Brain Wave Based Authentication

Kennet Fladby - 06HMISA

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2008

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Abstract

Authentication has become an essential part of our everyday lives through systems like passwords, PIN codes, card readers, fingerprint- , retina scanners. All designed with one purpose; to confirm a person's identity. Brain wave based authentication is another addition to the wide range of authentication systems, but with a brand new concept. The electrical activity in a human brain is used to confirm the identity. Instead of physically writing a password, one can think simply think about it. The password or ”pass-thought” can be anything that a human mind may think about, like a color, a feeling, an image, text or something else.

The benefits over other systems are many. With a standard password someone can watch or ”shoulder-surf” what others type, but no one can watch thoughts. Cards and keys can be lost, but the brain is always present. Handicaps can exclude people from systems like fingerprint- or retina scanners, but the brain still works. This thesis research the possibilities for a brain wave based authentication system.

We perform an experiment involving twelve participants using a head set with one sensor designed to record Electroencephalographic (EEG) signals (brain waves). The participants perform eight different tasks in three sessions. We analyze the recorded signals to see if there are enough similarities and differences to distinguish tasks and participants from one another. We look at EEG signals in both the time domain and the frequency domain and extract features in order to apply an algorithm called *Dynamic Time Warping* as well as a *feature based distance metric*.

The results show that similarities are most evident in the same sessions, meaning that the equipment have a noticeable impact on the performance because consecutive recordings are similar. We do end up with a complete authentication system, but based on what we have seen in related work and what we have been able to do with just one sensor, we believe that an implementation of a brain wave based authentication system is just a matter of time.

Sammendrag

Autentisering har blitt en vesentlig del av vår hverdag gjennom systemer som passord, PIN koder, kort lesere, fingeravtrykk-, og netthinne skannere. Alle er utviklet med *én* hensikt; å bekrefte identiteten til en person. Hjernebølge autentisering er enda en type autentiseringssystem i tillegg til de mange systemene som allerede finnes, men med et helt nytt konsept. Den elektriske aktiviteten i hjernen til et menneske blir brukt til å bekrefte identiteten. Istedet for å skrive passordet fysisk, kan man ganske enkelt tenke på det. Passordet eller ”pass-tanken” kan være hva som helst en hjerne kan tenke på, for eksempel en farge, en følelse, et bilde, tekst eller noe annet.

Det er mange fordeler over andre systemer. Ved bruk av standard passord kan noen se eller ”skulder-surfe” hva andre taster, mens ingen kan se andres tanker. Kort og nøkler kan mistes, mens hjernen alltid er med. Folk med handikap kan bli ekskludert fra systemer som for eksempel fingeravtrykk- og netthinne skannere, men hjernen fungerer fortsatt. Denne masteroppgaven forsker på mulighetene til et hjernebølge basert autentiseringssystem. Vi utfører et eksperiment med tolv deltakere med et ”head set” utstyrt med en sensor for å ta opp Elektroencefalogram (EEG) signaler (hjernebølger). Deltakerne utfører åtte forskjellige oppgaver i tre runder. Vi analyserer signalene for å se om det er nok likheter og forskjeller til å skille oppgaver og deltakere fra hverandre. Vi ser på signalene både i tids domenet og frekvens domenet og finner egenskaper såslik at vi kan bruke en algoritme som kalles *Dynamic Time Warping* (DTW) samt en *egenskap-basert sammenlignings metode* (”distance metric”).

Resultatene viser at likheter er mest fremtredene i de samme rundene, noe som betyr at utstyret har en vesentlig betydning på ytelsen, en følge av at signaler tatt opp rett etter hverandre er like. Vi ender ikke opp med et ferdig autentiseringssystem, men basert på hva vi har sett fra tidligere arbeid og hva vi klarer å få til med en sensor, tror vi det bare er et spørsmål om tid før et hjernebølge basert autentiseringssystem ser dagens lys.

Acknowledgments

First of all I would like to thank my supervisor Slobodan Petrović for guiding me in the right direction throughout this thesis, especially regarding statistical calculations and considerations. Patrick Bours deserves a special thanks for helping me with the authentication part of the thesis, as well as Arne Wold and Geir Olav Dyrkolbotn for their excellent knowledge in signal processing. Finally I would like to thank all of my classmates in both Information Security and Media Technology for all the interesting and helpful discussions, and particularly Kjetil Holien for lending me his implementation of the Dynamic Time Warping algorithm.

Revision history

Version #	Changelog	Date
0.1.0	First draft, added relevant parts from the pre-project	14.01.08
0.1.1	Authentication vs identification	14.01.08
0.1.2	Person identification and authentication based on EEG	17.01.08
0.1.3	Vogel, monozygotic twins	22.01.08
0.1.4	Low cost EEG equipment	14.02.08
0.2.0	experimental work	11.03.08
0.2.1	signal processing	02.05.08
0.2.2	feature extraction	03.05.08
0.2.3	distance metric	09.05.08
0.3.0	analysis	13.05.08
0.3.1	feature correlation	19.05.08
0.3.2	Chi-square goodness-of-fit test	20.05.08
0.3.3	Distance categories	22.05.08
0.3.4	DTW and feature based distance metric	23.05.08
0.4.1	Preliminary results	26.05.08
0.4.2	Appendix	26.05.08
0.5.0	First draft of the report	29.05.08
0.6.1	Contributions	27.06.08
0.6.2	Conclusion	29.06.08
0.6.3	Further Work	30.06.08
1.0.0	Final delivered version	30.06.08

Contents

List of Figures

List of Tables

1 Introduction

1.1 Topic

Authentication has become an essential part of our everyday lives through systems like passwords, pin codes, card readers, fingerprint- , and retina scanners. All designed with one purpose; to confirm a person's identity (Figure 1). Authentication has its use in several areas, but the main goal is to protect something of value where access is limited to just one or more individuals or groups. Some examples are airports, computers, homes or bank accounts. Brain wave based authentication is another addition to the wide range of authentication systems, but with a brand new concept. The electrical activity in a human brain is used to confirm the identity. Instead of physically writing a password, one can think simply think about it. The password or "pass-thought" can be anything that a human mind may think about, like a color, a feeling, an image, text or something else. The whole concept may sound sound a bit like science fiction, but the equipment to record brain waves is getting better, cheaper and more available as well as the methods to analyze them.

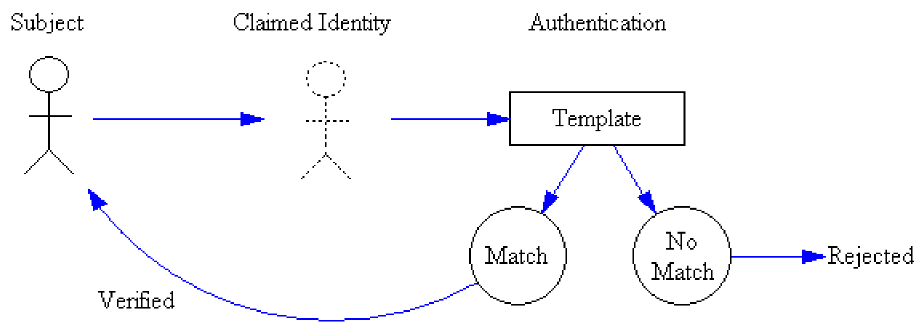


Figure 1: The principle of authentication; one-to-one matching

An adult brain contains about 100 billion neurons that each generates and leads electrical charges. The sum of all these very small electrical charges contributes to the generation of an electric field with fluctuating electrical potentials around our scalp. The fluctuating potentials are typically in the μV range and it is these fluctuations that can be measured [1]. The potentials are measured between two or more points called electrodes or sensors, which is placed on the scalp at different locations. The measurement have been named Electroencephalography (EEG) and resembles waves (Figure 2), which is why the term brain waves is used when referring EEG signals. This thesis research the possibilities for a brain wave based authentication system. We perform an experiment involving twelve participants using a head set with one sensor designed to record EEG signals (brain waves). The participants perform eight different tasks in three sessions.

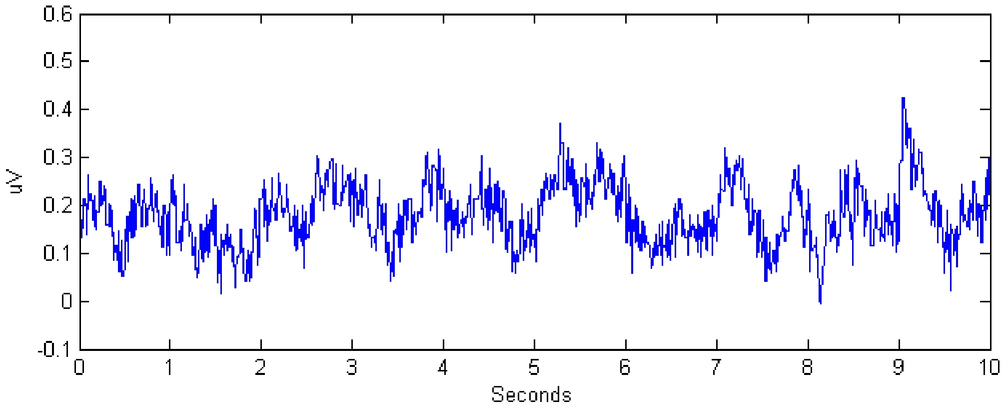


Figure 2: Example of a 10 second EEG signal captured with one sensor with 128Hz sample frequency

We analyze the recorded signals to see if there are enough similarities and differences to distinguish tasks and participants from one another.

1.2 Keywords

Security and protection, Authentication, Pattern recognition, Signal processing, Information security

1.3 Problem description

There are three basic forms of authentication; *something-you-have*, *something-you-know*, and *something-you-are* [2]. Something-you-have can be objects like a key or passport and people have to be very careful not to loose the object or get it stolen. Something-you-know is based on secret knowledge like passwords or PIN codes and the secret must never be written down, forgotten, or told to others. A quote from [3] gives a fun explanation of good password practice:

> *A password must be impossible to remember, and never written down.*

This is a difficult task, especially considering the huge amount of different passwords and codes we have to remember today. Something-you-are involves person specific features like fingerprints, voice, face, and gait. Authentication based on such features is called *biometric authentication*. Brain wave based authentication is a combination of something-you-know and something-you-are when the person involved has to think about something specific, but it can also be just something-you-are when the brain waves are used directly as a biometric.

The most important part of any authentication system is that true identities (clients) are verified and that false identities (impostors) are rejected. In a password system the password is either right or wrong, but with biometric authentication there is an uncertainty involved because the equipment that measure the biometric feature rarely provide exactly the same data twice. The reason is that external parameters like finger placement, head rotation, facial hair, location etc are present. The challenge is to overcome these problems in such a way that even two slightly different sets of data can be verified to originate from the same person. There is usually a threshold that decide how different two different sets of data is allowed to be before they are rejected, and as a consequence there is a chance that some clients are falsely rejected and some impostors are falsely verified. Biometric authentication therefore introduce two error rates; False Non-Match Rate (FNMR), the rate at which clients are falsely rejected by the system, and False Match Rate (FMR), the rate at which impostors are falsely verified by the system. As such the main problem in this thesis is two compare two or more EEG signals and decide whether they are from the same person or not, and get as low FNMR and FMR as possible.

1.4 Justification, motivation and benefits

The basis for why it is reasonable to believe that a brain wave based authentication system is possible dates back to the 1960's when Vogel discovered a direct connection between a person's EEG signals and his/hers genetic code (DNA) [4]. Monozygotic (identical) twins were shown to have the same EEG patterns in the same situations and even changes related to aging were similar. This is supported by [5] where the authors used EEG directly as a biometric with promising results.

In order to get an efficient biometric authentication system there are seven requirements that must be considered [6]:

- Universality: Every individual required to use the system should have the feature.
- Distinctiveness: The feature should be unique between all individuals.
- Permanence: The feature should not change significantly over time.
- Collectability: It should be easy to acquire measurements of the feature.
- Performance: Speed, accuracy and strength of the system used.
- Acceptability: The system and feature to be used has to be accepted by the public.
- Circumvention: How easy is it to evade the system.

A brain wave based authentication system has many benefits with this in mind. Every human has a brain (universality) so while cards and keys can be lost, the brain is always present. And while people with physical handicaps like missing limbs, total or partial paralysis can be excluded from systems like fingerprint- or retina scanners, the brain still works. With a standard password someone can watch or ”shoulder-surf” what the user is typing, but no one can watch a users thoughts. It has been demonstrated that fingerprint systems can be fooled by making fake prints (a simple search on the Internet results in many sites that actually describe how to do it), but is it possible to fake brain waves? The complexity of the brain (distinctiveness) implies that it is very hard for an impostor to mimic another person's brain (circumvention). So even if the ”pass-thought” is written down or shared it will be difficult for others to recreate the thought. These are all benefits that may reduce the increasing amount of ”identity-theft” where people get their identity ”stolen” by impostors that get a hold of other person's secret and personal information [7].

New equipment even makes it easy to capture brain waves (collectability) and the only required operation is to attach a head set on the client and press ”record”. There might be a problem with age as brain waves may change over time as persons get older (permanence) [4], so the system would have to adapt to these changes.

Because it is a new kind of technology with little research and no implementations yet, brain wave authentication may prove to be expensive first and even somewhat non user-friendly e.g. if it requires clients to wear unfamiliar equipment (acceptability). But over time, should it prove feasible to implement, every party involved in the authentication process will benefit from this technology. The company that wishes to protect something will have high security, those who manufacture the technology will have a huge customer base, and the people using the system will not have to remember the plethora of passwords that we have to today.

1.5 Research questions

1 Is it possible to authenticate by means of brain waves with only one EEG sensor?

2 What features should be extracted from the signals?

3 Do we have to authenticate based on a client's thoughts or can we just use a client's brain waves directly?

1.6 Contributions

Research directly related to authentication by means of brain waves is rather limited in terms of the number of published articles and related work, so any research in the field can be considered a contribution on some level. Our attempt is based on only one EEG sensor, which is a great challenge as this is the absolute minimum requirement to record EEG signals. In the analysis we apply an algorithm called *Dynamic Time Warping* (DTW) in the signal matching process, which has not yet been done with EEG signals. In addition to DTW we try a feature extraction approach based on previous work [8] where the authors did task classification with only two sensors. But instead of feeding the features into a neural network for task classification we attempt a distance metric approach aimed at authentication.

2 Related work and theoretical considerations

Brain waves can currently be recorded with sensors placed at locations classified into three groups:

- Invasive: The device is implanted directly into the ”gray-matter” of the brain.
- Partially-invasive: The device is attached to the inside of the skull, not touching the ”gray-”matter.
- Non-invasive: The device is attached directly to the scalp or at a distance to receive wireless brain waves.

In authentication it is very important that people accept the system (acceptability). With this in mind it is safe to say that a non-invasive method of capturing brain wave signals is the best approach.

An experiment performed in 1977 successfully used a non-invasive method to analyze electrical activity from the brain [1] by using EEG signals. The test subjects were able to navigate a symbol through a maze on a CRT display with a system that analyzed EEG signals in response to the direction of the subject’s gaze, based on external stimuli to the eye. Brain activity was recorded through EEG signals collected on the human scalp by placing electrodes on five scalp locations and both ears. The signals appeared to be confined to low frequencies, especially around 10 Hz alpha frequency, with amplitudes between 5-50mV. The authors found it difficult to extract any useful information by EEG signals alone because they generate a continuous electrical activity spatially distributed over the scalp. Instead they focused on *Event Related Potential* (ERP) that are microscopic potentials embedded in the continuous signals of EEG. The ERP potential occur when the brain responds to external stimuli [9], e.g. a flashlight directed to the eyes).

- Sensory ERPs: Responses that have been elicited by external stimuli. Their presence is most prominent at short ”latencies” (e.g., within 50 to 150 ms).
- Motor ERPs: Responses found accompanying voluntary movement that may in fact precede the actual behavioral event.
- ”Long Latency” Potentials: These refer to potential changes taking place some 250 to 450 ms after the initial event. Most prominent in the literature is a positive deflection occurring around 300 ms, today called P300 potentials.
- Artifacts: Potential fluctuations of non-neural origin are called artifacts. These include *electroocular potentials* (EOG) and muscle potentials from neck, scalp and face (including eye blinks), as well as *electrocardiographic signals* (ECG).

Even though it was brain activity that decided whether to move the symbol up, down, left, or right, it required the test subjects to physically move their eyes and react to external stimuli.

EEG recording through electrodes placed on the scalp is still the preferred way to measure electric activity in the brain because of its non-invasive nature and excellent temporal resolution (of the order of milliseconds) [10]. EEG equipment is also mobile and inexpensive [11].

