Wearable Smart Rings for Multi-Finger Gesture Recognition Using Supervised Learning

Seyed Ahmadreza Mousavi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Electrical and Computer Engineering)

at

Concordia University

Montréal, Québec, Canada

January 2023

Abstract

Wearable Smart Rings for Multi-Finger Gesture Recognition Using Supervised Learning

Seyed Ahmadreza Mousavi

This thesis presents a wearable, smart ring with an integrated Bluetooth low-energy (BLE) module. The system uses an accelerometer and a gyroscope to collect fingers motion data. A prototype was manufactured, and its performance was tested. To detect complex finger movements, two rings are worn on the point and thumb fingers while performing the gestures. Nine pre-defined finger movements were introduced to verify the feasibility of the proposed method. Data pre-processing techniques, including normalization, statistical feature extraction, random forest recursive feature elimination (RF-RFE), and k-nearest neighbors sequential forward floating selection (KNN-SFFS), were applied to select well-distinguished feature vectors to enhance gesture recognition accuracy. Three supervised machine learning algorithms were used for gesture classification purposes, namely Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes (NB). We demonstrated that when utilizing the KNN-SFFS recommended features as the machine learning input, our proposed finger gesture recognition approach not only significantly decreases the dimension of the feature vector, results in faster response time and prevents overfitted model, but also provides approximately similar machine learning prediction accuracy compared to when all elements of feature vectors were used. By using the KNN as the primary classifier, the system can accurately recognize six one-finger and three two-finger gestures with 97.1% and 97.0% accuracy, respectively.

Acknowledgments

My sincere thanks to my supervisor, Professor Rastko R. Selmic, for his continued support and guidance. I want to extend my genuine thankfulness to my wife for her support during the writing of this dissertation. My appreciation also extends to the support I received from Concordia University for the scholarships that enabled me to pursue this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Topic

This thesis presents a framework for multi-finger gesture recognition utilizing wearable smart rings equipped with inertial measurement units (IMUs). A pair of self-calibrating rings has been fabricated to be worn on the point and thumb fingers. Using Bluetooth low energy (BLE), the inertial data are transmitted to the Raspberry Pi Zero W. As part of this thesis, ten statistical features have been extracted from raw data for pre-processing before applying feature selection methods to improve machine learning accuracy and processing time. The classification of each gesture is further developed by comparing the results of three different machine learning algorithms, namely, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Naive Bayes (NB). As a result of its small size, our finger gesture recognition interface is sufficiently robust and adaptable to various hand and finger shapes. Furthermore, this sensor-based finger gesture recognition method does not suffer from the problems associated with vision-based gesture recognition, including background colour, texture, and illumination.

1.2 Motivation

The gesture recognition system recognizes human gestures and classifies them so they can be used as control commands. Two important applications of finger and hand gesture recognition include:

- Human-Computer Interaction (HCI) which can facilitate a wide spectrum of emerging applications, including smart home, virtual reality (VR), augmented reality (AR), and gaming [1]. Gesture recognition has been used to advance methods of human-robot interaction [2, 3, 4, 5, 6].
- As an assistive for people with disabilities such as motor impairments [7]. As another example, hand and finger gesture recognition can be broadly used by deaf and mute individuals as a communication tool through sign language recognition [8, 9].

In this project, we have focused on the Sensor-Based Recognition (SBR) approach due to its advantages compared with the Vision-Based Recognition (VBR) method, including:

- Lower computation burden [10]
- Faster dynamic response [11]
- Lower data storage [11]
- No limitations in environmental conditions such as light [8, 11].

1.3 Contribution

We propose an adaptable, wearable smart ring system to recognize and classify multi-finger gestures. Below are some of the key contributions of this work:

(1) Our system can detect multiple types of finger gestures, including upward, downward, and circular movements, as well as more complex movements such as zooming in and out.

(2) We have shown that an IMU can enhance machine learning accuracy by six to seven percent compared to an accelerometer alone and by eight to thirteen percent in contrast to a gyroscope alone. By combining accelerometer and gyroscope data for detecting each gesture, we were able to achieve significantly improved results compared to using one of them as a measurement device.

(3) Utilizing sophisticated data pre-processing methods before feeding the data to the machine learning model, the size of the feature vector is dramatically reduced from 60 to 10 for one-finger gestures and from 120 to 10 for two-finger gestures. Despite maintaining accuracy, this reduction in feature size improves processing time and prevents any possible overfitting.

(4) To improve the portability and user experience, we proposed a small-size ring powered by a rechargeable lithium polymer battery that can operate continuously for an extended period of time on a single charge and uses wireless connectivity (BLE) instead of a typical wired connection. Due to their small size (less than 20 mm $\times$ 20 mm) and low weight (3.2×10^{-3} kg), our smart rings can be worn on any finger.

The rest of this thesis is organized as follows: Chapter 2 presents an overview of the machine learning classification method and also hardware that are utilized in the field of sensor-based finger and hand gesture recognition along with a comprehensive literature review is introduced in which recent related works on the sensor-based finger and hand gesture recognition. Chapter 3 introduces the proposed multi-finger gesture recognition system and describes its hardware components and implementation. Chapter 4 is dedicated to the feature selection methods utilized in this project. Chapter 5 presents the training and testing details and the recognition results of the proposed system for multi-finger gesture recognition. Finally, Chapter 6 concludes the thesis and gives an overview of future works.

Chapter 2

Background and Related Works

2.1 Introduction

This chapter presents an overview of the machine learning fundamentals, classification methods, and sensors utilized in sensor-based finger and hand gesture recognition. Also, a comprehensive literature review is introduced in which recent related works on sensor-based finger and hand gesture recognition are summarized and compared.

2.2 Machine Learning Fundamentals

Figure 2.1 shows the general structure of a machine learning-based predictive model, where historical data is used as training data and new test data is used to generate outcomes.

In general, machine learning algorithms fall into four categories: Supervised learning, Unsupervised learning, Semi-supervised learning, and Reinforcement learning [12], as shown in Figure 2.2. Supervised learning is the most common algorithm utilized in hand and finger gesture recognition. It uses labeled training data and a collection of training examples to infer a function that maps an input to an output. Supervised learning is suitable in case of a task-driven approach, i.e. when there are certain goals that must be achieved from certain inputs. There are two types of supervised tasks that are commonly used: classification which separates the data, and regression which fits the data to the classes.

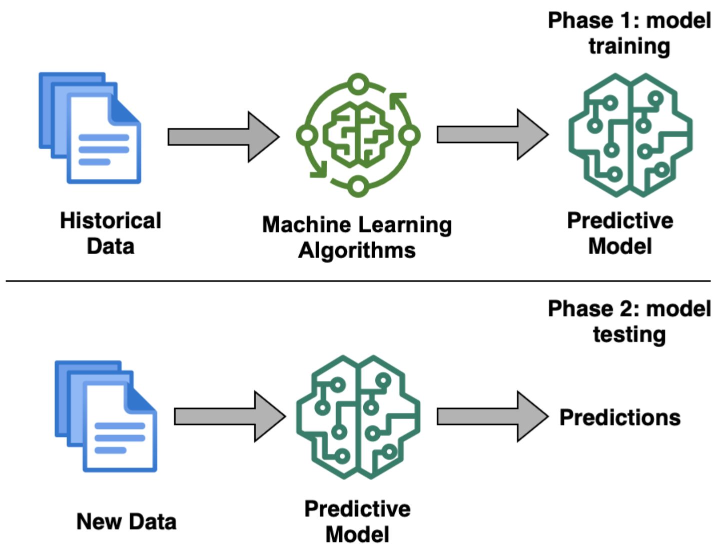


Figure 2.1: Training and testing phases of a machine learning-based predictive model.

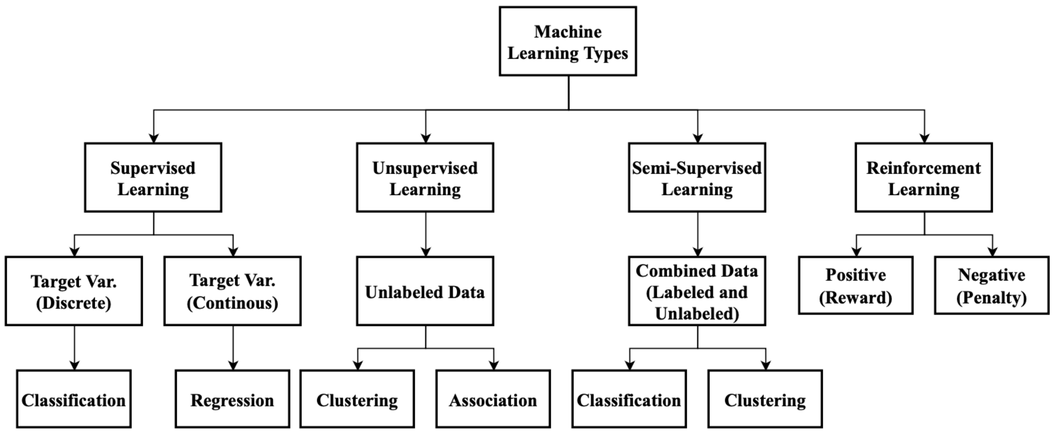


Figure 2.2: Types of machine learning techniques.

2.2.1 Classification Methods Used in Gesture Recognition

KNN:

Both classification and regression problems can be solved with KNNs in machine learning. Using the KNN algorithm, the new input data is classified based on its similarity to the previously trained data, which is grouped into coherent clusters or subsets. Inputs are assigned to classes based on the number of nearest neighbours they have. As a non-parametric classification algorithm, KNN does not assume anything about the elementary dataset. In spite of this classifier's simplicity, K plays a significant role in classifying data without labels. Increasing K reduces the net noise. It can be expensive to determine the optimal K in the case of large datasets [13].

NB:

Naive Bayes classifiers assign gesture classes based on feature vectors with each feature conditionally independent of another. Predictions can be made efficiently during both training and inference stages, as it requires less training time than other more complex models. Conditional independence is often observed to work well in many tasks, but it is also often untrue in most cases, limiting performance [8].

Relevance Vector Machine (RVM):

RVMs are Bayesian sparse kernel algorithms for regression and classification, which share many characteristics of SVMs while avoiding their primary drawbacks. SVMs produce decisions, while the outputs of RVM are posterior probabilities. In addition, the SVM was developed for two classes, so extending it to more would be problematic. RVM typically produces sparser models than SVM, resulting in faster performance on test data while maintaining comparable generalization error [14].

