Expertise
and insight
for the future

Aliisa Paulus

Development of Cost-Efficient and Accurate Tactile Sensing Wristband for Real-Time Static Gesture Recognition

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Electrical and Automation Engineering

Bachelor’s Thesis

13 November 2020

Author Title Number of Pages Date	Aliisa Paulus Development of Cost-Efficient and Accurate Tactile Sensing Wristband for Real-Time Static Gesture Recognition 51 pages + 2 appendices 13 November 2020
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Electronics
Instructors	Heikki Valmu, Principal Lecturer Matti Fischer, Principal Lecturer

There has been a tremendous increase in the use of gestures in human-computer interaction technology, where the most common approach is vision-based systems. However, a tactile sensing device around the wrist can give more information about a hand's static gestures than only motion tracking. The sensors can detect the pressure distribution on a surface between the sensor and the wrist. Hence, the thesis aimed to design and create a tactile sensing wristband prototype for a real-time hand gesture recognition system.

An in-depth analysis of current gesture recognition systems was conducted, from which a suitable architecture and machine learning algorithm was identified. After carefully selecting the appropriate components and techniques for the wristband, the final prototype was designed, constructed, and tested to obtain its performance. The main focus lies in the classification of the pressure values into gestures with the Support Vector Machine algorithms. The sensor values were scaled with a Min-Max normalization and then used as input for the Support Vector Machine, which included a radial base function kernel.

The result showed that all ten static gestures were clearly distinguished. The classification accuracy for the static gestures in the experiment was 98.0%. The result obtained shows that the wrist-based gesture recognition prototype adhered to the requirements presented in this thesis.

Keywords	Static Gesture Recognition, Tactile Sensing, Support Vector Machine algorithm

Tiivistelmä

Tekijä Otsikko Sivumäärä Päivämäärä	Aliisa Paulus Kustannustehokkaan ja tarkan kosketusta tunnistavan rannekkeen kehittäminen reaaliaikaisten staattisten eleiden tunnistamiseen 51 sivua + 2 liitettä 13.11.2020
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	elektroniikka
Ohjaajat	yliopettaja Heikki Valmu yliopettaja Matti Fischer

Eleiden käyttö ihmisen ja tietokoneen välisessä vuorovaikutuksen teknologiassa on yleistynyt huomattavasti. Tätä teknologiaa lähestytään yleisimmin näköhavaintoon perustuvilla järjestelmillä. Käden staattisista eleistä saataisiin kuitenkin tarkempaa tietoa kosketusantureilla varustetulla laitteella ranteen ympärillä kuin pelkällä liikkeen seurannalla. Kyseiset anturit havaitsevat paineen jakautumisen anturin ja ranteen pintojen välillä. Näin ollen, opinnäytetyön tavoitteena oli kosketusta tunnistavan rannekkeen prototyyppi reaaliaikaisen suunnittelu sekä toteutus eleiden tunnistamisjärjestelmää varten.

Nykyisiä eleiden tunnistusjärjestelmiä analysointiin perusteellisesti sopivan arkkitehtuurin ja koneoppimisalgoritmin valitsemiseksi. Rannekkeeseen valittiin huolellisen arvioinnin ja suunnittelun perusteella sopivat komponentit ja tekniikat, minkä jälkeen lopullinen prototyyppi suunniteltiin, toteutettiin ja testattiin sen suorituskykyominaisuuksien saavuttamiseksi. Rannekkeen sensorien arvot luokiteltiin eleiksi tukivektorikone-algoritmia käyttämällä. Itse tukivektorikone muodostettiin radiaaliin perustuvan funktion kernelillä ja sen syöttötietoina toimivat sensorien arvot, jotka skaalattiin Min-Max -normalisointia käyttämällä.

Tulos osoitti, että kaikki kymmenen staattista elettä pystyttiin erottamaan selkeästi. Testin luokitustarkkuus staattisille eleille oli 98,0 %. Tulos osoittaa rannetietoon perustuvan eleentunnistusprototyypin olevan tämän opinnäytetyön kriteereiden mukainen.

Avainsanat	staattisten eleiden tunnistaminen, kosketusanturi, tukivektorikone-algoritmi

Contents

List of Abbreviations

3D	Three-dimensional space
ANN	Artificial Neural Network
COVID-19	Coronavirus
ELM	Extreme Learning Machine
EMG	Electromyography
FSR	Force-sensitive resistor
HCI	Human-computer interface
HGR	Hand gesture recognition
I^{2}C	Inter-Integrated Circuit
IDE	Arduino Integrated Development Environment
IMU	Inertial measurement unit
I/O	Input/output
KKT	Karush-Kuhn-Tucker
KNN	K Nearest Neighbors
LDA	Linear Discriminant Analysis
MEMS	Microelectromechanical System
ML	Machine learning
PCB	Printed Circuit Board
PLA	Polylactic Acid
RBF	Radial Basis Function
SCL	Serial Clock Input
SDA	Serial Data
SVM	Support Vector Machine
TUI	Touchless user interface
VR	Virtual reality

1 Introduction

Interacting with hands allows advanced ways in both human-to-human and human-to-computer communication. Consequently, a hand gesture recognition (HGR) method is used more widely in technology nowadays. It provides a natural and intuitive contact-free input mechanism for smart devices. Another advantage is its feasibility, which lowers the disparity between intention and actual control. Hence, there has been a tremendous increase in the use of gestures in the field of human-computer interfaces (HCI). Applications as augmented reality [1], virtual reality (VR) [2], and sign language [3] have all taken advantage of different hand gesture recognition methods due to their broad positive impact.

In comparison to other HCI devices, one of the main advantages of HGR is that direct physical manipulation techniques, such as keyboards, can be eliminated. This method increases the intuitiveness of any interface. Furthermore, gesture recognition enables an accurate operating system, especially in human-robot interaction. A robot can be commanded and controlled remotely to perform a specific task by transmitting information from the gestures. A remote control can be necessary, for example, when human presence is not possible due to safety reasons. One situation can be facilities with excessive quantities of radioactive radiation that are dangerous for humans [4].

The gestures can be identified in multiple ways creating a wide-ranging effect field of the HGR devices. In a human-robot collaboration system, the interaction should be natural. For this purpose, the device attached to the controller's body should be unnoticeable. Ideally, the device is wireless, accurate, small, and compact. However, in terms of the development of a wearable HGR device, the design's accuracy and cost are two of the major problems. In addition, the accurate devices incline to be high-priced. Therefore, this work focuses on finding out if it is possible to create a cost-efficient and accurate HGR device with fewer sensors than usual.

1.1 Project Objectives

This project includes extensive background research and analysis of the already existing devices and methods, and continuous comparison to achieve the device's highest possible accuracy with affordable designing methods and fewer sensors.

This thesis takes advantage of the tactile sensing method for creating a wearable HCI gesture recognition device. Its approach lies in five pressure sensors that are improved to sense a hand's fine movements. The device is placed around the wrist, which is a natural location for wearable devices. The aim of this thesis is to have a precise wristband with fewer sensors to recognize the differences in static hand positions. In the testing stage, the data is treated with chosen machine learning algorithms to determine whether the above objectives are met or not. In addition to the discussion, the results include a risk analysis to describe the different factors affecting the wristband and its outcome.

2 Technical Review

As a result of the development of HCI technology, gesture recognition has become a trending topic. This section focuses on the most common techniques and methods for which the gestures are applied.

2.1 Categorization of Gesture Recognition Applications

Two of the most utilized methods in gesture recognition devices are the vision and the sensor approaches. The former method is based on visual technologies, for instance, cameras, while the latter perceives the gestures from contact sensors along the arm or around the wrist. Thus, the sensor-based HGR systems are usually wearable devices. However, there is a wide range of different types of sensors, which means that there are also many different HGR applications, such as Electromyography (EMG) [5], force-sensitive resistors (FSRs) [6], and gyro sensors [7]. The signals acquired from the sensors provide a broader recognition range than cameras. Therefore, only the most common method is discussed in chapter 2.1.2 and 2.1.3, and later, in chapter 2.3, other commonly used sensor methods are summarized together with machine learning algorithms.

2.1.1 Vision-Based Devices

The vision-based devices for gesture recognition have a touchless user interface (TUI). The gesture recognition occurs in a scene by decoding explicitly with different image processing algorithms. For the scene itself, the TUI method needs environmental cameras, which require a lot of space. Thus, these devices are bulky and may not be suitable for wearing or portable use in general. The additional problem is that cameras are fragile and susceptible to fluctuations in ambient illumination, disturbing image processing. Furthermore, cameras have less precise sensors than contact devices, leading to risk-off privacy issues. The more expensive camera the device is using, the higher the required resolution is. [3;8.]

Despite the listed disadvantages, these devices are easy to set up and use. Therefore, there are several vision-based devices on the market, where the most common one is a Leap Motion controller by Ultraleap [9]. The Leap Motion's primary function is to track hand movements in real-time in three-dimensional space (3D).

2.1.2 Electromyography-Based Devices

The most widely used sensor approach for HGR is the Electromyography signal measurement. As mentioned in chapter 2.1, this chapter focuses mainly on introducing this type of signal measurement.

The working principle is place electrodes on the skin to measure the electrical potentials in contracting muscles [3]. The data is then analyzed and processed to detect which muscles are active. The advantage of EMG is the cause-and-effect technique related to muscle activation. However, the device is typically located on the forearm where the majority of the hand's muscles are, decreasing the practicality. The electrodes also generate a lot of noise when the user sweats or moves a limb, which usually leads to a low signal-to-noise ratio [10].

The most well-known EMG device for gesture recognition is a Myo Gesture Control Armband produced by Thalmic Labs Co [11]. It provides an easy to use system at a low price point. The device is designed with eight EMG electrodes that are placed around the forearm to detect the forearm movements in the 3D space. It is used especially for prosthetics, although it suffers from a noisy signal and its weak intensity. Nevertheless, sales of the Myo bracelet ended in 2018, and there are no longer any similar low-cost gesture recognition products on the market. However, the CTRL-Labs company obtained several patents from Myo and developed a new CTRL-kit prototype that has a good accuracy in gesture recognition performance [12]. Since this product is still in development, it cannot yet be purchased at a reasonable price.

2.2 Tactile Sensing Applications

Tactile sensing technology is a new technique that consists of measuring a direct physical contact between two objects and provides an exact mapping. It is a combination of pressure and force sensing techniques, making the full complexity of the contact pressure profile possible. In tactile technology, two mostly used sensors are capacitive and resistive ones [13]. The first one is used to measure the resistance of a conductive material, and the capacitor measures its ability to store electrical charge. Also, the MEMS barometers can be modified to suit the tactile applications [14]. This approach is low-priced compared to the typical tactile array sensors.

The tactile sensors have good robustness and stability compared to other pressure or force sensors. They are mostly used in human-robot interaction to classify different types of touch [15]. The comparison of the measurement profiles between the pressure, force, and tactile sensor is illustrated in figure 1.

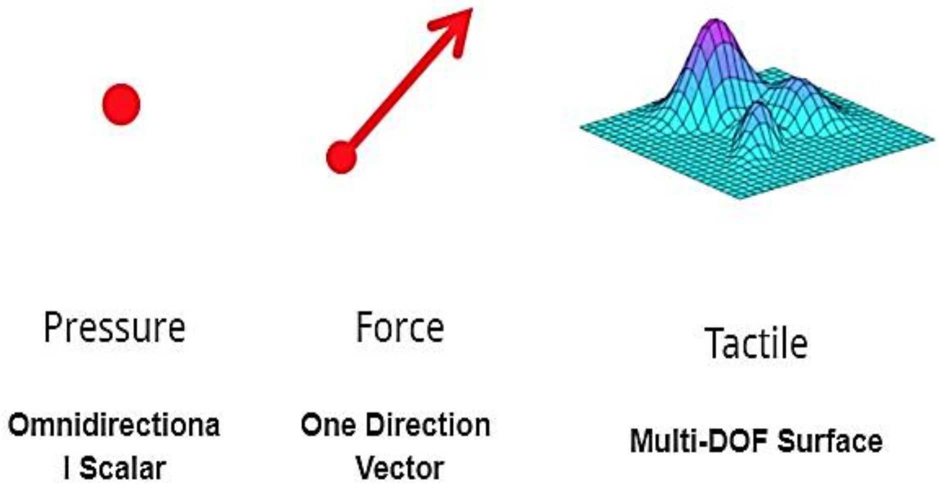


Figure 1. The measurement profiles of the pressure, force, and tactile sensors [13].

Another popular sensor designer in the robotics field is a company called RightHand Robotics, Inc. One of their product lines consist of TakkTile sensors that are low-cost and sensitive, but what makes them interesting is that they are based on the modified barometers [16]. The TakkTile sensors are suitable for flexible design due to their small size. In addition, they perform well in devices that require repeatability and temperature stability. However, the company has announced on their website [16] that their products are no longer available.

2.3 Machine Learning Algorithms in Hand Gesture Recognition

There is no exact distribution of what is and what is not machine learning (ML). In general, however, it can be defined as a computer science area, which intends to make computers learn and analyze data to make predictions and work independently. Most of the ML applications are used for data analyzing and processing, as well as gesture recognition. The computer's main task in ML is first to learn how to perform a task by studying a training set of samples and then perform the same task with the new data it has not seen yet. [17.]

Machine learning offers a variety of useful methods to train the system for gesture classifications. These systems use self-learning algorithms that allow them to generate more accurate predictions every time. This method improves the performance of the system after each training set. To understand how to use different ML algorithms, advanced mathematics and engineering skills are required. In this chapter, the most common algorithms and applications in hand gesture recognition are described.

In the field of HGR, the most common approaches in wearable devices apply to the forearm and wristband. Jiang *et al*. [18] created a wristband for real-time gesture recognition from surface and air gestures. For obtaining the data, the device uses the surface EMG sensors and an inertial measurement unit (IMU). The signal combinations were collected and analyzed by Linear Discriminant Analysis (LDA). Liang *et al.* [19] created a calibration method for their wrist-worn device based on the Support Vector Machine (SVM) algorithm. Likewise, Buyn and Lee [20] used SVM algorithms for HGR in their tactile sensor array that was integrated with a smartwatch. Ahsan *et al.* [21] utilized the Artificial Neural Network (ANN) to classify the EMG signals. Shull *et al.* [22] applied three machine learning algorithms: LDA, SVM, and the K Nearest Neighbors (KNN) to their wristband with the modified pressure sensors. Xiao *et al.* [23] developed an armband with eight FSR sensors and created a pattern recognition method with the Extreme Learning Machine (ELM). Zhang *et al.* [24] developed a framework for continuous gesture recognition for their wristband with four force-sensitive resistors. Benatti *et al.* [25] designed a prosthetic hand application based on an armband with four EMG sensors. They used the SVM algorithm as well.

Table 1. Comparison of gesture recognition studies based on wearable devices.