There are several companies that have developed modern devices to capture brain activity. *Brain Products* has a number of products for recording EEG signals [12]. Software, amplifiers, electrode caps, and accessories with a selection of devices in each group. *Emotiv Systems* has a product called Project Epoc [13].

> *Project Epoc is a headset that uses a set of sensors to tune into electric signals naturally produced by the brain to detect player thoughts, feelings and expression. It connects wirelessly with all game platforms from consoles to PCs.*

Another company, *NeuroSky*, has a similar technology called *ThinkGearTM* [14]:

> *Brainwave signals, eye movements, and other bio-signals are captured and amplified via our patented Dry-Active Sensor technology. Non-Invasive, Small Form Factor, Dry active sensors that do not use contact gels, Accuracy.*

2.1 Brain computer interface

A Brain Computer Interface (BCI) is an interface which allows communication directly between a human brain and an external device [15, 16] (Figure 3). The external device can be any device that requires input e.g. a computer game [17], a cellular phone [18], or a robotic arm [19]. People with physical limitations incapable of interacting normally with external devices would benefit greatly from such interfaces. Many people are reluctant to the technology because they feel it it is very unnatural, but a quote by Bach-y-Rita (Biomedical Engineering and Rehabilitation Medicine Professor) says something we tend to forget:

> *We don't see with our eyes, or feel with our hands; we see and feel with our brain*

So even though it might seem dangerous for some, the BCI systems are only working with something that is always present, the electrical activity in our brain.

In 1990 the authors of [21] showed it was possible to distinguish between five mental tasks, using only EEG with no physical action required. The authors started their research based on the *alpha band asymmetry* as explained in [22] were the authors found the alpha band power (8-13Hz) to be less in the left hemisphere- than the right hemisphere of the brain for verbal tasks, and vice versa with spatial tasks. Asymmetry was also found in non-motor tasks by [23] and [24], and such asymmetry suggested it would be possible to deduct enough information from EEG alone to distinguish between mental tasks. The different tasks they used was:

1 Baseline measurements: There was no mental task to be performed here. This task was used as a control and baseline measure.

2 Complex problem solving:The subject was given a nontrivial multiplication problem to solve without vocalizing or making overt movements.

3 Geometric figure rotation: The subject was instructed to visualize a rotating complex three dimensional block figure shown beforehand.

4 Mental letter composing: The subject was instructed to mentally compose a letter to a friend or relative without vocalizing.

5 Visual counting: The subject was asked to imagine a blackboard and to visualize numbers being written on the board sequentially, with the previous number being erased before the next number was written.

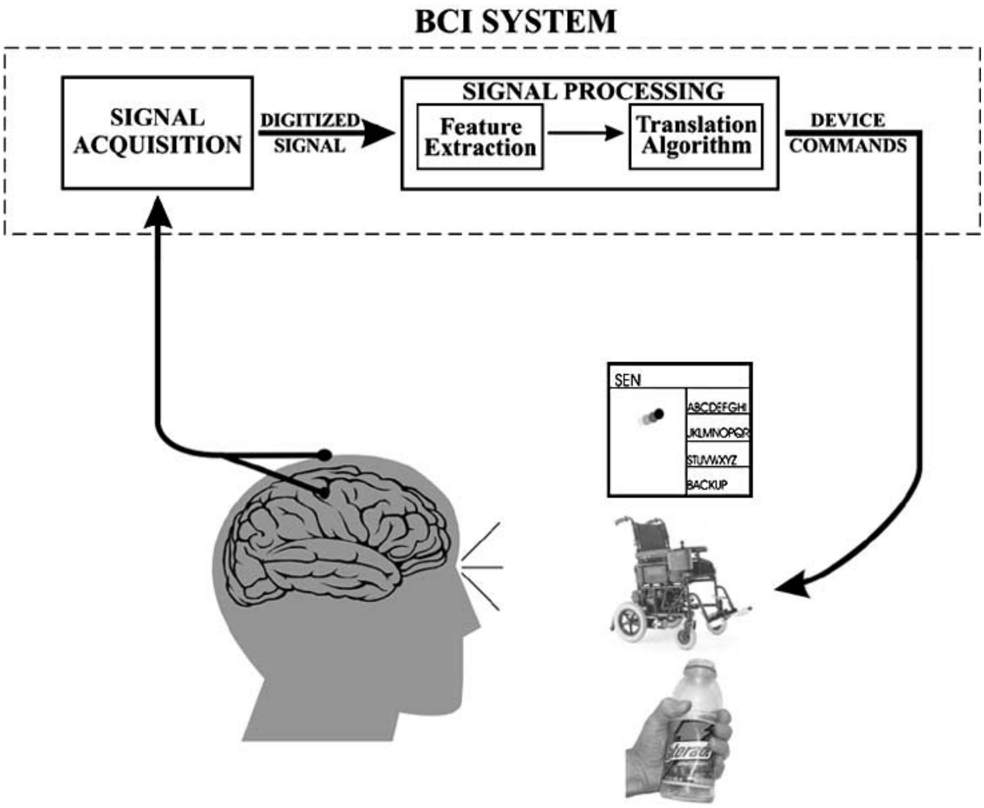


Figure 3: The basic design and operation of a BCI system. Signals from a person's brain is processed and translated into device commands like typing letters, controlling a wheelchair and grabbing a soda can. (Figure from [20])

The test results showed they were able to distinguish between the tasks with an average accuracy of 81.0%, 82.3%, and 84.6% using the *Wiener-Khinchine-*, *Burg Spectrum-* and the *Burg AR* coefficient methods respectively.

Most of the research in the domain of BCI have been performed with equipment capable of recording EEG signals with 32 electrodes or more, but this kind of equipment tend to be very expensive and cannot be bought of the shelf. This triggered a couple of researches to explore the possibilities of using low-cost equipment with fewer electrodes for task classification [8] were eight clients participated in two experiments with different setups. In the first experiment the clients had to keep their eyes closed and perform three different tasks; *Rest* (rest in a normal fashion), *Mental Arithmetic* (multiplication of two numbers), and *Mental Rotation* (mentally rotate a 3D object) as proposed in [21]. The second experiment required the clients to play a computer game with three different tasks; *Rest* (rest in a normal fashion), *Solo* (play the game alone), and *Play* (play against an opponent). Each task in both experiments was recorded over 14 seconds with 18 recordings per task. The authors used feature extraction on each recording to get the task data and used *Bayesian Network* classifier to classify each task. The average classification accuracy in the first experiment were 68.3% when they tested all three tasks against each other, and 84.4% when they tested two and two task against each other. In the second experiment the average classification accuracy was 78.2% when they tested all three tasks against each other, and 90.2% when they tested two and two task against each other. The best results were achieved when "rest" were classified against "solo" and "play" in the game experiment. In other words the best results were achieved when "no activity" was classified against "some activity".

Currently there are no official implementations of a system that utilizes brain waves to authenticate an individual, but an idea called *Pass-thoughts* was presented in [11]. The authors believe that recent advances in BCI technology give evidence that an implementation of a BCI authentication system is possible. The idea is to extract as much entropy as possible from a client's brain signals upon "transmitting" a thought. A thought can be anything and the size of the pass-thought space is not yet known. The number of neurons in a typical adult brain is approximately 100 billion. Assuming each neuron could only store one bit of information, a key space of up to 2^{36} bits could be achieved. To utilize the pass-thought the client has in mind for authentication, the authors propose to use *P300* potentials. A P300 potential is a positive potential in the signal which is evoked about 300ms after a surprising or exciting event for the client. For example if clients are shown the components of their pass-thought (e.g. a sequence of images or letters), P300 potentials is recorded as spikes in the signal. These spikes are stored in a set of P300 potentials that can be used to encrypt a key in a concept as shown in Figure 4. The signals, S, are processed into signal features, F, and the set of P300 potentials are represented as $\mathsf{F_r}$. In the enrollment process, cryptography is used to encrypt a key, K, using $\mathsf{F_r}$ to create $\mathsf{V_{F_r}}$. In the authentication process, $\mathsf{V_{F_r}}$ is decrypted to see if it provides the original key K. The authors do mention some problems though as the accuracy of signal recording and processing is still unknown and it is also somewhat slow. The current state of BCI technology using P300-based approaches run with a bit rate of 4.8 characters/minute, which is a problem that has to be solved for the system to be accepted by the general public.

It is of course possible to forget the very image or password one is supposed to think about, but we are likely to remember it when we see it again. This kind of authentication was proposed in [25] where clients do not have to remember the password or image until they are presented with it. It did not include measurements of brain waves, but P300 potentials are generated when we recognize something.

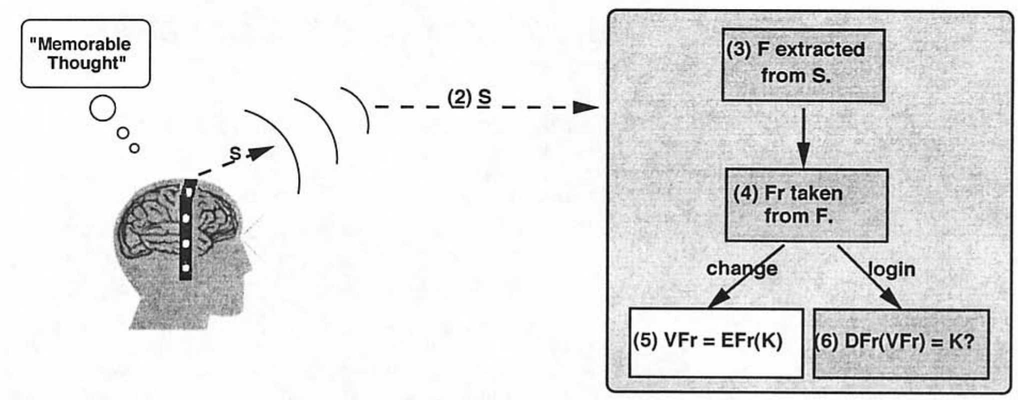


Figure 4: The pass-thought concept (Figure from [11])

2.2 EEG as identification and authentication

Research on person identification based on EEG were performed with promising results in [26, 27, 28, 29]. The authors used the Fast Fourier Transform (FFT) to calculate spectral power [30] and AutoRegressive (AR) parameters to extract features that were fed into a neural network in order to classify subjects. The authors used different methods to analyze the data, but research was based on the same recordings. 45 EEG recordings from 4 subjects (A,B,C,D) were used as training- and testing data in order to identify A,B,C and D amongst 75 different subjects (X) with one EEG recording each. The subjects were resting with eyes closed. The training data served as a template to match against the testing data and classify the subjects as either A,B,C,D or X. Correct classifications were achieved in over 80% of the cases showing that individuals can be identified based on brain waves.

Person authentication based on EEG was researched in 2007 with recordings from 9 subjects that were sitting with arms relaxed on their legs in 12 nonfeedback sessions over 3 days [31]. They performed three different mental tasks:

1 Imagination of repetitive self-paced left hand movements

2 Imagination of repetitive self-paced right hand movements

3 Generation of words beginning with the same random letter.

32 electrodes placed on the scalp were used to record EEG signals, with a sample frequency of 512Hz. The results showed that:

- there are some mental tasks that are more appropriate for person authentication than others
- the performance degrades over days
- using training data over two days increases the performance
- there is a potential for incremental learning

The subject pool was rather small and the authors plan on doing a larger experiment with more subjects and different mental tasks better suited for authentication.

Considering the high amount of data retrieved from EEG signals and the variations over time, an algorithm called *Dynamic Time Warping* (DTW) [32] could be applied. DTW is a technique that finds the optimal alignment between two time series that may vary in time. The first time series may be ”warped” non-linearly by stretching or shrinking it along its time axis to see if it is similar to the second time series. It is a rather slow algorithm and works best on small data sets, but research on a faster DTW algorithm is done in [33].

As mentioned in [10], a variety of tools exist to analyze EEG and ERP data. Principal component analysis (PCA) with The Dien PCA Toolbox [34], independent component analysis (ICA) and joint time-frequency analysis (TFA) with the Matlab toolbox EEGLAB [35], data cleaning, statistical extraction and visualization techniques with *Net Station* by *Electrical Geodesics, Inc* [36] that also offers a lot of information about EEG and EEG research products. Particularly interesting is their analysis environment, which has solutions for EEG/ERP analysis, source analysis, signal processing, and statistical analysis. Consistent, repeatable, and unique data between individuals are vital to authentication.

Without it, we cannot accurately verify a certain identity as true. As we could see in Figure 4, the authentication process would fail if *K* did not match the original *K*. Many BCI systems requires the subjects involved to undergo extensive training before they can generate fixed EEG patterns that can be accurately captured and given meaning in the form of external motion or mental state [37]. However, if brain waves should be useful in authentication, the clients cannot be expected or required to undergo such training. The system has to be user friendly.

2.3 Are we reading minds?

It is a difficult task to translate (interpret) patterns into their respective commands. In some cases, the BCI system has to be tweaked to fit individual clients. The idea proposed in [11] suggested that no such translation is needed in the pass-thought system. The authors said it would in fact reduce the entropy of the person's brain signals if such translation would be done. Instead, the signals should undergo feature extraction to filter out the non-repeatable parts. It actually makes sense to think like this, because authentication is primarily concerned with matching two sets of data rather than identifying the underlying meaning of the data. Let us say we extract signals from a client, which thinks about a certain color in the enrollment process. Then we extract the signals in the authentication process where the client thinks about the same color. The interesting part is whether the signals in both processes are similar enough to conclude they originate from the same client, not what the specific color is.

3 Experiment

This chapter explains how we performed the experiment and what we did up to the point of analysis. The purpose of this experiment is to see if it is possible to distinguish between clients and tasks based on EEG signals alone. We try two different approaches; A *Dynamic Time Warping* (DTW) and feature extraction. Initially we wanted to experiment with Event Related Potentials (ERP) and the P300 potential as well, but the SDK we use is too limited and does not include the necessary functions in order to control the events. We decided to conduct a small scale experiment involving just a few participants at first in order to learn how to proceed before conducting a larger scale experiment.

3.1 Equipment

We used the *ThinkGear* head set by *Neurosky* (Figure 5) with the capability of recording raw EEG signals from one sensor with sample frequency $Fs = 128Hz$. This is the absolute minimum requirement in order to record EEG signals. It runs with a 5V battery and records 8 bits of data through a serial port on the used computer. The recorded samples are in the μV range, Figure 6. The way it works were explained to us by a Neurosky representative:

> *The relationship between the sensor and the earclips are that the earclips work as ground/references. Basically, you are looking at one source (the sensor) that catches both brain waves, ambient noise and muscle movement. The other source (the ground and reference ear clips) look for a signal that still has proximity brainwaves and muscle movements, but is devoid of the direct frontal lobe brainwave readings. By intelligently filtering the two, the brainwave signal can be extracted. We know that it is an EEG signal through our comparison testing with medical devices and third party testing.*

3.2 Electrode placement

The 10-20 system is used to describe the placement of electrodes on a human scalp [38]. The scalp is divided into a grid that covers the top of the head relative to physical landmarks such as the nasion and inion (Figure 7). In our experiment the electrode is placed in scalp location Fp1 (Frontal Pole). It would be interesting to try other locations as well, but the design of the headset limits the location to Fp1 to ensure a good signal with effective recordings.

3.3 Tasks

The brain is a complex organ and tasks like vision, motor movement and emotions are processed at different locations [39, 40]. This means that some scalp locations and sensor placements are better suited to record certain thoughts than others. Higher order cognitive tasks like everyday planning, decision making, emotions, social- and moral reasoning is believed to be located at the frontal pole as explained in [41]. The EEG signals captured at the Fp1 location are other words related to subconscious thinking and personality. The authors experimented with tasks that required the subjects to talk about past and future plans, explaining the meaning of three words, and watching a fixed point for 60 seconds.