Logistic Regression (LR)

A standard probability-based model for classification problems in machine learning. LR uses a logistic function to estimate probabilities mathematically, also called a sigmoid function, as shown in Eq 1. It is possible to avoid over-fitting by using regularization techniques. LR has the disadvantage of assuming linear relationships between independent and dependent variables. Both classification and regression problems can be solved with LR, but the classification is more commonly used. LR, which is also known as maximum-entropy classification, is capable of both binary and multinomial classifications [14].

$$g(z) = \frac{1}{1+e^{-z}}. \quad (1)$$

SVM:

A SVM is a mechanism for separating feature vectors of different classes by generating hyperplanes in *n*-dimensional space (*n* = the number of input features). SVM uses the principal difference between the classes as a criterion to calculate the separating hyperplanes. SVMs are maximum margin classifiers, i.e. the hyperplanes are constructed to maximize the distance (margin) between the hyperplanes and the closest feature vectors on each side [15]. An example of SVM is depicted in Figure 2.3.

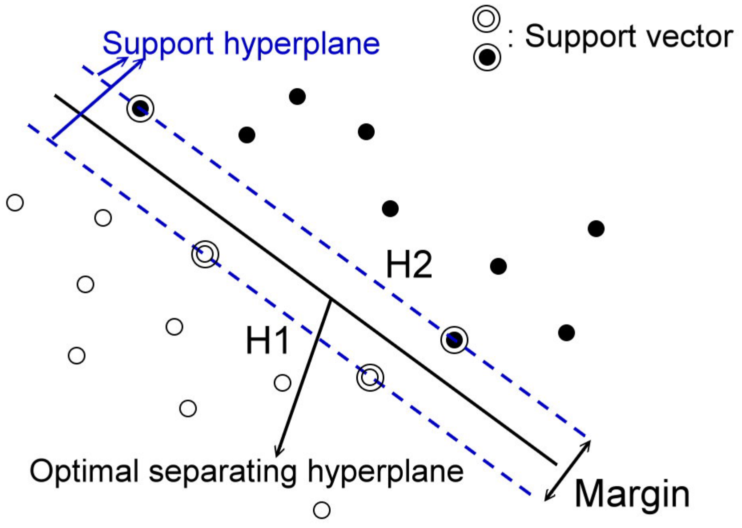


Figure 2.3: An example of SVM (source: [15]).

Decision Tree (DT):

The Decision Tree (DT) is one of the most popular classifiers due to its intelligible nature. Due to their flexibility, robustness to noise, low computational cost, and ability to handle redundant attributes, DT induction algorithms are preferred over other learning algorithms [16]. Both classification and regression tasks can be handled by decision trees. The prediction of discrete values is done through classification, while the prediction of continuous values is done through regression. There are numerus well known DT algorithms, including ID3, C4.5, and CART. As shown in Figure 2.4, the instances are classified by sorting down the tree from the root to some leaf nodes. Attributes of each node are checked for classification starting with the root node, then moving along the tree branches corresponding to the attribute value. It is common to split the branches based on Gini impurity and Entropy for information gain, which can be expressed mathematically as [17]:

$$Entropy : H(x) = -\sum_{i=1}^{n} p\left(x_i\right) \log 2p\left(xi\right)$$

$$\mathrm{Gini}(E) = 1 - \sum_{i=1}^{c} p_i^2. \tag{2}$$

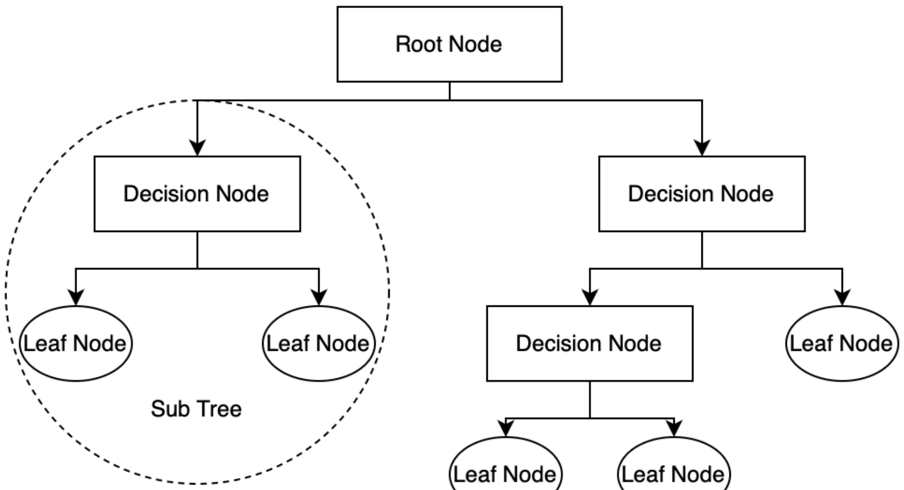


Figure 2.4: An example of a DT structure.

Random Forest (RF):

It has been found that growing an ensemble of trees and letting them vote for the most popular class can result in significant improvements in classification accuracy. A random vector is often generated to govern the growth of each tree in an ensemble in order to grow these ensembles. They vote for the most popular class after generating a large number of trees. Generally, these procedures are referred to as random forests Figure 2.5. Each tree in a random forest depends on the value of an independent and equal distribution random vector sampled independently for each tree. The use of random forests is an effective prediction tool. In random forests, there is no overfitting, since the Law of Large Numbers holds true. Random forest classifiers and regressors are more accurate when they are injected with the right kind of randomness [18].

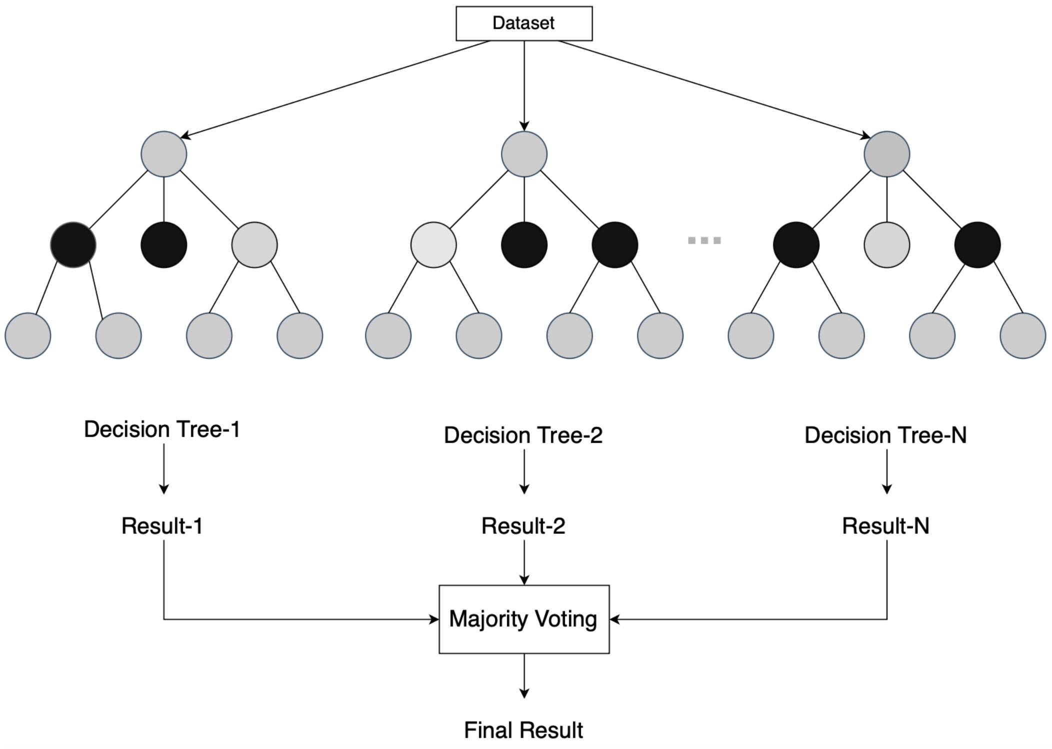


Figure 2.5: An example of a random forest structure considering multiple decision trees

2.3 Static vs. Dynamic Gesture Recognition

In finger or hand gesture recognition applications, the users move their hands or fingers to assume a specific pose or configuration. The literature review on gesture recognition has traditionally distinguished between two main types of gesture [19]:

- Static gestures refer to hand or finger positions that are held in a fixed position. Some examples of static gestures are pointing at a specific object, making a fist, or holding the hand in a specific position. These types of gestures are usually used to interact with digital devices, activate a certain function or to make selections on screens, etc. Static gestures are also known as pose-based or posture-based gestures.

- Dynamic gesture is a hand or finger movement that occurs over time. It is a gesture that changes over time, and can be detected using sensors such as accelerometers, gyroscopes, or cameras. Some examples of dynamic gestures are swiping, rotating, pinching or tapping. These types of gestures are typically used to interact with digital devices, such as scrolling through a webpage or zooming in on a map. Dynamic gestures are also known as motion-based or gesture-based gestures.

With the help of IMU sensors in our design, we are able to detect nine different dynamic gestures by utilizing machine learning algorithms.

2.4 Hardware for Finger and Hand Gesture Recognition Review

2.4.1 Visual-based and Sensor-based Hardware

Finger gesture recognition offers numerous benefits, including virtual typing and drawing, signature recognition, augmented reality, sign-language recognition, and real-time robot control. Finger gesture recognition approaches can be classified into Vision-Based Recognition (VBR) [20, 21] and Sensor-Based Recognition (SBR) [10, 11, 22] algorithms. Optical-based gesture recognition methods use optical sensors such as cameras [23, 24] or infrared (IR) sensors [25] to capture the movements of fingers and interpret them into commands. Although accurate classification is possible, high data storage and computation efforts are required to extract information from images for training and inference operations [10, 11]. The VBR approaches have also suffered from occlusion problems (no direct line of sight), different users' hand speed, the need for a proper environment with enough light to take photos [8, 11].

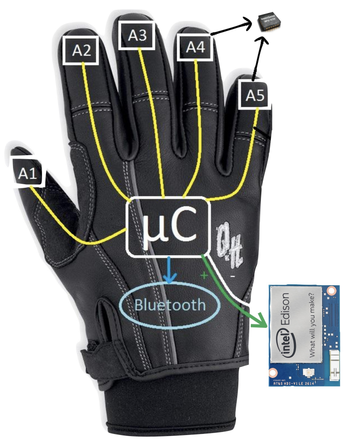
With recent technological advancements, batteries, circuit boards, microcontrollers, and sensors have become small and cost-effective. Sensors are less susceptible to environmental influences than cameras. Modern portable systems are practical for mobile and wearable applications due to their ability to store large amounts of sensor data. Sensor-based approaches have the primary disadvantage of causing discomfort to users or restricting their movements. The problem can be addressed by embedding sensors into wearable devices, such as gloves or rings [8]. These wearable interfaces are introduced in the next subsection.

