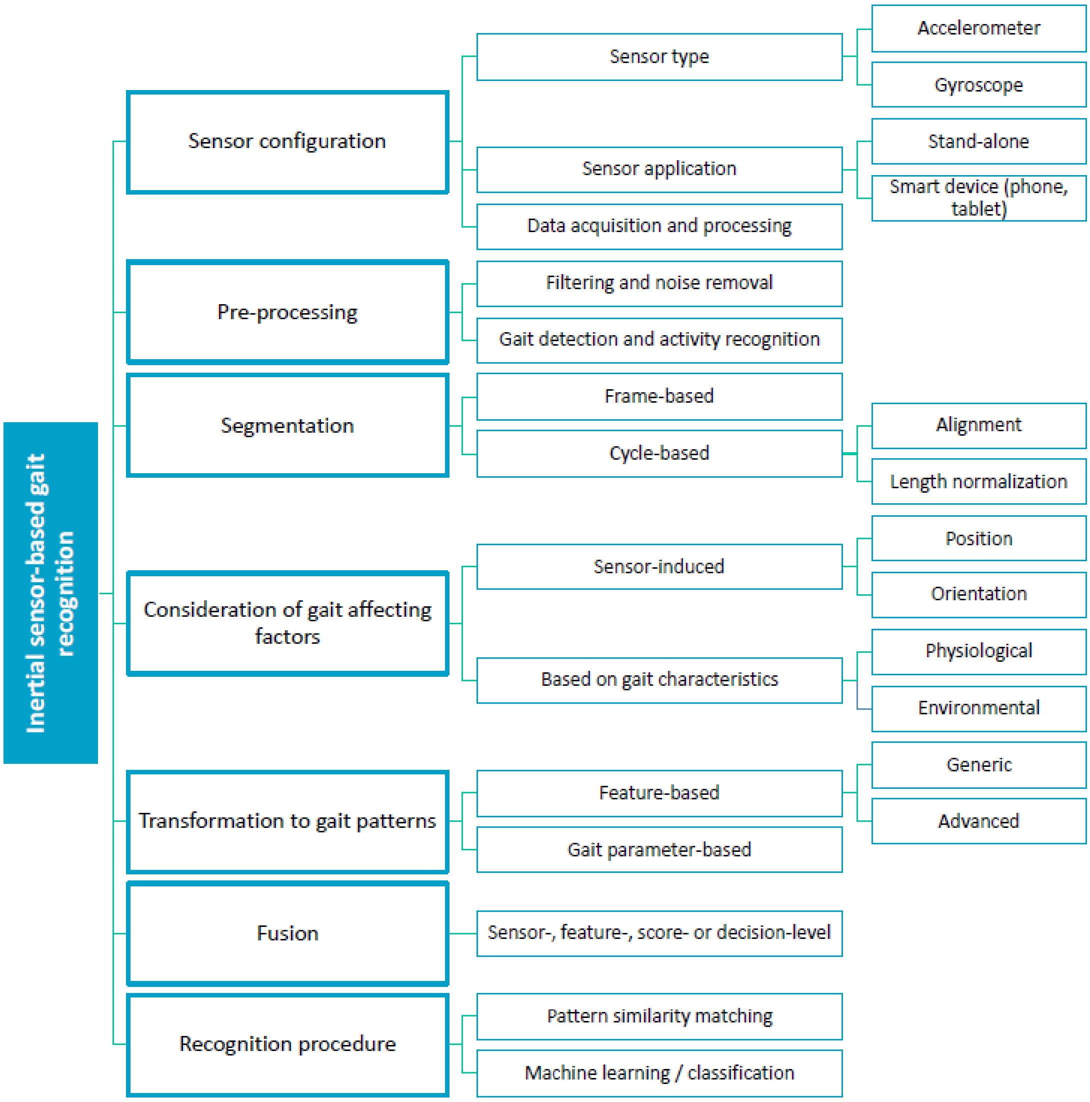


Figure 2.13: Methodological layout of existing Wearable Sensor Based approaches. [13]

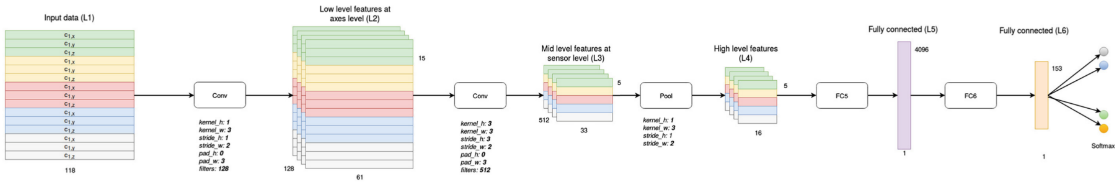


Figure 2.14: Neural network for gait recognition using Wearable Sensor Based approach. [14]

WSB works used specialized sensors, but it has been proven that commercially available sensors such as those that can be found in smartphones and smartwatches can also be effective.

Raw sensor data usually isn't captured on an invariant sample period, thus a first step is to interpolate the raw sensor data. This data is commonly unfit to be used in a learning algorithm directly, so a processing step is needed to transform raw noisy data into useful input data. Filtering can be used to remove sensor-induced gait-affecting factors [13]. High pass filters such as a moving average filter or a Gaussian filter are suitable for this task.

Next we divide gait signal into smaller parts. We defined in 2.3 a gait cycle which can be used as a segmentation method [5, 19, 26]. Other methods divide the data into fixed length segments which should contain at least one gait cycles. In recent works this fixed length method has been found to be more accurate than the more complex cycle detection method [13].

A next step is to transform the data into a feature space to be used by a machine learning classification method. This step is known as a hand-crafted feature extraction method. Finally a supervised traditional machine learning classifier can be constructed. A commonly used ML method is a SVM.

A newer and less common approach is to use DL. This method allows a DL architecture to learn a suitable feature-extraction technique. When used with an appropriate amount of training data, much time can be saved by no longer having to invest time in hand-crafting a feature extraction technique. A recent work [14] uses multiple sensors placed on the body (right wrist, left thigh, left upper arm,...) to extract gait cycles and concatenate them as input data in a CNN. They use data from the ZJU-GaitAcc dataset [27] which is a publicly available dataset of 175 users. The architecture of this network can be seen in figure 2.14. Although there are plenty of participants the walking distance is limited to 20m. In section 2.9 we compare datasets that can be used for gait recognition.

2.7 State-of-the-art Activity Recognition

Figure 2.15 shows the x-axis of an accelerometer and the performed activities of a subject during 60 seconds. The authentication system can only be used when the subject is walking. Therefore, if we want to use a gait-based authentication system continuously, we have to know when a subject is walking. Activity recognition or Human Activity Recognition (HAR) can help us detect when a person is walking or not. It is based on the fact that specific body movements translate into characteristic sensor signal patterns. The difficulty of state-of-the-art HAR activities can range from simple activities such as standing up to more complex activities such as ironing. In this project we are only interested in detection of locomotion activities i.e. walking and non-walking. We can use traditional machine learning to train a classifier on hand-crafted features extracted from our sensor data. Alternatively we can use deep learning to avoid the time-consuming process of manually crafting a feature.