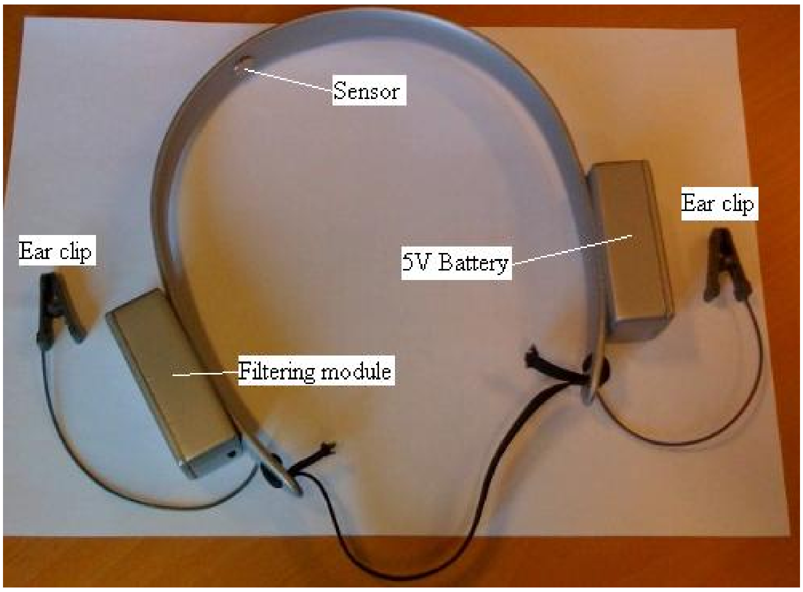


Figure 5: Picture of the ThinkGear head set by Neurosky

These tasks do not suit our experiment as eyes open and talking generate anomalies in the recorded EEG signals (Figure 8). If we had more sensors we could use Independent Component Analysis (ICA) to remove artifacts like physical movement [42], but for now we must try to avoid them manually. Instead we based the task selection on previous work where sensors were attached to locations O1,O2,P3, and P4, because these tasks are feasible choices for ”pass-thoughts” as opposed to those located at FP1. This means that when we analyze the tasks in this experiment, the similarities are not based on the tasks themselves, but rather the way they are performed due to the subconscious nature of tasks at FP1. So from an authentication perspective all tasks may be suitable for the Fp1 location.

The 8 tasks $(task_1, ..., task_8)$ we use are:

$task_1$ = Relax - client is told to sit comfortably and relax in a normal fashion

$task_2$ = Color - client is told to think about the red color

$task_3$ = Rotate - client is told to mentally rotate a house

$task_4$ = Password - client is told to think about the password ”BrainWaveS”

$task_5$ = Music - client is told to think about a song they know

$task_6$ = Words - client is told to generate random words in their native language starting with the letter ”M”.

$task_7$ = Count - client is told to count upwards in their native language, fast and starting with 1.

$task_8$ = Read - client is told to read from a provided random text.

They tasks are easy to perform, but difficult enough to ensure that the client has to concentrate in order to perform them. *Relax*, *rotate*, *words* and *count* was selected based

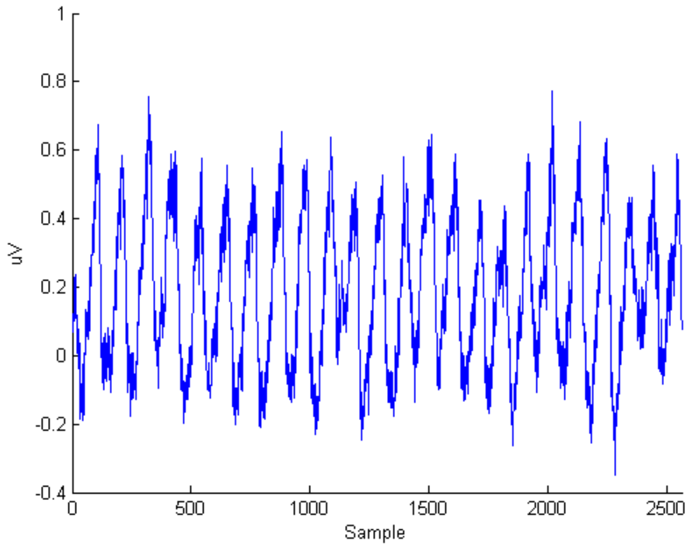


Figure 6: 20 seconds of an unfiltered EEG signal with sample frequency $Fs = 128Hz$ equals 2560 samples

on those used in [21] (although *count* were performed with eyes open in [21]). *Password* was selected based on the idea in [11] while *color* and *music* were selected out of our curiosity. *Read* was included to see that there actually is a difference between eyes open and eyes closed.

3.4 Location

We used the master lab at the university college in Gjøvik as the location for the experiment. The equipment is very sensitive to facial movement and we could often see that subjects reacted to abrupt sounds by involuntary eye movement that showed as peaks and drops in the signals. Therefore it was important to keep the location as quiet as pos-

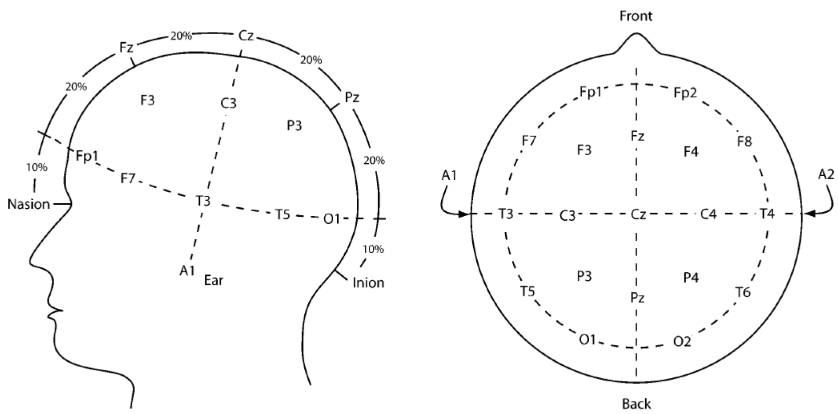


Figure 7: The international 10-20 Electrode Placement System (Figure from [8])

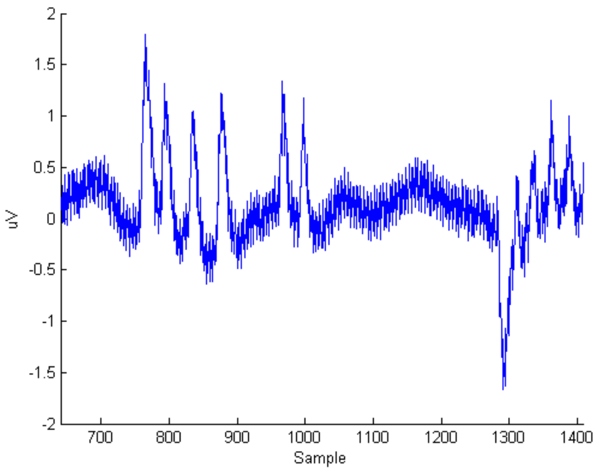


Figure 8: Example of anomalies in the EEG signal related to physical movement. Samples 750 - 1050 show six peaks related to six eye twinkles from the client. Samples 1250-1400 show both negative and positive peaks caused by the client head shaking.

sible, so we recorded only early in the morning and late evenings. Only two people were at the location; one researcher and one client.

3.5 Clients

Twelve clients $(client_1, ..., client_{12})$ were selected amongst friends, fellow students and supervisors who had time and interest to participate. All clients had to sign a participant agreement form (Appendix A). Age was not important as it does not change the EEG signal on a short-term basis (6 months in this case) [4]. Since we are dealing with signal matching rather than classification, we included both right and left handed participants even though the left side of the brain is dominant for right handed persons and vice versa [40]. The age ranged from 20-45 years including both right and left handed participants (1 female, 11 males).

3.6 Session

The client was told to sit in a relaxed position with eyes closed and arms resting in his/her lap. The headset was attached with aid from the researcher. The clients were presented with the 8 different tasks, one at a time. The tasks were not presented all at once to prevent the clients from drifting of and start doing the wrong tasks. Each task was recorded 3 times lasting 20 seconds with short breaks between each recording. All clients participated for three sessions $(session_1, ..., session_3)$ that lasted about 40 minutes total.

3.7 Samples

We used the NeuroView application (Figure 9) included with the MindKit SDK to record signals consisting of N samples. The application records raw data and performs filtering, spectrum analysis and calculates meditation, anxiety and attention values. In this experiment we only use the raw data $\mathbf{X} = (x_1, ..., x_N)$ because it contains all the information, but we stored the other data as well in case they could be valuable in further work. The raw data is stored in a text file containing two columns separated with a semicolon. The first column is the timestamps in seconds while the second column is the recorded EEG sample values in μV (Figure 10).

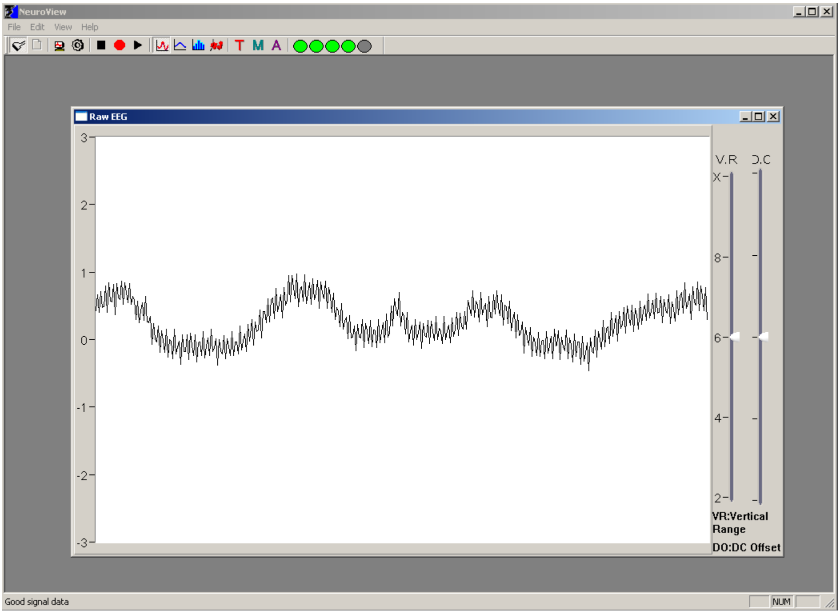


Figure 9: Screen capture of the NeuroView application

We did not store the timestamps because they are easy to calculate when we know the sample frequency. e.g. sample 705 was recorded at time $\frac{1s}{128Hz} \times 705 = 5.5078s$.

We named the files on the form <clientId>_<task>_<session>_<recordingNr>.<type> e.g. 3_Relax_1_2.raw. The NeuroView application did not include a parameter to set recording time so we had to manually time each recording to 20 seconds. In signal processing this was reduced to be exactly 20 seconds and 2561 samples (128Hz times 20 seconds). Each task was recorded 3 times $rec_1, ..., rec_3$ for 20 seconds in each session resulting in 72 recordings (3 recordings x 8 tasks x 3 sessions) for each client. After all session were complete we had 864 signals $(signal_1, ..., signal_{720})$ (72 recordings x 12 clients).

```
time;rawdata;
0;0.223945;
0.0078125;0.267891;
0.015625;0.248359;
0.0234375;0.287422;
0.03125;0.21418;
0.0390625;0.223945;
0.046875;0.228828;
0.0546875;0.184883;
0.0625;0.219063;
0.0703125;0.248359;
0.078125;0.306953;
0.0859375;0.282539;
0.09375;0.21418;
0.101563;0.263008;
0.109375;0.238594;
0.117188;0.287422;
```

Figure 10: Screen capture of the lines in a text file containing timestamps and EEG sample values

4 Signal processing

When clients are told to perform a task and the recording starts, it takes a few seconds before the client is focused. For this reason we removed the first 4 seconds of each recording leaving the remaining 16 seconds and 2048 samples intact. The equipment produced a lot of noise the first minute of recording that affects the *Relax* task the most. After one minute the signal stabilized and gradually improved during the remaining session (Figure 11). This problem becomes very evident when comparing signals in the time domain with DTW (Section 6.4), which is why we perform feature extraction (Section 5) as well.

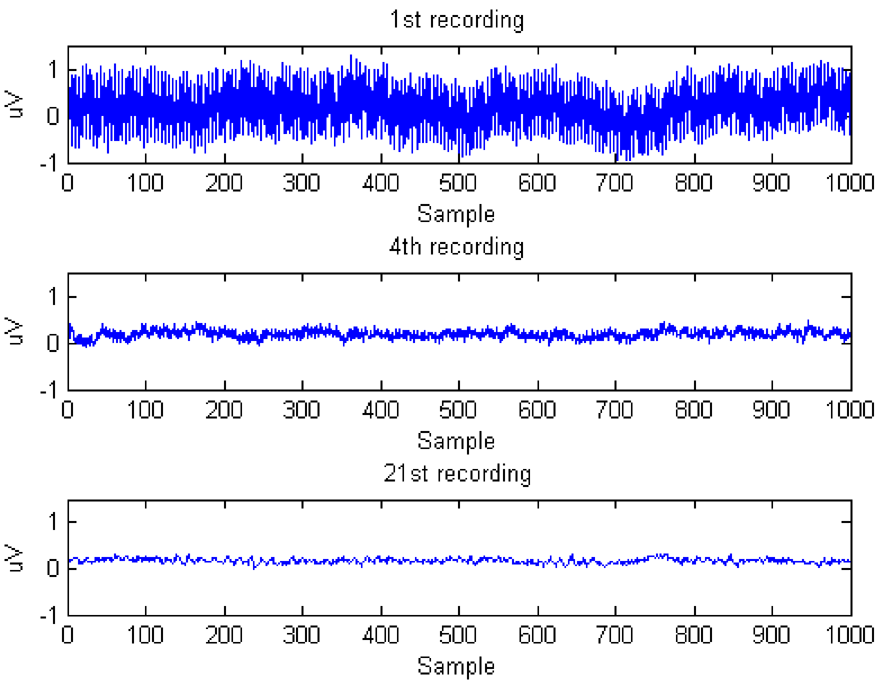


Figure 11: Equipment initialization period. Example of how signals look during the 1st, 4th and 21st recording

4.0.1 Frequency bands and the Discrete Fourier Transform

An EEG signal have a lot of information in the frequency domain as well as the time domain. The brain operates at low frequencies that range from 1-50Hz, which is usually divided into six frequency bands as explained in [8, 43] :

Delta: 1Hz - 4Hz

Theta: 4Hz - 8Hz

Alpha: 8Hz - 12Hz

Beta-Low: 12Hz - 20Hz

Beta-High: 20Hz - 30Hz

Gamma: 30Hz - 50Hz

The Discrete Fourier Transform (DFT) can be used to transform a signal with N samples from the time domain to the frequency domain (Figure 12). DFT is defined as

$$H_k = \sum_{n=0}^{N-1} X_n \ell^{-\frac{2\pi i}{N} kn} where k = 0, ..., N-1 \tag{4.1}$$

H_0 is the DC power (0Hz content) of the signal. **H** is symmetric around N/2 so $H_1 = H_N$ and $H_2 = H_{N-1}$ and so on. According to Kotelnikov's theorem the sample frequency have to be at least twice the value of the highest frequency we are interested in (50Hz in our case). Our sample frequency $Fs = 128Hz$ so we can get frequency information up $\frac{Fs}{2} = 64Hz$. The fast fourier transform (FFT) in Matlab performs this computation and provides the complex numbers of the DFT transform by running H = fft(X). The result is an array H with size N that contains complex numbers $\mathbf{H} = (h_1, ..., h_N)$. The indexing in Matlab starts at 1 so H(1) is the DC power and the array is symmetric around N/2+1. H(2:(N/2)+1) is the frequency content up to 64Hz. The absolute value of each complex number represents the signal power in dB at that specific frequency.

The frequency resolution df describes the frequency range of H_k and is defined as

$$df = \frac{Fs}{N} \tag{4.2}$$

In our case we have $df = \frac{128Hz}{2048} = 0.0625Hz$, which mean that the number of frequencies between 0Hz and 1Hz is $1Hz/df = 16$. Since Matlab starts indexing at 1 the frequency information of 1Hz start at H(1Hz / df + 1) = H(17). So if we want the values of the delta band (1Hz - 4Hz) we use H((1/df+1):(4/df+1)) = H(17:65), the theta band (4Hz - 8Hz) is H((4/df+1):(8/df+1)) = H(65:129) and so on.

4.0.2 Filtering

Most of our computations are based on the unfiltered samples X, but we store some filtered versions as well in order to test the DTW algorithm. We decided to do our own filtering instead of using the filtered signals by the NeuroView application to have complete control over the computations. Our filtering is based on the Inverse Fast Fourier transform (IFFT) that can be applied on H. Frequency information up to 50Hz is found in H(2:(50/df)+1) = H(2:801) and because of the symmetry around N/2+1 the same information is found at H((2048-(50/df)+1):2048) = H(1249:2048). In order to filter H such that frequency information above 50Hz is removed we set all the values between H(801) and H(1248) to 0 and apply IFFT to the result.