Study	Sensor(s)	Application	ML Algorithm	Accuracy
Jiang *et al.* [18]	4xsEMG & 4xIMU	Wrist	LDA	92.6%*
Liang *et al.* [19]	4xPressure (capacitive)	Wrist	SVM	90%
Buyn and Lee [20]	4xTactile (strain gauges)	Smart Watch	SVM	97.8%
Ahsan *et al.* [21]	5xEMG	Wrist	ANN	88.4%
Shull and Lee [22]	10xBarometers	Wrist	LDA	98.1%
Xiao *et al.* [23]	8xFSR	Forearm	ELM	92.33%
Zhang *et al.* [24]	4xFSR	Wrist	SVM	95.28%
Benatti *et al.* [25]	4xEMG	Forearm	SVM	98.4%**

*Only the air gestures form Lian e*t al.* [18] has been taken into account in this comparison.
** The accuracy is calculated from an error rate of 1.66% that corresponds the error of 1 in 60 gestures

The studies listed above are summarized in table 1 as a comparison of each's gesture recognition method. The table contains only studies that are based on wearable devices. The five columns, starting from the left include: the authors of the research paper, the sensors used, the application on the human arm, the ML algorithm, and the accuracy rating.

3 Theoretical Basics

This section contains the theories that need to be known for this project: an anatomy and a machine learning section. In order to design an accurate and practical wearable device, as described in the objective, it is essential to understand the structure and the physical phenomenon in the body part that will be measured. Another requirement was to decrease the number of sensors. Therefore, the locations need to be carefully designed with the intention of obtaining accurate data with a limited number of sensors. As a result, an anatomy part is included in this work. Later, in the actual recognition section, the machine learning method plays an important role in the relationship between sensor data and gestures. Since both topics are vast and hold a lot of information, only the essential aspects are summarized in the following paragraphs.

3.1 Gesture Definition

One of the main problems with gesture research is the lack of uniform definitions for commonly used terms. In general, gestures can be defined as non-vocal communication via the physical movement of the face, limbs, or the whole body [1]. The term 'gesture' is used throughout this thesis to describe one hand's posture only.

Roughly, gestures can be classified as dynamic or static contractions. The former is a signal over a time frame, and it has two components: the hand configuration and the location. The configuration is the shape of the hand, including the fingers and the wrist. The location is a combination of the hand orientations and positions in space. On the contrary, a static gesture has no temporal duration. It is a configuration at a specific point in time. It is also important to note that in a static gesture, the length of the muscle and the configuration of the hand do not change. [20.] The muscles and the tendons generate force, for example, when a hand is still but holding an object. As mentioned in the objectives, this work focuses on static gestures.

3.2 Hand Anatomy

A human hand is a complex body part that can perform very fine movements representing one of the most intuitive ways to interact with surroundings. Anatomically, it is described as the terminal, prehensile portion of an upper limb, consisting of the wrist, palm, and fingers [26]. This chapter is mainly based on the *Principles of Human Anatomy* [27, 227-241, 264-281, 304-341, 392-412].

3.2.1 Hand Motion

The configuration of bones, joints, and muscles allow 27 degrees of freedom in movements. This work focuses only on angular movements that can be subdivided into flexion, extension, abduction, and adduction. These movements are illustrated in figure 2. In flexion, the palm bends toward the forearm, decreasing the angle between articulating bones. For the thumb, flexion occurs when the thumb is touching the opposite side of the palm. The opposite movement is called extension, where the hand stretches out from the palm, increasing the angularity. In abduction, the movement points away from the midline: whereas adduction is towards the midline.

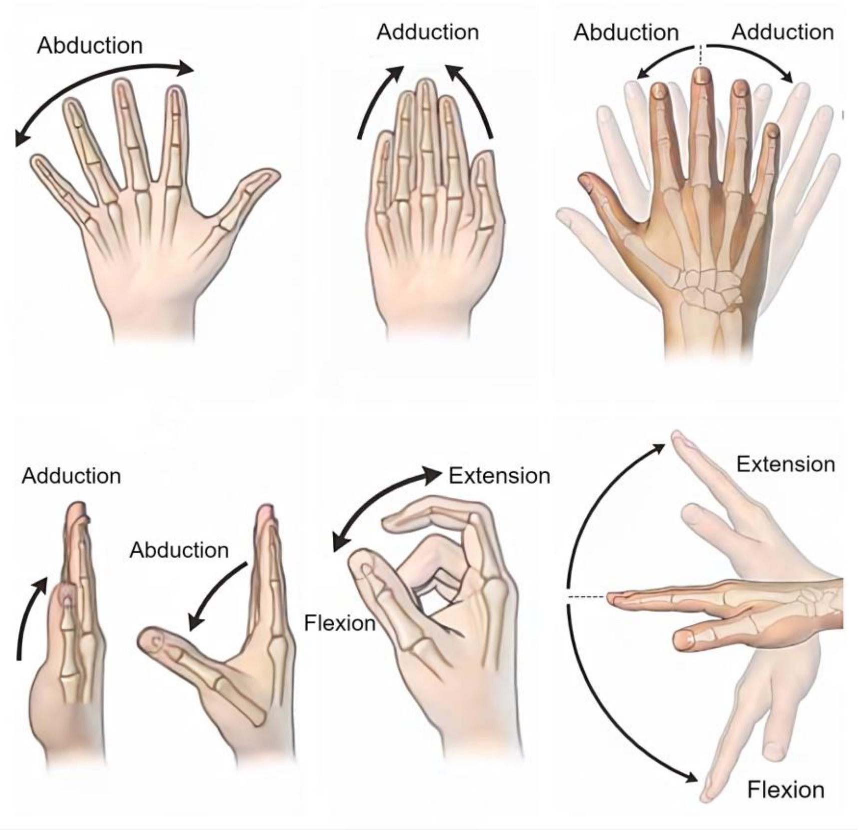


Figure 2. A visualization of angular movements. Reprinted and modified from Gross Anatomy (2011) [28, 353, 369].

There is another angular movement called circumduction that belongs to the dynamic classification group. It is a combination of flexion, abduction, extension, adduction, and rotating motions that produce a circular motion called circumduction. It occurs in the distal end of a body part. For instance, moving the hand in a circle at the wrist joint is a circumduction movement. Despite the fact that it is a dynamic gesture, in this work, circumduction motion is partly used to describe all the vague gestures, such as the intermediate form of flexion and adduction.

3.2.2 Osteology of the Hand

The bony structure of the hand, also called the osteology of the hand, can be broadly divided into three groups: the carpals, the metacarpals, and the phalanges. First 14 phalanges in the hand form the fingers. The thumb, index finger, middle finger, ring finger, and little finger each have three phalanges: a distal, a middle, and a proximal one. The thumb is the only finger without the intermediate phalange.

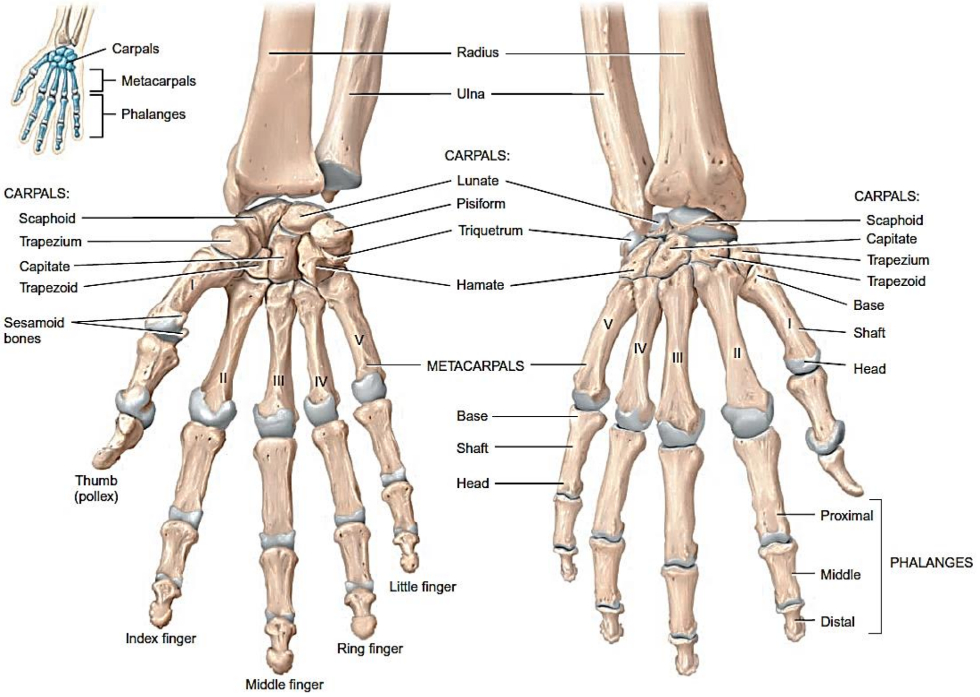


Figure 3. Osteology of the right hand. Reprinted from Principles of Human Anatomy (12th Edition) [27, 240].

Each of the five metacarpal bones make up the intermediate part of the hand called the palm. The wrist consists of a cluster of carpals. These eight small bones are physically located in two rows, which are as follows: the proximal and distal rows. The proximal row is attached to the two big bones called ulna and radius. The former bone, ulna, locates on the little finger's side, and the latter bone, radius, is parallel to the ulna, locating on the forearm's lateral side. The carpals in the distal row are more settled and immobile than the proximal carpals. In figure 3, the bones of the hand are illustrated in more detail.

3.2.3 Muscles of the Hand

The study of pressure sensing applies to skeletal muscle groups in the hand where the muscle tissue is connected to the bones through the tendons, making it possible to move the joints and create the hand gestures. This work focuses on the tendons and muscles in charge of flexing and extending the wrist and fingers. They respectively control the hand abduction and adduction illustrated in figure 4 and figure 5. More specifically, the placements of thenar muscles (explained later in this chapter), extensor pollicis brevis, flexor pollicis longus, extensor digiti minimi, flexor digitorum superficialis, flexor carpi radialis, and extensor carpi ulnaris are all studied [22;27]. These are highlighted in the pictures below.

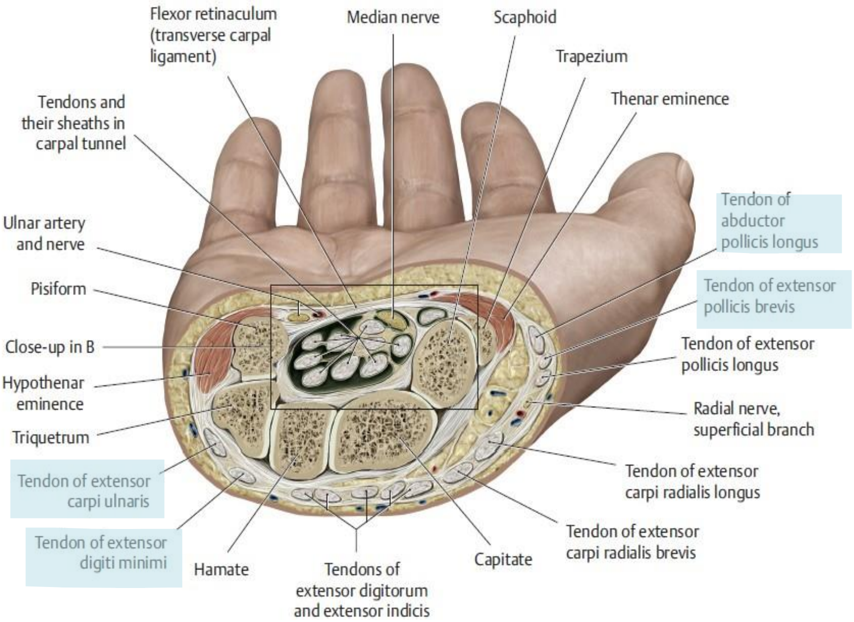

Flexor retinaculum (transverse carpal ligament)
Median nerve
Scaphoid
Trapezium
Thenar eminence
Tendons and their sheaths in carpal tunnel
Ulnar artery and nerve
Pisiform
Close-up in B
Hypothenar eminence
Triquetrum
Tendon of extensor carpi ulnaris
Tendon of extensor digiti minimi
Hamate
Tendons of extensor digitorum and extensor indicis
Capitate
Tendon of extensor carpi radialis brevis
Tendon of extensor carpi radialis longus
Radial nerve, superficial branch
Tendon of extensor pollicis longus
Tendon of extensor pollicis brevis
Tendon of abductor pollicis longus

Figure 4. Cross-section of a right wrist. Reprinted and modified from Musculoskeletal Key [29].

Extensor carpi ulnaris extends and adducts the hand. The thumb and the entire hand extension are controlled by extensor pollicis brevis. In contrast, extensor digiti minimi extends a little finger and the hand. Additionally, extensor carpi ulnaris extends and adducts the hand. The thenar muscle group is a combination of three muscles, that are as follows: the abductor pollicis brevis, flexor pollicis brevis, and opponens pollicis. They are responsible, for example, the adduction, abduction, and flexion of the thumb. They are all located at the bottom of the thumb. In figure 4, only the abductor pollicis longus is seen. However, hand anatomy is a complex field. Therefore, a rough generalization can be done by stating that the abductor pollicis longus usually acts together with the abductor pollicis brevis, allowing the thenar group to be measured at that point.

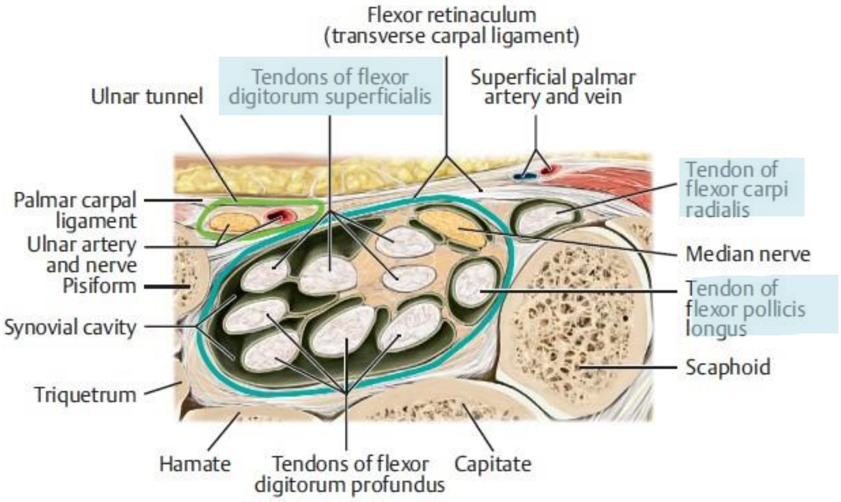

Flexor retinaculum
(transverse carpal ligament)
Tendons of flexor digitorum superficialis
Ulnar tunnel
Superficial palmar artery and vein
Palmar carpal ligament
Ulnar artery and nerve
Pisiform
Synovial cavity
Triquetrum
Tendon of flexor carpi radialis
Median nerve
Tendon of flexor pollicis longus
Scaphoid
Hamate
Tendons of flexor digitorum profundus
Capitate

Figure 5. The carpal tunnel on a wrist. Reprinted and modified from Musculoskeletal Key [29].