2.4.2 Types of interfaces for sensor-based gesture recognition

Although in numerous studies, no particular interface is used to mount the sensors on the body, e.g. [9, 26], in real-world applications, convenient interfaces are needed to facilitate the integration of motion sensors to the body. Four types of interfaces have been used in previous studies to attach the sensor needed for finger and hand gesture recognition in previous studies:

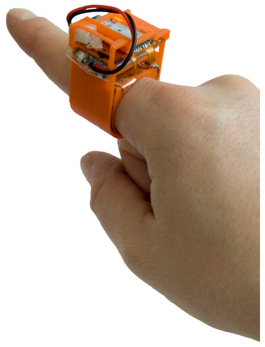
- Gloves [2, 9, 27]
- Rings [7, 28, 29, 30, 31]
- Smartwatches [29, 32, 33]
- Pens [22, 34]

Pens are more suitable for handwriting or signature recognition applications [9], but wearable equipment such as rings, smartwatches and gloves can be used for widespread applications. Smart Gloves are the first wearable interfaces and have been widely utilized since 1977 [27]. However, due to the miniaturization of electronic sensors and components, smaller and more fashionable interfaces such as smart rings and smartwatches are being developed. Rings are finger-mounted equipment, while smartwatches are wrist-mounted interfaces with different motion sensors. In [34, 35], an accelerometer-based digital pen, with a trajectory recognition algorithm, was designed to recognize handwritten digits and gestures. Users were asked to write words using a smartphone as a pen [36] and the handwriting was reconstructed using a gyroscope and accelerometer embedded in the phone.



(a) Glove [27]

(b) Smartwatch [37]



(c) Ring [38]

(d) Smart Pen [39]

Figure 2.6: Different types of interfaces for sensor-based hand and finger gesture recognition.

Wireless Interfaces

The presence of wires to transmit the signals of the sensors will cause some movement constraints for the users. To avoid these disadvantages, wireless sensors are becoming more favorable [40]. To manage energy consumption, BLE modules are a suitable choice for wireless interfaces [15, 38].

Our research for finger gesture recognition systems was identified through a comprehensive literature review of state-of-the-art techniques and technologies. Having conducted this research, we gained a better understanding of the advantages and limitations of different electronic devices, allowing us to make an informed decision regarding which devices to use. The following reasons led us to choose smart rings:

- Convenience: A smart ring is a convenient and portable device that can be worn on a finger, making it easy for users to carry and use at all times.
- Discreetness: A smart ring is small and subtle, which allows for a more discreet interaction with the device and can be worn in various social and professional situations.
- Ergonomics: A smart ring is worn on the finger, a natural and intuitive location for controlling digital devices, and it allows for easy and natural hand and finger movements.
- User needs: The smart ring is designed to cater to specific user needs, such as tracking fitness and health data, providing notifications, or controlling smart home devices.
- Technical feasibility: The smart ring is equipped with sensors and technologies that are compact and power-efficient, making it a technically feasible device to wear all day.

With recent advances in machine learning technologies, HCI products can now be used as assistive devices for people with disabilities. They can also be used as communication aids for people who are deaf or mute [7]. Our goal is to use this technology in the future to improve communication between people with disabilities of all kinds. Additionally, we intend to make people's interactions with their smartphones and smart homes more convenient and enjoyable. Smart ring systems have the potential to offer many possibilities.

2.4.3 Types of sensors used in gesture recognition

IMU-based sensors:

Wearable IMU-based sensors are probably the most common in gesture and human-motion recognition. IMU sensors benefit from efficiency in signal processing, and low latency [41]. IMU sensors have been augmented to all the different wearable interfaces used in hand and finger gesture recognitions, including gloves, rings, smartwatches and pens. Low weight, price, and compact size are among the advantages of IMU sensors [42]. These sensors have also become exceedingly power-efficient in recent years [27]. IMU-based sensors have three different types [43], namely:

- Accelerometers
- Gyroscopes
- Magnetometer

These types of IMU-based sensors can be used invaluably. However, in many applications, a combination of two or three are utilized.

The fusion of IMU-based sensors with other types of sensors introduced in the following is very common in hand and finger gesture recognition. A sensor-fusion algorithm typically uses inertial sensors to track hand motions and other sensors to detect additional information such as finger snapping, hand gripping, or fingerspelling [43]. However, it is suggested that the inertial-sensor-only approach potentially increases portability and mobility.

The inertial sensor improves portability and versatility with a reduced computation load compared to multiple wearable sensors or heavy algorithms. Motion energy measured at the smartwatch [33] is sufficient to recognize the user's finger gestures. In a similar study [32], a novel technique for recognizing fine-motor finger gestures using integrated motion sensors (accelerometer and gyroscope) in off-the-shelf smartwatches is proposed. The potential to distinguish five fine-motor gestures like pinching, tapping, and rubbing fingers, with an average F1-score of 87% is achieved. In [44], a steady hand gestures recognition using a three-axis accelerometer and gyroscope sensors in a smart device is developed. An automatic gesture spotting algorithm is also investigated to detect the start and endpoints of meaningful gesture segments.

Flex Sensors:

Flex sensors are widely used in hand and finger gesture recognition applications. These types of sensors are usually mounted on smart gloves. Figure 2.7 shows the utilization of flex sensors in fusion with IMU-based sensors for hand gesture recognition [9].

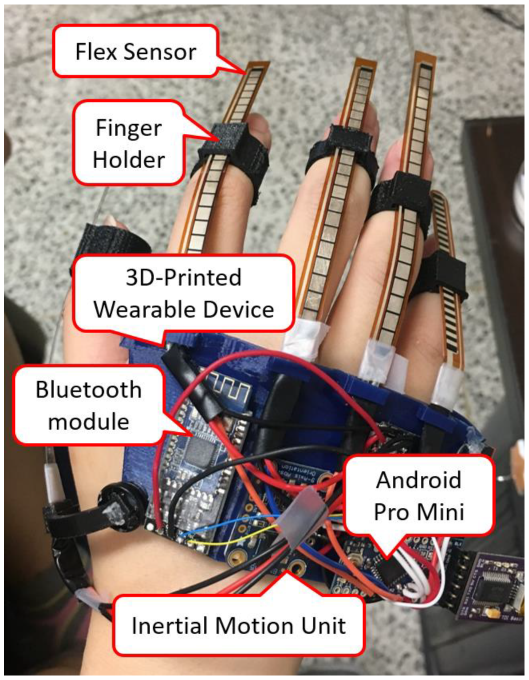


Figure 2.7: Flex sensors in fusion with IMUs [9].

Stretch Sensors:

Fusing stretch sensors with IMU-based sensors is also common in gesture recognition. For example, [40] has used stretch sensors in a combination of IMUs for multiple finger tracking (Figure 2.8).

Electromyography (EMG) Sensors:

In many applications, electromyography sensors have been combined with IMU-based sensors to recognize hand and finger gestures [42, 45]. As an example, [42] has utilized the fusion of surface electromyography (sEMG) sensors with wrist-worn IMU sensors for Persian sign language recognition, as is shown in Figure 2.9.

Pressure Sensors:

Pressure sensors mounted on the fingers are also used in gesture recognition applications.

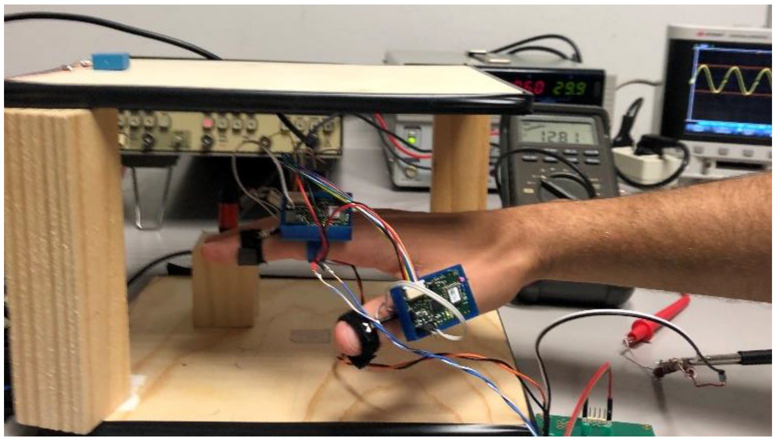


Figure 2.8: Stretch sensors in fusion with IMUs on wearable smart rings used for multiple finger tracking [40].

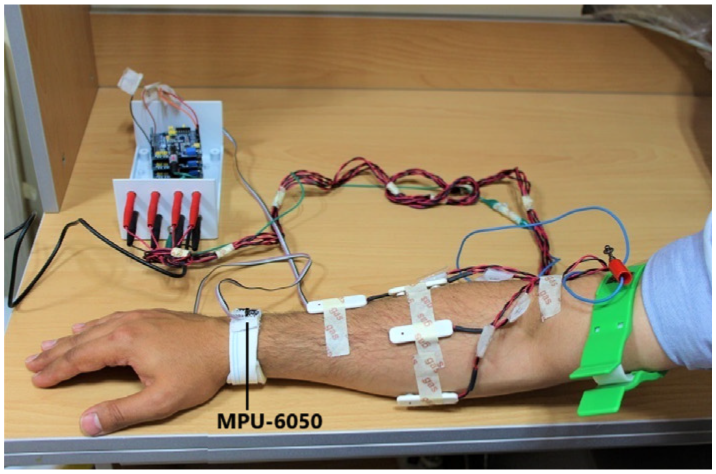


Figure 2.9: Fusion of surface electromyography (sEMG) sensors with wrist-worn IMU sensors [42].

As an example, [9] has shown that fusion of pressure sensors with flex and inertial sensors can improve the accuracy of sign language interpretation systems based on smart wearable hand devices. [46] has also used a fusion of pressure and IMU sensors for finger movement monitoring.