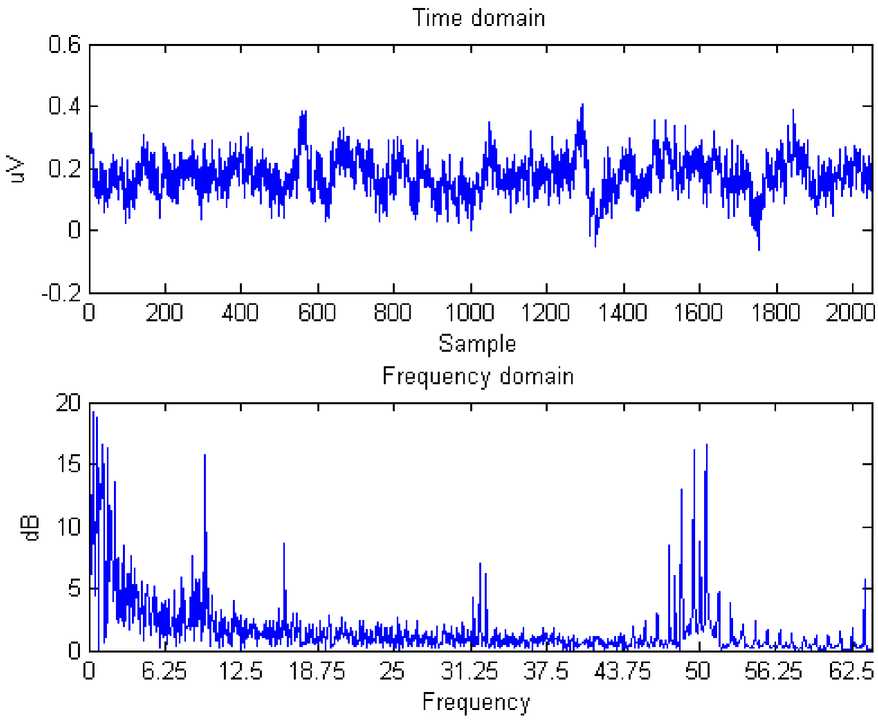


Figure 12: Representation of an EEG signal in the time domain and the frequency domain. Taken from the 4th recording in session 3 of client 4

We do the filtering in Matlab like this:

```
H = fft(X);
H(802:1248) = 0;
fX = ifft(H);
```

fX now contains the 1Hz-50Hz filtered version of **X** (Figure 13).The same principle can be applied if we want to filter a signal to contain information from each band only (Figure 14). For details on how we implemented the filtering in Matlab, the code is listed in Appendix B function *fourierFilter()*.

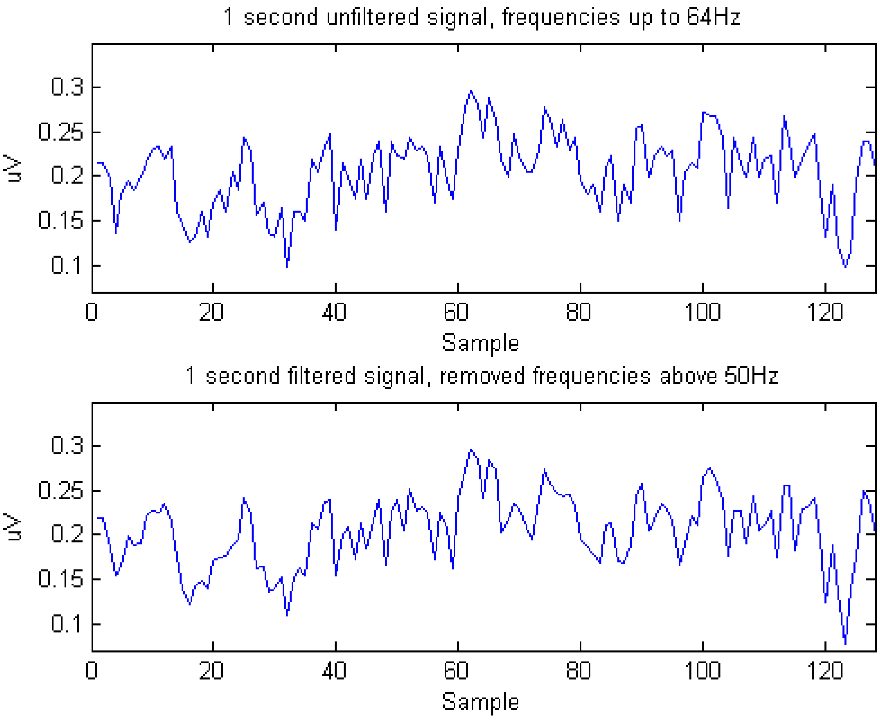


Figure 13: Signal filtering. 1 second of an unfiltered signal and the filtered version of the same signal

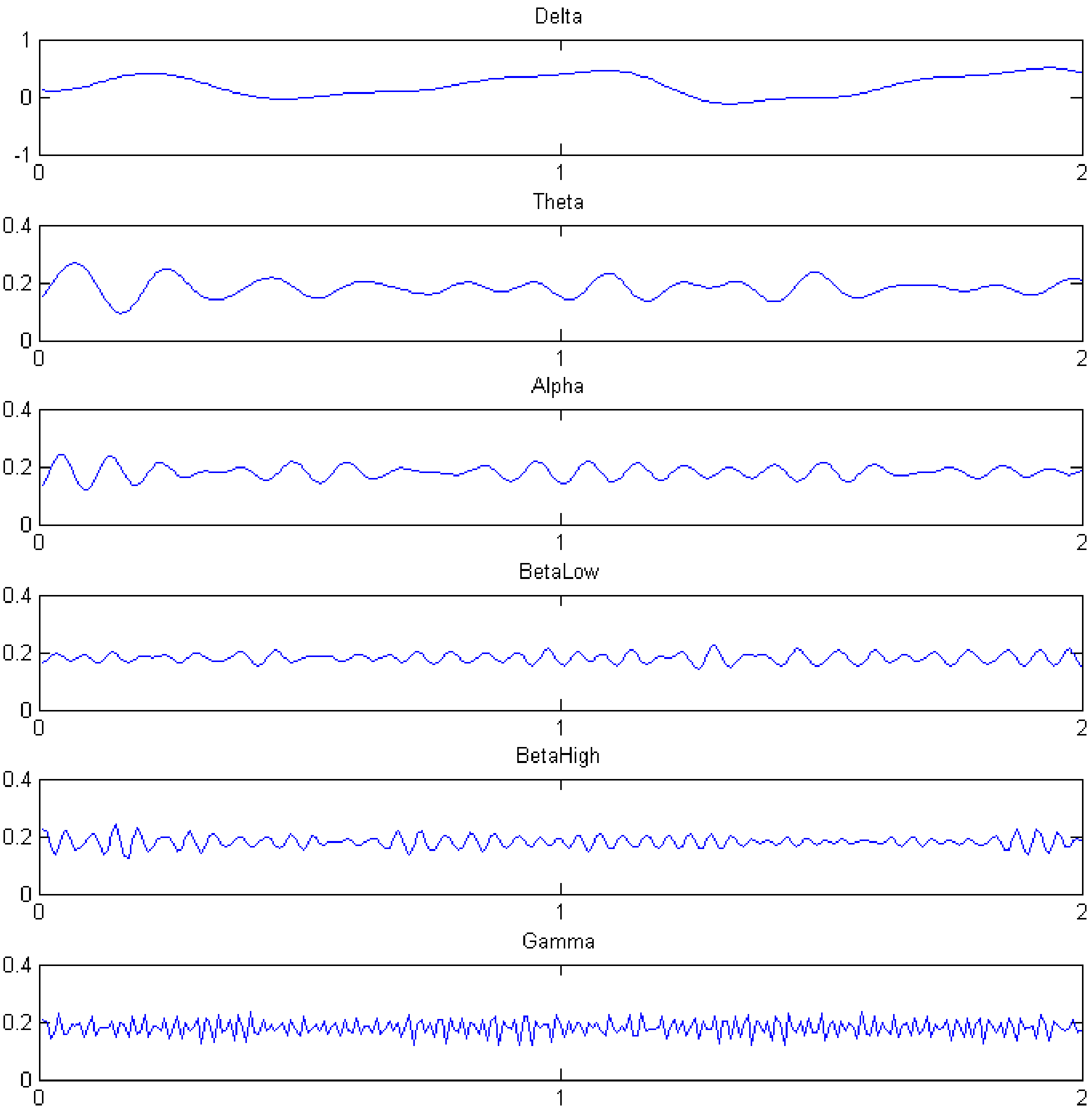


Figure 14: 2 seconds of a band filtered signal

5 Feature extraction

It is very difficult visually analyze what is going on in an EEG signal except for facial movement that may show as peaks. The information about thoughts is hidden within the signal and has to be extracted, so in addition to information in the time domain a lot of information can be found in the frequency domain. There are no limitations in feature extraction and the possibilities are endless.

We base our feature extraction on that described in [8] because the authors are able to do task classification with only two sensors, but we exclude the features that requires two sensors because we are limited to those that work with just one. We have $N = 2048$ number of samples and sample values $\mathbf{X} = (x_1, ..., x_N)$.

Mean sample value (MSV)

The mean of all sample values

$$MSV = \frac{1}{N} \times \sum_{n=1}^{N} x_n \tag{5.1}$$

Matlab:

```
MSV = mean(X).
```

Zero crossing rate (ZCR)

The rate at which values cross zero. If the product of two adjacent values is negative, they have opposite signs and a zero crossing has occurred. The rate is found by dividing the number of zero crossings by the number of samples

$$ZCR = \frac{1}{N} \times \sum_{n=1}^{N-1} 1\ \mathrm{if}(x_n \times x_{n+1} < 0),\ else\ 0 \tag{5.2}$$

Matlab:

```
for n=1:N-1
   if (X(n)*X(n+1) < 0)
      ZCR = ZCR + 1;
   end
end

ZCR = ZCR/N;
```

Values above zero (VAZ)

The number of values above zero

$$VAZ = \sum_{n=1}^{N-1} 1 \ if(x_n > 0), \ else \ 0 \tag{5.3}$$

Matlab:

```
for n=1:N-1
    if (X(n) > 0)
      VAZ = VAZ + 1;
    end
end
```

Mean phase angle (MPA)

The mean phase angle in the 1Hz - 50Hz frequency range. Complex numbers can be written as $H_n = a + bi$ or $H_n = A_n \angle \theta_n$ where A is the length and θ is the phase angle. The mean phase angle is then:

$$MPA = \frac{1}{N} \sum_{n=1}^{N} \theta_n. \tag{5.4}$$

Matlab:

$MPA = mean(angle(H(17:801)))$.

MPA is now in radians so we multiply $MPA \times \frac{180}{\pi}$ to get it in degrees.

Signal power in the six frequency bands (Pdelta,...,Pgamma)

Signal power P in a sampled signal is found by dividing the signal energy E by the number of samples N

$$E = \sum_{n=1}^{N} |x_n|^2 \tag{5.5}$$

$$P = \frac{1}{N} \times E \tag{5.6}$$

Signal power in the frequency domain is computed the same way. If we want to find the signal power in a specific frequency band, we simply provide the range as parameters in the function. As explained in Section 4.0.1 the delta band range is H(17:65). To compute the signal power in that range we can use

$$P_{delta} = \frac{1}{65 - 17} \sum_{n=17}^{65} |h_n|^2 \tag{5.7}$$

Matlab:

$P_{delta} = mean(abs(H(17:65)).^2)$.

This is done for all the frequency bands; delta, theta, alpha, betaLow, betaHigh, and gamma.

Peak frequency magnitude (PFM,PFMdelta,...,PFMgamma)

We compute the power of all frequencies in each frequency band, as well as between 0Hz - 50Hz in **H** and store the highest values.

Matlab:

```
PFM = max(abs(H(2:800)))
```

This is also done for all the frequency bands; delta, theta, alpha, betaLow, betaHigh, and gamma.

Peak frequency (PF,PFdelta,...,PFgamma)

The frequency where the highest magnitude in each band as well as between 0Hz - 50Hz is located.

Matlab:

```
PFM, PF = max(abs(H(2:800)));
PF = PF * df + 1;
```

This is done for all the frequency bands; delta, theta, alpha, betaLow, betaHigh, and gamma.

Mean spectral power (MSP)

The mean power of the six frequency bands.

Matlab:

$MSP = mean(P_{delta} + P_{theta} + P_{alpha} + P_{beta-low} + P_{beta-high} + P_{gamma})$

5.1 Feature summary

Now we have 25 features $(feature_1, ..., feature_{25})$ each signal:

$feature_1$ = MSV - Mean sample value

$feature_2$ = ZCR - Zero Crossing rate

$feature_3$ = VAZ - Values above zero

$feature_4$ = PF - Peak Frequency

$feature_5$ = PFM - Peak Frequency Magnitude

$feature_6$ = MSP - Mean spectral power

$feature_7$ = MPA - Mean phase angle

$feature_8$ = P_{delta} - Delta band power

$feature_9$ = PF_{delta} - Delta band peak frequency

$feature_{10}$ = PFM_{delta} - Delta band peak frequency magnitude

$feature_{11}$ = P_{theta} - Theta band power

$feature_{12}$ = PF_{theta} - Theta band peak frequency

$feature_{13}$ = PFM_{theta} - Theta band peak frequency magnitude

$feature_{14}$ = P_{alpha} - Alpha band power

$feature_{15}$ = PF_{alpha} - Alpha band peak frequency

$feature_{16}$ = PFM_{alpha} - Alpha band peak frequency magnitude

$feature_{17}$ = $P_{betaLow}$ - BetaLow band power

$feature_{18}$ = $PF_{betaLow}$ - BetaLow band peak frequency

$feature_{19}$ = $PFM_{betaLow}$ - BetaLow band peak frequency magnitude

$feature_{20}$ = $P_{betaHigh}$ - BetaHigh band power

$feature_{21}$ = $PF_{betaHigh}$ - BetaHigh band peak frequency

$feature_{22}$ = $PFM_{betaHigh}$ - BetaHigh band peak frequency magnitude

$feature_{23}$ = P_{gamma} - Gamma band power

$feature_{24}$ = PF_{gamma} - Gamma band peak frequency

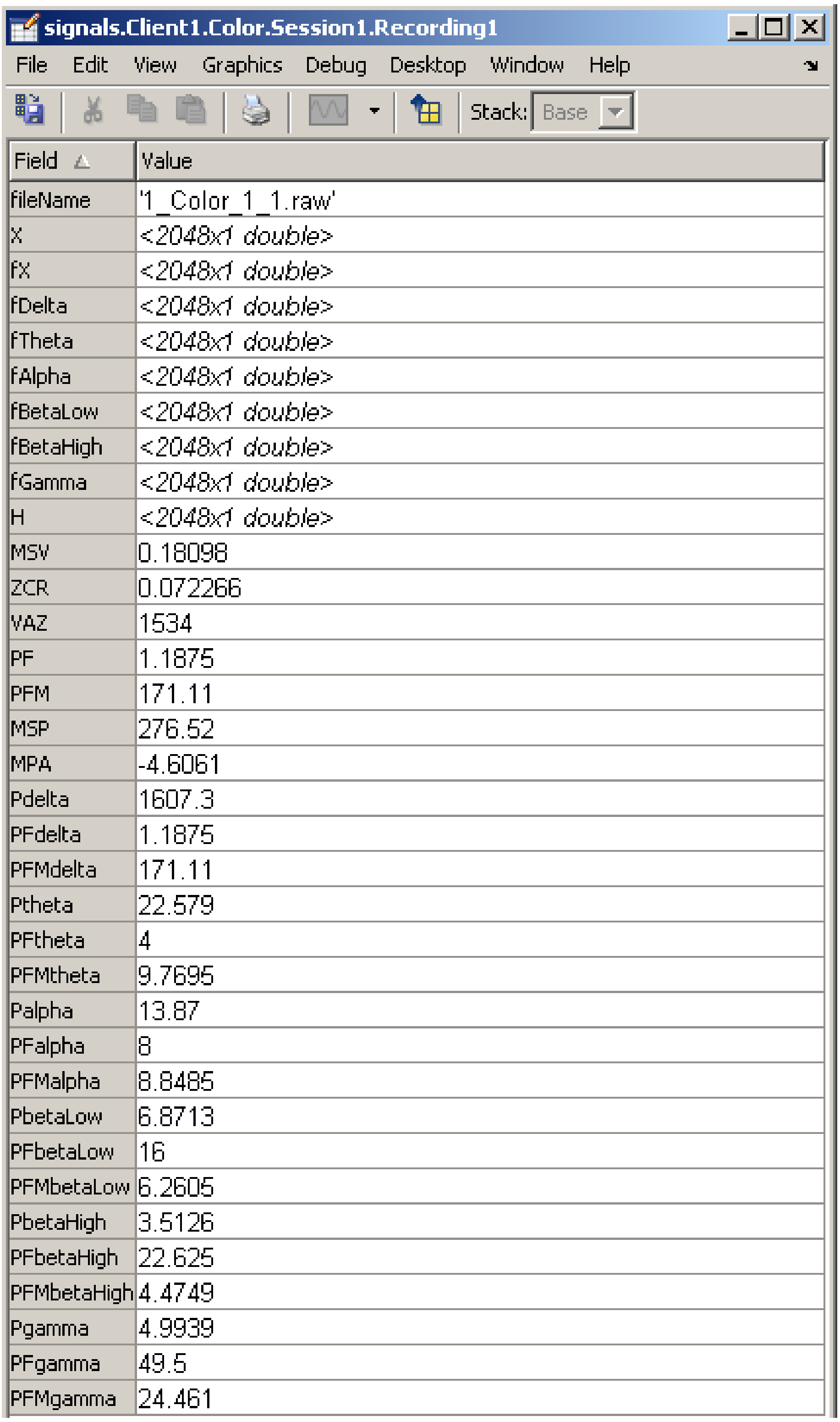
$feature_{25}$ = PFM_{gamma} - Gamma band peak frequency magnitude

5.2 Signal representation

For the rest of this thesis we use i,j,k,m, n, and o to refer to a *signal* consisting of *client*, *task*, *session*, *recording*, and *features* respectively:

$$signal_i = (client_j, task_k, session_m, rec_n, feature_o)$$

We store signals in Matlab by *struct* (code is listed in Appendix B). We have a *struct* called $signals$ with fields $Client_j.Task_k.Session_m.Recording_n$ to represent one recording. The struct containes the $fileName$, unfiltered samples **X**, filtered samples **fX** to **fGamma**, and each $feature_o$ (Figure 15). To get the **X** values we can write $X = signals.Client1.Session1.Recording1.X$. To get a feature we can write $Pdelta = signals.Client1.Session1Recording1.feature.Pdelta$. If we refer to a specific feature we use the name e.g. $signal.DC$ while any feature is referred to as $s_i.feature_o$. The filtered samples are referred to by name e.g. $s_i.fX$ or $s_i.fGamma$.