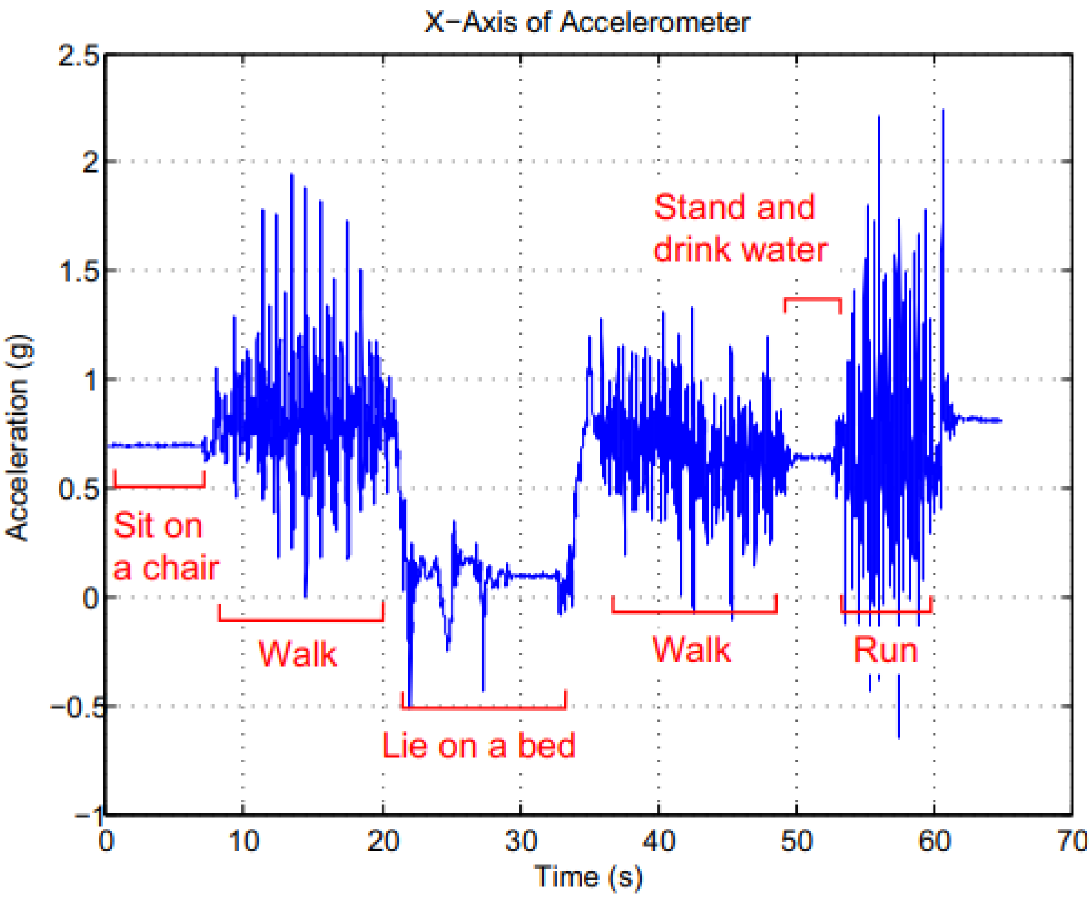


Figure 2.15: An example of activity data from the x-axis of the 3-axis accelerometer [2]

Table 2.1: z-score for different confidence level

Desired confidence level	z-score
99%	2.58
95%	1.96
90%	1.65

2.8 Statistically Significant Validation Population Size

The current validation result is limited to a very specific and a small group of people i.e. 5 technical people from Vasco Data Security in Wemmel. Therefore, we are not sure about the result whether it remains valid if it is extrapolated to a larger group of people. To have more confidence on the validation result, the validation needs to be performed on a larger group of people. It is neither practical nor feasible to study the whole population in any study [28]. Hence, a set of participants is selected from the population, which is less in number, but adequately represents the population from which it is drawn, so that true inferences about the population can be made from the results obtained. In statistics, there exists a way to calculate a sample size from a margin of error and a confidence level. The formula that is widely used for a finite population size is [28]:

$$samplesize = \frac{\frac{z^2 p(1-p)}{e^2}}{1 + \frac{z^2 p(1-p)}{e^2 N}} \tag{2.8}$$

where N is population size, e is margin of error and z is z-score for desired confidence interval. The z-score is the number of standard deviations a given proportion is away from the mean. To find the right z-score to use, refer to the table 2.1:

The definition of various terminologies are explained as follows:

Population size (N) The total number of people in the group we are trying to study. It is speculated that there will be 2.5 billion people by 2019 using the smartphone, assuming that the smartphone will be used to access many resources. Therefore, this value is used to calculate the sample size. N = 2,500,000,000. Since the population size is large, it has small influence on the sample size.

Confidence level (p) The confidence level tells how sure we can be about the result. The most common confidence intervals are 90%, 95%, and 99%. Most researchers use the 95% confidence level. For example, what a 95 percent confidence level is saying is that if the study/experiment were repeated over and over again, the results would match the results from the actual population 95 percent of the time.

Margin of error (e) The margin of error is the range of values below and above the sample statistic in a confidence level. The smaller the margin of error, the closer we are to having the exact answer at a given confidence level. For example, a 95% confidence interval with a 3 percent margin of error means that the statistic will be within 3 percentage points of the real population value 95% of the time.

Table 2.2 gives the sample size for different confidence level and margin of error. Most of the

Table 2.2: sample size for different confidence level and margin of error for p=0.5

	99%	95%	90%
1%	16,641	9,504	6,724
3%	1,849	1,068	748
5%	666	**385**	269

Table 2.3: sample size for different confidence level and margin of error for p=0.95

	99%	95%	90%
1%	3,161	1,824	1,293
3%	351	202	143
5%	126	72	**51**

research uses the value of 95% confidence level and margin of error of 5% and the value indicated in bold in the table can be used as the sample size for validation.

The value p has influence on the sample size. Since in our experiment the participant do not really have to answer any question, they only needs to be in their routine life and not be traveling on holidays. We can assume that the most of the participants are in their routine level. Hence, we can use a higher value of p=0.95. Table 2.3 gives the sample size for the different confidence level and margin of error.

2.9 Datasets

To evaluate our models to solve our two problems discussed in 2.6 and 2.7, we are in need of datasets. We can record our own and/or use a publicly available dataset. In table 2.4 are listed some of the requirements. First, we need the data to be recorded with commercially or equivalent quality wearable sensors. Ideally we'd like the sensors to be positioned on the wrist, alternatively data recorded from a front pocket or lower hip will suffice. Using data from both accelerometer and gyroscope data is preferred and a sufficient amount of data for each user is required. For walk detection we are looking for a minimum of 5 users and the most common locomotion activities. Whereas for authentication we are aiming for 51 users (see 2.8).

In table 2.5 we show a comparison of publicly available datasets that can be used to evaluate methods to solve our gait detection and gait recognition. To evaluate our gait detection methods we can use the IDNet [1] dataset. For gait detection we can best evaluate on PAMAP2 [3] and the

Table 2.4: Requirements of datasets for gait detection and gait recognition

	Gait detection	**Gait recognition**
Size	>=5	>=50
Sensors	accelerometer and/or gyroscope	
Hardware	smartwatch, smartphone	
Position	wrist, front pocket, hip	
Duration	1 minute per activity	5 minutes
Activity	sitting, lying, standing, walking	walking

Table 2.5: Comparison of publicly available datasets

Dataset	**IDNet [1]**	**Z-JU [29]**	**PAMAP2 [3]**	**G. Wu [30]**	**USC-HAD [2]**
Sensors	both	only accelerometer	both	both	both
Position	front pocket	wrist+others	wrist+others	fingertip	hip
Duration	>5 minutes	1 minute	5 minutes	>5 minutes	>5 minutes
Activity	walking	walking	multiple	multiple	multiple
Conditions	some outside	inside	some outside	inside	inside
Subjects	50	175	9	40	14

USC-HAD [2] dataset.

2.10 Conclusion

In the literature study we discussed approaches to build a continuous gait-based authentication system with wearable sensors. We need to solve two problems, gait detection and gait recognition. For gait detection we looked at Human Recognition and for gait recognition we looked at gait recognition problems. For both problems, techniques from traditional machine learning and alternatively deep learning can be used. A dataset for both problems can be recorded or we can use one of the publicly available ones discussed in 2.9.

Chapter 3

Approaches to improvement

In chapter 2 we discussed a general approach to build a continuous gait-based authentication system with wearable sensors. The following sections describe the goal of the project, the implementation of a gait-based authentication model at the start of the project and the approach towards an improved implementation.

3.1 Goal

More specifically than described in 1.2 the main goal is to improve an existing barebone implementation which we will describe in 3.2). We wish to research a computationally efficient and accurate method for gait authentication system that can be implemented on an Android based wearable.

3.2 Beginning

The dataset at the beginning of the project consisted of only a few participants. This data was collected from a variety of Android-based smart watches using the above mentioned application. Unfortunately, this dataset was inadequate and first we obtained a new dataset of which the process is described in section 3.3.

Our initial implementation exists of two applications, one application on an Android wearable and a second application written in Python which runs in a desktop environment.

Initially the functionality of this application on an Android wearable device is limited to only allow the recording of gait data. A user can input their username and press a button to start/stop recording. The User Interface (UI) can be seen in 3.1.

The desktop application was written in Python and consists of a walk detection model in cascade with a binary authentication model. A visual overview is given in figure 3.2. To authenticate a user based on his walking pattern, we have to be certain that the user is walking. We train a walk detection model to differentiate between walking data and other activities. These two subsystems, gait detection and gait recognition are each described in the following subsections 3.2.3 and 3.2.4. Both systems use similar preprocessing and feature extraction techniques, so we will begin by describing these in 3.2.1 and 3.2.2.

Figure 3.1: Initial Android User Interface (UI)

Figure 3.2: Overview of walk detection model and the authentication model

3.2.1 Preprocessing

We linearly interpolate raw data collected from an accelerometer and a gyroscope with a sampling rate of 50Hz to a rate of 100Hz. This ensures that there will be enough data points for our feature extraction technique to work properly. We divide data into segments with length of 10 seconds. Next, we apply a wide box filter of size 20 to reduce some of the noise and improve our signal-to-noise ratio.