The most important muscle in the wrist is the flexor carpi radialis. It flexes and adducts the whole hand at the wrist joint and precisely controls the flexion of the thumb. The flexor pollicis longus is responsible for flexing the thumb. Lastly, the vital place for measurement is the flexor digitorum superficialis due to its high density of tendons. Flexor digitorum superficialis is in charge of flexing all the fingers, while flexor carpi radialis flexes the thumb.

3.3 Data Preprocessing in Machine Learning

It is essential to prepare the raw data before processing or analyzing them with any machine learning method. For large measurement scales and to avoid dependence in different measurement units, the data needs to be normalized. There are several data normalization methods. Nonetheless, the main idea of all of them is to transform all values into the same range for each dimension. Typically, the transformation is done into a smaller range, such as [−1, 1] or [0, 1]. The Min-Max normalization is an excellent choice for treating the ratios of the data values equally. Namely, it performs a linear transformation on the raw data, and therefore, it is used in this work.

The raw data stream is divided into a window where features can be calculated over a series of points. These features represent the data that is organized into columns (v_i'). This method is applied for every feature, and thus, all income values will get a new, scaled value in the range of [0,1]. The idea in the Min-Max normalization [30] is that in column v_i', the smallest value (min_A) of an attribute, A, is transformed into a 0, and the highest value (max_A) of A into a 1, respectively. Then, the rest of the values in the column are scaled between 0 and 1. The Min-max normalization can be written as the following:

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \qquad (1)$$

Here, it can be seen that the Min-Max maps the feature v_i of the attribute, A to the column v_i' in the range $[new\_max_A\,,\,new\_min_A]$. Where new_max_A is the maximum of the normalized dataset and new_min_A is the minimum of the normalized dataset.

3.4 Support Vector Machine Classifier

Finding a good model for gesture recognition is the challenging part. The machine learning models can easily have limited and low accuracy, for example, if the samples are independent from time-series signals or if the dimension is too small [31]. The latter typically leads to overfitting the model. Another big drawback is that when a wearable device is re-used, the sensor locations change meaning the data also changes and thus, requires retraining the model each time. A profound background in linear algebra is essential since it creates the underlying mathematics of all ML models. It also helps to understand the algorithms and how to use them in a project.

This work uses Support Vector Machine algorithms due to its excellent performance in previous research studies, as shown in table 1. The SVM is a classifier derived from statistical learning theory by Vapnik in 1992. It is a classical supervised learning model for both regression and classification analysis. [31] The SVM has a low model complexity in a non-linear classification problem which is mathematically explained in the following subsections. Intuitively the goal is to construct an optimal hyperplane in multidimensional space in an iterative manner to separate the dataset into two classes. Chapters 3.4.1, 3.4.2, and 3.4.3 are based on the two sources: [32;33].

3.4.1 Linear Separation

This chapter starts with the linear separability concept, which is the most straightforward case of the SVM classification. Later, in chapter 3.4.2, the kernel trick is used to map the same inputs into high-dimensional feature spaces. This derives the non-linearly separable cases which occur in most of the practical cases.

The first task is to find a line, called a hyperplane, which separates the two classes accurately. The classified classes are created from the labeled data distinguished with a classifier the positive '+ 1' and negative '-1' data points. These data points are referred to as the support vectors that locate in a vector space. As seen in figure 6, a support vector is a normal vector with several names and is often referred to as a weight vector in ML literature. Nevertheless, due to its perpendicularity to the hyperplane, it can be considered as a unit vector. This perpendicular distance between the support vector and the hyperplane is called a margin. The main idea is to minimize errors in classifications by finding a maximum marginal hyperplane that is far away from any data point. The hyperplane which meets all the above conditions is called an optimal hyperplane, as seen in figure 6.

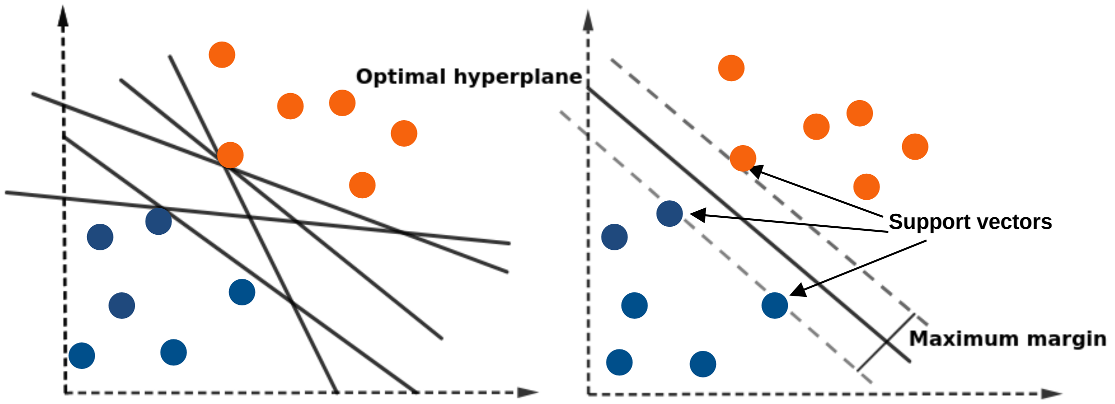


Figure 6. Finding the optimal hyperplane to separate two classes

The desired hyperplane is mathematically defined as following:

$$\boldsymbol{w}^T \varphi(\boldsymbol{x}) + b = 0 \tag{2}$$

where $\boldsymbol{w}$ is the weight vector, $\varphi(\boldsymbol{x})$ is a fixed feature-space transformation, and b is a bias. More specifically, the bias is an unknown constant which is the reason why the hyperplane does not have to go through the origin. This helps fit the model in the best way. The equation above, and especially the term $\boldsymbol{w}^T$ comes from the linear models for regression models.

A training dataset is defined as $D = (\boldsymbol{x}_i, y_i)$. It comprises *N* input vectors $\boldsymbol{x}_i$ ∊ (1, …, N) with the corresponding desired label y_i ∊ (+1, -1). It is used for a dual representation that includes the kernel functions. These are later defined in detailed. However, in the training dataset $D = (\boldsymbol{x}_i, y_i)$, each data points $(\boldsymbol{x})$ are classified according to the decision function $f(\boldsymbol{x})$, that is:

$$f(\boldsymbol{x}) = sign(\boldsymbol{w}^T \varphi(\boldsymbol{x}) + b) \tag{3}$$

This means, the sign of f is positive when the point is correctly classified and negative when classification is incorrect. The equation (3) needs to be calculated for each training sample. The multiplication with the respective class y_i ensures that the term $y_i(\,\boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b)$ is always positive in the case of correct classification. The constraint for this is, $\boldsymbol{w}^T\varphi(\boldsymbol{x}) + b \geq 1$ and $\boldsymbol{w}^T\varphi(\boldsymbol{x}) + b < -1$ for the labelled classes "+1" and "-1". This means, negative and positive samples should fall on different sides of the hyperplane. The data points now satisfy the constraints:

$$y_i(\,\boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b) \geq 1, \qquad \forall\ i = 1, \dots, N \tag{4}$$

This is called the canonical representation of the optimal hyperplane, illustrated in figure 7, satisfying the condition $f(\boldsymbol{x}_i) = y_i$. Now, an important point must be taken into account: there is always at least one active constraint. The hyperplane has always one support vector and when margins are added and maximized, there are at least two of them. To reiterate, two support vectors mean two constraints.

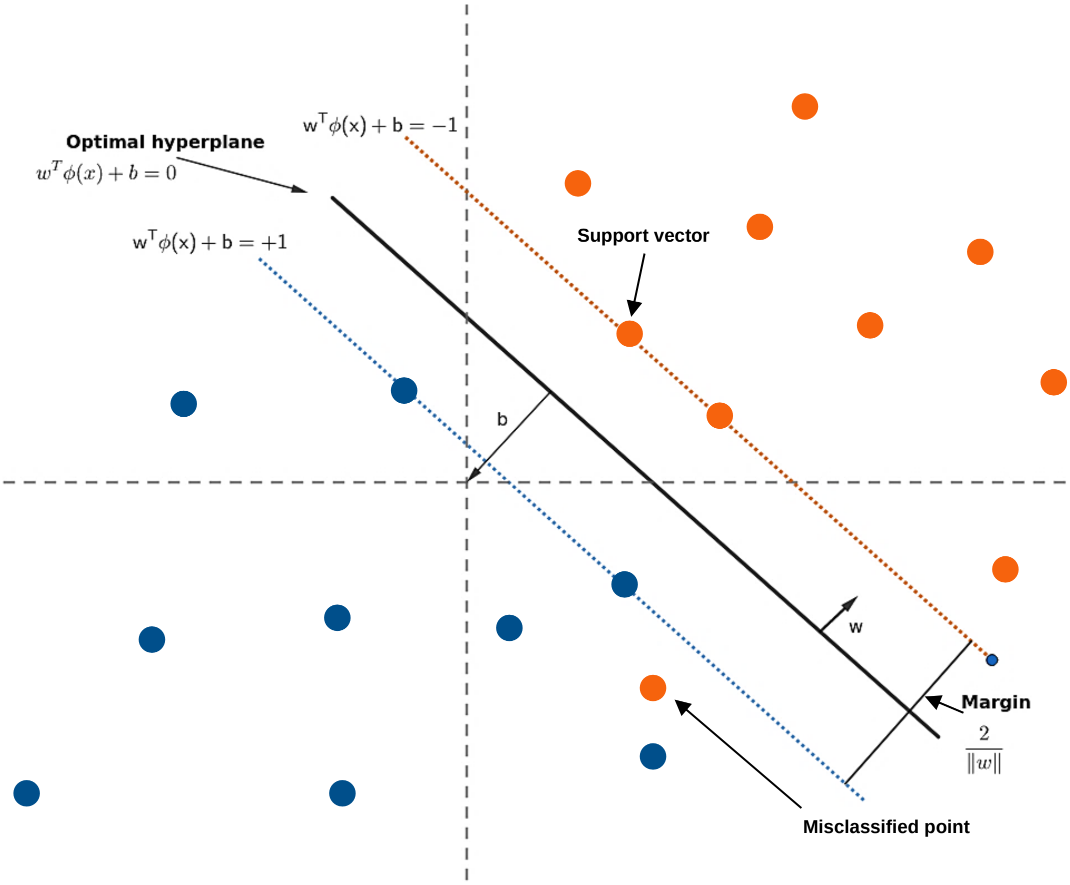

$\mathrm{w}^T\phi(\mathrm{x}) + \mathrm{b} = -1$
Optimal hyperplane
$w^T\phi(x) + b = 0$
$\mathrm{w}^T\phi(\mathrm{x}) + \mathrm{b} = +1$
Support vector
b
w
Margin
$\frac{2}{\|w\|}$
Misclassified point

Figure 7. The canonical representation of the optimal hyperplane.

There can be several different ways to classify the training data set. Thus, the best one can be determined by identifying the one with the smallest generalization error. In SVM, this means finding the maximized margin that has the smallest distance between the maximum marginal hyperplane and any support vectors. This requires knowledge off the statistical learning theory. From geometry, the margin γ can be defined as the following:

$$\gamma = \frac{\boldsymbol{w}^T(\boldsymbol{x}_{pos} - \boldsymbol{x}_{neg})}{||\boldsymbol{w}||} = \frac{2}{||\boldsymbol{w}||} \tag{5}$$

Next, the margin γ needs to be found and maximized for a better performance. To maximize the distance between the support vectors, the parameters $\boldsymbol{w}$ and b need to be optimized. This is a necessary step in the training set to give a lower error rate from yet unobserved data classification. Another advantage lies in the fact that it requires only a few of the support vectors and not all the samples.

To maximize the margin γ and to minimize the length of the vector $\boldsymbol{w}$, knowledge in the field of linear algebra is required. Since maximizing $\frac{1}{||\boldsymbol{w}||}$ is equivalent to minimizing $||\boldsymbol{w}||$, it is also equivalent to minimize $\frac{1}{2}||\boldsymbol{w}||^2$ to solve the optimization problem. Furthermore, this leads to more compact classifiers:

$$\underset{w,b}{\arg} min \ \frac{1}{2}||\boldsymbol{w}||^2 \tag{6}$$

The equation underlies the conditions (10) and can now be called a quadratic programming problem in SVM. Now, The Lagrange multiplier method needs to be introduced to restate the previous problem. The basic idea in Lagrange function is to find the minimum of f under the equality constraint g, which is as follows:

$$\nabla f(x) - \alpha \nabla g(x) = 0 \tag{7}$$

Where α is called the Lagrange multiplier. With the constraints above, the corresponding Lagrange function is now:

$$L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 - \sum_{i=1}^{n} \alpha_i (y_i\ (\boldsymbol{w}^T \varphi(\boldsymbol{x}_i) + b) - 1) \tag{8}$$

Here, α = (α_1, …, α_N)^T and α_i ≥ 0. In order to solve the Lagrangian multipliers analytically, the number of samples must be small. This can be done by maximizing $L(\boldsymbol{w}, b, \alpha)$ for all samples and because there is $(\boldsymbol{w}, b)$ for each hyperplane, the max $L(\boldsymbol{w}, b, \alpha)$ needs to be minimized. Maximizing α can be done by the following: differentiating w.r.t a vector $\boldsymbol{w}$ and differentiating w.r.t a scalar b, and both need to be equal to 0.

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \varphi(\boldsymbol{x}_i) \tag{9}$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i \tag{10}$$

Now, $\boldsymbol{w}$ and b from $L(\boldsymbol{w}, b, \alpha)$ can be eliminated under the mentioned conditions. This leads to the dual representation of the maximum margin problem. However, it also needs to be maximized. Parameters $\boldsymbol{w}$ and *b* are replaced by the Lagrange multipliers and the max $L(\boldsymbol{w}, b, \alpha)$ is represented by $\tilde{L}(\alpha)$. Now, the Lagrange function is as follows:

$$\tilde{L}(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{11}$$

The equation (11) is under the constraints below:

$$\alpha_i \ \geq 0, \quad \forall\, i = 1, \dots, N \tag{12}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{13}$$

In the next step, a kernel function needs to be defined. This is presented in next, in chapter 3.4.2.

3.4.2 Non-Linear Separation

When the training data is non-linearly separable, the previously applied hyperplane separation method needs some tricks and procedures. The basic idea is to use a function, the Kernel method, to transform the vector space into a higher-dimensional space. The Kernel allows the training data to be separated linearly. Thus, the canonically separating hyperplane can be determined similarly to the linear case. However, during the retransformation into the original space, the hyperplane becomes a non-linear interface. The transformation is shown in figure 8 below.