Field	Value
fileName	'1_Color_1_1.raw'
X	<2048x1 double>
fX	<2048x1 double>
fDelta	<2048x1 double>
fTheta	<2048x1 double>
fAlpha	<2048x1 double>
fBetaLow	<2048x1 double>
fBetaHigh	<2048x1 double>
fGamma	<2048x1 double>
H	<2048x1 double>
MSV	0.18098
ZCR	0.072266
VAZ	1534
PF	1.1875
PFM	171.11
MSP	276.52
MPA	-4.6061
Pdelta	1607.3
PFdelta	1.1875
PFMdelta	171.11
Ptheta	22.579
PFtheta	4
PFMtheta	9.7695
Palpha	13.87
PFalpha	8
PFMalpha	8.8485
PbetaLow	6.8713
PFbetaLow	16
PFMbetaLow	6.2605
PbetaHigh	3.5126
PFbetaHigh	22.625
PFMbetaHigh	4.4749
Pgamma	4.9939
PFgamma	49.5
PFMgamma	24.461

Figure 15: Matlab signal representation. The header show how the previous fields in the *struct*

6 Analysis

At this point we have many of signals and features that we have to compare against each other. The first step is to check the statistics of each feature by calculating their minimum, maximum, mean (Equation 6.1), and standard deviation (Equation 6.2) values.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{6.1}$$

$$\sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{6.2}$$

We also calculate the percentage difference between the standard deviation and mean:

$$\%\ deviation = \frac{\sigma_x}{\bar{x}} * 100 \tag{6.3}$$

E.g. Mean Sample Value (MSV) has $\sigma_x = 0.025$ and $\bar{x} = 0.179$ such that $\%\ deviation = \frac{0.025}{0.179} * 100 = 14\%$ The average % deviation per client is calculated the same way, but calculated based on each client separately. A good feature will have values that are close together for the same client, but far apart for different clients. From Table 1 we can see that the average % deviation per client of Peak Frequency in the gamma band (PFgamma) is only 2%, which is good, but the % deviation is only 3% meaning that the values of PFgamma is very similar even for different clients. The most interesting feature appears to be PbetaHigh as the difference between % deviation and average % deviation per client is high ($654\% - 112\% = 542\%$). The worst feature is Mean Phase Angle (MPA) because it deviates more for each client on average than it does between clients. It is also worth noting that Pdelta have the highest mean power (497.70) followed by Pgamma (97.37) meaning that the EEG signal activity is strongest in the delta- and gamma band.

6.1 Chi-square goodness-of-fit test

We investigate whether our samples, filtered samples and features follow a normal distribution or not. We apply a chi-square goodness-of-fit test as explained in [44] (Matlab code in Appendix C). The requirement is that we have a sequence of values **X** of size n where the probability distribution is unknown. The n observations are arranged in a frequency histogram with k bins or intervals. O_i is the observed frequency of the ith interval. A normal distribution have equal probabilities for each interval, so $p_i = \frac{1}{k}$. Expected frequency in each interval is then $E_i = n \times p_i$. With these values we can calculate the chi-square statistic

$$X_0^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{6.4}$$

The boundaries $a_0, a_1, ..., a_k$ of each interval $(a_0, a_1)_1, ..., [a_{k-1}, a_k)_k$ must be selected such that all probabilities p_i are equal according to

Table 1: Feature statistics for all signals

Feature	Min	Max	$\bar{x}$	σ_x	% deviation	Average % deviation per client
MSV	-0.012	0.423	0.179	0.025	14%	12%
ZCR	0	0.773	0.108	0.173	160%	130%
VAZ	968	2048	1794	255	14%	11%
PF	0.06	49.56	9.98	19.20	192%	184%
PFM	14	1710.6	127.12	146.68	115%	78%
MSP	6.21	2368	109.57	193.60	177%	127%
MPA	-47.13	72.65	0.48	13.98	2928	18920%
Pdelta	16.03	11804	497.70	882.84	177%	130%
Ptheta	4.83	1926.3	40.02	98.89	247%	151%
Palpha	2.08	385.33	13.79	20.05	145%	85%
PbetaLow	1.15	90.76	5.03	6.01	119%	70%
PbetaHigh	0.65	615.99	3.50	22.94	654%	112%
Pgamma	0.39	11456	97.37	595.71	612%	181%
PFdelta	1	3.44	1.25	0.29	23%	21%
PFtheta	4	8	4.91	0.98	20%	18%
PFalpha	8	11.94	9.40	1.05	11%	10%
PFbetaLow	12	19.81	15.76	1.15	7%	6%
PFbetaHigh	20	30	23.66	3.15	13%	12%
PFgamma	31.13	49.94	48.90	1.54	3%	2%
PFMdelta	9.45	344.19	61.35	52.86	86%	66%
PFMtheta	4.31	88.18	12.88	8.86	69%	54%
PFMalpha	2.88	50.28	8.24	3.88	47%	35%
PFMbetaLow	2.36	32.62	7.93	2.32	29%	25%
PFMbetaHigh	1.70	109.74	4.00	4.79	120%	40%
PFMgamma	1.77	1710.6	57.81	125.25	217%	85%

$p_i = P(a_{i-1} \le x \le a_i) = \int_{a_{i-1}}^{a_i} f(x)dx$, where $p_{i-1} = p_i$

In our case we select $k = 10$ intervals so $p_i = \frac{1}{10} = 0.1$. Using the *cumulative standard normal distribution* table from [44] we find k intervals with probability 0.1 for a normal distribution to be

$(-\infty, -1.29)_1, [-1.29, -0.85)_2, [-0.85, -0.53)_3, [-0.53, -0.26)_4, [-0.26, 0)_5,$
$[0, 0.26)_6, [0.26, 0.53)_7, [0.53, 0.85)_8, [0.85, 1.29)_9, [1.29, \infty)_{10}$

To find the intervals for **X** we need to calculate $\bar{x}$ and σ_x to get the boundaries of the k intervals

$(\bar{x} + a_0 \times \sigma_x, \bar{x} + a_1 \times \sigma_x)_1, ..., [\bar{x} + a_{i-1} \times \sigma_x, \bar{x} + a_i \times \sigma_x)_k$

Our hypothesis is

H_0: **X** has normal distribution

H_1: **X** does not have normal distribution

We have $k - p - 1 = 10 - 2 - 1 = 7$ degrees of freedom ν select confidence interval $\alpha = 0.05$. Using the *chi-squared distribution* table from [44] we find that $X^2_{\alpha,\nu} = X^2_{0.05,7}$ is 14.07 meaning that H_0 is rejected if $X^2_0 > 14.07$. If $X^2_0 <= 14.07$ we cannot reject H_0

and X might have a normal distribution. We apply the X_0^2 test on the filtered samples and each feature and count the number of non-rejects of H_0. From the Table 2 we can

Table 2: Chi-square goodness of fit test results

X	Probable normal distributions
$s_i.X$	71
$s_i.fX$	175
$s_i.fDelta$	29
$s_i.fTheta$	343
$s_i.fAlpha$	436
$s_i.fBetaLow$	597
$s_i.fBetaHigh$	555
$s_i.fGamma$	121
$s_1.feature_o, ..., s_{72}.feature_o$ per client	35
$s_1.feature_o, ..., s_{864}.feature_o$	0

see that the samples in fBetaLow have the highest count of probable normal distributions because the H_0 hypothesis could not be rejected in 597 out of 864 signals. It is interesting to see that normal distributions are most common in the theta, alpha, betaLow and betaHigh band as opposed to delta and gamma that have noticeably fewer probable normal distributions. This indicates that the signals are more stable in the mid range frequency bands, but this is not necessarily better in terms of authentication because what we want is similar values for the same client but different values for different clients. Finally we can see that when we test each feature for each client separately only 35 feature tests out of 300 (25 features times 12 clients) had a probable normal distribution. When features are tested across all signals there are 0 out of 25 tests (25 features) with a probable normal distribution. The ideal result would be that all features followed a normal distribution per client, but not across all signals, because then we would know something about what values to expect per client. But until this is tested on 30 clients or more we do not conclude with anything. The results are only considered as preliminary and used as a guide.

6.2 Correlation

We compute the correlation between features in order to see how they relate to each other. If two or more features have a high correlation, a combination of them can be utilized when creating a distance metric. To find the correlation between two sequences $\mathbf{X} = (x_1, ..., x_n)$ and $\mathbf{Y} = (y_1, ..., y_m)$ we need the mean (equation 6.1), standard deviation (equation 6.2), and covariance (equation 6.5)

$$\sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{6.5}$$

The correlation between **X** and **Y** is then defined as

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{6.6}$$

Correlation ρ_{xy} have values in the range $[-1, 1]$. The further away from 0 the stronger the correlation. Positive correlation implies that when X increase Y increase, and when X decrese Y decrease. Negative correlation implies that when X increases Y decreases, and when X decreases Y increases. We set the correlation threshold to $\geq |0.6|$ such that only features with a high correlation are found. Table 3 show the average correlation between two features per client, and the number of clients with high correlation between the two features (we only those with 5 clients or more to reduce the size of the table). We can see

Table 3: Correlation between features per client

X	Y	Number of clients	Average ρ_{xy}
ZCR	Pgamma	10	0.81
ZCR	PFMgamma	10	0.82
VAZ	PFM	12	-0.79
VAZ	PFMdelta	6	-0.72
PFM	MSP	6	0.79
PFM	PFMdelta	7	0.73
MSP	Pdelta	11	0.99
MSP	PFMdelta	11	0.78
MSP	Ptheta	9	0.83
MSP	PFMtheta	9	0.77
MSP	PbetaLow	6	0.83
Pdelta	PFMdelta	12	0.81
Pdelta	Ptheta	9	0.79
Pdelta	PFMTheta	9	0.76
Ptheta	PFMtheta	12	0.89
Ptheta	Palpha	10	0.85
Ptheta	PbetaLow	7	0.84
PFMtheta	PbetaLow	7	0.70
Palpha	PFMalpha	11	0.85
PbetaLow	PbetaHigh	12	0.78
PbetaLow	PFMbetaHigh	6	0.72
PbetaHigh	PFMbetaHigh	12	0.82
Pgamma	PFMgamma	12	0.94
Ptheta	PFMalpha	6	0.79
PFMtheta	Palpha	8	0.79
PFMtheta	PFMalpha	6	0.75
Palpha	PbetaLow	7	0.81

that Zero Crossing Rate (ZCR) is closely related to the signal power in the gamma band (Pgamma), which implies that activity in the gamma band tend to produce an increase in negative values. ZCR is also connected to the peak frequency magnitude in the delta band (PFMgamma), but this is only a natural effect due PFMgamma's computation based on Pgamma. Values Above Zero (VAZ) is decreasing as the peak frequency magnitude (PFM) is increasing, which indicates that peaks are higher with more samples below zero. The high correlation between the Mean Sample Power(MSP) and the power in the delta band (Pdelta) is telling us that most clients have highest activity in the delta band, followed by the activity in the theta band. The most interesting result is the high correlation between the power in the betaLow band (PbetaLow) and the betaHigh band (PbetaHigh) for all 12 clients. Maybe these two features are tied to the activity of specific clients.

We have not mentioned correlations like Pdelta and PFMdelta, or Ptheta and PFMtheta because these correlations are also natural because the way they are computed.

6.3 Distance metric

The main problem in this thesis is to compare two EEG signals and determine how similar they are. A distance metric is a common method in authentication to compute the distance between two sequences $\mathbf{X} = (x_1, ..., x_p)$ and $\mathbf{Y} = (y_1, ..., y_q)$ and must have the following properties:

$d(\mathbf{X}, \mathbf{Y}) \geq 0)$, distance must be positive

$d(\mathbf{X}, \mathbf{Y}) = 0 \text{ if } \mathbf{X} = \mathbf{Y}$, equal sequences give the best result

$d(\mathbf{X}, \mathbf{Y}) = d(\mathbf{Y}, \mathbf{X})$, order independent symmetry

$d(\mathbf{X}, \mathbf{Z}) \leq d(\mathbf{X}, \mathbf{Y}) + d(\mathbf{Y}, \mathbf{Z})$, triangle inequality

There are no limitations other than the properties above when designing a distance metric. The possibilities are endless and there is no ”best way” to create a distance metric that provide the best results. Usually a template is created for each client which contain features values that best describe that client. All signals recorded from the same client should then contain feature values in the same range as those in the template within a certain threshold th. Ideally $d(s_1, s_2) < th$ indicate that the two signals originate from the same client while $d(s_1, s_2) \geq th$ indicates different clients. A good template is dependent on the distance metric and vice versa.

In our experiment, we have to consider that signals are recorded from different clients in addition to different tasks. This means that the distance $d(s_1, s_2)$ fall into one of four categories:

- **A**: Signals are from the same client doing the same task;
  $d((client_j, task_k, session_m, rec_n), (client_j, task_1, session_m, rec_{n+1}))$.
  Distances in category A are referred to as $\mathbf{A}_a$ where $a = 1, ..., 8$ (Table 4).
- **B**: Signals are from the same client doing two different tasks;
  $d((client_j, task_k, session_m, rec_n), (client_j, task_{k+1}, session_m, rec_n))$.
  Distances in category B are referred to as $\mathbf{B}_b$ where $b = 1, ..., 28$ (Table 4).
- **C**: Signals are from different clients doing the same task;
  $d((client_j, task_k, session_m, rec_n), (client_{j+1}, task_k, session_m, rec_n))$.
  Distances in category C are referred to as $\mathbf{C}_c$ where $c = 1, ..., 8$ (Table 5).
- **D**: Signals are from different clients doing different tasks;
  $d((client_j, task_k, session_m, rec_n), (client_{j+1}, task_{k+1}, session_m, rec_n))$.
  Distances in category D are referred to as $\mathbf{D}_d$ where $d = 1, ..., 56$ (Table 5).

A set of distances that should be verified as originating from the same client is called *genuine attempts* while a set of distances that should be rejected because they originate from different clients is called fraudulent attempts. Hence the number of genuine attempts is always lower than fraudulent attempts.