3.2.2 Feature Extraction Method

We applied some of the state-of-the-art features such as a feature based on Fast Fourier transform (FFT) and a feature based on Euclidian Distances (ED). Both feature extraction methods are described in pseudo-code in algorithm 1 and algorithm 2, respectively.

Algorithm 1 Euclidian Distances (ED) feature computation

Initialize multiple (*N*) *gait_windows (gw)*
for $i \in \{0, \dots, N\}$ **do**
    for $j \in \{i+1, \dots, N\}$ **do**
        $xscores \leftarrow$ XCORR($gw[i], gw[j]$) ▷ xcorr computes max(sqrt(x[i]-y[i])^2) for each value i for each channel of the accelerometer and gyroscope
    end for
end for ▷ $xscores$ contains the concatenated feature

Algorithm 2 Fast Fourier transform (FFT) feature computation

Initialize a *window_duration* with a duration of 10 seconds and *template_duration* with a duration of 1 second.
function SCORES($signal, template$)
    for $i \in \{1, \dots, 9\}$ **do**
        $fft\_scores \leftarrow$ EUCLIDEAN_DISTANCES($signal[i : i+10], template$);
    end for
    return fft_scores
end function
$template \leftarrow signal[: template\_duration]$ ▷ Select first second of signal as template
for each signal channel data $signal$ **do**
    $scores \leftarrow$ SCORES($signal, template$)
    $fft\_coef \leftarrow$ FFT($scores, sample\_freq, fft\_n = 1024$)
end for ▷ fft_coef contains the concatenated feature

3.2.3 Initial gait detection

The initial gait detection model uses the preprocessing method described in subsection 3.2.1 and the Fast Fourier transform (FFT) feature extraction method described in subsection 3.2.2. The initial gait detection model is a logistic regression model.

3.2.4 Initial gait recognition

The initial gait recognition model uses the preprocessing method described in subsection 3.2.1 and the Euclidian Distances (ED) feature extraction method described in subsection 3.2.2. The initial model uses a Support Vector Machine (SVM).

3.3 Data collection

We possess a dataset of walking data from 5 participants and data from other activities. Data was collected using commercial grade sensors from a single smart watch. They were asked to walk outside for a duration of 10 minutes and repeat this over a period 5 days. The participants were trusted volunteers and were not supervised during their activities. This dataset contain the following activities:

- Lift: stand in the lift while it moves up and down
- Cycling: cycle at a normal pace
- Stairs : walk up/down a few steps along a staircase
- Sit down: sit in the chair carrying out normal routines
- Walking: walk in a short straight line

3.4 Improving with Traditional machine learning

Both approaches described in section 3.2 use traditional machine learning. To improve such a model we can take a look at the architecture as shown in figure 2.5. The most effective area to improve upon in our case is the feature extraction method, but also preprocessing and choosing a different learning algorithm can be used.

3.4.1 Preprocessing

In literature, preprocessing is almost always used to improve the signal-to-noise ratio to gain better results. Instead of using a box filter we apply a Gaussian filter with size 13 to reduce noise.

3.4.2 Derivatives

In state-of-the-art gait recognition and gait detection many different kind of features are used. A recurring idea is to use feature extraction methods to capture frequency components that are unique to gait. A feature that computes the Fast Fourier Transformation (FFT) of a signal can be used, but this method is computationally expensive. We can improve speed by crafting our own feature based on derivatives.

3.5 Deep learning

In section 3.4 we mentioned crafting a new feature. An alternative to this approach is to use CNNs to automatically learn a best feature-extraction technique. CNNs have achieved great results in several applications of pattern recognitions such as gait recognition as proven in [1]. We can try a similar architecture and try improving on our traditional machine learning approach.

3.6 Conclusion

We must begin by collecting more data to apply the above described approaches and evaluate which performs best. Next we can apply changes to our server-side application. Initially some of the state-of-the-art features such as a feature based on Fast Fourier transform (FFT) and a feature based on Euclidian Distances (ED) are applied. We concluded that we can improve by changing the preprocessing pipeline, crafting our own feature based on derivatives and alternatively use CNNs to automatically learn a best feature-extraction technique. Finally we can further develop the wearable application to allow for processing of the data and create a demo.

Chapter 4

Experiments

In this chapter we describe our experiments following the approaches described in the previous chapter 3. The evaluation of these techniques is done on the dataset described in 3.3. We will go over the experiments and their results to improve the gait recognition and then gait detection.

4.1 Gait recognition

The results of the initial gait recognition implementation can be seen in 4.1. We reach a TPR/FPR of 70% and 22%, and an EER of rougly 25%. This is far from ideal, so we can try changing the feature extraction technique. Instead of using algorithm 1, we try algorithm 2 and crafting our own technique.

4.1.1 Derivatives

We craft our own extraction technique based on the derivative filter from computer vision. In this derivative feature extraction method we assume a segmentation length of 10 seconds, which will be referred to as a gait window. We use additional filtering in combination with down-sampling to apply a low-pass filter to our data. This way we can extract different frequencies when applying our derivative filter. Next we slide a derivative filter of [-1 0 1] across our 10 seconds of data. A final step is to do an analysis based on the frequencies extracted by our derivative filter. With a $\sum a_x a_y$ we want to capture the relation of the accelerometer data in the x direction with respect to the y direction.

After multiple experiments where we modified things such as increasing the filter length and increasing amount of repetitions we settled on a best algorithm. A visual of this process can be seen in figure 4.2, where each line represents a different feature extraction technique. In the figure FFT represents the algorithm 2 and the others are modified versions of the derivative feature. The best algorithm is described in pseudocode in algorithm 3.

4.1.2 Filter

A visual representation of using a Gaussian filter compared to a common wide box filter is shown in figure 4.4. A Gaussian filter is preferred as it contains more of the useful data than using a box filter. We can see the results of this in figure 4.3.