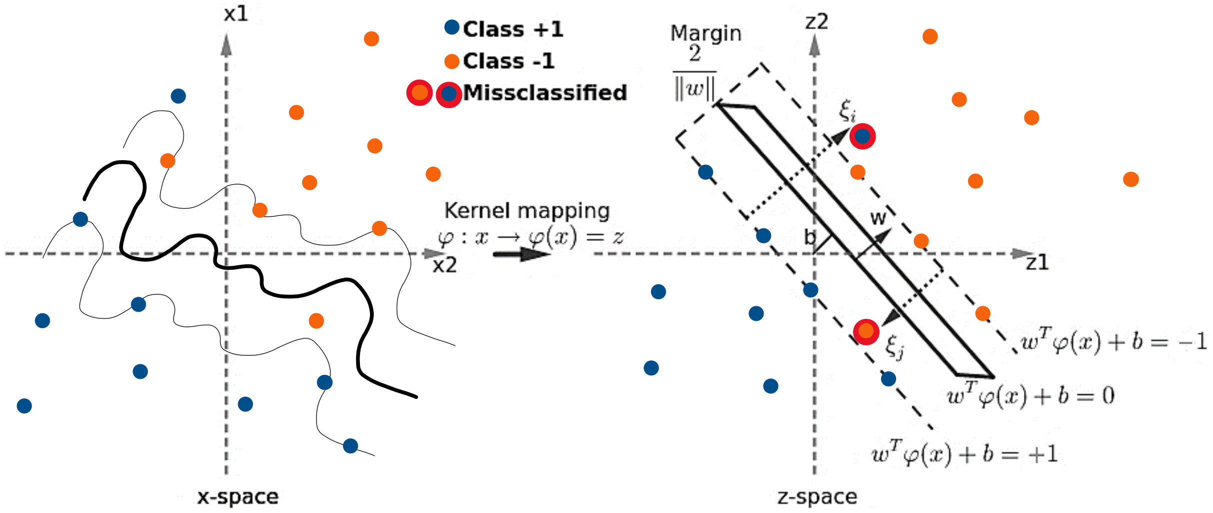


Figure 8. Non-linear mapping from two-dimensional input space into a linear separable feature space. Modelled from Khan *et al.* [34] research article.

On the left side in figure 8, the binary pattern classification lies in input space. As seen, it is complex and usually the classification is difficult to execute. After the Kernel mapping, the graph is now in high-dimensional feature space. As seen on the right side in figure 8, the data classification occurs as linearly separable.

With mathematical methods, subsequently rescaling is done as follows: $\boldsymbol{w} \to k\boldsymbol{w}$ and $b \to kb$. This needs to be done to ensure the distance from any point $\boldsymbol{x}_n$ to the hyperplane is unchanged. The kernel function used here, is defined as:

$$k(\boldsymbol{x},\boldsymbol{x}') = \left(\varphi(\ \boldsymbol{x})^T, \varphi(\ \boldsymbol{x}')\right) \tag{14}$$

Now, the kernel function can be used for substituting $\boldsymbol{w}$ in equation (8). This leads to:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{15}$$

In the hand gesture recognition, it makes more sense in practice to develop an algorithm that allows a certain number of outliers. Therefore, misclassifications need to be allowed. The Karush-Kuhn-Tucker (KKT) method fits well for this problem. It includes a Slack variable ξ_i, called Xi, that is needed to the constraints of the optimization problem by allowing the outliers. Also, KKT includes a regularization parameter C that is a positive error weight value. The parameter C determines the trade-off between the maximal margin and training error minimization. The constraints are now:

$$0 \ \leq \ \alpha_i \leq \ C \tag{16}$$

$$y_i(\ \boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i \tag{17}$$

Both for $i = 1, \dots, N$. The constraints (15) are also known as box constraints. The role of the ξ_i is defined as the following: $\xi_i = 0$ occurs when the training data is correctly classified, data inside the margin is characterized by $0 < \xi_i \leq 1$ and finally, $\xi_i > 1$ when the point is on the wrong side of the hyperplane.

3.4.3 Advantages and Disadvantages of the SVM

Finally, all pros and cons of the Support Vector Machine classifier are summarized in this chapter.

Advantages:

1. Capable of fast classification
2. Requires less data points
3. Scales the data well to high dimensional space
4. Decent performance for unknown datasets.
5. In the field of SVM, the generalization occurs in practice and therefore, the risk of over-fitting is lower.

Disadvantages:

1. Sensitive to noise
2. Difficult to choose the most suitable kernel function
3. Problematic for large datasets or at least it requires long training time
4. The SVM model is not suitable for fine-tuning parameters > a new training is required each time

In summary, the Support Vector Machines only have a few drawbacks that would negatively impact this work. It is capable of high-dimensional pattern recognition with fewer samples representing the key concept of gesture recognition. With the SVM, the raw sensor signals from the non-linearity profiles can be transformed into a gesture.

4 System Architecture

There is a trade-off between practicality and accuracy when designing wearable devices based on biomechanical measurands. Using multiple sensor combinations in suitable locations provide more accurate measurements with higher amounts of hand gestures; however, these devices become larger and uncomfortable to wear in the applications. At the moment, there are no commercially available wearable pressure sensing devices for accurate real-time gesture recognition.

4.1 Sensor Array

In chapter 3.2, it was shown that the signals around the wrist are precise to sensitive sensors due to the high density of tendons on a wrist. It is also known that the tactile sensors can accurately measure the pressure between the wrist and sensor surfaces and this way, map the hand and finger movements [3]. Pressure sensors can generally detect both dynamic and static contractions and they are robust to noise which makes them good candidates for HGR devices. Therefore, this work aims to meet all above mentioned requirements and create a tactile sensor array form five modified barometric sensors.

4.1.1 Miniature Pressure Sensor with Microelectromechanical System

A pressure sensor can be defined as a device that converts an obtained pressure signal into an output electric signal. There are five types of pressure sensors: piezoresistive, capacitive, optical fiber, resonant, and piezoelectric ones [35]. For the wristband, a miniature sensor model MPL115A2 [36] from NXP Semiconductors Inc. was chosen. It is a piezoresistive pressure sensor with a Microelectromechanical System (MEMS) technology. The MEMS pressure sensors can be used as barometers as well. Therefore, the MPL115A2 is called as a miniature barometer [36]. In figure 9, the MPL115A2 sensor's appearance and block diagram are shown.

The modern MEMS pressure sensors have low power consumption [36] and are very small. They are also resistant to sweat, allowing them to provide relatively stable signals. Additionally, they are inexpensive components and practical due to their compatibility with standard printed circuit boards. The MPL115A2 sensor has a small package size, 5 x 3 x 1.2 mm, which makes it fit well for sensing a complex surface, such as the wrist. The sensor consists of a MEMS diaphragm with a Wheatstone bridge, temperature sensor, high-quality instrumentation amplifier, multiplexer, an analog-to-digital converter, and an Inter-Integrated Circuit (I^2C) bus interface [16;36]. All the listed features are covered with a metal case, as seen in figure 9.

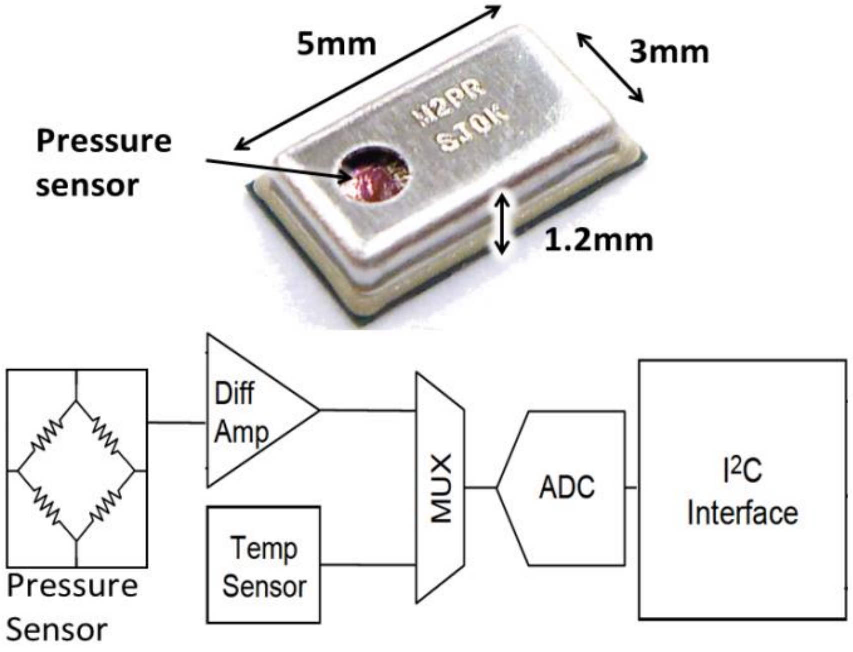


Figure 9. The MPL115A2 sensor's appearance and its block diagram. Reprinted from Tenzer *et al.* [14,1] synopsis.

Due to the integrated ADC, the sensor gives the digitized temperature and pressure outputs via its I^{2}C port. The MPL115A2 sensor perceives a pressure signal in range 50 to 115 kPa [36]. For the mentioned features, the MPL115A2 model is an excellent choice for cost-efficient wearable applications.

4.2 Circuit Design

A sensor reduction is one of the main objectives concerning the design of the wristband. The advantage is that the smaller number of sensors makes data processing more resource efficient. However, this is a significant risk in terms of accuracy, since each sensor must perform in a stable and error-free manner. As mentioned in the previous chapter, the MPL115A2 sensor uses the I^{2}C bus interface. The circuit design takes into account that each sensor has the same predetermined I^{2}C address. Therefore, different protocols are needed to successfully read the data from all the sensors via the same I^{2}C bus. Figure 10 shows the circuit of the MPL115A2 sensor.

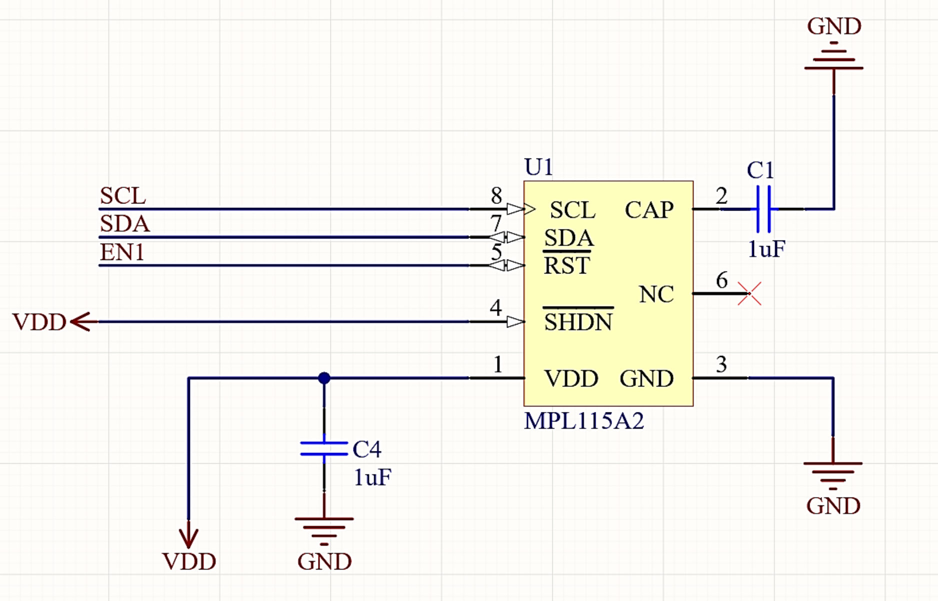


Figure 10. The MPL115A2 pressure sensor circuit.

Each sensor has two capacitors. One is connected to the ground from CAP, and one is for the VDD power supply connection. The recommended value for each capacitor is 1 µF [36] used in this work. Pin 3 is connected to the ground, while pin 4 is connected to VDD for normal operation, and pin 6 is not used in this work. Pin 5, 7, and 8 are all connected to the Arduino NANO 33 BLE. The Serial Data (SDA) input/output line and the Serial Clock Input (SCL) are both connected with the pull-up resistors to disable the I^{2}C communication [36]. In figure 11, the I^{2}C communication on the Arduino NANO 33 BLE's side is shown.

The Arduino NANO 33 BLE is a microcontroller board, which is chosen to be used due to its simple and user-friendly interface and integrated Bluetooth feature that allows wireless communication. There are five sensors in this work separately connected to the Arduino's input/output (I/O) pins. For the I^{2}C communication, there are two pull-up resistors with recommended values of 4.7 kΩ [36]. This method enables the Arduino NANO 33 BLE to share the I^{2}C bus with all the sensors. Thus, only four lines are necessary to communicate with all sensors: two communication and two power lines. The schematic of the Arduino NANO 33 BLE is illustrated in figure 11 below.

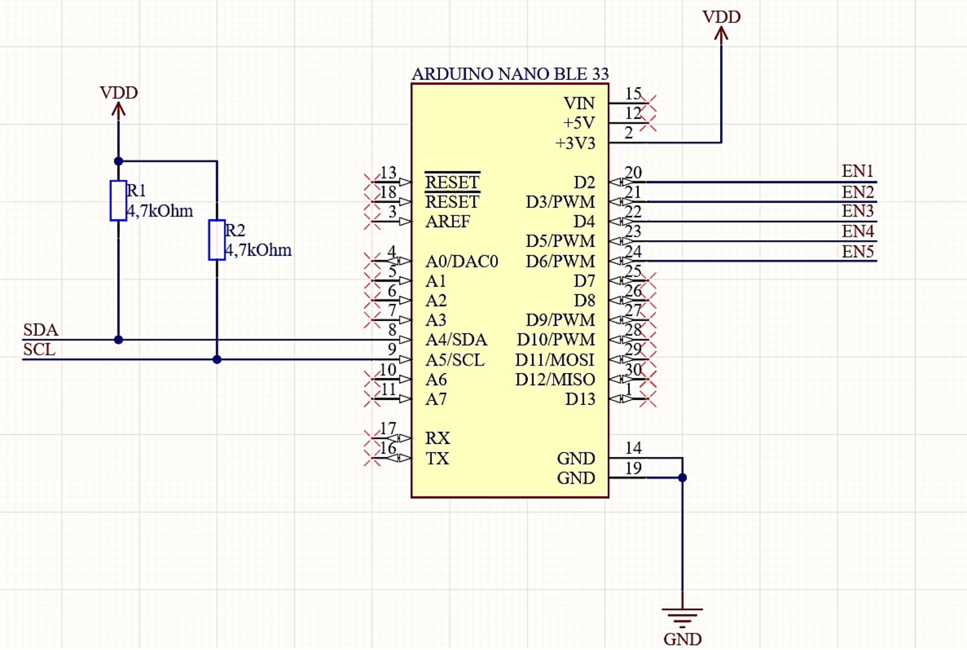


Figure 11. The Arduino NANO 33 BLE circuit.

The overall circuitry design includes five times the MPL115A2 circuits and one Arduino NANO BLE 33 circuit. The schematic is attached to Appendix 1.