Capacitive Proximity Sensors:

A low-cost, effective capacitive sensor device is proposed in [47] for recognizing hand gestures. A prototype is fabricated and equipped with a capacitive sensing unit to capture capacitance values from electrodes attached to finger phalanges. A recognition rate of 97% is reached for the compressed feature dataset using Error Correction Output Code Support

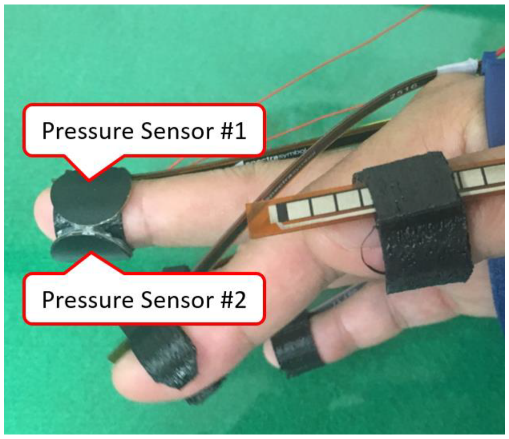


Figure 2.10: Wearable pressure sensors used for hand gesture recognition [9].

Vector Machines (ECOC-SVM) and KNN classifiers. Capacitive proximity sensing can detect conductive objects by measuring the effect of an electrode on the electrical field. TypingRing [28] distinguishes finger taps on a surface to allow text input. It uses proximity sensors to detect if the adjacent finger is next to the ring or not. In finger gesture recognition, smart rings equipped with capacitive proximity sensors have been used to measure the approximate distances to other fingers, enabling multi-finger gesture recognition [38]. Figure 2.11 depicts an example of multi-finger gesture recognition using capacitive proximity sensors.

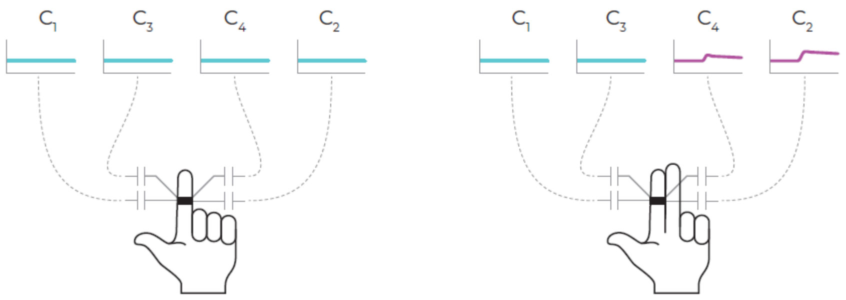


Figure 2.11: Capacitive proximity sensing working principles [38].

Microphones:

Microphones are sensors that are capable of recording sound signals. In some finger recognition systems, microphones are also utilized in fusion with other sensors to improve accuracy. As an example [48] has used a smart ring equipped with a contact microphone in fusion with an IMU for unistroke thumb gestures using a ring.

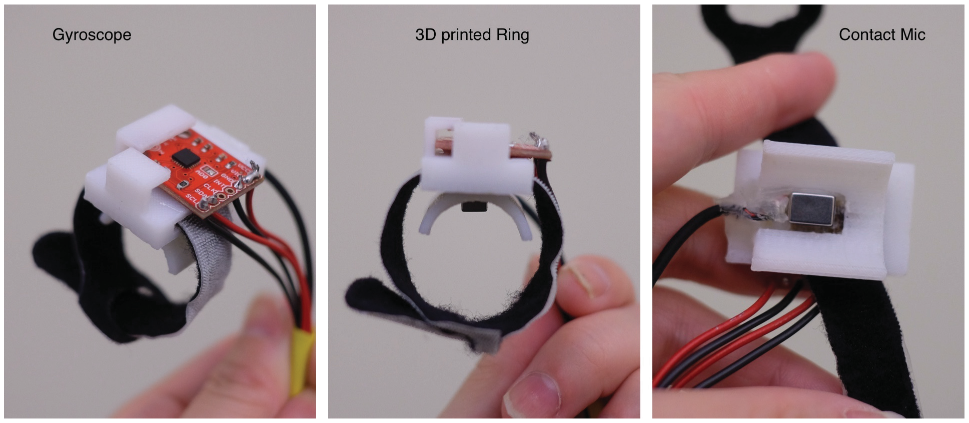


Figure 2.12: Wearbale ring equipped with contact microphone in fusion with an IMU sensor [48].

2.4.4 Types of Target Hand and Finger Gestures

Different studies have chosen various target hand and finger gestures to recognize. Directional gestures, such as up, down, right, left, left up, left down, right up, right down, clockwise circle, and counterclockwise circle, are among the common gestures due to their wide application in human-computer interaction [15]. A critical application of hand and finger gesture recognition, as illustrated in Figure 2.13, is in sign language recognition (SLR) systems, which deaf individuals have extensively used to communicate [42]. Numbers and alphabet can also be recognized as unistroke finger gestures using smart rings, as shown in Figure 2.14 and Figure 2.15 [48].

2.5 Related Works

This section reviews previous work in sensor-based hand and finger gesture recognition. Table 2.1 presents a summarized comparison of the mentioned works.

In [49], the authors have investigated the use of smartwatches for finger gesture recognition. Recorded signals of IMU-based sensors integrated into smartwatches, including accelerometers and gyroscopes, are utilized for finger gesture recognition. Three machine

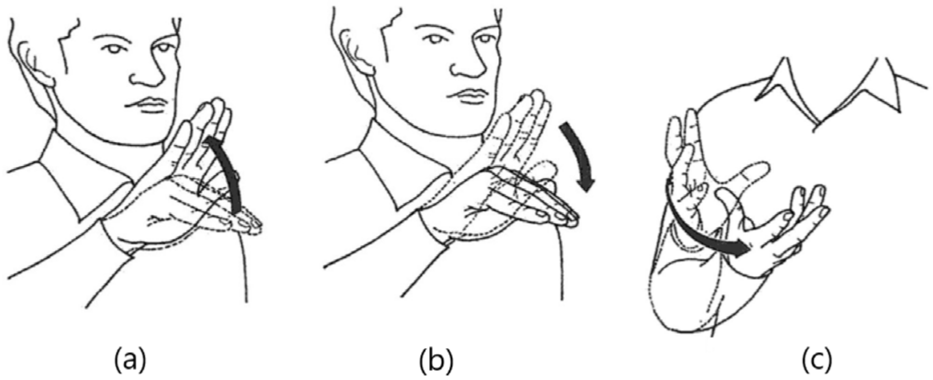


Figure 2.13: Examples of the typical hand gestures in sign language recognition. (a) NO. (b) Yes. (c) How are you [42].

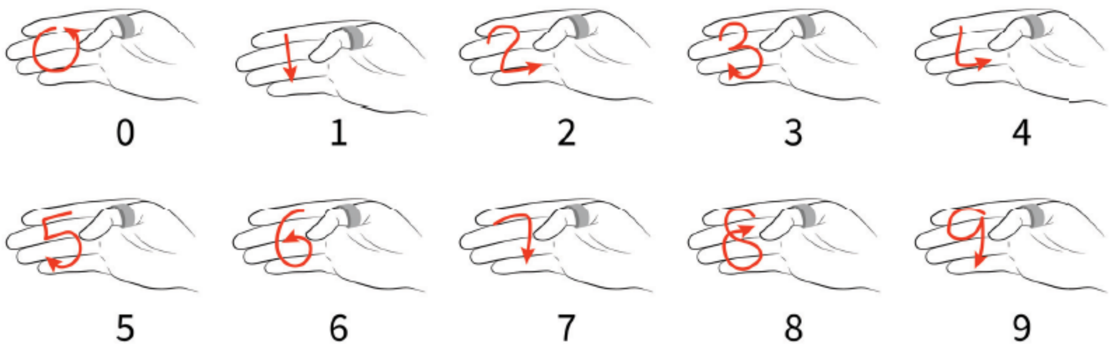


Figure 2.14: Recognition of numbers as unistroke finger gestures [48].

learning classifiers are utilized and compared for finger gesture recognition: KNN, LR and DA.

A real-time continuous hand gesture recognition with wireless wrist-worn IMU sensors has been introduced in [15]. This study used 3-axis accelerometers and 3-axis gyroscope sensors to recognize ten hand movement gestures. SVM is used in this work as the classification algorithm. The authors have achieved recognition accuracy at 86.99% in fixed combinational gestures and 60% in arbitrary combinational gestures.

A gyroscope-based continuous human hand gesture recognition has been proposed by [43]. Instead of machine learning methods, the gestures are classified based on normalized covariance and threshold adjustment in this study. A smart ring equipped with capacitive proximity sensing in fusion with IMU-based sensors for multi-finger gesture recognition

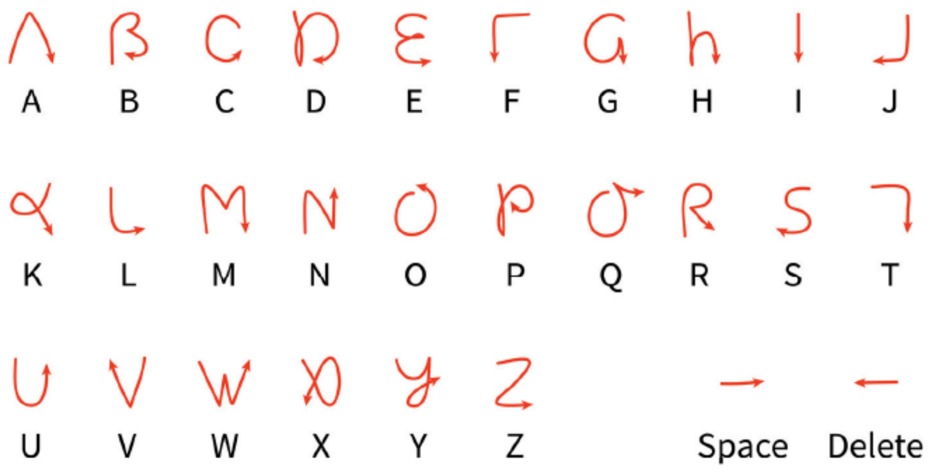


Figure 2.15: Recognition of alphabets as unistroke finger gestures [48].

has been developed in [38].

An approach to hand gesture recognition using surface electromyography (sEMG) sensors mounted on the arm and wrist-worn IMU sensors is described in [42]. The IMU-based sensors are a 3-axis accelerometer and a 3-axis gyroscope fused with four sEMG sensors. The performances of 4 various classifiers for the targeted gesture recognition are compared, including KNN, LDA, DT and SVM.

The performance of ring and smartwatch-mounted IMU sensors for hand and finger gesture recognition has been examined in [37]. This study investigates two IMU-based sensors, namely an accelerometer and a gyroscope. The results are presented for RF, SVM, KNN and NB classification algorithms. RF has shown the highest accuracy among all methods.

A study of the effect of fusing pressure sensors with IMU and flex sensors on the accuracy of hand gesture recognition in American Sign Language is reported in [9]. This study shows that without using the pressure sensor, 67.5% accuracy is achieved while adding the pressure sensor dramatically improves the recognition efficiency to 98.2%.