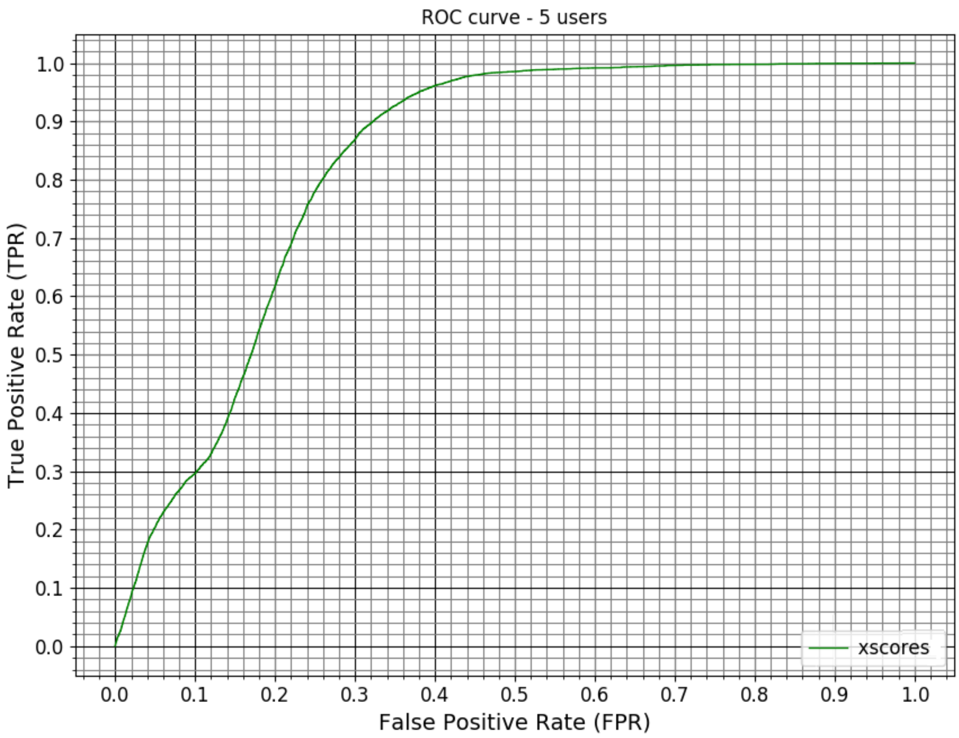


Figure 4.1: ROC curve given for collected dataset of 5 users

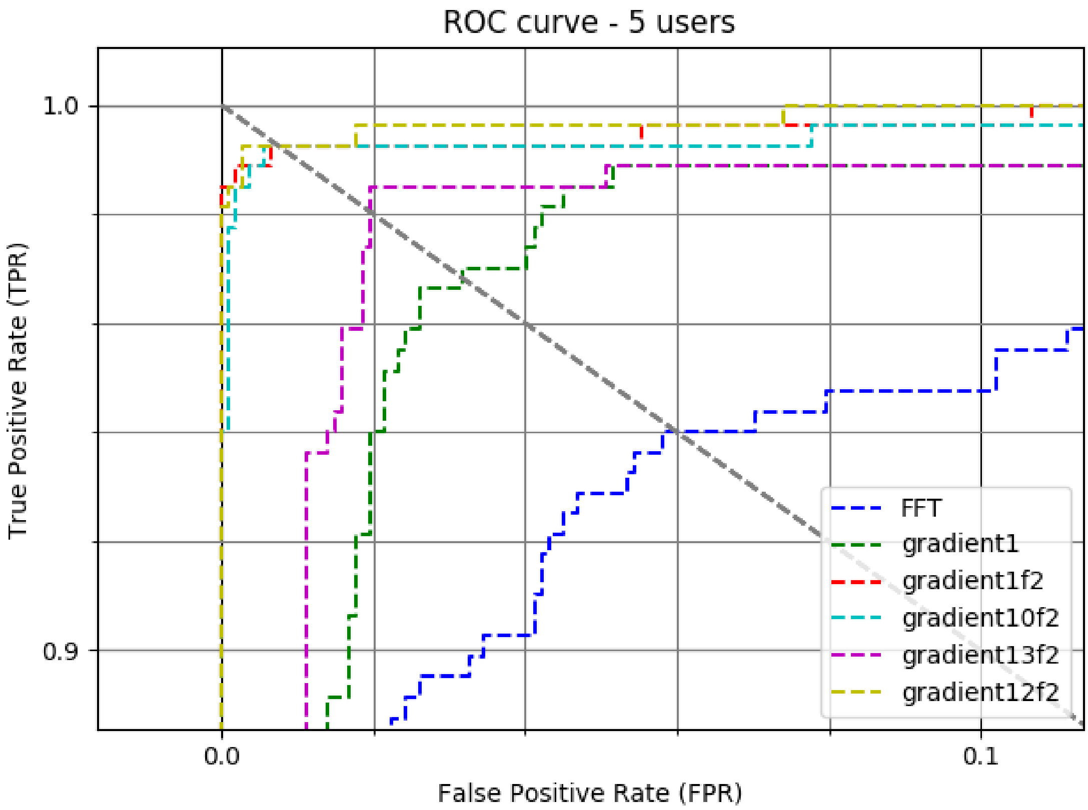


Figure 4.2: ROC curve given for collected dataset of 5 users using different feature extraction techniques

Algorithm 3 Derivatives feature computation

$\sigma \leftarrow 1.6$
$downsample\_rate \leftarrow 2$
function FIRST_ORDER_DERIVATIVE($signal$)
  for each signal channel data $channel$ **do**
    APPLY_CONVOLUTION1D($channel, [-1\ 0\ +1]$)
  end for
end function
function COMPUTE($signal$)
  sum(ax) ▷ accelerometer channels(ax,ay,az)
  sum(|ax|)
  sum(ay)
  sum(|ay|)
  sum(az)
  sum(|az|)
  sum(multiply(ax, ay))
  sum(multiply(ax, az))
  sum(multiply(ay, az))
  sum(|multiply(ax, ay)|)
  sum(|multiply(ax, az)|)
  sum(|multiply(ay, az)|)
  repeat instructions above for gyroscope channels(gx,gy,gz)
end function
$signal \leftarrow$ FIRST_ORDER_DERIVATIVE($signal$)
$full\_res\_coef \leftarrow$ COMPUTE($signal$)
$signal \leftarrow down\_sample(signal, downsample\_rate)$
$signal \leftarrow gaussian\_filter1d(signal, \sigma)$
$1st\_order \leftarrow$ FIRST_ORDER_DERIVATIVE($signal$)
$half\_res\_coef \leftarrow$ COMPUTE($signal$)
$signal \leftarrow down\_sample(signal, downsample\_rate)$
$signal \leftarrow gaussian\_filter1d(signal, \sigma)$
$1st\_order \leftarrow$ FIRST_ORDER_DERIVATIVE($signal$)
$quarter\_res\_coef \leftarrow$ COMPUTE($signal$)
$feature \leftarrow$ Concatenate full, half and quarter resolution ▷ $feature$ contains the derivatives filter

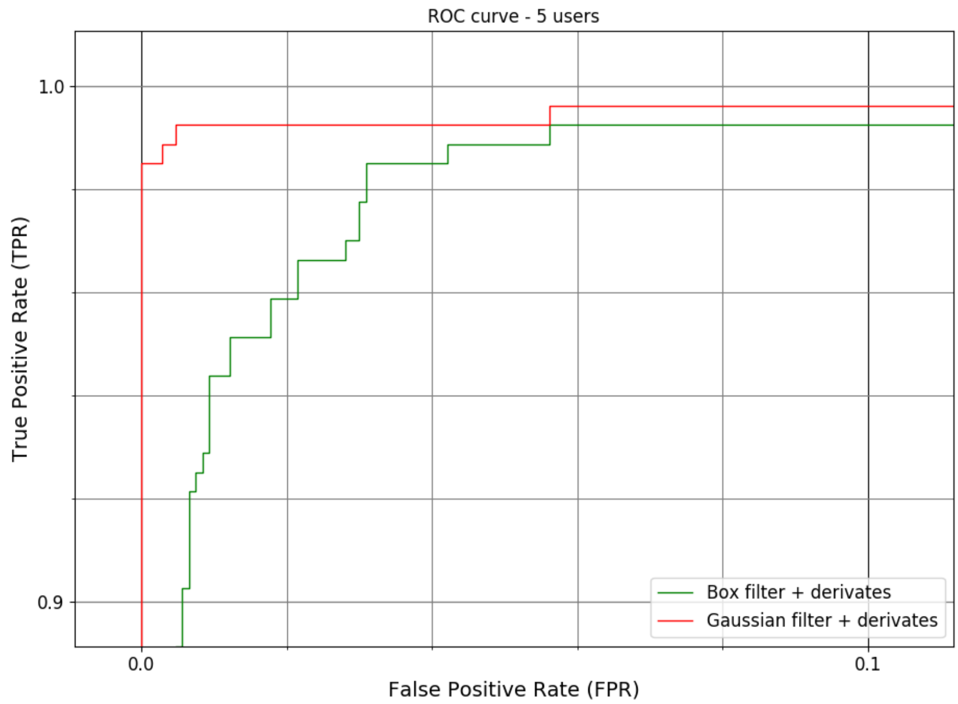


Figure 4.3: ROC curve given for collected dataset of 5 users using different filters and derivative feature

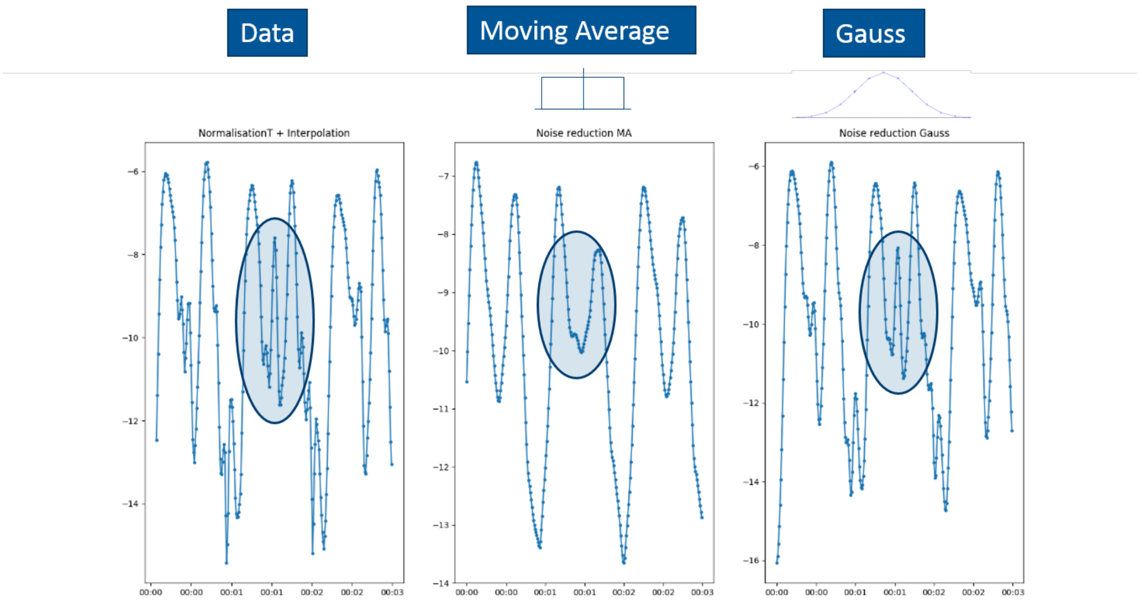


Figure 4.4: Comparison of using a Moving Average (MA) filter with width 20 and a Gaussian filter with size 13. In blue is shown useful information of gait