4.3 Sensor Array Sampling Rate

The main challenge concerning performance is the sensor data conversion time, which is 1.6 ms [36]. A typical serial approach has to go through each sensor in the array and wait until the data is available. Therefore, an algorithm from Tenzer *et al.* research [14] has been used to calculate a faster sensor array sampling rate. The study [14] compared both serial and double-loop methods, where the latter could provide a better performance. The double-loop method is based on the waiting time used to communicate a start-conversion command to all the sensors. The algorithm is interested in the speed of the sensors, microcontroller, and bus connection. After alternately reading the data from each sensor in a suitable time interval, the sensor array sampling time can be calculated from the following equation:

$$\left(\frac{2C_b + S_b + R_b}{bus\ speed}\right) \cdot N + T_c \tag{18}$$

Here C_b is the bits required to command the microcontroller's sensor selection, S_b is the bits required to command data conversion, R_b is the bits required to read the data, T_c is the conversion time of the sensors, and N is the number of sensors in the device. The bus communication speed is the maximum operating value in kHz for the I^{2}C interface. [14.]

4.4 System Layout

The general distribution of the system is illustrated as a block diagram in figure 12.

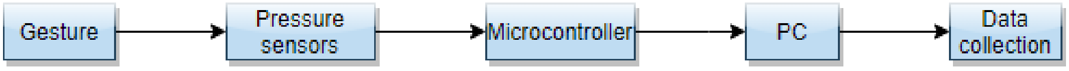


Figure 12. The block diagram of the architecture system.

When a gesture occurs, the device records the wrist tactile profile with five modified barometric pressure sensors. All five recorded pressure sensor data values are read on the Arduino NANO 33 BLE microcontroller board via the I^{2}C bus communication. The code is written in the Arduino Integrated Development Environment, also known as Arduino Software (IDE). Then, the sensor values are transmitted via Bluetooth or Miro USB to the local PC, where a gesture database is created.

5 Wristband Design

This chapter focuses on designing a complete and functional prototype that implements the requirements of the previous chapters. Additionally to the earlier prerequisites concerning the objectives, the design focuses on creating a practical, ergonomic, and wearable prototype. A device that goes around the wrist should have a natural and comfortable approach for HGR applications.

5.1 Wristband Materials

The prototype is mainly used for research purposes. Therefore, the wristband material needs to be durable and easy to build and use. The wristband requires a flexible and rigid material to fit properly on the curved surface to fulfill the conditions. The outer layer around the electronics should ensure a tight contact between the sensors and the wrist. Therefore, a custom faux suede and Velcro strap combination is used here. The material is low-cost, easy to attach, and helps to hold the sensors in place. The printed circuit board (PCB) should be manufactured on a flexible polyimide due to its bending feature. Also, additional silicone is used for the sensors' rubber casting. For over-molding the liquid rubber, the object is to print the 3D-printed models using Polylactic Acid (PLA) plastic. The rubber casting method is explained more specifically in chapter 5.2.3.

5.2 Placements of the Sensors

The information from anatomy and other studies can be applied in the acquisition of sensor models and determine the sensors' locations. The number and location of the wristband sensors were chosen, considering the fact that the classification accuracy can still be 90% with only five sensors [22]. Therefore, in this work, the prototype measures the contact pressure distribution with five modified tactile sensors that are later classified into hand gestures.

A high density of tendons underside the wrist makes it the most important source for the measurand. The sensors are strategically placed on estimating physical pressure changes from a surface. The sensors are located as follows: three sensors are located in line, close to each other on the underside of the wrist. The two other sensors lie on both sides of the wrist. Studies [22] have shown that measuring tendons located on the posterior side reflects the tendons' movements on the anterior side. This information is mostly redundant, and therefore no sensors were placed on the posterior side. In summary, only the muscles and tendons mentioned in chapter 3.2.3 were directly measured. The sensor placements on the wristband are illustrated in figure 13.

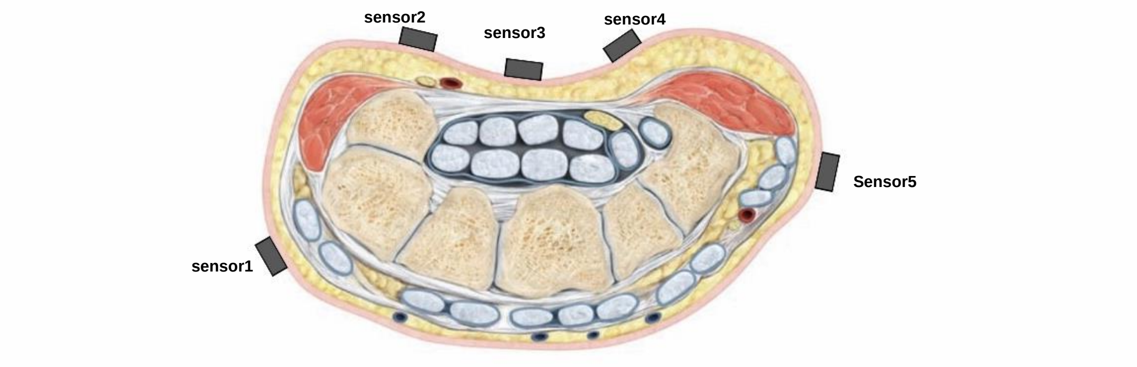


Figure 13. Hand-sided view of transverse section with five MPL115A2 sensors. Printed and modified from Principles of Human Anatomy (12th Edition) [27, 403]. Not in scale.

The sensors in the wristband have been named to distinguish the data values in the following process. Figure 13 shows that *sensor1* represents the left side sensor, closest to the little finger, on the underside locating *sensor2*, *sensor3,* and *sensor4*, and on the right side, close to the thumb, is *sensor 5*.

5.3 Printed Circuit Board Design

Hand movements change the shape of the wrist surface, making it challenging to place the sensor. Therefore, the wristband needs to be compact and tight enough to execute an accurate hand gesture classification with five pressure sensors. Because of this, the board should contain all the electronics used in the wristband. It connects all the five pressure sensors with the Arduino NANO 33 BLE and other components, namely resistors and capacitors. The printed circuit board, seen in figure 14, was designed with Altium. The 3D picture of the designed PCB is attached to Appendix 2.

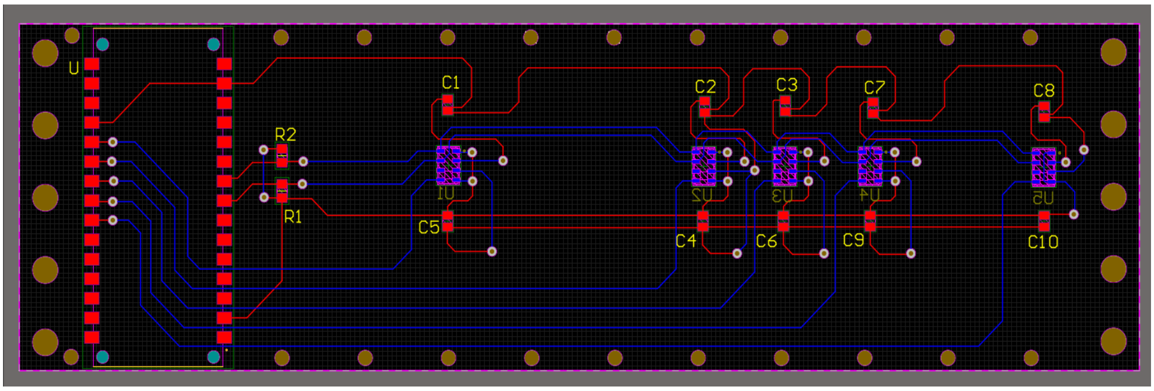


Figure 14. A two-dimensional layout of the printed circuit board. Designed with Altium.

Due to the fact that the Arduino NANO 33 BLE is not bendable and the wrist itself is boney, the sensors are placed slightly above the centerline, allowing the wristband to fit better around the wrist. They are also located on the backside of the wristband, as seen in figure 14. As mentioned in chapter 4.2, each sensor has two ceramic capacitors, and since all sensors are connected in series, they share two pull-up resistors for the I^2C communication. The microcontroller on the Arduino NANO 33 BLE can be powered from a local PC via the Micro USB cable connection since it has a built-in voltage regulator. In this project, the microcontroller runs at 3.3V.

The manufacturing of the PCB took place at PCBWay, in China. In table 2, the PCB specifications are shown in detail.

Table 2. The PCB specifications.

Type	Flexible Circuit Board
Board Size	153 x 45 mm
Board Thickness	0.1 mm
Base Material	Copper
Layers	2Layer
Solder mask	Yellow
Material	Polyimide Flex
Manufacture	PCBWay - China

Concerning the wristband design, the PCB requires an extra fabric to hold it still. The holes on the upper and bottom sides were designed for sewing purposes. That way, the board can be attached to a fabric. Respectively, the holes on the side are for wires to connect the PCB to their ends.

5.4 Mechanical Design

The prototype design corresponds to already existing ergonomic shapes, which contribute to the practicality of the device. Moreover, the wristband should provide reliable mechanical support for the sensors to ensure the sensors are correctly attached to the wrist.

The board's width is determined by the Arduino NANO 33 BLE board size, that is a 45x18mm. A smaller model would also be suitable; nevertheless, the Bluetooth feature allows testing the data's wireless transmission with an easy-to-use interface. The custom two-sided printed circuit board's size and weight are, without the components, 153 × 45 mm and 0.7 g, respectively. In figure 15, the final result of the PCB is shown.

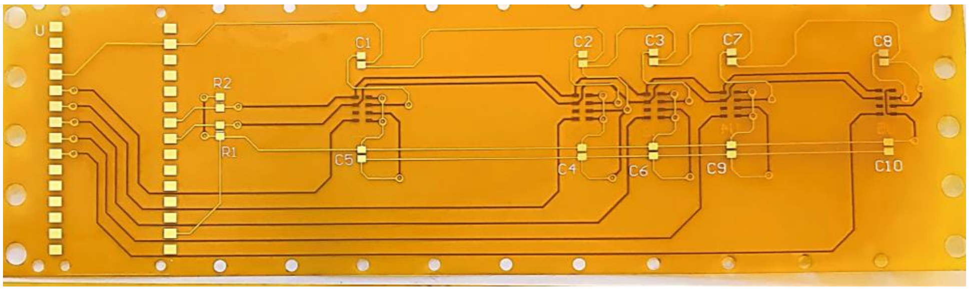


Figure 15. The manufactured Printed Circuit Board.

The tendon and muscle movements apply motion and pressure to the sensors. Therefore, the sensors are integrated with the PCB to stabilize their positions. The components are soldered on the flexible PCB with standard integrated circuit surface-mount techniques. This circuit board is suitable for over-molding with liquid polymer that is explained in chapter 5.3.1. It is also important to mention that the wristband is ideally made for a wrist with a circumference of 17 cm, and therefore, the use of the wrist strap is limited. However, the connection holes on the side allow an attachment of a slightly larger circumference, maximum for approximately 20 cm.

5.4.1 Rubber Casting Process

The sensors' function is to identify the contact pressure profile between the device and the wrist surface. For this, each sensor requires a high density. Therefore, the MEMS sensors need to be encased in liquid rubber to give better measurements and provide a stable contact surface. With rubber casting, the sensor can convert a physical measurand from the tendons and muscles to an electric signal. According to studies [14], the best result is expected to be obtained using liquid urethane and vacuum degassing. However, small quantities of liquid urethane are rare on the market, and the vacuum degassing method is too expensive for the realization of this project. Instead, soft silicon was used.

A ventilation hole is only with a 1 mm diameter, and the casting is carried out at normal atmospheric pressure. This will leave air caught in the metal case's ventilation hole. The rubber's pressure is too small to change air volume between the rubber and the sensor, and therefore, the trapped air would cause a very low sensitivity. Thus, the air needs to be removed from inside the case. As mentioned above, one way to implement the air remover method is to use vacuum degassing. However, it is an expensive method and thus would not be used for a cost-efficient prototype. Another way would be the removal of the metal case, but this creates a high risk of damaging the sensor itself. The most suitable approach would be to use a small syringe to spread the rubber into the metal case through the ventilation hole before it cures. This method allows the surface contact pressure to be transmitted through the rubber to the sensor.

First, the soldered sensors were isolated from the printed circuit board by a 3D printed mold. Two different sized molds are used in the process. The dimensions for the sensor units located on the side are 7 x 8 x 2.5 mm, and for three sensors located in the middle, the sizes are 30 x 8 x 2.5 mm, respectively. The molds and rubber casting process is shown in figure 16. The thickness affects the sensor's sensitivity, and therefore, the minimum size was used. The rubber thickness of 2.5 mm had an excellent performance in the tests. Consequently, it was chosen to be the final thickness.

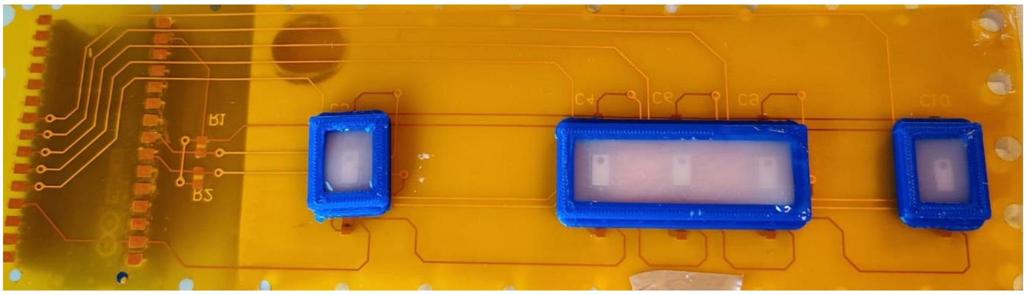


Figure 16. The rubber casting process.

Extruding the silicone inside the metal housing is more complicated than expected before. Often, big air bubbles remain in the sensor, which in turn impairs pressure detection. As a result, the method required three PCBs, where the last one showed good performance.

5.4.2 Sensor Performance Evaluation

A sensitivity test determines each sensor's new sensitivity after the rubber casting process to ensure that the sensor is sensitive enough for HGR. First, the sensors are measured without applying any load. Then, the weight is applied incrementally above the ventilation hole. At the same time, the sensor response is recorded and saved to a local PC. The modified MEMS sensors were tested, and it was seen that they have good temperature stability, high sensitivity, low hysteresis, and good linearity, as was shown in the study [14]. These qualities also apply to tactile sensors, which is why these pressure sensors can be considered as tactile sensors.

However, the sensors should be sensitive enough to recognize the gentle forces that occur during the static gestures. Therefore, the wristband is attached to the wrist while the hand is doing a circumduction movement slowly. At the same time, the pressure signals are captured and analyzed. Figure 17 shows the signals from each sensor during the circumduction movement. The x-axis represents time in milliseconds, and the y-axis represents the sensor value.