Let us say that we use A_1 as genuine attempts and C_1 as fraudulent attempts for $client_1$ and $client_2$. All clients have 3 recordings of each task in each session so if we compute the distances per session A_1 consists of $\frac{(3 \times 3 - 3)}{2} = 3$ genuine attempts per client

Table 4: Category **A** and **B** distances for $\mathtt{client}_j$

		$\mathtt{client}_j$							
		Relax	Color	Rotate	Password	Music	Words	Count	Read
$\mathtt{client}_j$	Relax	$\mathbf{A}_1$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_3$	$\mathbf{B}_4$	$\mathbf{B}_5$	$\mathbf{B}_6$	$\mathbf{B}_7$
	Color	$\mathbf{B}_1$	$\mathbf{A}_2$	$\mathbf{B}_8$	$\mathbf{B}_9$	$\mathbf{B}_{10}$	$\mathbf{B}_{11}$	$\mathbf{B}_{12}$	$\mathbf{B}_{13}$
	Rotate	$\mathbf{B}_2$	$\mathbf{B}_8$	$\mathbf{A}_3$	$\mathbf{B}_{14}$	$\mathbf{B}_{15}$	$\mathbf{B}_{16}$	$\mathbf{B}_{17}$	$\mathbf{B}_{18}$
	Password	$\mathbf{B}_3$	$\mathbf{B}_9$	$\mathbf{B}_{14}$	$\mathbf{A}_4$	$\mathbf{B}_{19}$	$\mathbf{B}_{20}$	$\mathbf{B}_{21}$	$\mathbf{B}_{22}$
	Music	$\mathbf{B}_4$	$\mathbf{B}_{10}$	$\mathbf{B}_{15}$	$\mathbf{B}_{19}$	$\mathbf{A}_5$	$\mathbf{B}_{23}$	$\mathbf{B}_{24}$	$\mathbf{B}_{25}$
	Words	$\mathbf{B}_5$	$\mathbf{B}_{11}$	$\mathbf{B}_{16}$	$\mathbf{B}_{20}$	$\mathbf{B}_{23}$	$\mathbf{A}_6$	$\mathbf{B}_{26}$	$\mathbf{B}_{27}$
	Count	$\mathbf{B}_6$	$\mathbf{B}_{12}$	$\mathbf{B}_{17}$	$\mathbf{B}_{21}$	$\mathbf{B}_{24}$	$\mathbf{B}_{26}$	$\mathbf{A}_7$	$\mathbf{B}_{28}$
	Read	$\mathbf{B}_7$	$\mathbf{B}_{13}$	$\mathbf{B}_{18}$	$\mathbf{B}_{22}$	$\mathbf{B}_{25}$	$\mathbf{B}_{27}$	$\mathbf{B}_{28}$	$\mathbf{A}_8$

Table 5: Category **C** and **D** distances for $\mathtt{client}_j$ and $\mathtt{client}_{j+1}$

		$\mathtt{client}_{j+1}$							
		Relax	Color	Rotate	Password	Music	Words	Count	Read
$\mathtt{client}_j$	Relax	$\mathbf{C}_1$	$\mathbf{D}_{29}$	$\mathbf{D}_{30}$	$\mathbf{D}_{32}$	$\mathbf{D}_{35}$	$\mathbf{D}_{39}$	$\mathbf{D}_{44}$	$\mathbf{D}_{50}$
	Color	$\mathbf{D}_1$	$\mathbf{C}_2$	$\mathbf{D}_{31}$	$\mathbf{D}_{33}$	$\mathbf{D}_{36}$	$\mathbf{D}_{40}$	$\mathbf{D}_{45}$	$\mathbf{D}_{51}$
	Rotate	$\mathbf{D}_2$	$\mathbf{D}_8$	$\mathbf{C}_3$	$\mathbf{D}_{34}$	$\mathbf{D}_{37}$	$\mathbf{D}_{41}$	$\mathbf{D}_{46}$	$\mathbf{D}_{52}$
	Password	$\mathbf{D}_3$	$\mathbf{D}_9$	$\mathbf{D}_{14}$	$\mathbf{C}_4$	$\mathbf{D}_{38}$	$\mathbf{D}_{42}$	$\mathbf{D}_{47}$	$\mathbf{D}_{53}$
	Music	$\mathbf{D}_4$	$\mathbf{D}_{10}$	$\mathbf{D}_{15}$	$\mathbf{D}_{19}$	$\mathbf{C}_5$	$\mathbf{D}_{43}$	$\mathbf{D}_{48}$	$\mathbf{D}_{54}$
	Words	$\mathbf{D}_5$	$\mathbf{D}_{11}$	$\mathbf{D}_{16}$	$\mathbf{D}_{20}$	$\mathbf{D}_{23}$	$\mathbf{C}_6$	$\mathbf{D}_{49}$	$\mathbf{D}_{55}$
	Count	$\mathbf{D}_6$	$\mathbf{D}_{12}$	$\mathbf{D}_{17}$	$\mathbf{D}_{21}$	$\mathbf{D}_{24}$	$\mathbf{D}_{26}$	$\mathbf{C}_7$	$\mathbf{D}_{56}$
	Read	$\mathbf{D}_7$	$\mathbf{D}_{13}$	$\mathbf{D}_{18}$	$\mathbf{D}_{22}$	$\mathbf{D}_{25}$	$\mathbf{D}_{27}$	$\mathbf{D}_{28}$	$\mathbf{C}_8$

while C_1 consists of $3 \times 3 = 9$ fraudulent attempts per client. If we compute the distances across all sessions A_1 consists of $\frac{(9\times9-9)}{2} = 36$ genuine attempts per client, while C_1 consists of $9 \times 9 = 81$ fraudulent attempts per client. The total number of genuine and fraudulent attempts computed for two clients is then 6 and 18 per session, and 72 and 162 across all sessions respectively.

A *false match* has occurred if a distance is accepted (below th) even though it should be rejected. A *false non-match* has occurred if a distance is rejected (above th) even though it should be accepted. As a measurement of how well a distance metric perform we use False Match Rate (FMR), False Non-Match Rate (FNMR) and Equal Error Rate (EER) for several th.

$$FMR = \frac{number\ of\ false\ matches}{number\ of\ fraudulent\ attempts} \tag{6.7}$$

$$FNMR = \frac{number\ of\ false\ non-matches}{number\ of\ genuine\ attempts} \tag{6.8}$$

$$EER = The\ rate\ where\ FMR = FNMR \tag{6.9}$$

The FMR and FNMR for different thresholds th can be plotted in a Detection Error Trade-Off Curve (DET-Curve) (e.g. Figure 16) that show how they relate to each other (Matlab code in Appendix E).

6.4 Dynamic time warping based distance metric

Dynamic time warping (DTW) is a technique that simulates the human ability to match patterns by stretching the two time series in time, called warping. It is a very useful technique that has been applied in several fields like gesture and speech recognition. The downside is that DTW has a quadratic time and space complexity making DTW algorithms slow as the size of the time series increase. This is especially noticeable when we have a lot of time series to compare. We tried the FastDTW algorithm as found in [33], which has linear complexity in time and space. Unfortunately the time series input had to be read from file, which made it cumbersome and slow to use with Matlab. Instead we used a DTW algorithm written in Java by a fellow student (Kjetil Holien) and his supervisor (Patric Bours). The input to the algorithm is the two time series we want to compare and the output is the distance. We apply DTW on X, fX, fDelta, fTheta, fAlpha, fBetaLow, fBetaHigh, and fGamma e.g:

$$d(s_1, s_2) = DTW(s_1.fAlpha, s_2.fAlpha) \tag{6.10}$$

6.4.1 Results

DTW would take a long time to run if we were to calculate the distance between all signals. Instead we calculate the distances in what we call a ”best-case-scenario”. This is based on selecting the task with the best performance in each session per client by taking the distance between the three recordings of each task and select the one with the best average, in other words the category A_a with the best average for each client. This task is used as genuine attempts, while other client’s *read* task category $D_{50}, ..., D_{56}$ (depending on A_a) is used as fraudulent attempts. This is a very unrealistic case that are based on few values, but by doing this we actually design a case where the EER should be low if DTW has any potential at all.

The results are shown in Table 6 and we can see that EER is generally very high, but the samples in fGamma stands out (22.2%, 22.2%, and 14.7% EER) so we decided to test DTW on fGamma across all sessions. We call this distance metric *DTW fGamma* and Table 7 show the category A_a task with the best average distance for each client, and we can see that the DTW fGamma fails to detect the same task in each session. For testing purposes we use the best session 1 A_a as genuine attempts across all sessions, and $D_{50}, ..., D_{56}$ from all sessions as fraudulent attempts. The result was an EER of 34.30 (Figure 16 which is to high for authentication purposes, and the EER is even calculated based on an unrealistic ”best-scenario-case”.

Table 6: EER in each session using DTW as a distance metric on the filtered samples

Samples	Session 1	Session 2	Session 3
X	42.8%	46.4%	54.5%
fX	38.9%	48.3%	50.0%
fDelta	44.4%	42.9%	62.9%
fTheta	49.9%	67.6%	62.2%
fAlpha	40.4%	47.2%	42.6%
fBetaLow	39.9%	43.1%	42.4%
fBetaHigh	29.7%	37.0%	38.0%
fGamma	22.2%	22.2%	14.7%

Table 7: Best performing A_a tasks with DTW fGamma as distance metric in each session

Client	Session 1	Session 2	Session 3
1	Rotate	Music	Count
2	Rotate	Password	Color
3	Words	Music	Relax
4	Rotate	Password	Relax
5	Relax	Color	Words
6	Words	Music	Count
7	Rotate	Music	Music
8	Password	Rotate	Relax
9	Relax	Relax	Color
10	Rotate	Rotate	Words
11	Relax	Relax	Color
12	Count	Password	Password

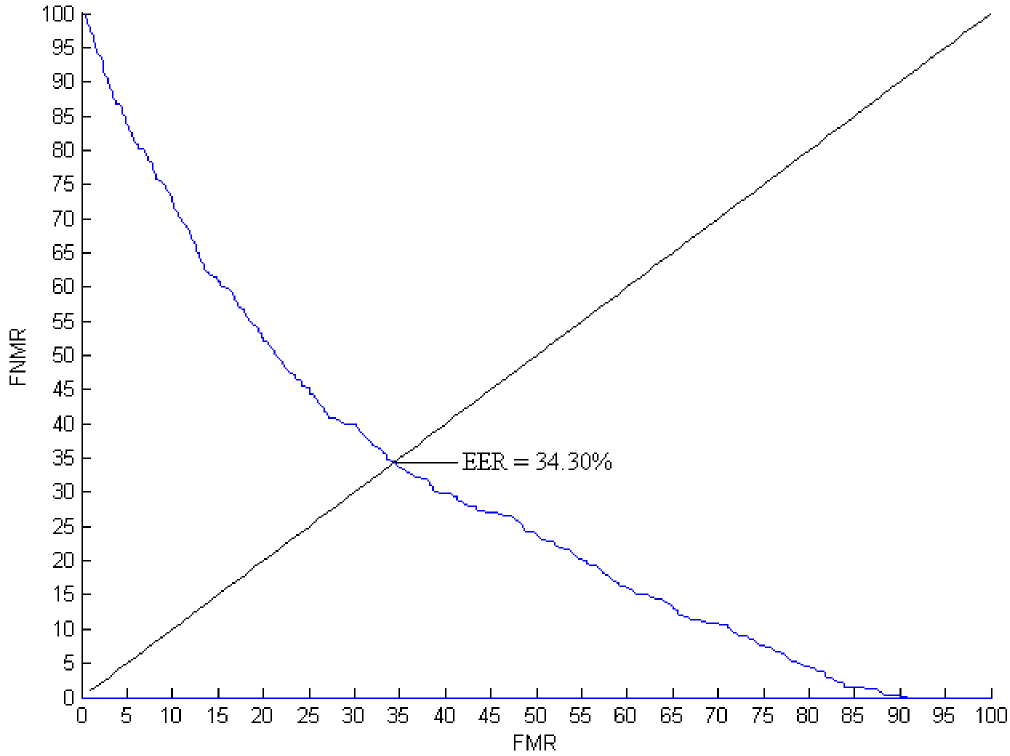


Figure 16: DET-Curve with DTW fGamma as distance metric across all sessions

6.5 Feature based distance metric

After the poor results with DTW we move on to the features and create our own distance metric. We have 25 features to work with and the distance metric can utilize a combination of all or just some of them. Our challenge is to find the best one.

6.5.1 Feature to task relation

The first step is to compute how each the feature relate to the same task across all sessions per client. A distance metric based on the Manhatten distance of each feature is used

$$d(s_1, s_2) = |s_1.feature_o - s_2.feature_o| \tag{6.11}$$

This way we find can the feature with the best relation to each task. We compute the average category A_a distance per client across all sessions using each feature in the distance metric. The results in Table 8 show that the values in two different rows vary allot because of the inherent differences in feature values (Table 1), so it makes no sense to compare two rows and the table should be read row by row only. The general trend is that distances decrease from *relax* to *count* and then rise again with *read*. This is probably due to the initialization period and implies that the equipment actually does have an impact on the performance. The PbetaHigh and Pgamma features gets a noticeable drop in average distance from the *Relax* task to the *Color* task (16.90 to 1.69 and 426.44 to 99.25 respectively), which means that a lot of the noise in the first minute of recording is originating from the betaHigh band and gamma band. All in all there is no real evidence that some features relate to specific tasks.

6.5.2 Feature performance

The next step is to test the performance of each feature without considering specific tasks by doing an "all vs all" approach instead, excluding the read task as this is only a reference task. We use the manhatten distance (Equation 6.11) and category $A_1, ..., A_7$ and $B_1, ..., B_{26}$ as genuine attempts and $C_1, ..., C_7$ and $D_1, ..., D_{49}$ as fraudulent attempts. The results in Table 9 show that EER is generally very high, but that each session has lower EER than when all sessions are included. E.g. the EER when using ZCR in the distance metric is e.g 33.07%, 33.87%, and 31.26% when session 1, 2, and 3 are computed separately while the EER is 42.74% when computed across all sessions. In fact, there are no features that provide an EER lower than 39.36% (PbetaHigh) when computing distances across all sessions. Therefore we have to create a distance metric that utilize a combination of the features to reach an EER lower than 39.36%

Table 8: The average A_a distance across all sessions

	Task							
Feature	Relax	Color	Rotate	Password	Music	Words	Count	Read
MSV	0.0225	0.0115	0.0110	0.0159	0.0210	0.0144	0.0167	0.0362
ZCR	0.12	0.08	0.08	0.07	0.06	0.07	0.07	0.08
VAZ	210.19	150.48	173.00	170.60	187.41	180.06	190.26	200.15
PF	14.51	13.45	12.45	8.84	10.46	6.70	5.95	6.92
PFM	139.99	68.77	76.30	72.33	82.74	73.34	74.63	119.41
MSP	202.29	109.70	71.59	74.76	74.19	92.23	74.34	112.19
MPA	13.06	12.71	14.56	15.74	14.58	13.82	14.04	17.64
Pdelta	730.88	534.29	381.03	386.96	402.55	511.02	408.82	617.93
PFdelta	0.36	0.28	0.23	0.19	0.23	0.26	0.18	0.21
PFMdelta	39.81	32.67	37.52	42.09	40.36	50.13	47.50	47.23
Ptheta	69.59	54.80	16.24	17.47	14.44	16.12	13.00	61.30
PFtheta	0.83	0.95	1.02	1.01	1.01	0.81	1.14	0.78
PFMtheta	7.69	6.06	4.51	4.70	3.81	4.38	3.95	8.69
Palpha	16.03	12.34	4.21	4.80	4.51	3.89	4.21	9.49
PFalpha	1.05	1.19	1.09	1.09	1.06	1.06	0.97	0.90
PFMalpha	3.62	2.67	2.13	2.18	2.14	1.97	2.23	2.59
PbetaLow	6.50	3.21	1.45	1.54	1.20	1.30	1.21	2.03
PFbetaLow	1.11	0.76	0.64	0.69	0.54	0.86	0.59	0.97
PFMbetaLow	2.63	2.44	1.84	1.65	1.89	1.75	1.67	1.67
PbetaHigh	16.90	1.69	1.13	1.14	0.91	0.86	0.77	1.18
PFbetaHigh	2.98	3.11	2.95	3.09	2.74	3.14	3.03	3.31
PFMbetaHigh	4.24	1.35	1.16	1.10	1.02	0.86	0.79	1.07
Pgamma	426.44	99.26	82.4881	61.12	47.35	42.78	41.98	52.22
PFgamma	0.31	0.43	1.02	0.52	0.66	0.53	0.77	1.25
PFMgamma	94.02	46.47	40.20	35.67	28.43	27.17	25.38	35.24

Table 9: EER in each session for each feature in a manhatten distance metric

Feature	Session 1	Session 2	Session 3	All
MSV	46.40%	46.83%	41.32%	47.90%
ZCR	33.07%	33.87%	31.26%	42.74%
VAZ	32.07%	33.29%	33.14%	40.55%
PF	36.42%	40.15%	35.76%	42.95%
PFM	33.70%	36.02%	38.50%	44.14%
MSP	34.03%	38.81%	37.36%	42.92%
MPA	46.51%	49.23%	50.02%	49.06%
Pdelta	35.01%	38.92%	39.89%	44.32%
PFdelta	41.08%	40.00%	45.75%	47.45%
PFMdelta	33.71%	37.78%	40.02%	45.37%
Ptheta	37.34%	40.33%	41.52%	43.66%
PFtheta	47.43%	48.58%	47.62%	48.64%
PFMtheta	36.34%	41.70%	43.02%	44.50%
Palpha	41.14%	37.75%	38.79%	41.51%
PFalpha	44.64%	43.40%	44.00%	45.16%
PFMalpha	41.49%	40.08%	40.13%	42.51%
PbetaLow	41.38%	37.62%	37.01%	42.65%
PFbetaLow	49.53%	49.52%	44.85%	49.19%
PFMbetaLow	49.20%	49.87%	35.03%	48.01%
PbetaHigh	35.26%	34.52%	36.82%	39.36%
PFbetaHigh	46.85%	47.44%	41.54%	46.04%
PFMbetaHigh	40.59%	38.31%	37.92%	41.88%
Pgamma	38.47%	34.36%	36.57%	42.46%
PFgamma	35.88%	31.48%	39.59%	41.86%
PFMgamma	39.99%	32.01%	34.97%	42.15%

6.5.3 Results

We have tried lots of different feature combinations in our distance metric with lots of unsatisfactory results. So far the best one is based on the correlation between PbetaLow and PbetaHigh, and Ptheta and Palpha as we discovered in Table 3. We call it *feature based distance metric* and it looks like this (Matlab code in Appendix D):

$$d_1 = |(X.PbetaLow/X.PbetaHigh - Y.PbetaLow/Y.PbetaHigh)|$$
$$d_2 = |(X.PbetaLow/Y.PbetaLow - Y.PbetaLow/X.PbetaLow)|$$
$$d_3 = |(X.PbetaHigh/Y.PbetaHigh - Y.PbetaHigh/X.PbetaHigh)|$$
$$d_4 = |(X.Ptheta/X.Palpha - Y.Ptheta/Y.Palpha)|$$
$$d_5 = |(X.Ptheta/Y.Ptheta - Y.Ptheta/X.Ptheta)|$$
$$d_6 = |(X.Palpha/Y.Palpha - Y.Palpha/X.Palpha)|$$
$$d(s_1, s_2) = d_1 + d_2 + d_3 + d_4 + d_5 + d_6$$

To show that the feature based distance metric perform better than DTW fGamma, we try the same ”best-case-scenario” as we did with DTW fGamma. As we can see from Table 10 the feature based distance metric also fails to detect the same task in each session, but the overall result is a lot better with the feature based distance metric (Table 11). It is important to note that the ”best-case-scenario” is based on too few values (432 genuine attempts and 10692 fraudulent attempts) to be conclusive, it is just a case to show that the feature based distance metric has better performance than DTW fGamma.