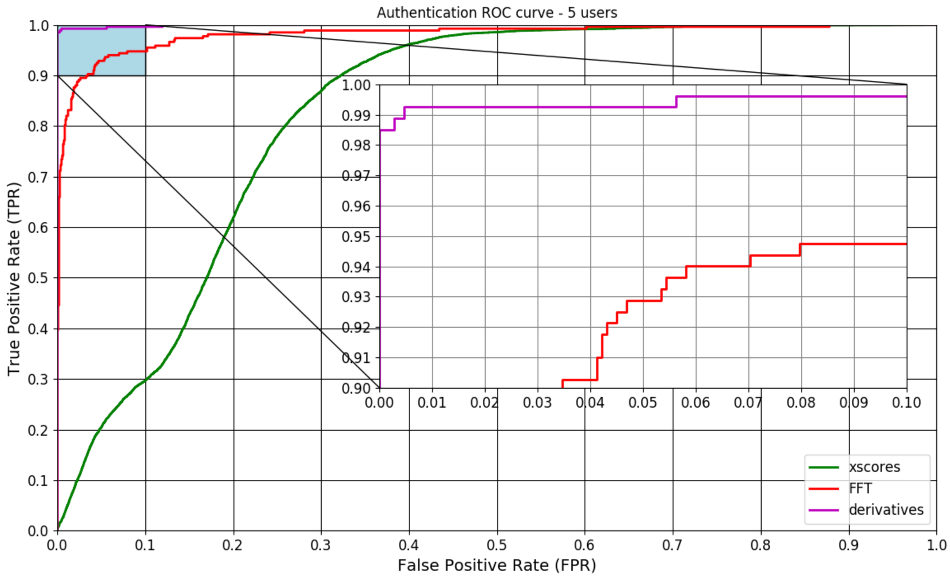


Figure 4.5: ROC curve given for collected dataset of 5 users using ED, FFT and derivative features

4.1.3 Result

We can compare the 3 algorithms 1, 2, 3 in one graph 4.5. We can see the combination of a gaussian filter and both FFT and derivatives filter perform much better than the original ED (xscores) feature using the original preprocessing.

4.2 Gait detection

The problem of gait detection is inherently easier than gait recognition. We applied the same preprocessing technique as described in subsection 4.1.2. We already had good results using algorithm 2 as our feature extraction technique. We experimented with slightly modified version of the derivatives algorithm to select a best feature extraction technique (see figure 4.6). Using our derivative feature described in algorithm 3 we gain even better results. A comparison of these two features is given in figure 4.7. Similarly to our gait recognition problem, the derivative feature and better preprocessing give a more accurate result.

4.3 Android implementation

To the android application on the wearable we've added the inference of the gait detection (walk detection) and gait recognition (authentication) models. Figure 4.8 shows the face of the watch after recording and processing a walk of 23 segments of 10 seconds.

The SenseID wearable (see figure 4.9) has at the time of writing not been completed. Therefore the integration on this new hardware cannot be done. However, the SenseID wearable will also run android and therefore the developed application should be compatible with the new hardware.

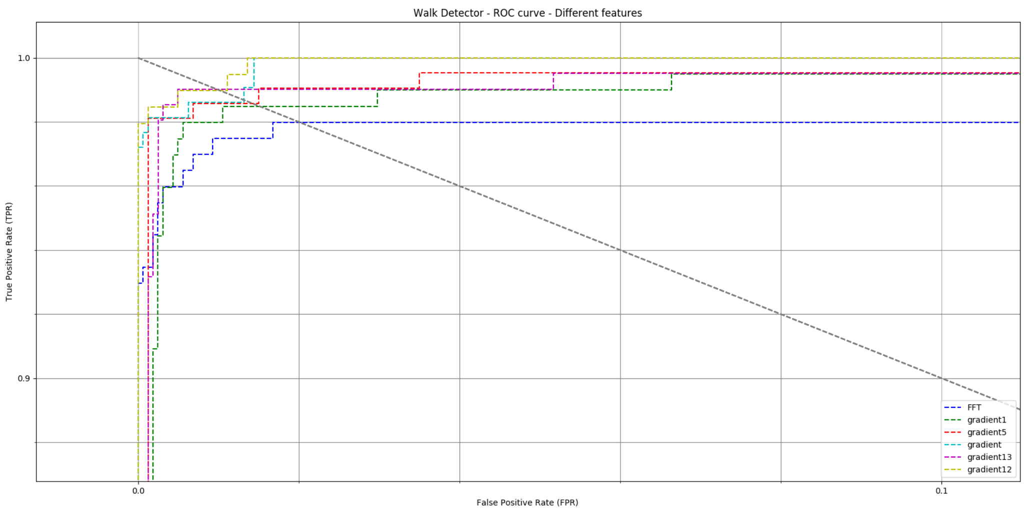


Figure 4.6: ROC curve given for collected dataset of 5 users using FFT and derivative features

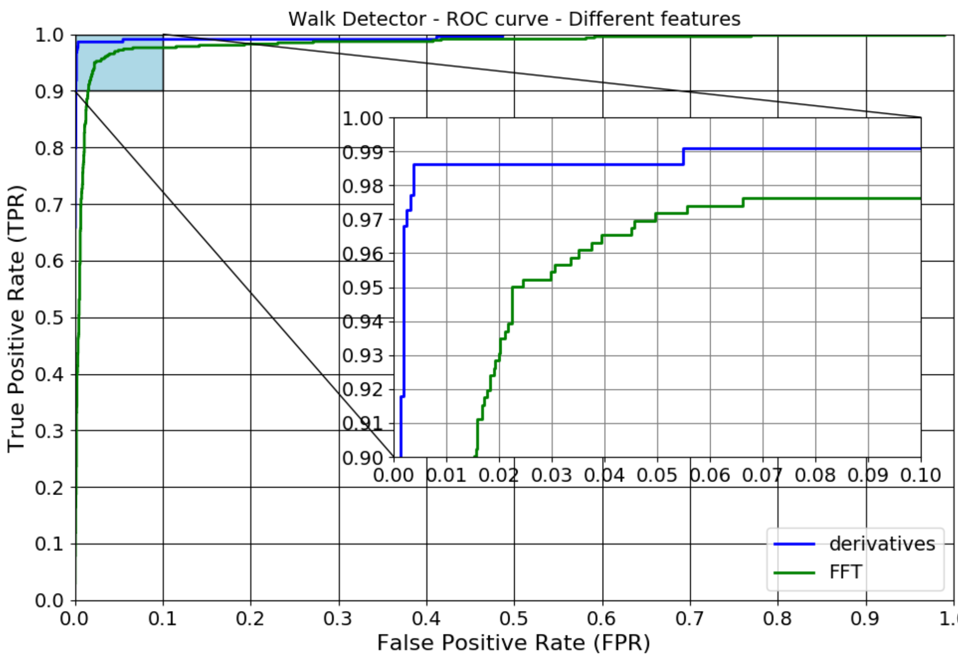


Figure 4.7: ROC curve given for collected dataset of 5 users using FFT and derivative features

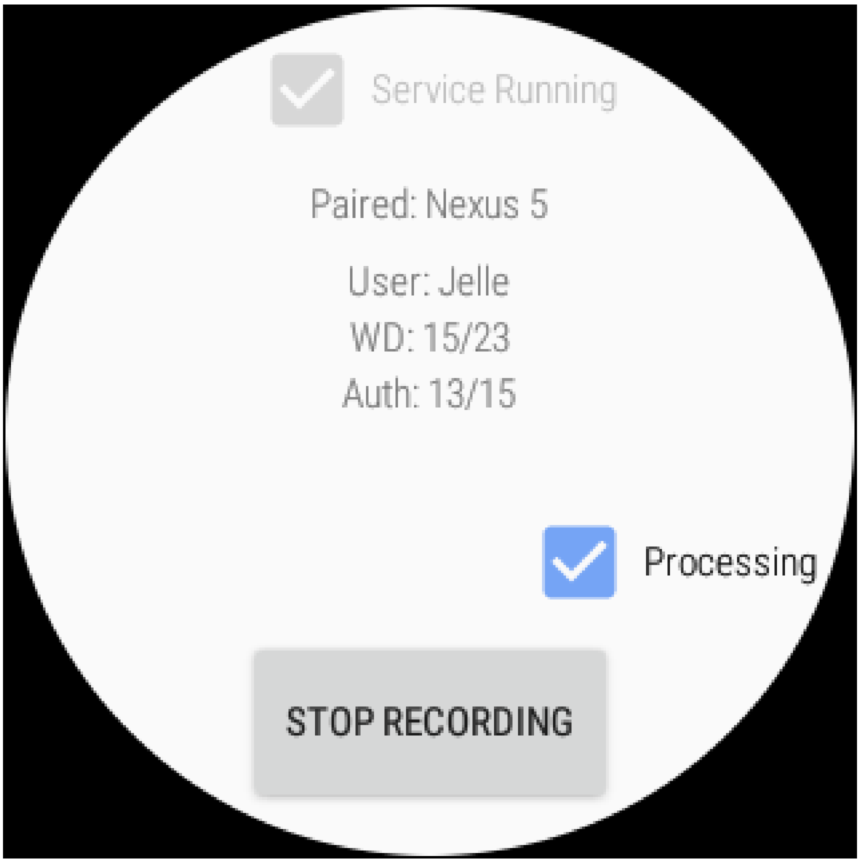


Figure 4.8: Android User Interface (UI)