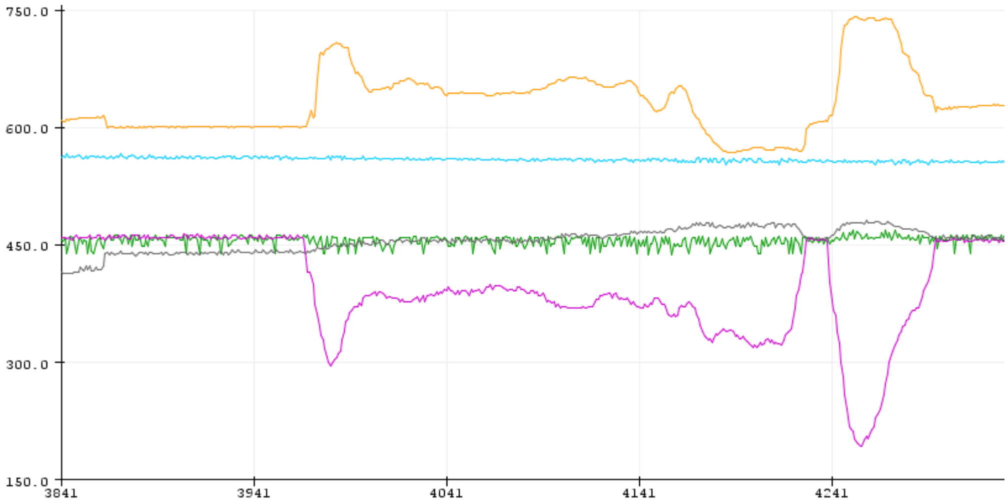


Figure 17. Sensor output values versus time for the sensors with a silicone layer of 2.5 mm thickness in the circumduction movement. The orange color represents sensor1, the blue is sensor2, the grey is sensor3, the green is sensor4, and the pink color represents sensor5.

As seen in figure 17, the system with rubbers of 2.5 mm thickness is stable enough to recognize the pressure changes.

5.5 Dataset

This chapter describes the conditions and characteristics under which the measurements are performed. It includes the trial set, where gestures for recognition are introduced. Here it is also described how the dataset is created.

The experimental test involved one subject with a wrist circumference of 17cm. The participant sat comfortably in a chair, leaning against a fixed seatback. Then, the participant was asked to place the arm on the chair's armrest, forming a 90º angle between the forearm and the armrest. The wristband (figure 18) was then placed on the participant's right wrist, and the power cable of the wristband was connected to the local PC. The wire has no adverse effect on this test. Then, the participant was asked to perform a trial set and repeat it ten times.

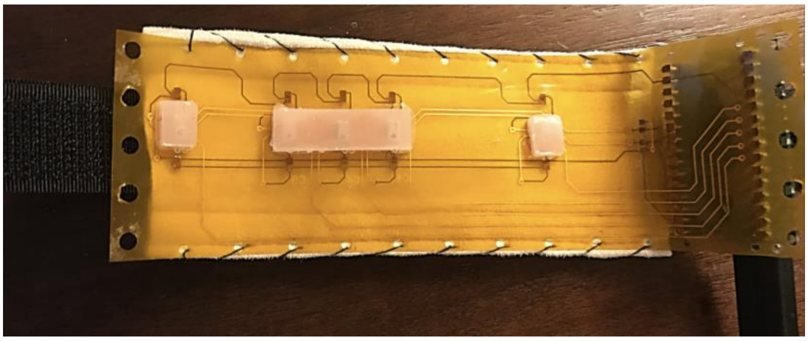


Figure 18. The prototype of the wristband.

The testing trial consists of ten static gestures. They are the followings: Fingers Adduction (1), Wrist Flexion (2), Wrist Extension (3), Wrist Adduction (4), Wrist Abduction (5), Fingers Abduction (6), Clenched Fist (7), Holding a Phone (8), an OK Sign (9), and a Shaka Sign (10). These gestures are illustrated in figure 20. A hypothesis is that these gestures have different motion states, denoting that they are easy to distinguish from each other.

The trial was performed as follows: fingers adduction, wrist flexion, wrist extension, wrist adduction, wrist abduction, fingers abduction, clenched fist, holding a phone, the OK sign, and the shaka sign. Each gesture was recorded for approximately two seconds, and the transformation between the gestures needed two seconds. For creating a good base with enough data, the general rule of thumb is used. It means that each additional variable requires 20 subjects. If the total number of variables is five, then the minimum sample size is a hundred. It means a trial needs to be repeated ten times.

To reiterate, the signals were simultaneously recorded from the five pressure sensors. The data values were then collected straight in an excel spreadsheet. The rows represent individual data points, and the six columns represent the features and a class. For machine learning, the data representation must be a table, and therefore, the files were saved as a .csv format. This layout is now a two-dimensional numerical array, also known as matrix.

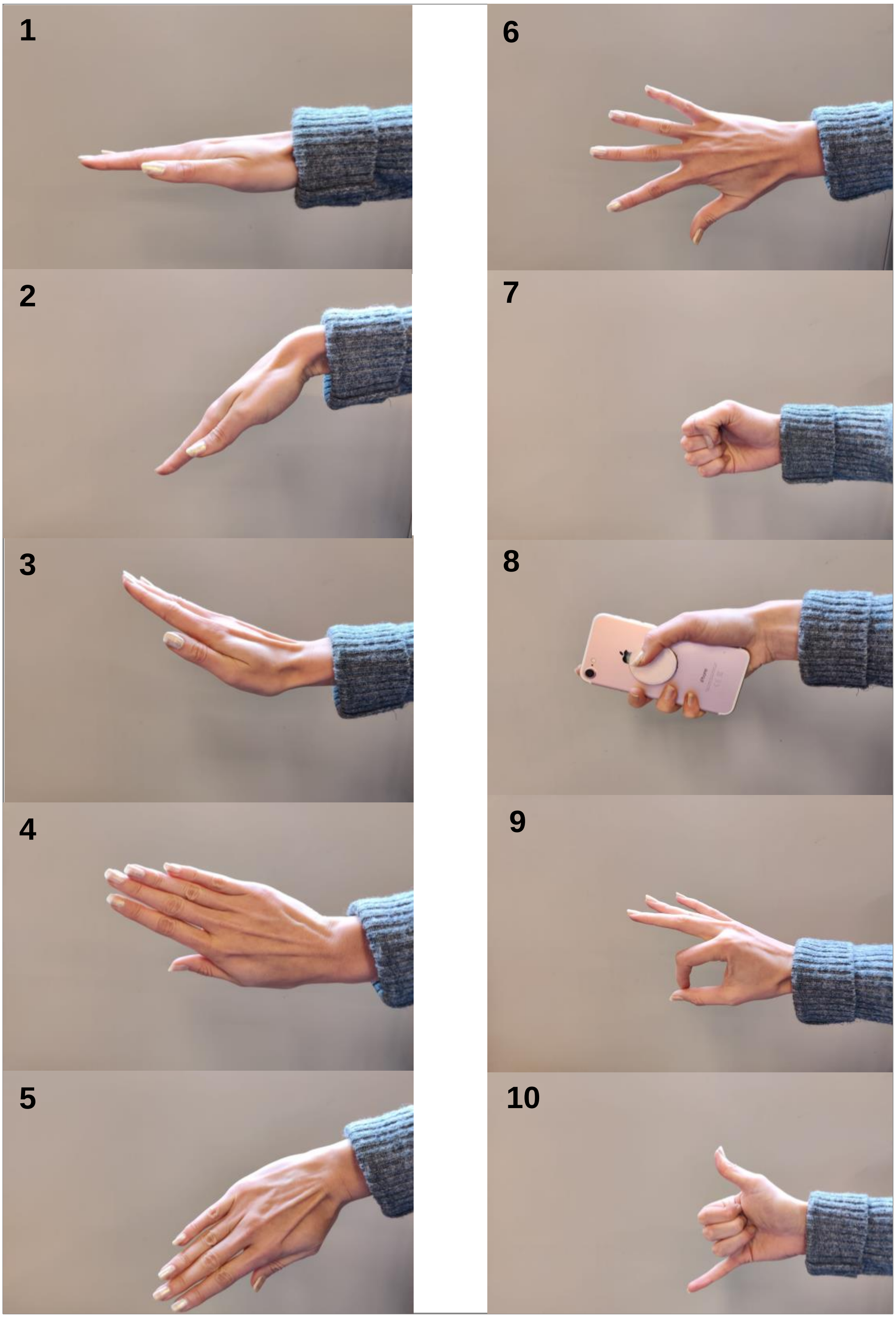


Figure 19. Performed hand gestures.

Moreover, integrating the edge cases in the database is essential. It was done by recording four trials with the exaggeration and the light postures of each gesture. For example, when measuring the wrist extension gesture, the hand was measured in a similar position six times in a row. Then, two times the hand's position was left deficient in flexion, and the values were recorded. In the two-last recording, the hand was in the excessive deflection. These events create the edge cases for the hand extension gesture. Sometimes the subject performed a circumduction gesture, and thus, an extra round was performed. Therefore, the trial was done 11 times in total, and the best ten gestures, defined by the recorder, were stored in the database.

6 Testing the SVM Algorithms

The device is intended for real-time applications. Therefore, an excellent computational efficiency is essential. According to comparison table 1, the Support Vector Machine algorithm performed well with good accuracy. The SVM is also easy to use for training. However, it requires a lot of empirical testing to get the best model for a project.

6.1 Gesture Recognition Framework

In the following, the most important processes of the framework are described, starting from the inclusion of the data up to the result of the full training. First, the database was created with 10000 data points, as was explained in chapter 5.5. The database is then split into training and testing datasets to train and evaluate the models, respectively. In order to find the model's optimized hyperparameters, a grid search algorithm is used with cross-validation. Finally, the performance is evaluated. All the steps are then repeated with tuned parameters in order to find the most suited model to classify the gestures. The object framework is illustrated with a flowchart in figure 20.

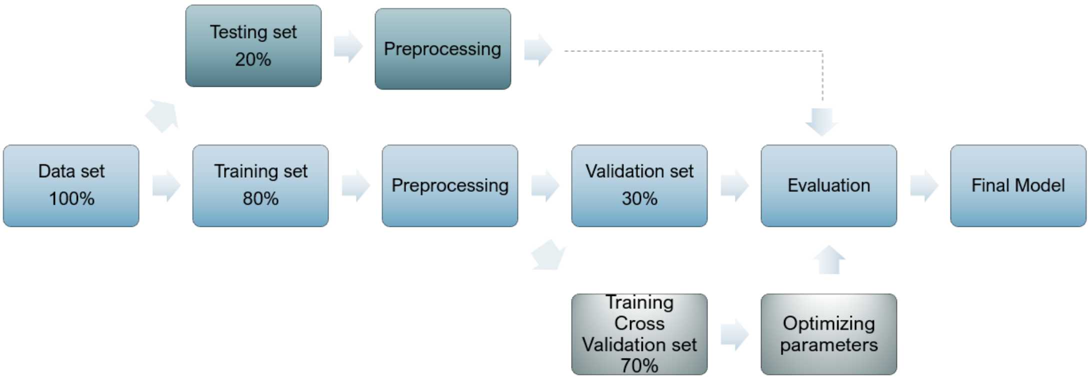


Figure 20. A Flowchart of the data classification and evaluation process.

First, the database was created with 10000 data points. Then the database is split into training and testing sets to obtain independent results. Usually, the features are extracted from the database before using a classifier. Reducing the number of data features allows us to work with large datasets by reducing a model's complexity. Hence, it enables the SVM algorithm to train faster and to avoid overfitting. This project includes a detailed study in sensor placements, and the sensor number is already reduced. Therefore, all the sensors, which are the features, are used.

6.1.1 Data Splitting

The data is split into two parts: training and testing datasets to obtain independent results. The dataset's division is as follows: a test set is 20%, and a training set is 80%, which is executed with a random seed. The training set's primary function is to train the classifier that later the testing set tests. The final results show the performance of the classifier to properly differentiate the gestures from the test dataset.

6.1.2 Data Preprocessing

Before using the Support Vector Machine, the data needs to be centered and scaled. It is done separately to the testing and training datasets to avoid data leakage, which occurs when information about the training dataset corrupts or influences the testing dataset. The data is normalized by scaling the measurement values from 0 to 1. The normalization was done with the equation (1), which was presented in chapter 3.3.

A feature histogram (figure 21) is created to indicate the statistical frequency distribution of the values from the sensors with respect to the feature values. The histogram shows a clear differences in the values between the sensors' performance.

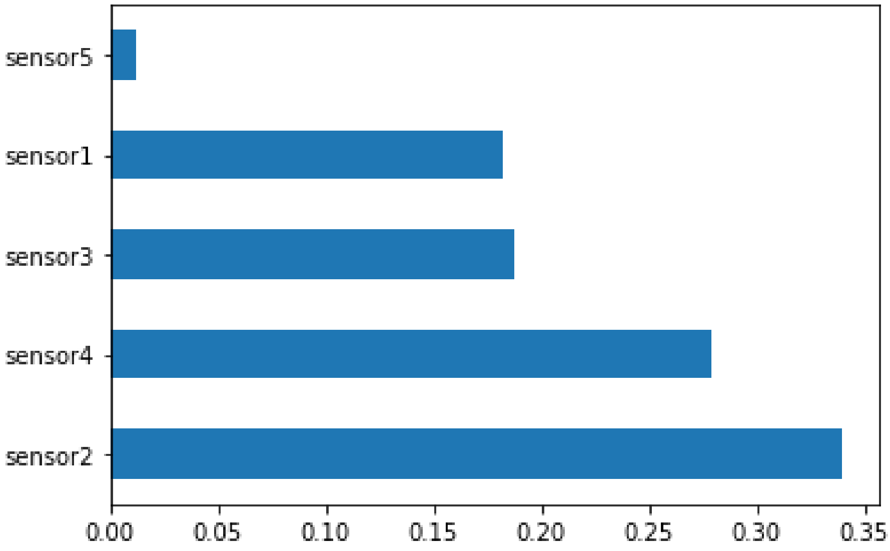


Figure 21. Feature Histogram of the sensor. The X-axis describes the explained variance, and the Y-axis represents the features.

As shown in figure 21, sensor5 has a weak performance. Each sensor's effect on gesture recognition is distributed as follows: sensor5 approximately 1.2 %, whereas sensor1 has 18.2 % and sensor3 has 18.8 % impact, sensor4 has the second most with 27.9 %, and the best-performed feature was sensor2 with 33.9% impact.

6.1.3 Feature Selection

This process selects the features with strong performances. As seen in figure 22, sensor5 has a low performance, and thus, the use of all features as input would provide weaker accuracy in the results. Therefore, only the features with strong performance in differentiation should serve as input.

Before leaving out sensor5, one more analysis needs to be done. The feature value differences of all the classes are illustrated graphically with the scatter plots. When comparing, for example, sensor1 and sensor2, there are exact distributions between all the gestures. This is shown in figure 22.