Table 10: Best performing category A task with feature based distance metric

Client	Session 1	Session 2	Session 3
1	Count	Words	Words
2	Count	Color	Password
3	Color	Count	Password
4	Music	Color	Music
5	Rotate	Password	Words
6	Count	Words	Words
7	Music	Music	Music
8	Rotate	Words	Count
9	Rotate	Color	Words
10	Rotate	Music	Music
11	Relax	Color	Rotate
12	Music	Color	Count

Table 11: Comparison between DTW fGamma and feature based distance metric in the ”best-case-scenario”

Distance metric	Session 1	Session 2	Session 3	All
DTW fGamma	22.2%	22.2%	14.7%	34.30%
feature based	5.51%	7.55%	9.0488%	21.42%

In order to really test the feature based distance metric, we have to include more genuine- and fraudulent attempts. We try an ”all-vs-all’ computation (excluding the *read* task) where we use category $A_1, ..., A_7$ and $B_1, ..., B_{26}$ as genuine attempts and $C_1, ..., C_7$ and $D_1, ..., D_{49}$ as fraudulent attempts. This resulted in an EER of 25.34%, 28.03%, and

28.38% in session 1, 2 and 3 respectivetly, which means that a combination of features is clearly an improvement over using just one of the features as we did in Table 9. The EER calculated across all sessions (Figure 17) based on 23436 genuine attempts and 261954 fraudulent attempts is 4.95% better than using PbetaHigh alone (39.36 % - 34.41%).

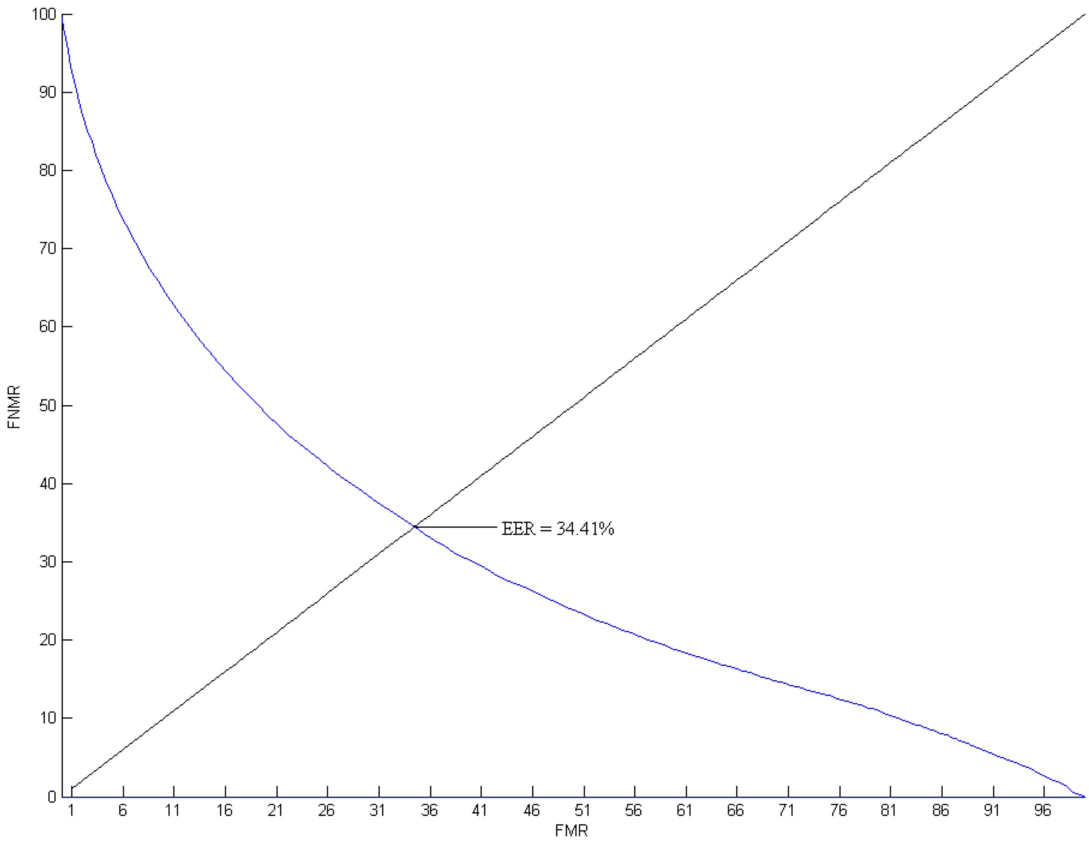


Figure 17: DET-Curve with feature based distance metric across all sessions

We have seen that DTW alone has poor performance on EEG signals, but maybe it can work in combination with the feature based distance metric. Our final attempt is based on a combination of both:

$$
\begin{aligned}
d_1 &= |(X.PbetaLow/X.PbetaHigh - Y.PbetaLow/Y.PbetaHigh)| \\
d_2 &= |(X.PbetaLow/Y.PbetaLow - Y.PbetaLow/X.PbetaLow)| \\
d_3 &= |(X.PbetaHigh/Y.PbetaHigh - Y.PbetaHigh/X.PbetaHigh)| \\
d_4 &= |(X.Ptheta/X.Palpha - Y.Ptheta/Y.Palpha)| \\
d_5 &= |(X.Ptheta/Y.Ptheta - Y.Ptheta/X.Ptheta)| \\
d_6 &= |(X.Palpha/Y.Palpha - Y.Palpha/X.Palpha)| \\
d_7 &= DTW(X.fGamma, Y.fGamma) \\
d(s_1, s_2) &= d_1 + d_2 + d_3 + d_4 + d_5 + d_6 + d_7
\end{aligned}
$$

With this distance metric we achieved an EER based of 34.40%, which only 0.01% better than without DTW and may just be due to chance. And while the computation based on the feature based distance metric alone use 35 seconds to complete, the distance metric including DTW takes about 35 hours to complete.

7 Conclusion

We found the EEG signal activity of our clients to be strongest in the delta band (1Hz - 4Hz) and the gamma band (4Hz - 8Hz) with the *ThinkGear* head set by *Neurosky*. The equipment suffered from an initialization period where the signal gradually improved during a client's recording session, which made it difficult to analyze the signal in the time domain. As a consequence our attempt to apply Dynamic Time Warping (DTW) on filtered versions of the time domain signal did not produce good results. With DTW we only achieved an EER of 34.30% in an unrealistic ”best-case-scenario” with few values were we selected the best performing tasks for each client, a case that should get a low EER for DTW to have any potential. To overcome this problem we transferred the signal into the frequency domain and extracted features from both domains. Initially we wanted to be able to distinguish between tasks as well as clients, but we have not been able to do this as we did not find any relation between a specific task and feature. By utilizing the correlations we found between the power in the betaLow- and betaHigh band as well as the theta- and alpha band in what we called a *feature based distance metric*, we managed to get an EER of 21.42% in the ”best-case-scenario”, a clear improvement over DTW. When we applied the feature based distance metric in an ”all-vs-all” computation we got an EER of 34.41%, which is the best result so far.

In our final attempt we combined the feature based distance metric with DTW in the same ”all-vs-all” computation and got an EER of 34.40%, which is such a marginal improvement that we cannot justify it. In the end it looks like DTW does not perform well on EEG signals.

We have to ask ourselves, why do we get an EER lower than 50% (the worst case)? Is it because we actually detect client specific similarities and differences, or is it in fact session based such that two consequtive signals are similar because of the equipment? The results strongly suggests that the similarities are session based. The reason is that one sensor with only 128Hz sample frequency fails to extract enough client specific information in order to extract features with high entropy. The only reliable information we can extract with one sensor is the power in each frequency band, which is not enough to distinguish between clients or tasks.

Finally we have to say that based on what we have seen in related work and what we have been able to do with just one sensor, we believe that an implementation of a brain wave based authentication system is just a matter of time.

8 Further work

Considering the infinite possibilities in a distance metric creation there is probably a feature combination that may perform even better than the one we ended up with. The challenge is to find it, and we have been analyzing our data to the very end of this thesis, trying to improve the results. The next step would be to learn about Autoregressive (AR) parameters and neural networks like in [8] and their performance on authentication with one EEG sensor. One possibility is to use WEKA, a collection of machine learning algorithms for data mining tasks [45].

We could also try different tasks that are more suitable for the FP1 location as well as different sensor placements, but overall we do not believe that tasks has such a huge impact in authentication. What we really want to find is the EEG to DNA relation as Vogel discovered in the 1960's [4].

The most evident improvement that can be done in further work is to get better equipment that has more sensors and a higher sample frequency. There has been lots of improvements to EEG equipment during the writing of this thesis and it has become cheap enough for the public to buy. Emotiv [13] recently announced their new head set:

> *Based on the latest developments in neuro-technology, Emotiv has developed a new personal interface for human computer interaction.*
>
> *The Emotiv EPOC uses a set of sensors to tune into electric signals naturally produced by the brain to detect player thoughts, feelings and expression and connects wirelessly to most PCs.*
>
> *The Emotiv neuroheadset now makes it possible for games to be controlled and influenced by the player's mind.*

It would be very exciting to test this head set in terms of authentication. Even having two sensors is a great improvement over just one as it allows analysis on the relation between the sensors. Other possibilities include those we mentioned in the related work chapter, Princippal Component analysis (PCA) with The Dien PCA Toolbox [34], independent component analysis (ICA) and joint time-frequency analysis (TFA) with the Matlab toolbox EEGLAB [35], data cleaning, statistical extraction and visualization techniques with Net Station [36].

Bibliography

[1] Vidal, J. 1977. Real-time detection of brain events in eeg. *Proceedings of the IEEE*, 65.

[2] Merkow, M. & Breithaupt, J. 2006. *Information Security: Principles and Practices*. Pearson Prentice Hall.

[3] Smith, R. E. 2002. *Authentication: From Passwords to Public Keys*. Addison Wesley.

[4] Vogel. 1970. The genetic basis of the normal human electroencephalogram (eeg). *Humangenetik*, 10, 91–114.

[5] Benedicenti, L., Koles', Z., Mahovsky, J., & Paranjape, R. 2001. The electroencephalogram as a biometric. *Electrical and Computer Engineering, 2001. Canadian Conference on*, 2.

[6] Jain, A. K., Ross, A., , & Prabhakar, S. 2004. An introduction to biometric recognition. *IEEE Transactions On Circuits and Systems for Video Technology*, 14, 4–20.

[7] Loibl, T. R. 2005. Identity theft, spyware and the law. In *InfoSecCD '05: Proceedings of the 2nd annual conference on Information security curriculum development*, 118–121, New York, NY, USA. ACM.

[8] Lee, J. C. & Tan, D. S. 2006. Using a low-cost electroencephalograph for task classification in hci research. *19th ACM Symposium on User Interface Software and Technology*, 81–90.

[9] Childers, D. G., Halpeny, O. S., Mesa, W., & N. W. Perry, J. 1972. Simulation of spatio-temporal visual evoked eeg's. In *ACM '72: Proceedings of the ACM annual conference*, 884–898, New York, NY, USA. ACM.

[10] Dou, D., Frishkoff, G., Rong, J., Frank, R., Malony, A., & Tucker, D. 2007. Development of neuroelectromagnetic ontologies(nemo): a framework for mining brainwave ontologies. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 270–279, New York, NY, USA. ACM Press.

[11] Thorpe, J., van Oorschot, P. C., & Somayaji, A. 2005. Pass-thoughts: authenticating with our minds. In *NSPW '05: Proceedings of the 2005 workshop on New security paradigms*, 45–56, New York, NY, USA. ACM Press.

[12] Brain products.
`http://www.brainproducts.com` (last visited June 30th 2008).

[13] Emotiv systems.
`http://www.emotiv.com/` (last visited June 30th 2008).

[14] Neurosky.
http://www.neurosky.com/ (last visited June 30th 2008).

[15] JM, C., MA, L., RE, C., JE, D., & DM, S. 2003. Learning to control a brainmachine interface for reaching and grasping by primates. *PLoS Biology*, 1.

[16] Brower, V. 2005. When mind meets machine. *EMBO Reports*, 6.

[17] Kaul, P. 2006. Neurological gaming environments. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Educators program*, 25, New York, NY, USA. ACM.

[18] Chen, D. & Vertegaal, R. 2004. Using mental load for managing interruptions in physiologically attentive user interfaces. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, 1513–1516, New York, NY, USA. ACM.

[19] Monkeys control a robot arm with thoughts.
http://www.iht.com/articles/2008/05/29/healthscience/29brain.php (last visited June 30th 2008).

[20] Birbaumer, N., Hinterberger, T., McFarland, D., Schalk, G., & Wolpaw, J. June 2004. Bci2000: a general-purpose brain-computer interface (bci) system. *Biomedical Engineering, Transactions on*, 51, 1034 – 1043.

[21] Keirn, Z. A. & Aunon, J. I. 1990. A new mode of communication between man and his surroundings. *IEEE transactions on bio-medical engineering.*, 37.

[22] Doyle, J. C., Omstein, R., & Galin, D. 1974. Lateral specialization of cognitive mode: I1 eeg frequency analysis. *Psychophysiol.*, 11, 567–578.

[23] Butler, S. R. & Glass, A. 1974. Asymmetries in the electroencephalogram associated with cerebral dominance. *Electroencephalogr. clin. Neurophysiol.*, 36, 481–491.

[24] Ehrlichman, H. & Wiener, M. S. 1980. Eeg asymmetry during covert mental activity. *Psychophysiol.*, 17, 228–235.

[25] Weinshall, D. & Kirkpatrick, S. 2004. Passwords you'll never forget, but can't recall. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, 1399–1402, New York, NY, USA. ACM Press.

[26] Alexandris, N., Poulos, M., & Rangoussi, M. 1999. Neural network based person identification using eeg features. *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, 2.

[27] Chrissikopoulos, V., Evangelou, A., Poulos, M., & Rangoussi, M. 1999. Person identification based on parametric processing of the eeg. *Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on*, 1.

[28] Chrissikopoulos, V., Evangelou, A., Poulos, M., & Rangoussi, M. 1999. Parametric person identification from the eeg using computational geometry. *Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on*, 2.

[29] Poulos, M., Rangoussi, M., Alexandris, N., & Evangelou, A. 2002. Person identification from the eeg using nonlinear signal classification. *Methods of information in medicine.*, 41.

[30] Welch, P. 1967. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *Audio and Electroacoustics, IEEE Transactions on*, 15, 70–73.

[31] Marcel, S. & del R. Millan, J. 2007. Person authentication using brainwaves (eeg) and maximum a posteriori model adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4), 743–752.

[32] Keogh, E. & Pazzani, M. 2001. Derivative dynamic time warping. *First SIAM International Conference on Data Mining, (Chicago, IL, 2001)*.