In [26], a 9-axis IMU, including an accelerometer, gyroscope and magnetometer, is mounted on the arm used for hand gesture recognition. SVM classifier is used in this study, and the accuracies of 99.63% and 88.43 are achieved in user-dependent and user-independent cases, respectively.

As a result of the related research, and considering the advantages of these types of classification over the use of a neural network, we decided to use KNN, SVM, and NB machine learning algorithms for our finger gesture recognition system and compare their performance. Furthermore, our experiments in comparing all of these when dealing with accelerometer and gyroscope data confirms the results of the related research and demonstrates the effectiveness of these methods. Compared to neural networks, these algorithms offer the following advantages [50]:

- The KNN, SVM, and NB algorithms are relatively simple and easy to interpret.
- The theoretical foundations of these algorithms have been extensively studied. Comparatively to a neural network, which may be considered to be a black box model, this can provide a greater level of confidence in the model's performance.
- These algorithms require less computational resources to train and predict than neural networks. Particularly in real-time applications, where a quick response is essential, this is extremely important.
- In comparison with neural networks, these algorithms are less sensitive to the quality of data, and they can handle noise and missing values more effectively.

There are several commercial hand and finger gesture recognition devices on the market that use various technologies such as infrared sensors, cameras, and IMUs to detect and recognize hand and finger gestures.

- The Myo technology is a device that uses EMG sensors to detect muscle activity in the arm, allowing it to recognize a wide range of hand and finger gestures, as well as arm movements [51]. Advantages of Myo include high precision , the ability to recognize a wide range of gestures. On the other hand, Myo requiring to wear the device on the arm. This can make it less comfortable to wear, more expensive.

- Rokoko smart glove is a commercial device that uses IMUs to track hand and finger movements [52]. It has advantages such as high accuracy in tracking hand and finger movements, providing realistic interactions, allowing for hands-free control of devices, and integration with other technologies. The system, however, has some disadvantages, such as the fact that it requires significant computational resources in order to process visual data and recognize gestures.

- Leap Motion is a commercial device that uses infrared cameras and LED lights to track the movement of hands and fingers, and it can recognize a wide range of gestures such as finger movements and hand poses [53]. It has the ability to recognize a wide range of gestures and can be integrated into various devices such as robots. Nevertheless, they are sensitive to variations in lighting and appearance as they rely on visual features such as colour, shape, and texture to recognize gestures.

Compared with commercially available motion detection and gesture recognition approaches, our smart ring system offers an acceptable level of accuracy for recognizing dynamic finger gestures, while also being robust against adverse environmental conditions, such as poor lightning. Additionally, low computational intensity required to detect our predefined finger gestures. the IMU sensor utilized in our design provides real-time feedback and responsiveness to finger gestures, making it suitable for applications that require high-speed interaction.

Table 2.1: Comparison of Previous Related Work

Ref	Dynamic/Static	Recognized Gestures	Interface	Sensor Type	Classification Method	Detection Accuracy
[49]	Static	5	Smartwatch	IMU	KNN LR DA	81% 66% 84%
[15]	Dynamic	10	Wrist-worn	IMU	SVM	86.99%
[43]	Dynamic	6	Hand-worn	IMU	Normalized covariance & Threshold adjustment	96%
[38]	Static	8	Ring	Capacitive IMU	-	95%
[54]	Static	12	Ring	Accelerometer	DT KNN NB	87% 86% 86%
[42]	Static	20	Hand-worn	sEMG IMU	KNN	96.13%
[48]	static	42	Ring	Gyroscope Microphone	DTW-KNN	92%-98%
[37]	Dynamic	12	Ring/Smartwatch	IMU	RF SVM KNN NB	93% 92% 88% 86%
[9]	Static	26	Hand-worn	IMU Flex Pressure	SVM	98%
[26]	Static	8	Arm-worn	IMU	SVM	99.6%
[35]	Dynamic	10	Pen	Accelerometer	NN	98%
[34]	Dynamic	8	Pen	IMU	DTW	97.9%
[33]	Static	37	Smartwatch	IMU	NB SL DT	90% 94.62% 88%
[32]	Static	5	Smartwatch	IMU	DTW-KNN	87%
[44]	Dynamic	6	Smartphone	IMU	DTW	-
[28]	Dynamic	QWERTY Keyboard	Ring	Accelerometer Multi-line Detector Optical Displacement	HMM	98%

Chapter 3

System Components and Implementation

3.1 Introduction

This chapter presents an overview of the proposed system for multi-finger gesture recognition and describes its components and implementation.

3.2 System Structure

Figure 3.1 illustrates the structure of the proposed system for multi-finger gesture recognition in this project. The system contains three main parts, namely:

- The thumb finger ring
- The point finger ring
- The wrist band

Figure 3.2 shows how these three parts are worn on the fingers and wrist in a real-life application.

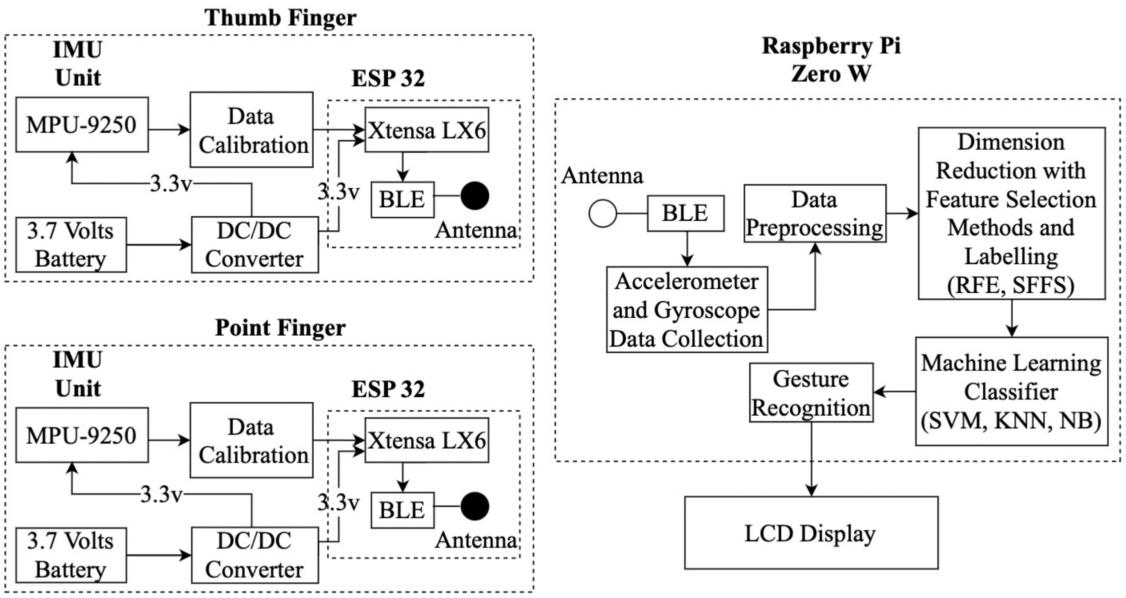


Figure 3.1: Finger gesture recognition system functional block diagram.

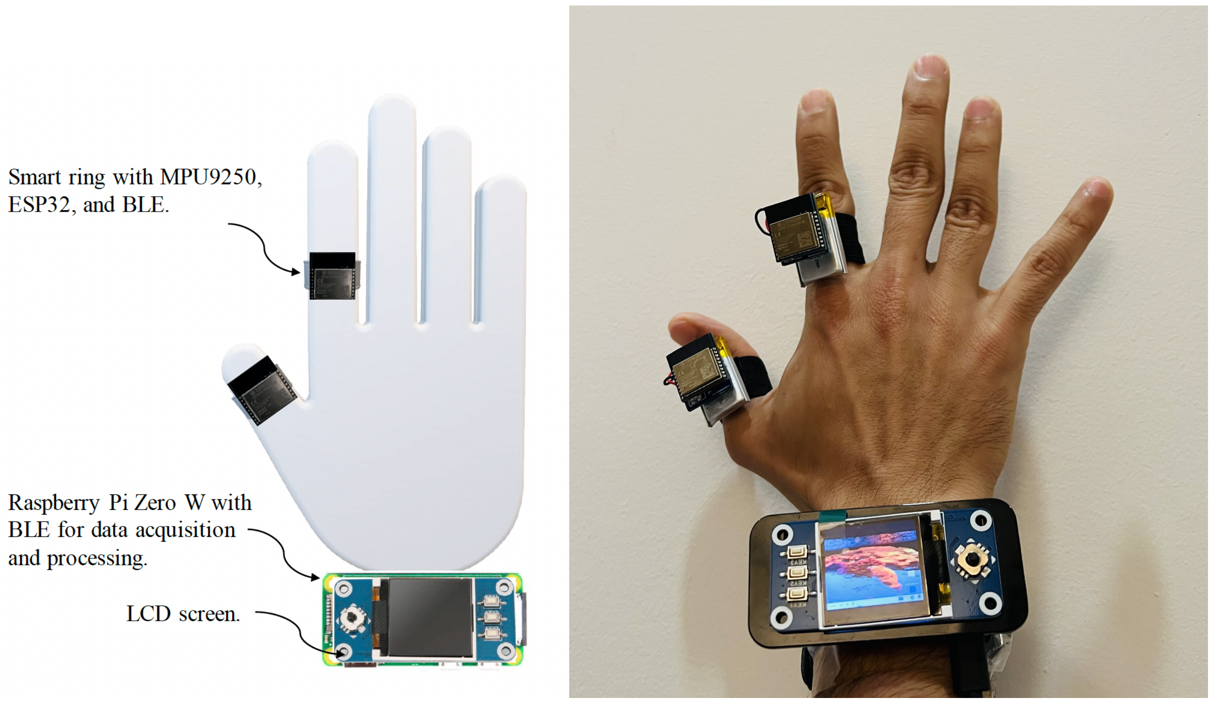

Smart ring with MPU9250, ESP32, and BLE.
Raspberry Pi Zero W with BLE for data acquisition and processing.
LCD screen.

Figure 3.2: Overall fingers gesture detection and recognition system.

3.2.1 Smart Rings

This project fabricated two identical smart rings to be worn on the thumb and point fingers, as shown in Figure 3.2. Each of these smart rings is equipped with an IMU sensor and Bluetooth transmitters to measure and send data needed for finger gesture recognition to the wristband. The components used in the smart rings are introduced in detail in Section 3.3. The designed smart rings in this project benefit from low weight and small size and are convenient to use. Their wireless connectivity feature eliminates any wire that could limit user motions. Figure 3.3 shows the appearance of the fabricated rings.