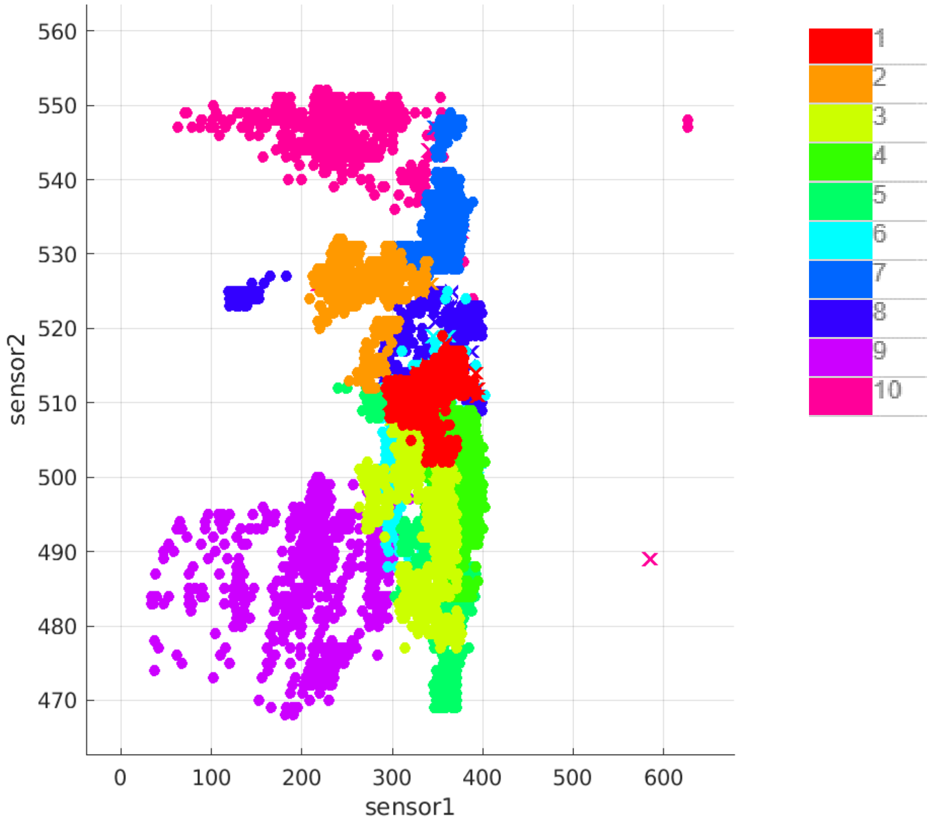


Figure 22. Distributions between the gestures with sensor1's and sensor2's values. The X-axis describes sensor1's sensor values, and the Y-axis represents sensor2's sensor values. On the right, each gesture (from 1 to 10) are color-coded.

In figure 22, the dots represent the correctly classified gestures while the crosses represent the wrong classified gestures.

Figure 23 shows the distributions between all the gestures with sensor1 and sensor5 values. As can be seen, it is not easy to distinguish the gestures. The dots and the crosses lie on the same line. The same occurs when comparing sensor5 with the rest of the sensors.

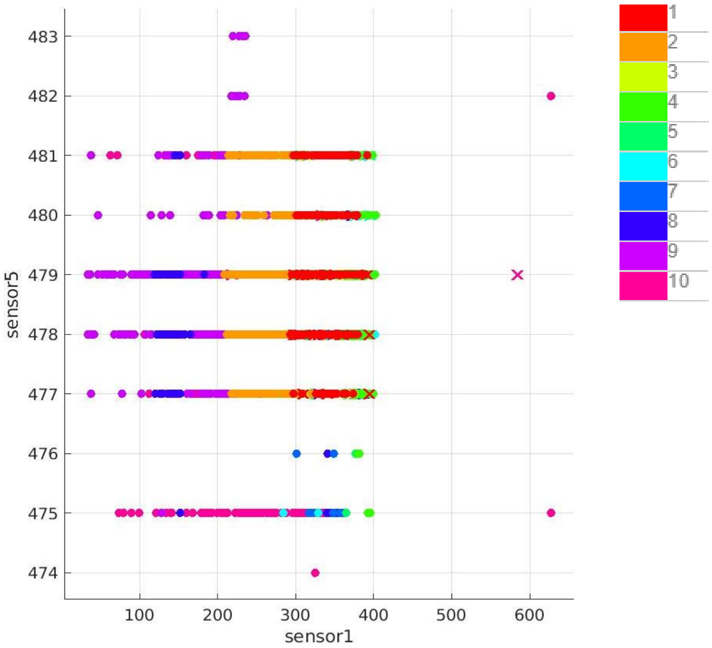


Figure 23. Distributions between the gestures with sensor1's and sensor5's values. The X-axis describes sensor1's sensor values, and the Y-axis represents sensor5's sensor values. On the right, each gesture (from 1 to 10) are color-coded.

Based on the previous analysis and empirical testing, it was decided that leaving out sensor5 would improve the SVM classifier's results. Thus, the following features have been selected for the input: sensor1, sensor2, sensor3, and sensor4.

6.1.4 Cross Validation

Before the cross validation, it is essential to redistribute the training dataset, which corresponds to 80 %. It is split into training cross validation- and validation set. The distribution related to the training dataset is as follows: the training cross validation set is 70 %, and the validation set is 30 %. With cross validation, the hyperparameters can be optimized to improve a model's performance and predictions.

Optimizing the Support Vector Machine is finding the best value for the error regularization parameter C, the kernel coefficient gamma, and the best fitting kernel function. These all are done manually with the cross validation and a grid search algorithm. However, a grid search took a long time to calculate, and thus, MATLAB was used for repeating the task. After extensive examination of the parameters and testing of the accuracies, the following parameters were identified as the most effective ones:

Table 3. Parameter adjustment of the SVM.

Parameter	Value
C	10
Gamma	scale
Kernel	Gaussian radial basis function (RBF)
Others	default

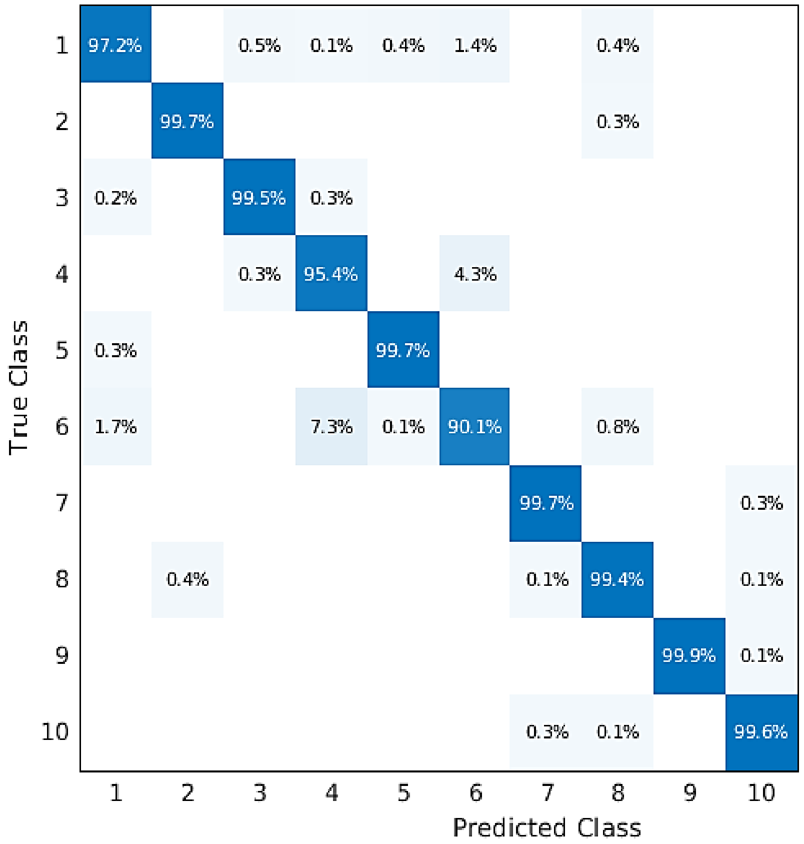
The parameters were calculated ten times, where eight of ten were the ones that are listed in table 3. The Gaussian radial basis function kernel was used in this algorithm. It can be mathematically described as follows:

$$k(\boldsymbol{x}, \boldsymbol{x}') = e^{-\gamma||\varphi(x)-\varphi(x')||^2}, \qquad \gamma > 0 \tag{19}$$

For the formula (19), the parameters are the same as described in chapter 3.3, Support Vector Machine Classifier. The ‘scale’ value means that the algorithm uses 1/features*the variance of the training set as value of gamma.

7 Results and Discussion

In this chapter, the full training results are presented, and the wristband's overall performance is analyzed. The performance of the classifier is evaluated with a confusion matrix in figure 24. Each row represents an actual gesture (from 1 to 10), and each column represents a predicted gesture.



True Class \ Predicted Class	1	2	3	4	5	6	7	8	9	10
1	97.2%		0.5%	0.1%	0.4%	1.4%		0.4%		
2		99.7%						0.3%		
3	0.2%		99.5%	0.3%						
4			0.3%	95.4%		4.3%				
5	0.3%				99.7%					
6	1.7%			7.3%	0.1%	90.1%		0.8%		
7							99.7%			0.3%
8		0.4%					0.1%	99.4%		0.1%
9									99.9%	0.1%
10							0.3%	0.1%		99.6%

Figure 24. Confusion matrix of the Gaussian SVM.

In the confusion matrix, the diagonal elements represent how many percentages are labeled correctly by the classifiers. These values are also called true-positive rates. The off-diagonal elements are mislabeled and thus called as false-negative rates.

The overall accuracy of full training is **98.0%** for four sensors. With five sensors, the best classifier accuracy was 97.2%.

7.1 Cost Analysis

One of the main goals of this project is to develop a cost-efficient wristband for static gesture detection. The approach started with a lot of background researching of the existing systems and devices. The focus was on searching the available sensor types, their accuracy, and the features' relation. This chapter reviews the cost breakdown of the project, which can be seen in table 4. The components are summarized and calculated together to get the overall prize for the device.

Table 4. Production Development Costs.

Type	Model	Company	Cost	Quantity	Total Cost
Electronics:					
Pressure Sensor	MPL115A2	NXP	5.63 CHF	15	84.45 CHF
Capacitor	1 uF	TDK	0.072 CHF	30	2.16 CHF
Resistor	4.7 kΩ	VISHAY	0.04 EUR	6	0.24 EUR
Microcontroller	Arduino NANO 33 BLE	Arduino	19.50 EUR	1	19.50 EUR
Printed Circuit Board	Flexible	PCBWay	20.8 USD	5	104.0 USD
Others:					
Liquid Silicone	Gingifast Elastic 2x50 ml	Zhermack	58.30 CHF	1	58.30 CHF
Faux Suede	Clothing Belt	Reused	0	1	0
Velcro	Stripes	MIGROS	6.90 CHF	1	6.90 CHF
Shipping & Handling					9.50 EUR
	**Currency converter* ([151.81 x 0.93 EUR] + [104.0 x 0.85 EUR] + 29.24 EUR)				
	Costs total				**≈ 258,82 EUR**

***Currency rates were taken from Morningstar on 26th Oct 2020, 15:51 UTC.**

The component selections can reduce the value of the final product. However, the accuracy of the components must be taken into account when replacing them. The most expensive component of the product is the printed circuit board. Other expensive parts were the liquid silicone and the sensors. There are usually low-priced liquid polymer products on the market. However, they were not available during the prototype design. In addition, there are no alternative, less expensive models available for the sensors, according to the present information. However, as this project shows, the number of sensors can be reduced.

According to table 4, the overall cost for the wristband development is ~260€. Further, it can be estimated that the final price for one prototype is around ~140 €, which is reasonable. The prototype is hence more affordable than the other wearable gesture recognition prototypes. The device's price can be reduced by 20 % with a more precise design, for example, by reducing the quantities of the PCBs and using a low-cost liquid polymer for the rubber casting process. Also, the Arduino NANO 33 BLE should be changed ideally to another, smaller microcontroller.

7.2 Risk Analysis

This chapter includes a risk analysis, which evaluates the factors that affect the performance of the wristband. In this project, the risk is defined as an event that was identifiable in advance and can affect the design, performance, and objectives. Each risk is analyzed to identify its qualitative effect on the wristband. The qualitative risk analysis additionally considers the probability and impact of each risk. This project's impact is classified into four groups, which are low, medium, high, and extreme. These are shown in table 5. The probability has three levels: low, medium, and high, and is presented in table 6. All the tables in this chapter are created from the source [37].

Furthermore, the features are combined to create a probability/impact assessment. This graphical report is called a risk matrix. There are different types of risk analyzing methods and risk matrices. Therefore, all tables and definitions are tailored solely to serve this thesis.

Table 5. The impact levels.

Impact	Low	Medium	High	Extreme
Description	**Acceptable** Not affecting the objectives	**Moderate** The act of mitigating	**Ineligible** Not suitable	**Intolerable** Place the event on hold.

As seen in table 5, the impact is the effect of the risk. All of them are associated with the objectives that are the cost and quality. When the impact is low, it is acceptable to continue. Medium impact needs mitigation efforts, and all high impacts require some actions. The extreme is intolerable, and therefore, the event must be placed on hold.

The probability levels are the likelihood that a risk will happen. These are illustrated in table 6.

Table 6. The probability levels.

Probability	Low	Medium	High
Description	**Acceptable** Risk occurs very rarely	**Moderate** Risk will likely occur	**Ineligible** Risk is expected

The graphical report includes only the risks with the highest impact and probability for the wristband to fail its goals. The risk matrix is shown in table 7 to visualizing the overall levels of the risks.

Table 7. A 3 x 4 risk analysis matrix for determination of qualitative severity levels and probability levels.

Impact →	**Low**	**Medium**	**High**	**Extreme**
Probability ↓	**Little to no effect on event**	**Effects are felt but not critical to outcome**	**Serious impact to the outcome**	**Could result in disaster**
Low	-1-	-4-	-6-	-10- **PCB design**
Medium	-2-	-5- **Soldering**	-8- **Sensor placement**	-11- **Delays in orders**
High	-3-	-7-	-9- **Machine Learning**	-12- **Rubber casting**

The most extreme risk is the rubber casting process, as it is the most difficult and challenging phase of the project. It requires careful work and the right tools. Thus, the likelihood that something will go wrong is high. If this happens, the result will be a non-functioning wristband, as air will remain between the sensor and the rubber, and the sensor will no longer be able to detect hand movements accurately. The second extreme risk is the delay in transporting the products. Usually, this does not happen, as products are ordered on time. In addition to that, it is possible to trace each order step by step. However, during the Coronavirus (COVID-19) pandemic, there were many delays in different sectors. For this work, the manufacturing and transportation of the PCB were two months behind the scheduled time. This affected the overall outcome, as there was no time left to refine the design.

According to table 5, designing the PCB, choosing the machine learning algorithm, and placing the sensors are classified as high impact risks. There is a very high probability that the used algorithm is not good enough in machine learning, leading to a high-profile error susceptibility. It would negatively affect the accuracy of the device. The limited number of sensors requires exact measurement positions. If this changes too much, the device is not capable of measuring accurately. The same principle applies to the PCB design. If there are errors, the measurement cannot be done. Although the probability of the PCB design error is low, its consequences are truly extreme. Therefore, it is classified as the third riskiest process.

The least risky is the soldering of the electron components process. There is always a risk of errors in the manufacturing of the components and the soldering itself. There are many components and PCB boards, and thus in the event of errors, these can be replaced.