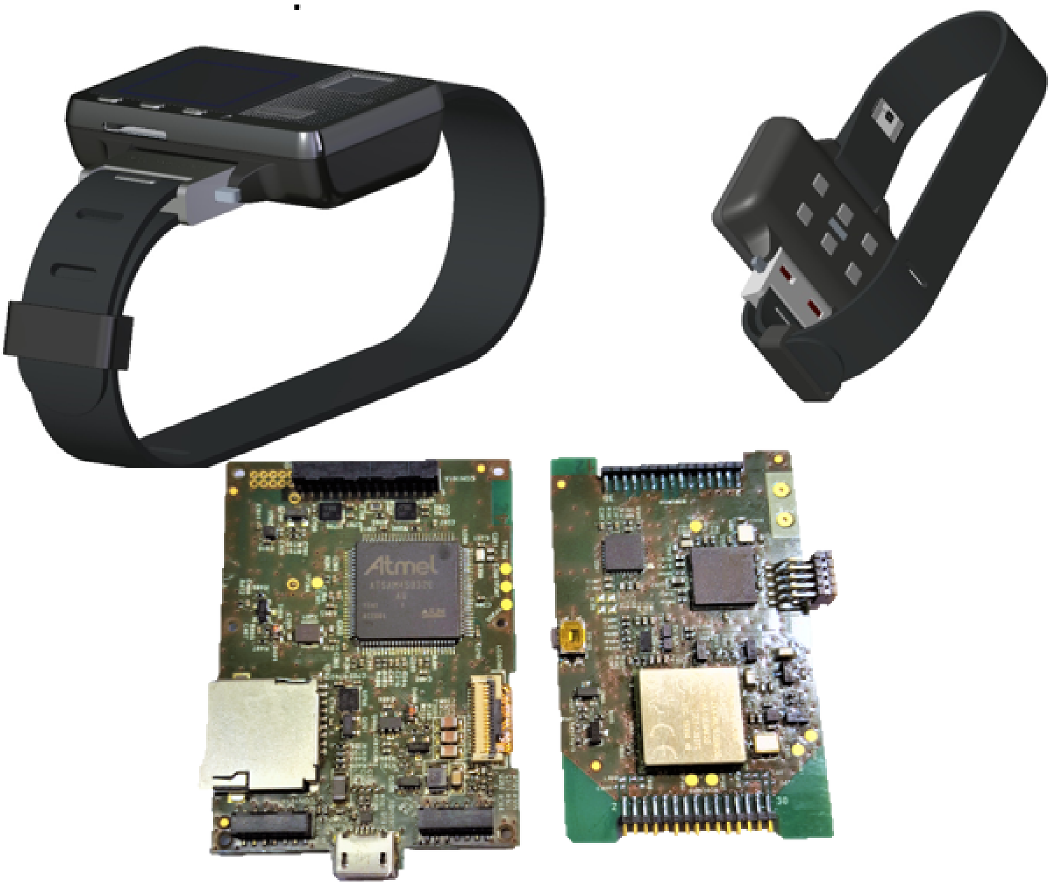


Figure 4.9: SenseID wearable research platform

Table 4.1: Computation time of features ED, FFT and derivatives

Feature	Preprocessing (ms)	Computation Auth Time (ms)	Total (ms)
ED	11	540	551
FFT	11	170	181
derivatives	11	13	24

4.4 Computation time

On the server-side application in Python, we calculated different features on the same data and measured how long they took to compute. The results of which are shown in table 4.1. When comparing the computation time of ED, FFT and the derivative feature we can conclude that derivative feature is a much more computationally efficient method.

4.5 Conclusion

After recording our own dataset, we've succesfully improved upon the initial result by changing our preprocessing pipeline and crafting a new feature extraction technique. However, a dataset of only 5 participants is unsufficient. In chapter 5 we will validate our results on public available datasets described in section 2.9.

Chapter 5

Validation

Currently our dataset has sufficient walking data of only 5 users, which is according to state-of-the-art publications a small dataset. This makes the classification task easier and therefore must be tested with more participants. In 2.9 we listed some feasible datasets on which we can evaluate our techniques. In this chapter we describe how the implemented methods from chapter 4 perform on public datasets.

5.1 Gait recognition

To test our gait recognition methods we validate on the IDNet [1] dataset. The resulting ROC graph is shown in figure 5.1. We can conclude that when we increase the pool of users to a more representative size, the accuracy of our method does not suffer.

5.2 Gait detection

To validate our gait detection methods we use the PAMAP2 [3] dataset and the USC-HAD [2] dataset.

We selected walk data as positive samples and other activities as negative samples. To properly make the trained model generalize we chose to use a cross validation method. Here we choose the data of n amount of subjects for training and the rest for testing. Next, we measure the accuracy and choose a different set of subjects for training. We then measure the accuracy, and so forth.

5.3 Conclusion

We can conclude that the methods in chapter 4 do work on suitable public datasets.

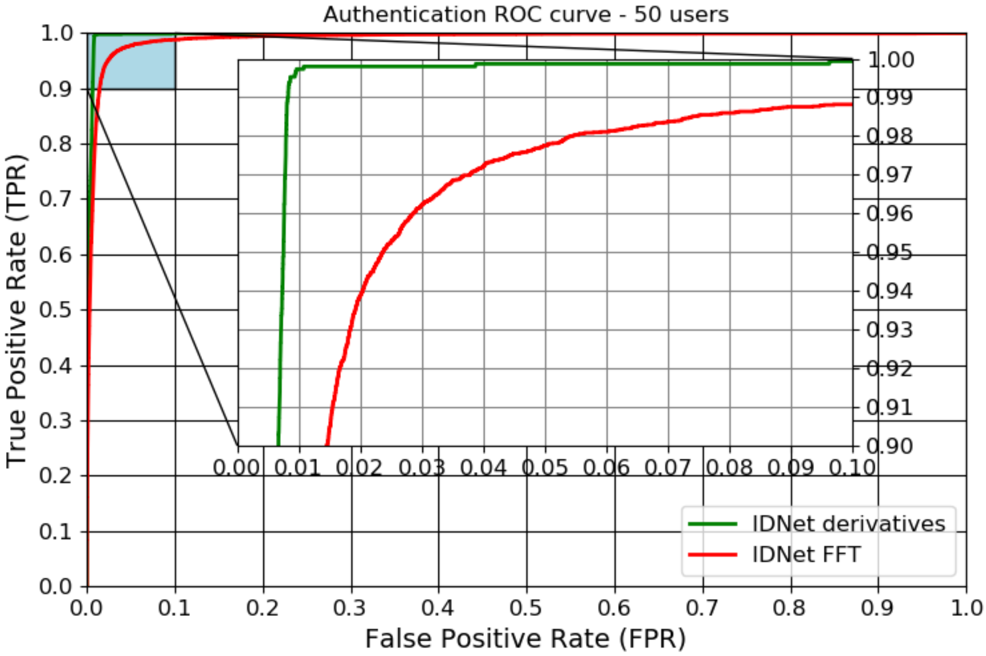


Figure 5.1: ROC curve Authentication IDNet dataset using FFT and derivatives features