The modular approach we follow is similar to [40], which leads to one or multiple measurement devices that can be worn depending on the application where rings are used. In comparison with current glove-based systems, which are inherently non-modular [55, 56], physical constraints are reduced, especially when monitoring all five fingers is not necessary.

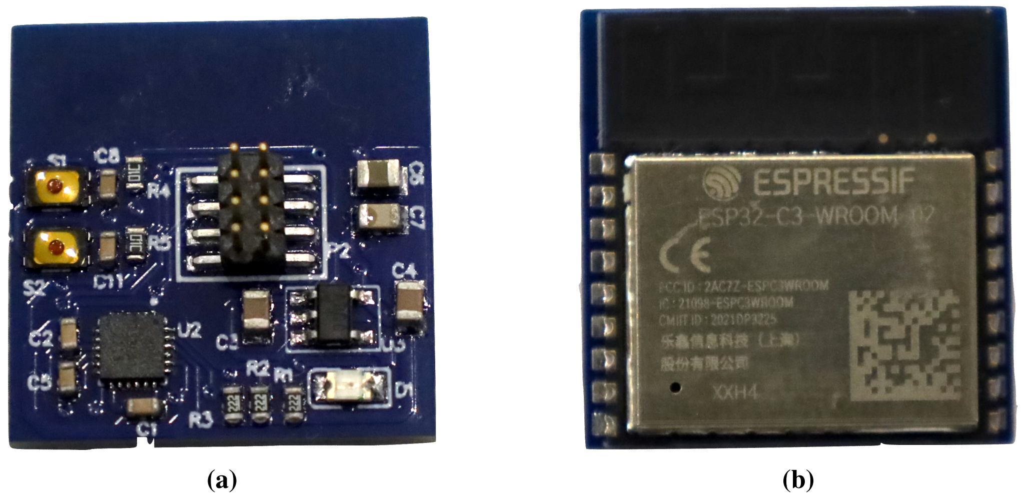


Figure 3.3: smart ring hardware module: (a) Main board, back view; (b) Main board, front view.

3.2.2 Wrist Band

In order to recognize the multi-finger gestures in this study, a processing unit for machine learning algorithm implementation and an LCD system to show the results are fabricated to a wrist-worn band. This wristband receives the data transmitted by both smart rings, processes them through the implemented machine learning algorithms and displays the recognized gesture on the screen as a result. A Raspberry Pi processor is used as the processing unit. The utilized Raspberry Pi processor and the LCD system are introduced in more detail in Section 3.3.

3.2.3 Wireless and BLE Features

The proposed smart rings in this project benefit from wireless connectivity and BLE features. Wireless connectivity provides the following major advantages [57]:

- Mobility: The length of the wires is not a limitation. The mobility of wireless devices allows users to stay connected while moving around. This is an essential feature in many applications involving the users' motion.
- Convenience: Using wireless devices is much more convenient than their wired counterparts. This feature provides a user-friendly experience.
- Almost zero setup time: A wireless device usually needs only to be turned on and used to become set up.

Bluetooth is a global wireless technology standard that allows devices to communicate over radio waves at short distances and low power. This technology was initially developed to replace serial data cables that connect various devices. Almost all mobile phones, tablets, and laptops are equipped with Bluetooth technology [57]. Therefore, our proposed smart rings can be compatible with many other smart electronic devices using Bluetooth technology. BLE is a new technological enhancement to Bluetooth that was added as a Bluetooth 4.0 specification. In keeping with its name, it is targeted at ”ultra-low power” devices. Bluetooth's low power features are extended even further with BLE. Devices compliant with BLE standards consume very low power so that they can operate much longer on smaller batteries and with less frequent charging. This is useful in applications where battery life is critical and frequent charging is difficult [57].

3.2.4 Programming and Charging Boards

Besides the smart rings and wristband system, another PCB board is used for charging and programming the measurement boards. The programming board modifies or debugs software throughout the designated pin headers. As an additional benefit, it enables a sensor board powered by USB to start up and operate without a battery for a considerable period. The battery board, whose pin header aligns with the main board, is also fabricated to increase the convenience of charging and powering up the board while keeping its size as small as possible. Figure 3.4 illustrates the top and back views of the battery charger and programming board. When the battery has been charged, as shown in Figure 3.5, the measurement board and battery board are connected, and the measurement device is ready to begin transmitting data.

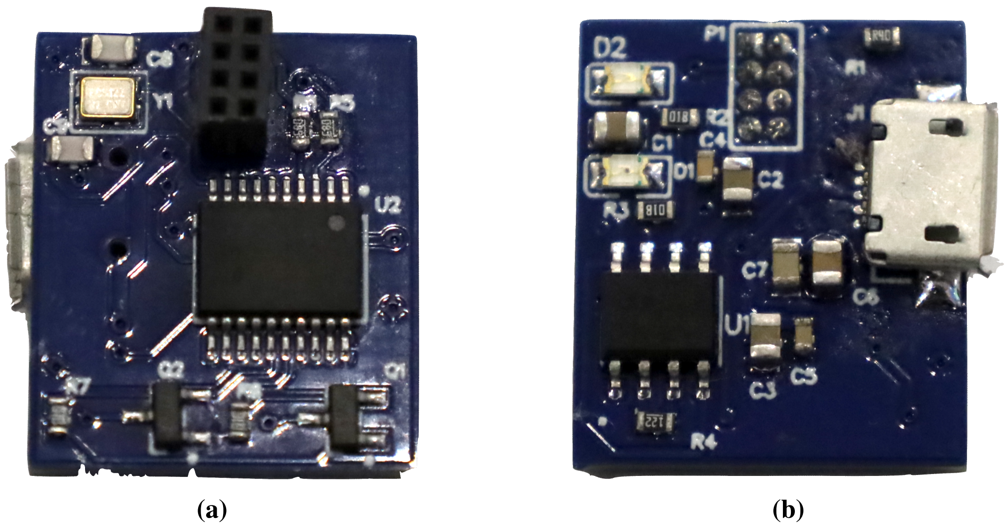


Figure 3.4: programming and charger board: (a) back view; (b) front view.

3.3 System Components

3.3.1 Raspberry Pi Zero W

The proposed system utilizes a Raspberry Pi Zero W as its microprocessor unit to process the data transmitted by the smart rings and execute the machine learning algorithms to recognize and report finger gestures. The Raspberry Pi Zero W extends the Pi Zero family by adding wireless LAN and Bluetooth connectivity. Thus, it is a suitable choice for this project which features wireless smart rings. Figure 3.6 shows the appearance of Raspberry

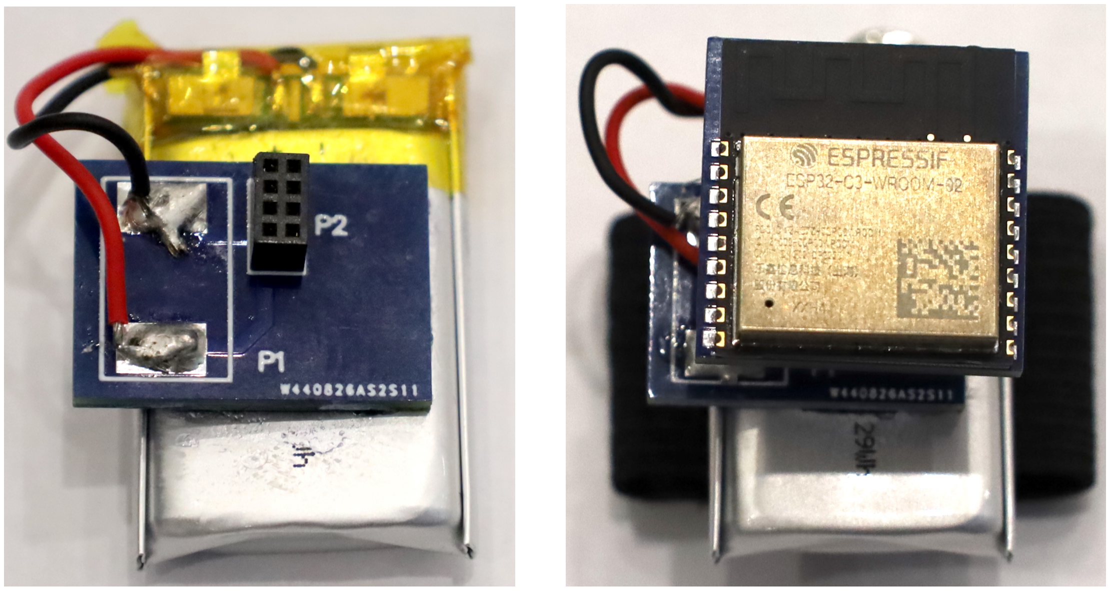


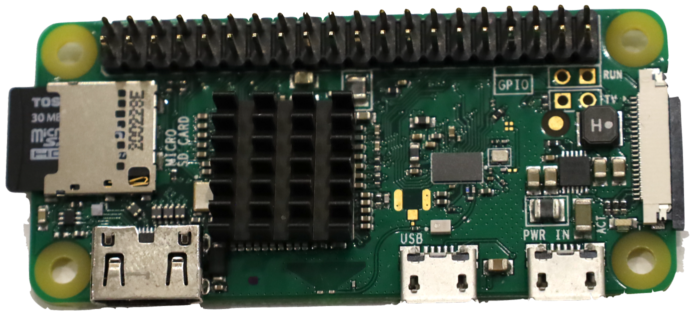
Figure 3.5: fabricated battery board.