8 Conclusion and Future Improvement

The aim of developing a wristband, which recognizes statistic gestures accurately with fewer sensors, has been achieved. Three main objectives nominated for the project were: affordability, accuracy, and fewer sensors. A cost breakdown was made for the device, which shows that its final price is only around 140 €. With small changes, the price of the device can be reduced by another 20 %. The same accuracy of 98.0 % for four sensors was obtained several times when predicting the right gestures using the RBF classifier. Therefore, the accuracy and fewer sensors -objectives are also met. Additionally, the SVM performed fast and thus can be used in real-time applications.

While the prototype meets the main objectives, many details in the design stage need to be redesigned. The biggest drawback was the PCB board, as sensor5 was not robust. It did not have a proper connection when the wristband was placed around the wrist in the trial set. The Arduino NANO is a thick and rigid component, whereas the PCB itself is very thin and bendable. Therefore, the SDA and SCL connections to the boards were unstable. Additionally, the wristband, as such, is not the ideal model for practical gesture recognition. For instance, the size of the wristband needs to be adjustable. Small sensing units needs to be designed and integrated with an already existing design, such as a smartwatch, to overcome the mentioned challenges.

The database for gesture recognition was collected from the same participant. Hence, in the development of the model, the training dataset should be created from different participants in a future project.

References

[1] Murhij Y, Serebrenny V. Hand gestures recognition model for Augmented reality robotic applications. Moscow, Russia; Bauman Moscow State Technical University; 2020.

[2] Li Y, Huang J, Tian F, Wang HA, Dai GZ. Gesture interaction in virtual reality. Beijing, China; Chinese Academy of Science; 2018.

[3] Galván-Ruiz J, Travieso-González CM, Tejera-Fettmilch A, Pinan-Roescher A, Esteban-Hernández L, Domínguez-Quintana L. Perspective and Evolution of Gesture Recognition for Sign Language: A Review. MDPI; 2020.

[4] Mario Di Castro. *Robotic Solutions for CERN Accelerator Harsh Environments.* [online]. Developer Works; 21 May 2019.
URL:
https://indico.cern.ch/event/814717/attachments/1849470/3035661/Robotic_Solutions_for_CERN_Accelerator_Harsh_Environments_EP_DT_Seminar_21_5_19.pdf
Accessed 14 September 2020.

[5] Morais GD, Neves LC, Masiero AA, Castro MCF. Application of Myo Armband System to Control a Robot Interface. Brazil; São Bernardo do Campo; 2016.

[6] Esposito D, Andreozzi E, Gargiulo GD, Fratini A, D'Addio G, Naik GR, Bifulco P. A Piezoresistive Array Armband With Reduced Number of Sensors for Hand Gesture Recognition. Frontiers in Neurorobotics; 2020.

[7] Gupta HP, Chudgar HS, Mukherjee S, Dutta T, Sharma K. A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors. IEEE; 2016.

[8] Dardas N. *Real-time Hand Gesture Detection and Recognition for Human Computer Interaction*. Ottawa, Canada; University of Ottawa; 2012.

[9] Digital Worlds that Feel Human | Ultraleap.
URL: https://www.ultraleap.com/
Accessed 13 August 2020.

[10] McIntosh J, McNeill C, Fraser M, Kerber F, Löchtefeld M, Krüger A. EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing. CHI; 2016

[11] Visconti P, Gaetani F, Zappatore GA, Primiceri P. Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses. Exeley; 2018

[12] CTRL-Labs.
URL: https://www.ctrl-labs.com/
Accessed 13 August 2020.

[13] Pressure Profile Systems, Inc.
URL: https://pressureprofile.com/
Accessed 13 August 2020.

[14] Tenzer Y, Jentoft LP, Howe RD. Inexpensive and Easily Customized Tactile Array Sensors using MEMS Barometers Chips. Harvard School of Engineering and Applied Sciences. IEEE; 2012.

[15] Weiner P, Neef C, Shibata Y, Nakamura Y, Asfour T. An Embedded, Multi-Modal Sensor System for Scalable Robotic and Prosthetic Hand Fingers. MDPI; 2019.

[16] TakkTile by RightHand Robotics, Inc.
URL: https://www.labs.righthandrobotics.com/
Accessed 13 August 2020.

[17] ScienceDirect: Machine Learning
URL: https://www.sciencedirect.com/topics/psychology/machine-learning
Accessed 15 October 2020.

[18] Jiang S, Lv B, Guo W, Zhang C, Wang H, Sheng X, Shull PB. *Feasibility of Wrist-Worn, Real-Time Hand, and Surface Gesture Recognition via sEMG and IMU Sensing*. IEEE; 2018

[19] Lian X, Ghannam R, Heidari H. Wrist-Worn Gesture Sensing with Wearable Intelligence. IEEE; 2019.

[20] Byun SW, Lee SP. Implementation of Hand Gesture Recognition Device Applicable to Smart Watch Based on Flexible Epidermal Tactile Sensor Array. MDPI; 2019.

[21] Ahsan M, Ibrahimy M, Khalifa O. Electromygraphy (EMG) Signal Based Hand Gesture Recognition Using Artificial Neural Network (ANN). ICOM; 2011.

[22] Shull PB, Jiang S, Zhu Y, Zhu X. Hand Gesture Recognition and Finger Angle Estimation via Wrist-Worn Modified Barometric Pressure Sensing. IEEE; 2019.

[23] Xiao ZG, Menon C. Towards the development of a wearable feedback system for monitoring the activities of the upper-extremities. Journal of Neuro Engineering and Rehabilitation; 2014.

[24] Zhang Y, Liu B, Liu Z. Recognizing Hand Gestures with Pressure-Sensor-Based Motion Sensing. IEEE. 2019

[25] Benatti S, Milosevic B, Farella E, Gruppioni E, Benini L. A Prosthetic Hand Body Area Controller Based on Efficient Pattern Recognition Control Strategies. Zürich, Switzerland; ETH Research Collection; 2019.

[26] Dictionary: A Hand definition.
URL: https://www.dictionary.com/browse/hand
Accessed 14 August 2020.

[27] Tortora GJ, Nielsen MT. Principles of Human Anatomy. 12th ed. USA; John Wiley & Sons; 2012. p. 227-241, 264-281, 304-341, 392-412.

[28] Morton D, Albertine K, Foreman B. The Big Picture: Gross Anatomy. The McGraw-Hill Companies; 2011. p. 353-369.

[29] Musculoskeletal Key: 6 The Wrist and Hand: Diagnostic Imaging
URL: https://musculoskeletalkey.com/6-the-wrist-and-hand-diagnostic-imaging
Accessed 10 October 2020.

[30] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques: Data Preprocessing. 3rd ed. USA; Elsevier; 2012. p. 83-124.

[31] Shuzhan Fan. Understanding the mathematics behind Support Vector Machines
URL: https: //shuzhanfan.github.io/2018/05/understanding-mathematics-behind-support-vector-machines/
Accessed 10 October 2020.

[32] Christianini N, Shawe-Taylor J. An Introduction To Support Vector Machines And Other Kerlne-based Learning Methods. Cambridge University Press; 2000. p. 1-148.

[33] Christopher M. Bishop. Pattern Recognition And Machine Learning. Springer; 2006. p.1-220, 291-319, 605-646.

[34] Khan F, Enzmann F, Kersten M. Multi-phase classification by a least-squares support vector machine approach in tomography images of geological samples. Germany; Johannes Gutenberg University; 2015.

[35] Song P, Ma Z, Ma J, Yang L, Wei J, Zhao Y, Zhang M, Yang F, Wang X. Recent Progress of Miniature MEMS Pressure Sensors. MDPI; 2020.

[36] Semiconductor Freescale, "MPL115A2 Miniature I2C Digital Barometer," Datasheet.
URL: https://www.nxp.com/docs/en/data-sheet/MPL115A2.pdf
Accessed 28 July 2020.

[37] Project Risk Coach
URL: https://projectriskcoach.com/perform-qualitative-risk-analysis/
Accessed 23 October 2020.

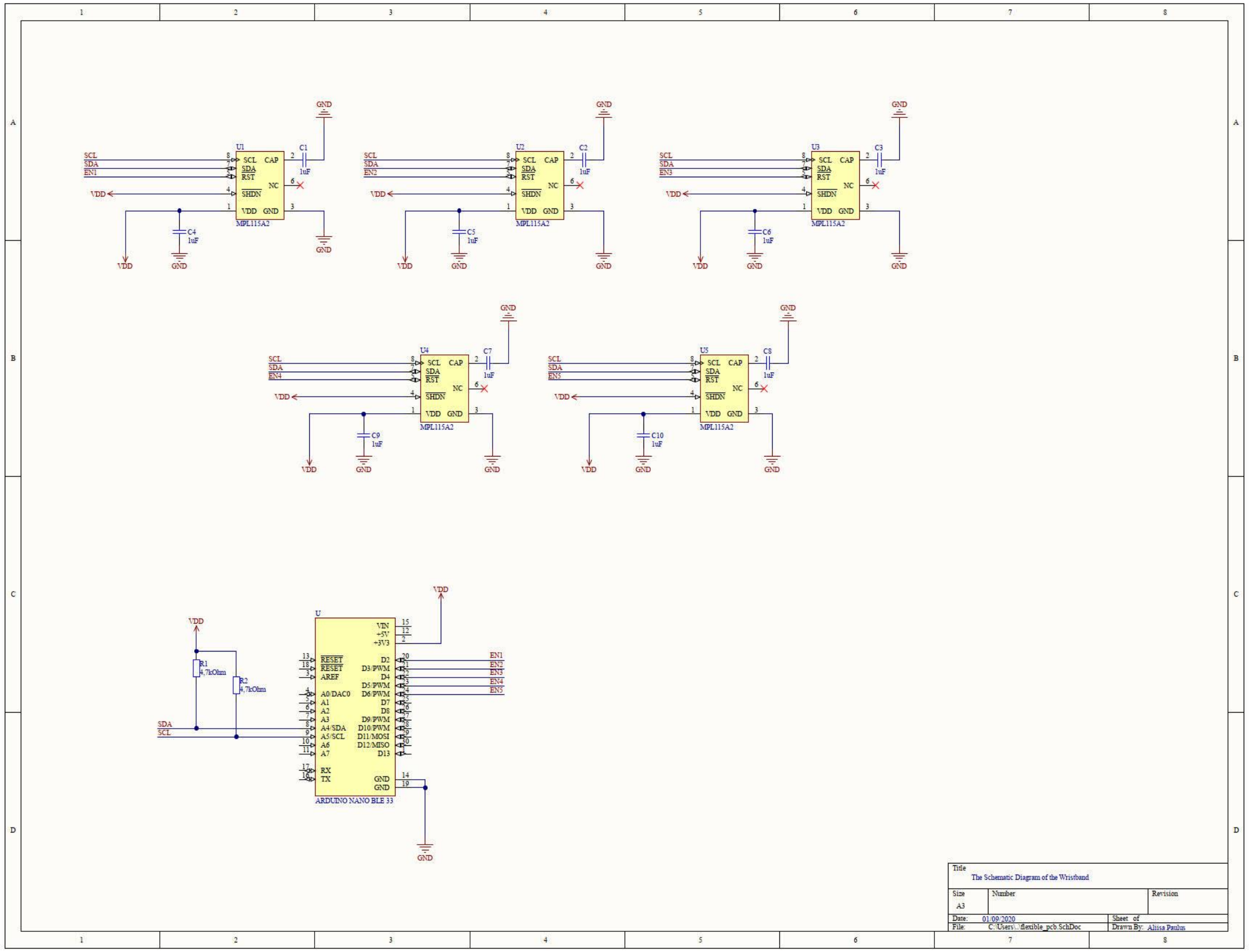
Acknowledgements

I would like to sincerely thank my supervisors Heikki Valmu and Matti Fischer, whom convincingly guided me during this project's development and continuously encouraged me in my studies at Metropolia University of Applied Sciences.

I wish to express my deepest gratitude to CERN (the European Organization for Nuclear Research) for providing access to all the research papers and the textbooks that were needed for this work. A very special appreciation goes to the Section Leader of the Mechatronics, Robotics, and Operations Section (EN-SMM-MRO) at CERN, Mario Di Castro, and my supervisor at CERN, Luca Rosario Buonocore, for giving me the possibility to join the robotics team and for providing equipment and support when developing this project. Also, I want to thank Manuel Sanchez Suarez, who helped me with the assembly of the printed circuit boards presented in this thesis.

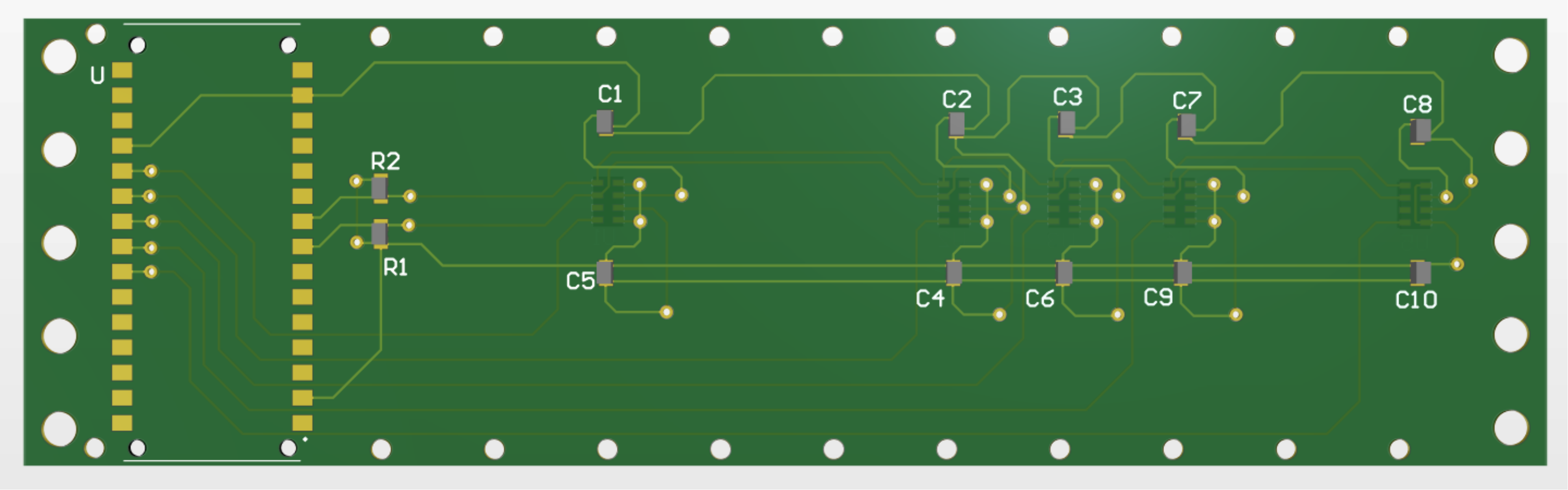
Finally, I want to express my appreciation to my colleagues at CERN, David Forkel and Victor Norrild, and my colleague at Metropolia, Daniel Fasel. Their constant invaluable feedback, assistance, and endless support throughout the work were precious.

The Schematic Diagram of the Wristband



Schematic for the sensor inputs of the control board.

Printed Circuit Board of the Wristband



3D layout of the PCB.