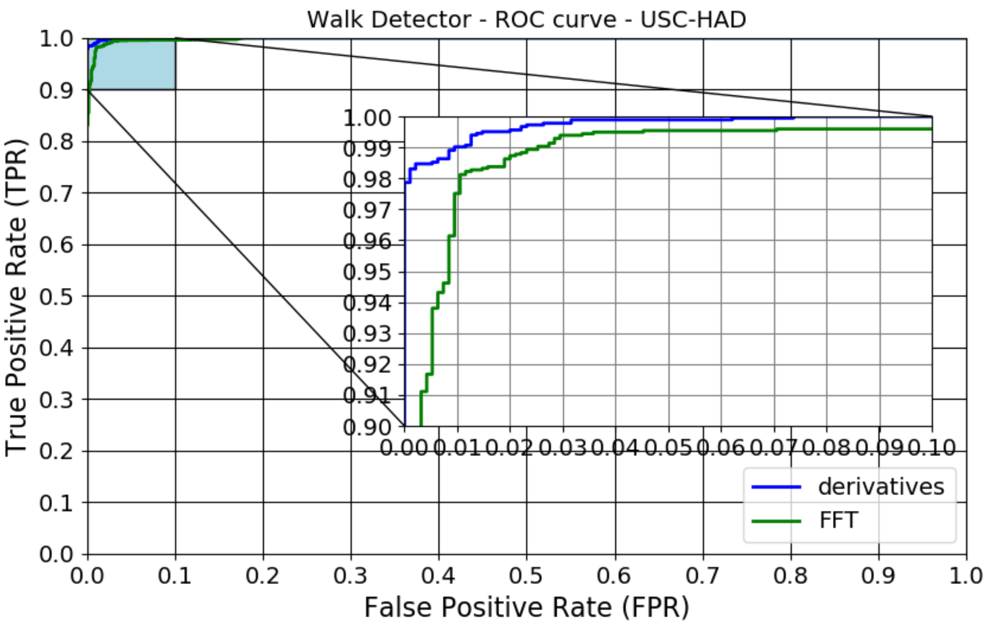


Figure 5.2: ROC curve Gait detection USC-HAD dataset using FFT and derivatives features

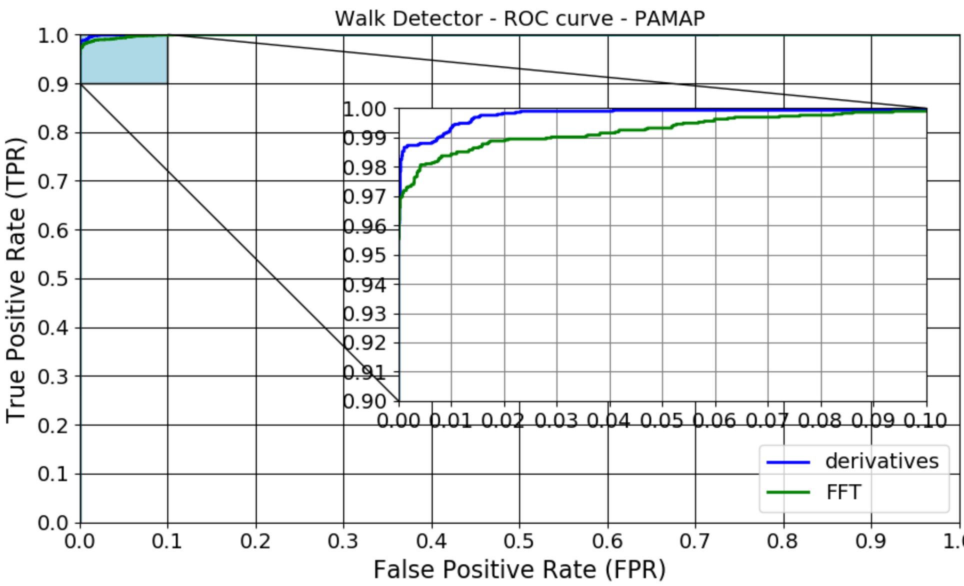


Figure 5.3: ROC curve Gait detection PAMAP2 dataset using FFT and derivatives features

Chapter 6

Discussion and Future Work

The goal of this thesis project was to take a look at existing unobtrusive gait recognition methods and attempt to improve the state-of-the-art algorithms to provide a more computationally efficient and accurate method for computationally limited devices.

We developed a continuous gait-based authentication system by combining two traditional machine learning models to do gait detection and gait recognition. We implemented new preprocessing and a feature extraction technique to improve accuracy and computational speed. The implemented techniques show good results on a new recorded dataset and on public available datasets.

6.1 Future Work

We managed to craft a good feature extraction technique which shows good results, so there is little improvement to be made in terms of accuracy and computational speed. More work can be done to reduce memory and cpu usage as little to no work has been done to optimize the resources on both the wearable application and the server-side application. Furthermore, the training of models has to be done offline, so more work can be done to transfer data from the wearable to the server to enroll new users more easily. Once the development of the SenseID wearable is completed, we will be able to collect walking data for a lot more users over a long duration and test our current results in a realistic environment.

Bibliography

[1] M. Gadaleta and M. Rossi, "Idnet: Smartphone-based gait recognition with convolutional neural networks," *arXiv preprint arXiv:1606.03238*, 2016.

[2] M. Zhang and A. A. Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 1036–1043.

[3] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2012, p. 40.

[4] A. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1262027/

[5] A. Kharb, V. Saini, Y. K. Jain, and S. Dhiman, "A review of gait cycle and its parameters," *IJCEM International Journal of Computational Engineering & Management*, vol. 13, pp. 78–83, 2011.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] K. P. Bennett and E. J. Bredensteiner, "Duality and geometry in SVM classifiers," in *ICML*, 2000, pp. 57–64.

[8] P. V. V, *network with multiple layers feeding-forward the non-linearity to encapsulate complex behavior from simpler neural units*, 2017.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. [Online]. Available: http://ieeexplore.ieee.org/document/726791/

[10] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 406–413, 2016.

[11] L. Hertel, E. Barth, T. Käster, and T. Martinetz, "Deep convolutional neural networks as generic feature extractors," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–4.

[12] A. Muro-de-la Herran, B. García-Zapirain, and A. Méndez-Zorrilla, "Gait Analysis Methods: An Overview of Wearable and Non-Wearable Systems, Highlighting Clinical Applications," *Sensors*, vol. 14, no. 2, pp. 3362–3394, Feb. 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/2/3362/

[13] Y. Zhao and S. Zhou, “Wearable Device-Based Gait Recognition Using Angle Embedded Gait Dynamic Images and a Convolutional Neural Network,” *Sensors*, vol. 17, no. 3, p. 478, Feb. 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/3/478

[14] K.-T. Nguyen, T.-L. Vo-Tran, D.-T. Dinh, and M.-T. Tran, “Gait Recognition with Multi-region Size Convolutional Neural Network for Authentication with Wearable Sensors,” in *Future Data and Security Engineering*, T. K. Dang, R. Wagner, J. Küng, N. Thoai, M. Takizawa, and E. J. Neuhold, Eds. Cham: Springer International Publishing, 2017, vol. 10646, pp. 197–212. [Online]. Available: http://link.springer.com/10.1007/978-3-319-70004-5_14

[15] F. Breitinger and C. Nickel, “User Survey on Phone Security and Usage.” in *BIOSIG*, 2010, pp. 139–144.

[16] D. Todorov, *Mechanics of user identification and authentication: fundamentals of identity management*. Boca Raton: Auerbach Publications, 2007, oCLC: ocm77716961.

[17] M. Whittle, *Gait analysis: an introduction*. Oxford ; Boston: Butterworth-Heinemann, 1991.

[18] K. Holien, “Gait recognition under non-standard circumstances,” Master’s thesis, 2008.

[19] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, “Unobtrusive User-Authentication on Mobile Phones Using Biometric Gait Recognition.” IEEE, Oct. 2010, pp. 306–311. [Online]. Available: http://ieeexplore.ieee.org/document/5638036/

[20] T. Segaran, *Programming collective intelligence: building smart web 2.0 applications*, 1st ed. Beijing ; Sebastapol [CA]: O’Reilly, 2007, oCLC: ocn166886837.

[21] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0893608014002135

[22] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980. [Online]. Available: http://link.springer.com/10.1007/BF00344251

[23] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735

[24] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[25] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.” *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. Makela, and H. Ailisto, “Identifying Users of Portable Devices from Gait Pattern with Accelerometers,” vol. 2. IEEE, 2005, pp. 973–976. [Online]. Available: http://ieeexplore.ieee.org/document/1415569/

[27] Yuting Zhang, Gang Pan, Kui Jia, Minlong Lu, Yueming Wang, and Zhaohui Wu, “Accelerometer-Based Gait Recognition by Sparse Representation of Signature Points With Clusters,” *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1864–1875, Sep. 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6963443

[28] P. Kadam and S. Bhalerao, “Sample size calculation,” *International journal of Ayurveda research*, vol. 1, no. 1, p. 55, 2010.

[29] Y. Zhang, G. Pan, K. Jia, M. Lu, Y. Wang, and Z. Wu, “Accelerometer-based gait recognition by sparse representation of signature points with clusters,” *IEEE transactions on cybernetics*, vol. 45, no. 9, pp. 1864–1875, 2015.