[33] Salvador, S. & Chan, P. 2007. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5), 561–580.

[34] Dien. J. pca toolbox (version 1.7).
`http://people.ku.edu/~jdien` (last visited June 30th 2008).

[35] Eeg lab.
`http://www.sccn.ucsd.edu/eeglab` (last visited June 30th 2008).

[36] Electrical geodesics, inc.
`http://www.egi.com` (last visited June 30th 2008).

[37] del R., M. J. 2003. Adaptive brain interfaces. *Communications of the ACM*, 46.

[38] Herwig, U., Satrapi, P., & Schonfeldt-Lecuona, C. December, 2003. Using the international 10-20 eeg system for positioning of transcranial magnetic stimulation. *Brain Topography*, 16(2), 95–99.

[39] Woolsey, T. A., Hanaway, J., & Gado, M. H. 2008. *the Brain Atlas, A Visual Guide to the Human Central Nervous System, 3rd edition*. Wiley.

[40] Dietrichs, E. & Gjerstad, L. 2007. *Vaar Fantastiske Hjerne*. Gylendal.

[41] Okuda, J., Fujii, T., Ohtake, H., Tsukiura, T., Tanji, K., Suzuki, K., Kawashima, R., Fukuda, H., Itoh, M., & Yamadori, A. 2003. Thinking of the future and past: the roles of the frontal pole and the medial temporal lobes. *NeuroImage.*, 19.

[42] Sarikaya, B., Alim, M. A., & Rezaei, S. 2006. Integrating wireless eegs into medical sensor networks. In *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*, 1369–1374, New York, NY, USA. ACM.

[43] Leyman, A. & Sanei, S. 2001. Eeg brain map reconstruction using blind source separation. *Statistical Signal Processing, 2001. Proceedings of the 11th IEEE Signal Processing Workshop on*.

[44] Montgomery, D. C. & Runger, G. C. 2006. *Applied Statistics and Probability for Engineers, 4th Ed.* Wiley.

[45] Weka, a collection of machine learning algorithms for data mining tasks. `http://www.cs.waikato.ac.nz/~ml/weka/` (last visited June 30th 2008).

A Participant Agreement Form

The Participation Agreement Form is in its original form on the next page.

Participant Agreement Form

Purpose and explanation:

In this experiment you will wear a headset with the capability of recording a signal that emits from your brain (EEG signal), captured on your forehead by a sensor attached to the headset. There will be no signals sent into your brain. With the aid from a researcher you will be required to think about different thoughts with your eyes closed and arms resting in your lap. The purpose of the experiment is to research similarities and differences between thoughts and how they can be separated from one another. One session consists of 24 recordings that lasts 20 seconds each to a total of 8 minutes. Breaks will be given when needed. We hope you will participate for 3 sessions separated with a few days.

Participation in acquisition of EEG data:

I am participating in the acquisition of EEG data on a voluntarily basis. The purpose of this experiment is described above.

The only data related to me that will be stored on a computer is the raw EEG signal recorded by the headset and an identification number that will be provided by the researcher. My name and signature will exist on this paper only.

With my signature I confirm the following:

1) I have been informed in oral and written form about the purpose of the experiment.
2) I allow the researcher use the headset to collect EEG data from me
3) The data may be used in future research, i.e. other master thesis experiments.
4) I am allowed to reject to sign the agreement.
5) I can request to receive insight in the collected data and get further explanation of its purpose.
6) I can withdraw my participation at anytime without giving any explanation and all collected data from me will be permanently deleted.

Full name: ______________________

Gjøvik, date: _____________________signature: ______________________

B Matlab signal processing code

```
% readSignals(folder)
%
% Author: Kennet Fladby
%
% parameter: folder - read all the signal files in the specified folder.
% This folder must contain raw signal files only and no subfolders.
%
% returns: signals - a struct containing samples and features of each client's recordin
function [signals] = readSignals(folder)

    Fs = 128;          % Sample rate
    t = 20;            % Recording duration in seconds
    cut = 4;           % Used to remove the first specified seconds
    N = (t-cut) * Fs;  % Number of samples
    df = Fs/N;         % The resolution of the fourier transform

    % Six frequency bands and their location in the fft transform.
    % E.g. frequency 4 is located at 4/df, but because matlab stores the
    % DC value at index 1 so the correct location is 4/df + 1.
    deltaBand =    [1/df+1,4/df+1];    %1-4Hz
    thetaBand =    [4/df+1,8/df+1];    %4-8Hz
    alphaBand =    [8/df+1,12/df+1];   %8-12Hz
    betaLowBand =  [12/df+1,20/df+1];  %12-20Hz
    betaHighBand = [20/df+1,30/df+1];  %20-30Hz
    gammaBand =    [30/df+1,50/df+1];  %30-50Hz

    %Retrieve all the filenames in the specified folder
    files = dir(folder);

    files(1:2) = [];

    signals = struct;

    %Loop through all the signal files
    for i=1:numel(files)

        %Get signal name information
        [start_idx, end_idx, extents, matches, tokens] = regexp(files(i).name,'(\d+)\_(
        client = ['Client',tokens{1}{1}];
        task = tokens{1}{2};
        session = ['Session',tokens{1}{3}];
        recording = ['Recording',tokens{1}{4}];

        %Opens a signal file
        fileName = [folder,files(i).name];
        fid = fopen(fileName,'r');

        %Ignore the first line which is just text
        textscan(fid,'%*s %*s',1);

        %Ignore the first specified number of samples
```

```
        textscan(fid,'%n %n',cut*Fs,'delimiter',';');

        %Store the remaining specified number of samples
        v = textscan(fid,'%n %n',N,'delimiter',';');
        X = double(v{2});
        fclose(fid);

        %Loop through all samples to find values above zero.
        VAZ = 0;

        for n=1:N
            if (X(n) > 0)
                VAZ = VAZ + 1;
            end
        end

        %Loop through all samples to find zero crossing rate
        ZCR = 0;

        for n=1:N-1

           %If the product of two adjacent samples are negative,
           %a zero-crossing has occured.
           if (X(n)*X(n+1) < 0)
               ZCR = ZCR + 1;
           end

        end

        ZCR = ZCR/N;

        % Transform the samples to the frequency domain
        H = fft(X);

        %Filters the signal to only include band frequencies
        fX = fourierFilter(H, 2, gammaBand(2));
        fDelta = fourierFilter(H, deltaBand(1), deltaBand(2));
        fTheta = fourierFilter(H, thetaBand(1), thetaBand(2));
        fAlpha = fourierFilter(H, alphaBand(1), alphaBand(2));
        fBetaLow = fourierFilter(H, betaLowBand(1), betaLowBand(2));
        fBetaHigh = fourierFilter(H, betaHighBand(1), betaHighBand(2));
        fGamma = fourierFilter(H, gammaBand(1), gammaBand(2));

        % Mean phase angle
        MPA = mean(angle(H(2:gammaBand(2))))*180/pi;

        % Signal power in each of the six frequency bands.
        Pdelta = mean(abs(H(deltaBand(1):deltaBand(2))).^2);
        Ptheta = mean(abs(H(thetaBand(1):thetaBand(2))).^2);
        Palpha = mean(abs(H(alphaBand(1):alphaBand(2))).^2);
        PbetaLow = mean(abs(H(betaLowBand(1):betaLowBand(2))).^2);
        PbetaHigh = mean(abs(H(betaHighBand(1):betaHighBand(2))).^2);
        Pgamma = mean(abs(H(gammaBand(1):gammaBand(2))).^2);

        % Mean spectral power
        MSP = (Pdelta+Ptheta+Palpha+PbetaLow+PbetaHigh+Pgamma) / 6;

        % Find peak frequency and peak frequency magnitude in the delta
```

```
        % to gamma range 1-50Hz.
        [PFM, maxLoc] = max(abs(H(2:gammaBand(2))));
        PF = maxLoc * df;

        % Find peak frequency and peak frequency magnitude in all bands
        [PFMdelta,maxLoc] = max(abs(H(deltaBand(1):deltaBand(2))));
        [PFdelta] = (deltaBand(1) + maxLoc - 2) * df;

        [PFMtheta,maxLoc] = max(abs(H(thetaBand(1):thetaBand(2))));
        [PFtheta] = (thetaBand(1) + maxLoc - 2) * df;

        [PFMalpha,maxLoc] = max(abs(H(alphaBand(1):alphaBand(2))));
        [PFalpha] = (alphaBand(1) + maxLoc - 2) * df;

        [PFMbetaLow,maxLoc] = max(abs(H(betaLowBand(1):betaLowBand(2))));
        [PFbetaLow] = (betaLowBand(1) + maxLoc - 2) * df;

        [PFMbetaHigh,maxLoc] = max(abs(H(betaHighBand(1):betaHighBand(2))));
        [PFbetaHigh] = (betaHighBand(1) + maxLoc - 2) * df;

        [PFMgamma,maxLoc] = max(abs(H(gammaBand(1):gammaBand(2))));
        [PFgamma] = (gammaBand(1) + maxLoc - 2) * df;

        %Store signal information sorted by client, session, task and
        %recording.
        signals.(client).(task).(session).(recording).fileName = files(i).name;
        signals.(client).(task).(session).(recording).X = X;
        signals.(client).(task).(session).(recording).fX = fX;
        signals.(client).(task).(session).(recording).fDelta = fDelta;
        signals.(client).(task).(session).(recording).fTheta = fTheta;
        signals.(client).(task).(session).(recording).fAlpha = fAlpha;
        signals.(client).(task).(session).(recording).fBetaLow = fBetaLow;
        signals.(client).(task).(session).(recording).fBetaHigh = fBetaHigh;
        signals.(client).(task).(session).(recording).fGamma = fGamma;
        signals.(client).(task).(session).(recording).MSV = H(1)/N;
        signals.(client).(task).(session).(recording).ZCR = ZCR;
        signals.(client).(task).(session).(recording).VAZ = VAZ;
        signals.(client).(task).(session).(recording).PF = PF;
        signals.(client).(task).(session).(recording).PFM = PFM;
        signals.(client).(task).(session).(recording).MSP = MSP;
        signals.(client).(task).(session).(recording).MPA = MPA;
        signals.(client).(task).(session).(recording).Pdelta = Pdelta;
        signals.(client).(task).(session).(recording).PFdelta = PFdelta;
        signals.(client).(task).(session).(recording).PFMdelta = PFMdelta;
        signals.(client).(task).(session).(recording).Ptheta = Ptheta;
        signals.(client).(task).(session).(recording).PFtheta = PFtheta;
        signals.(client).(task).(session).(recording).PFMtheta = PFMtheta;
        signals.(client).(task).(session).(recording).Palpha = Palpha;
        signals.(client).(task).(session).(recording).PFalpha = PFalpha;
        signals.(client).(task).(session).(recording).PFMalpha = PFMalpha;
        signals.(client).(task).(session).(recording).PbetaLow = PbetaLow;
        signals.(client).(task).(session).(recording).PFbetaLow = PFbetaLow;
        signals.(client).(task).(session).(recording).PFMbetaLow = PFMbetaLow;
        signals.(client).(task).(session).(recording).PbetaHigh = PbetaHigh;
        signals.(client).(task).(session).(recording).PFbetaHigh = PFbetaHigh;
        signals.(client).(task).(session).(recording).PFMbetaHigh = PFMbetaHigh;
        signals.(client).(task).(session).(recording).Pgamma = Pgamma;
        signals.(client).(task).(session).(recording).PFgamma = PFgamma;
```

```
            signals.(client).(task).(session).(recording).PFMgamma = PFMgamma;

      end

end

% Filters a fast fourier transformed sequence to include only frequencies
% in the range: lowerFreq,...,upperFreq.
function [filteredX] = fourierFilter(H, lowerFreq, upperFreq)

            N = numel(H);

            if (lowerFreq > 2)
                  H(2:(lowerFreq-1)) = 0;
                  H((N - lowerFreq+3):N) = 0;
            end

            H((upperFreq+1):(N-upperFreq+1)) = 0;

            filteredX = ifft(H);

end
```

C Matlab Chi-square goodness-of-fit test

```
% chisquare(X)
%
% Author: Kennet Fladby
%
% Applies a goodness of fit test with confidence interval 0.05 to test
% whether X follows a normal distribution or not. X should contain at least
% 30 values as the significance of the results increase with larger X.
%
% Parameters: X - an array containing the samples to test
%
% Returns: isNormal - [true] if X have a normal distribution
%                     [false] if X does not have a normal distribution
%
function [isNormal] = chisquare(X)

    k = 10;
    p = 1/k;
    E = numel(X) * p;
    rejectTh = 14.07;

    meanX = mean(X);
    stdX = std(X);

    %Found from a table
    chiCell = [-inf,-1.29; -1.29,-0.85; -0.85,-0.53; -0.53, -0.26; -0.26,0; 0,0.26; 0.2

    %Calculates the lower and upper range of the k bins
    bins = meanX + chiCell.*stdX;

    %Counts the number of samples in each bin
    O = zeros(k,1);

    % Put the values in X in their respective bins
    for n=1:numel(X)

        for i=1:k

            if (X(n) >= bins(i,1) && X(n) < bins(i,2))
                O(i) = O(i) + 1;
            end
        end

    end

    X2 = 0;

    %Calculate the X2 value
    for i=1:k

        X2 = X2 + ((O(i) - E)^2)/E;

```

```
    end

    %Decide whether X has a normal distribution or not
    if (X2 >= rejectTh)
        isNormal = false;
    else
        isNormal = true;
    end

end
```

D Matlab feature based distance metric

```
% getDistance(X,Y)
%
% Author: Kennet Fladby
%
% Parameters: X,Y - Calculates the distance between X and Y using this distance metric
%
% Returns: d - the distance
function [d] = getDistance(X, Y)

    d1 = abs(X.PbetaLow/X.PbetaHigh - Y.PbetaLow/Y.PbetaHigh);
    d2 = abs(X.PbetaLow/Y.PbetaLow - Y.PbetaLow/X.PbetaLow);
    d3 = abs(X.PbetaHigh/Y.PbetaHigh - Y.PbetaHigh/X.PbetaHigh);
    d4 = abs(X.Ptheta/X.Palpha - Y.Ptheta/Y.Palpha);
    d5 = abs(X.Ptheta/Y.Ptheta - Y.Ptheta/X.Ptheta);
    d6 = abs(X.Palpha/Y.Palpha - Y.Palpha/X.Palpha);
    dist2 = d1+d2+d3+d4+d5+d6;

    d = dist2;

end
```

E Matlab DET-Curve and EER computation

```
% calculateDET(intraValues, interValues)
%
% Author: Kennet Fladby
%
% Calculates FMR, FNMR and EER
%
% Returns: DET - Contains a matrix with FMR, FNMR and threshold values
%          EER - The calculated EER
function [DET, EER] = calculateDET(intraValues, interValues)

        %Count the number of genuine and fraudulent attempts
        intraAttempts = numel(intraValues);
        interAttempts = numel(interValues);

        lastFMR = 0;
        lastFNMR = 0;

        findEER = true;

        intraMax = max(intraValues);
        interMax = max(interValues);
        intraMin = min(intraValues);
        interMin = min(interValues);

        %Find the threshold range
        if (intraMax >= interMax)
            tMax = double(intraMax);
        else
            tMax = double(interMax);
        end

        if (intraMin <= interMin)
            tMin = double(intraMin);
        else
            tMin = double(interMin);
        end

        tStep = (tMin + tMax)/10000;

        %Adjust the threshold range to include the edges
        tMax = tMax + tStep;

        tStep = (tMin + tMax)/10000;

        count = 1;
        DET = zeros(10000,3);

        %Calculate FMR and FNMR
        for t=tMin:tStep:tMax

            FMCount = 0;
```

```
            FNMCount = intraAttempts;

            % Counts the number of false matches (inter values below
            % the threshold)
            for k=1:interAttempts
                if (interValues(k) < t)
                    FMCount = FMCount + 1;
                end
            end

            % Counts the number of false non-matches (intra values above
            % the threshold)
            for k=1:intraAttempts

                if (intraValues(k) < t)
                   FNMCount = FNMCount - 1;
                end
            end

            % Compute the rates
            FMR = (FMCount/interAttempts)*100;
            FNMR = (FNMCount/intraAttempts)*100;

            %Store the result
            DET(count,1) = FMR;
            DET(count,2) = FNMR;
            DET(count,3) = t;

            count = count + 1;

            % Find the EER.
            if (findEER)

                if ((FMR == FNMR) || ((FMR > FNMR) && (lastFMR < lastFNMR)) || ((FMR < FNMR) &&
                    EER = (FMR + FNMR + lastFMR + lastFNMR) / 4;
                    findEER = false;
                end

                lastFMR = FMR;
                lastFNMR = FNMR;

            end

        end

end
```