Pi Zero W. The technical specification of this component is as follows [58]:

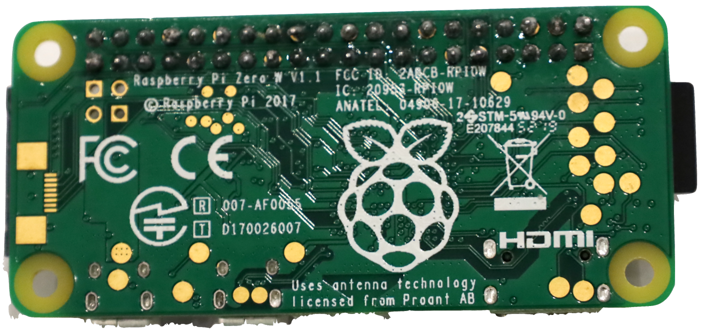
- 802.11 b/g/n wireless LAN
- Bluetooth 4.1
- BLE
- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI port and micro USB On-The-Go (OTG) port
- Micro USB power
- HAT-compatible 40-pin header

3.3.2 MPU 9250

The smart rings in this project utilize MPU 9250, IMU-based sensors. These sensors consist of three independent vibratory MEMS rate gyroscopes and a three-axis MEMS



(a)



(b)

Figure 3.6: Raspberry Pi Zero W used as the system microprocessor: (a) top view; (b) back view.

accelerometer. The vibratory MEMS rate gyroscopes detect rotation about the *x*-, *y*-, and *z*-axis as shown in Figure 3.7. A capacitive pickoff detects vibrations caused by the Coriolis Effect when the gyros are rotated about one of the sense axes. The signal is amplified, demodulated, and filtered to produce a voltage proportional to the angular rate. Individual 16-bit Analog-to-Digital Converters (ADCs) are used on the chip to sample each axis. It is possible to program the full-scale range of the gyro sensors as ±250, ±500, ±1000, or ±2000 degrees per second (DPS). Moreover, a wide range of cut-off frequencies is available, and the ADC sample rate can be set from 8,000 down to 3.9 samples per second [59].

There is a separate proof mass for each axis of the 3-Axis MPU-9250 accelerometer. To measure the acceleration in each axis, capacitive sensors detect the differential displacement on the corresponding proof mass. The architecture of MPU-9250 is so that the susceptibility of an accelerometer to fabrication variations and thermal drift is reduced. When placed on a level surface, the device will measure 0g on the *x*-, *y*- axes and +1g on the *z*-axis. Factory calibration provides nominal voltage independence for the accelerometers' scale factor. Sigma-delta ADCs provide digital outputs for each sensor. It is possible to adjust the digital output range to ±2g, ±4g, ±8g, or ±16g. Using the full-scale gyroscope range of ±500°/sec (DPS) and the full-scale accelerometer range of ±2g, it is possible to acquire sufficient precision while tracking finger motions. Table 3.1 summarizes the MPU-9250 sensor specification [59].

We calibrated the MPU9250 is by holding the hand parallel to the ground for a few seconds. In this situation, change in angular velocity in all axes should be minor since there is no movement, and accelerometer data should be equal to 1g on the *z*-axis and 0g for the *x*-axis and *y*-axis.

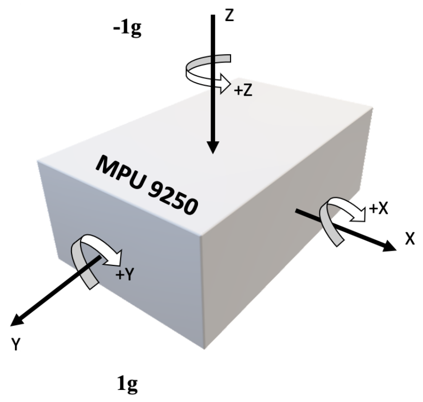


Figure 3.7: Orientation of MPU 9250 sensor and its 3 axis data reading.

Table 3.1: MPU-9250 Technical Specification [59].

MPU 9250	Accelerometer	Gyroscope
Scale Range	±2g, ±4g, ±8g and ±16g	±250, ±500, ±1000, and ±2000°/sec
Operatin Current	450 μA	3.2mA
Sleep Mode Current	8μA	8μA
Supply Voltage Range	2.4 – 3.6V	2.4 – 3.6V

It is possible to determine the orientation of an object in relation to Earth's gravity by using a gyroscope. A crucial difference between an accelerometer and a gyroscope is that the gyroscope can detect rotation movements. In our design, since the IMU sensor is mounted in the back of the main board, the orientation of the IMU unit is depicted in Figure. 3.7. For instance, while performing 90 deg clockwise gesture, we understand that a rotation around a specific axis, such as the y-axis, could cause both accelerometer and gyroscope data to change. The gyroscope exhibits fluctuating angular velocity, starting with an increase during rotation around the y-axis and decreasing to zero after the gesture is completed. In the final point, the accelerometer data on the x-axis indicates 1g as we align with the Earth's gravity, while the accelerometer data on the other two axes indicates zero. Using this approach to interpret the IMU data enables us to define and differentiate more complex gestures, either static or dynamic.

3.3.3 ESP32 Module

Each smart ring contains an ESP32 module with an Xtensa LX6 microprocessor and a BLE module to transmit its collected data to the wristband, where the processing unit is located. We have used the ESP32-C3-WROOM-02 module in this project. It supports BLE for wireless data transmission. Figure 3.8 and 3.9 show the external appearance and internal block diagram of this module, respectively.

Figure 3.8: External casing of ESP32-C3-WROOM-02 module [60].

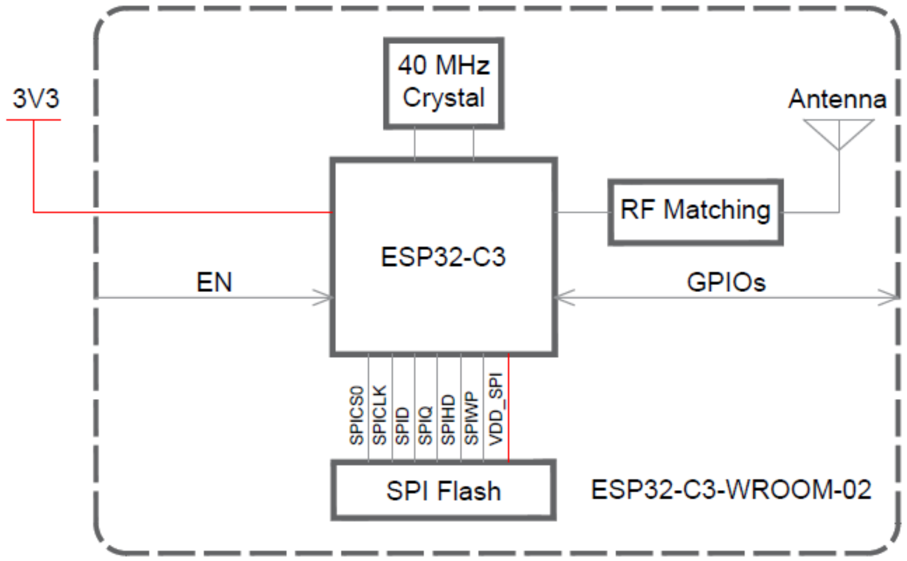


Figure 3.9: Internal block diagram of ESP32-C3-WROOM-02 module (source: [60]).

The technical specification of the utilized ESP32 module are as follows [60]:

- 2.4 GHz WiFi
- (802.11 b/g/n) and Bluetooth 5 module
- Built around ESP32C3 series of SoCs, RISCV singlecore microprocessor
- 4 MB flash
- 15 GPIOs
- Onboard PCB antenna

3.4 Target Finger Gesture and Corresponding Signals

In Table 3.2, you will find an overview of the most common finger and hand gestures performed in recent studies. Smart rings, gloves, arm bands, and wrist bands are commonly used as interfaces to recognize these gestures. With the combination of accelerometer and gyroscope in our design, we are able to identify the common gestures observed in recent studies, which are finger movements up and down, right and left. Furthermore, like similar studies, our one-ring system is capable of detecting angular movements such as rotations of 90 degrees clockwise and counterclockwise. However, when two rings are used together, three new finger gestures are introduced, including zooming in, zooming out, and rotating two fingers (Figure 3.10). As well as these gestures, our design has demonstrated the capability of interpreting and collecting two rings data, which may be easily expanded to include additional rings with different functionality and even define more complex multi-finger gestures such as those used in sign language or other similar studies.

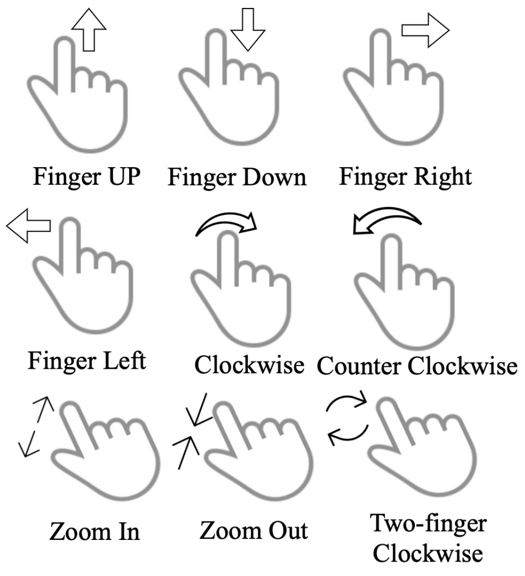


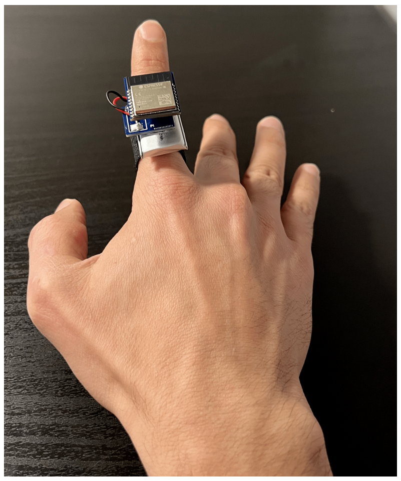
Figure 3.10: Target gestures.

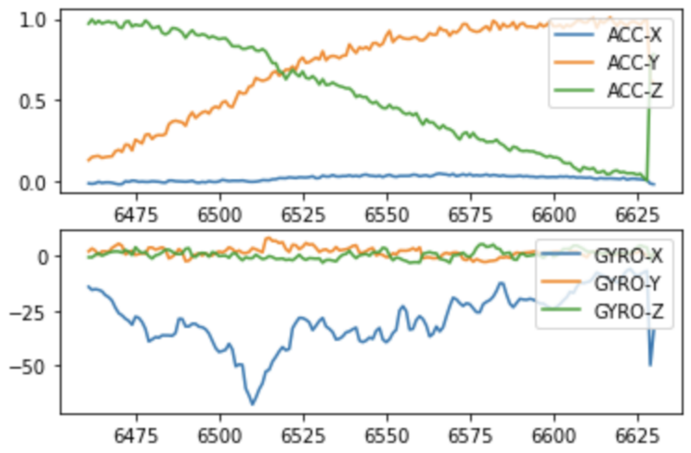
Figures 3.11 - 3.19 illustrate real-time signal variation produced by the IMU units while performing the nine target gestures of this project. With each gesture, the data shows a combination of both gyroscope and accelerometer fluctuations in all the 3 axes.

The next chapter discusses feature extraction and selection from the recorded IMU-based signals to effectively recognize multi-finger gestures using machine learning algorithms.