[30] G. Wu, J. Wang, Y. Zhang, and S. Jiang, “A continuous identity authentication scheme based on physiological and behavioral characteristics,” *Sensors*, vol. 18, no. 1, p. 179, 2018.

Tasks

Name	Begin date	End date
Msc Project	24/7/17	27/6/18
Internship	24/7/17	18/8/17
Study existing code	24/7/17	14/8/17
Python implementation	24/7/17	18/8/17
Android implementation	1/8/17	18/8/17
1st Activity Rapport	21/8/17	2/10/17
1st Activity Rapport Deadline	11/10/17	11/10/17
Msc Handbook	3/10/17	7/5/18
Background and Goals	3/10/17	17/11/17
Literature Study	3/10/17	19/2/18
Add literature study, CNN/1d sign processing	12/2/18	6/4/18
Write implementation, start + after dec	5/2/18	6/4/18
Results+etc	1/2/18	7/5/18
Traditional machine learning approach	24/7/17	13/4/18
Study feature extraction techniques	24/7/17	10/11/17
Learning algorithms	24/7/17	14/11/17
Implement Python	24/7/17	18/12/17
Implement Android	31/7/17	18/12/17
Public DS Test WD	15/2/18	13/4/18
Public DS Test AUTH	15/2/18	13/4/18
Deep learning approach	3/10/17	24/5/18
Study literature	3/10/17	29/11/17
Tools installation	3/10/17	16/10/17
DL reading	3/10/17	29/11/17

Tasks

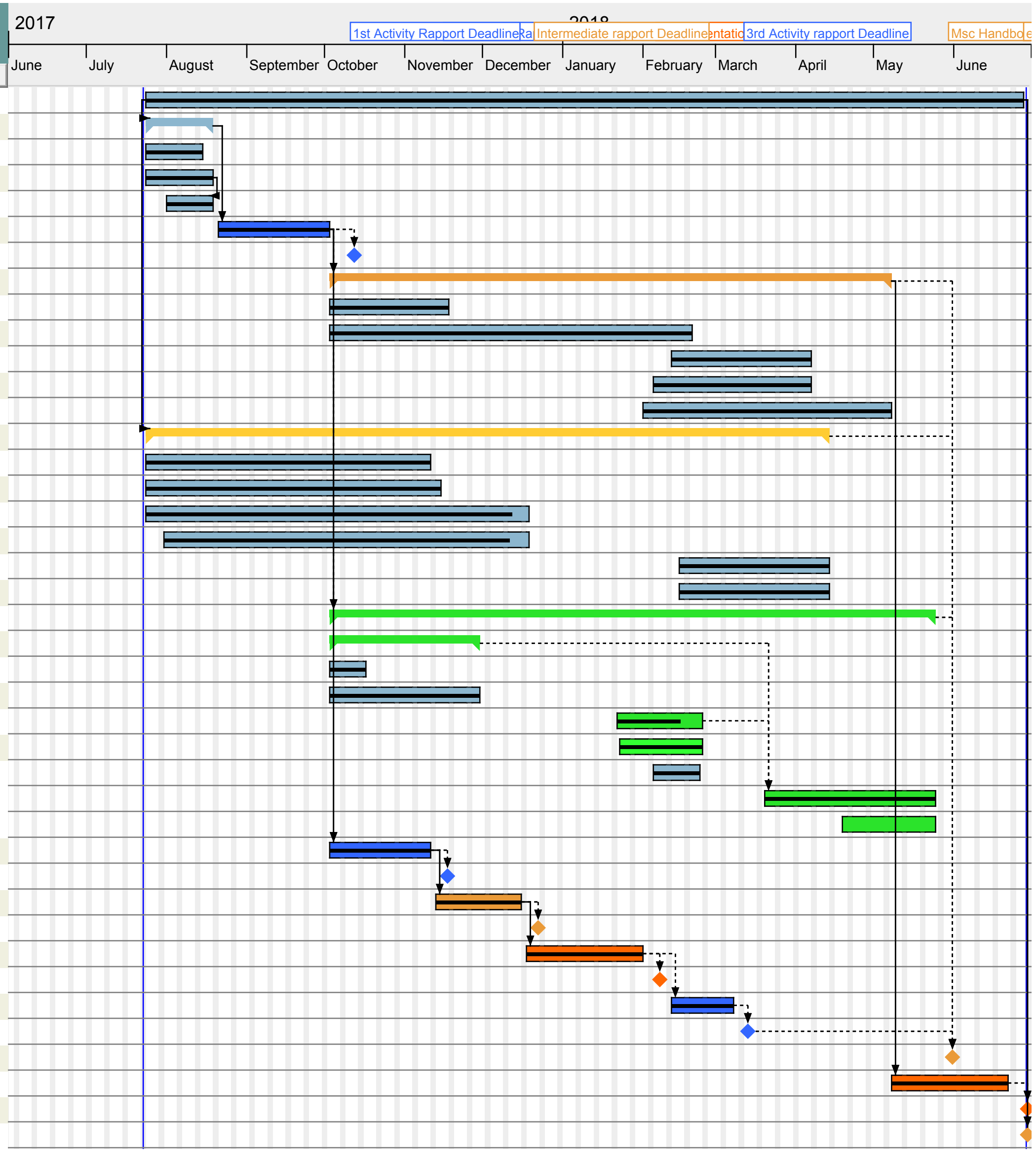
Name	Begin date	End date
Learn DL tools	22/1/18	23/2/18
1d signal processing study	23/1/18	23/2/18
elementary model	5/2/18	22/2/18
Apply DL techniques	20/3/18	24/5/18
Implement Android	19/4/18	24/5/18
2nd Activity Rapport	3/10/17	10/11/17
2nd Activity Rapport Deadline	16/11/17	16/11/17
Intermediate rapport	13/11/17	15/12/17
Intermediate rapport Deadline	21/12/17	21/12/17
1st Presentation	18/12/17	31/1/18
1st Presentation Deadline	6/2/18	6/2/18
3rd Activity rapport	12/2/18	7/3/18
3rd Activity rapport Deadline	12/3/18	12/3/18
Msc Handbook Deadline	30/5/18	30/5/18
2nd Presentation	8/5/18	21/6/18
2nd Presentation Deadline	28/6/18	28/6/18
end MsC	28/6/18	28/6/18

MSc Project

May 30, 2018

Gantt Chart

Name	Begin date	End date
Msc Project	24/7/17	27/6/18
Internship	24/7/17	18/8/17
Study existing code	24/7/17	14/8/17
Python implementation	24/7/17	18/8/17
Android implementation	1/8/17	18/8/17
1st Activity Rapport	21/8/17	2/10/17
1st Activity Rapport Deadline	11/10/17	11/10/17
Msc Handbook	3/10/17	7/5/18
Background and Goals	3/10/17	17/11/17
Literature Study	3/10/17	19/2/18
Add literature study, CNN/1d sign processing	12/2/18	6/4/18
Write implementation, start + after dec	5/2/18	6/4/18
Results+etc	1/2/18	7/5/18
Traditional machine learning approach	24/7/17	13/4/18
Study feature extraction techniques	24/7/17	10/11/17
Learning algorithms	24/7/17	14/11/17
Implement Python	24/7/17	18/12/17
Implement Android	31/7/17	18/12/17
Public DS Test WD	15/2/18	13/4/18
Public DS Test AUTH	15/2/18	13/4/18
Deep learning approach	3/10/17	24/5/18
Study literature	3/10/17	29/11/17
Tools installation	3/10/17	16/10/17
DL reading	3/10/17	29/11/17
Learn DL tools	22/1/18	23/2/18
1d signal processing study	23/1/18	23/2/18
elementary model	5/2/18	22/2/18
Apply DL techniques	20/3/18	24/5/18
Implement Android	19/4/18	24/5/18
2nd Activity Rapport	3/10/17	10/11/17
2nd Activity Rapport Deadline	16/11/17	16/11/17
Intermediate rapport	13/11/17	15/12/17
Intermediate rapport Deadline	21/12/17	21/12/17
1st Presentation	18/12/17	31/1/18
1st Presentation Deadline	6/2/18	6/2/18
3rd Activity rapport	12/2/18	7/3/18
3rd Activity rapport Deadline	12/3/18	12/3/18
Msc Handbook Deadline	30/5/18	30/5/18
2nd Presentation	8/5/18	21/6/18
2nd Presentation Deadline	28/6/18	28/6/18
end MsC	28/6/18	28/6/18



FACULTY OF ENGINEERING TECHNOLOGY
TECHNOLOGY CAMPUS DE NAYER
Jan De Nayerlaan 5
2860 SINT-KATELIJNE-WAVER, Belgium
tel. + 32 15 31 69 44
fet.denayer@kuleuven.be
www.fet.kuleuven.be