Table 3.2: Hand and finger gestures used in similar studies

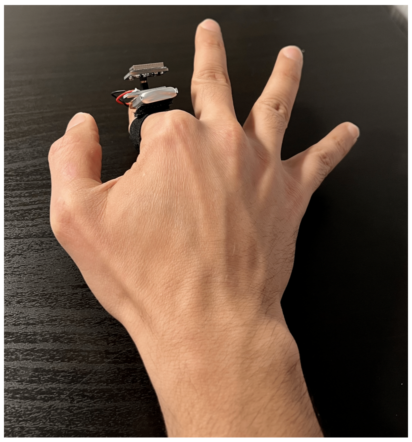
Hand and finger gestures	**Reference**
Wrist: up, down, left, right, left up, left down, right up, right down, rotation right,rotation left	[15]
Hand: pinch, tap, rub finger, squeeze, wave	[32]
Pen: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 numeric trajectory	[35]
Ring: circle, rectangle, bloom, scissors, slide thumb over nail, slide thumb over middle finger, flick, snap	[38]
Hand: up, down, left, right, clockwise, counter clockwise	[43]
Wrist: hip, press, shake, beat, curve	[49]
Ring: crook, unbend,finger down, wrist down, l-rotate, finger l-shift, wrist l-shift, r-rotate, finger r-shift, wrist r-shift, finger up, wrist up	[54]

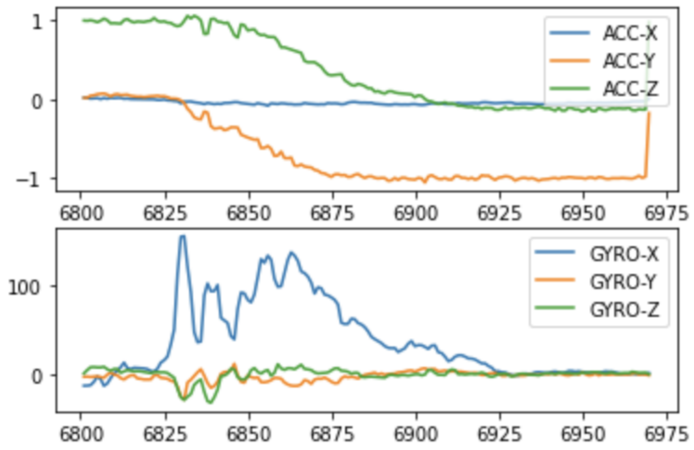

(a) Performed finger-up gesture.



(b) Accelerometer and gyroscope raw data signals during the finger-up gesture.

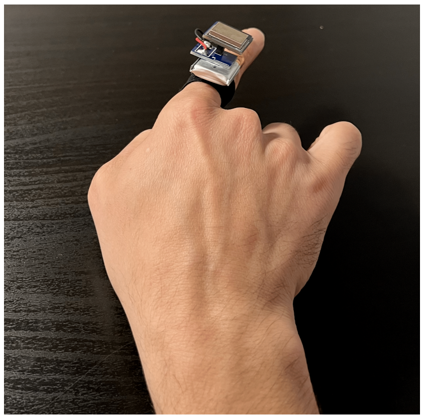
Figure 3.11: Finger-up signal demonstration.

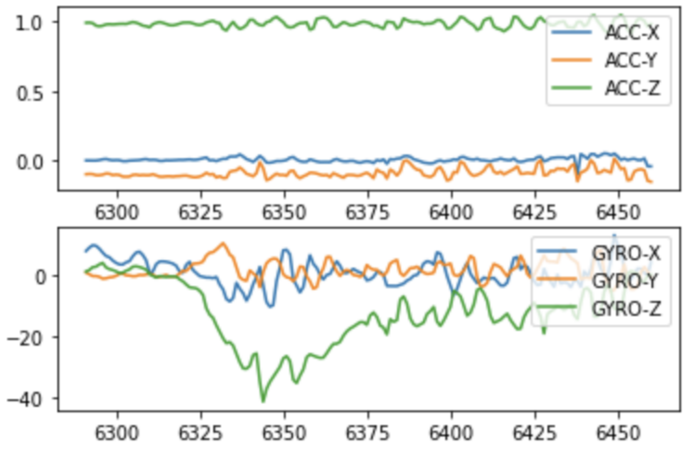

(a) Performed finger-down gesture.



(b) Accelerometer and gyroscope raw data signals during the finger-down gesture.

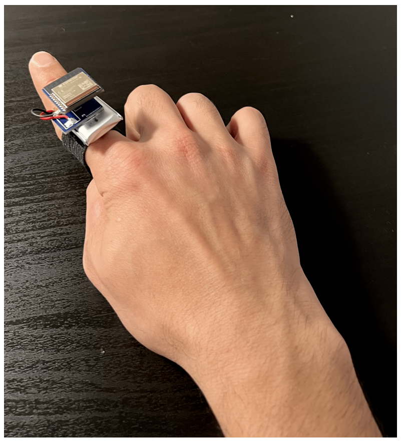
Figure 3.12: Finger-down signal demonstration.


(a) Performed finger-right gesture.

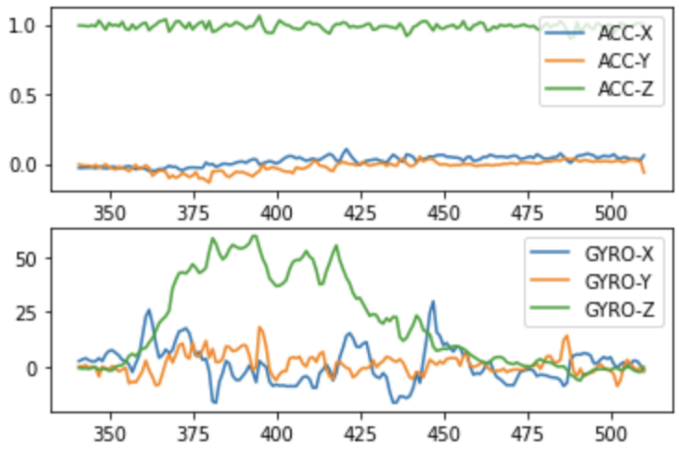


(b) Accelerometer and gyroscope raw data signals during the finger-right gesture.

Figure 3.13: Finger-right signal demonstration.

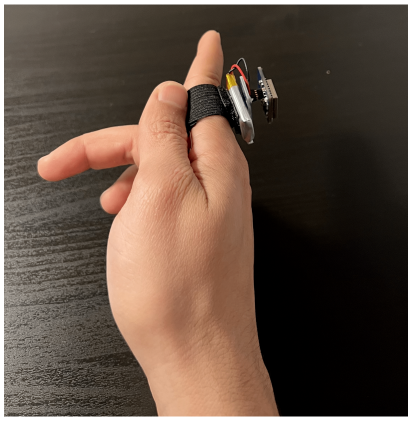


(a) Performed finger-left gesture.

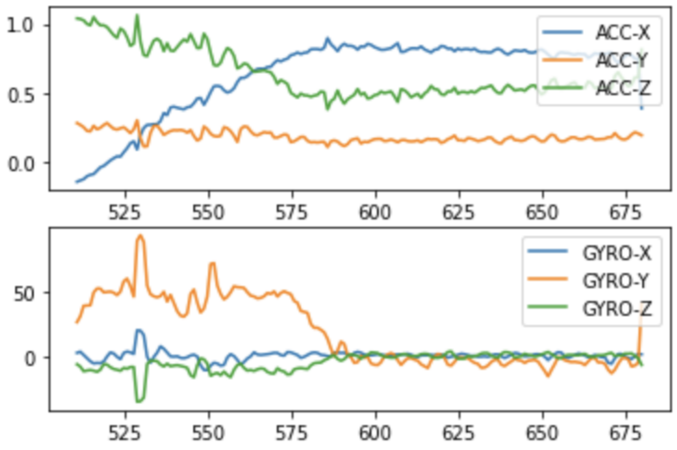


(b) Accelerometer and gyroscope raw data signals during the finger-left gesture.

Figure 3.14: Finger-left signal demonstration.

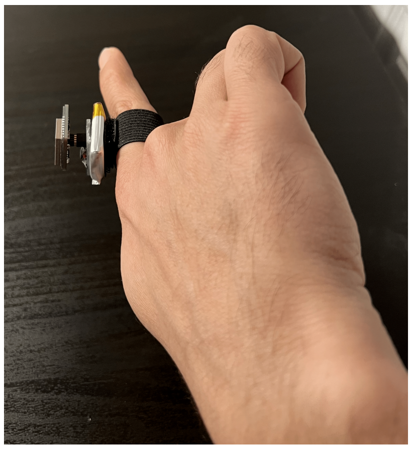


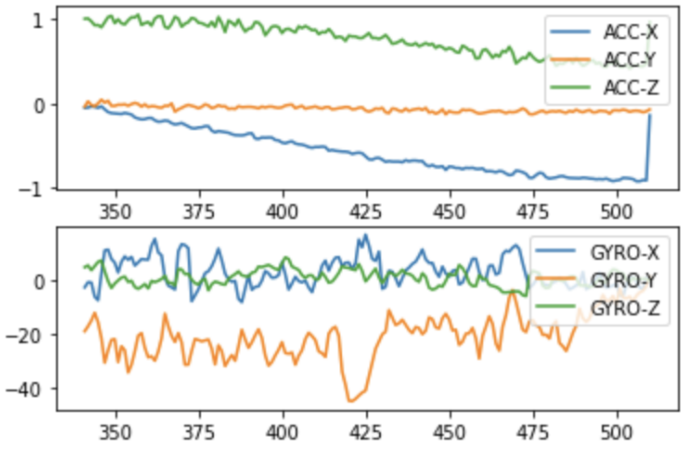
(a) Performed finger-clockwise gesture.



(b) Accelerometer and gyroscope raw data signals during the finger-clockwise gesture.

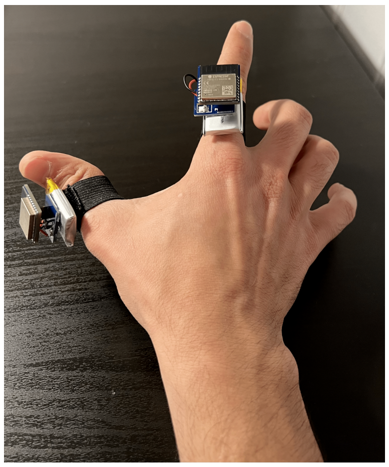
Figure 3.15: Finger-clockwise signal demonstration.

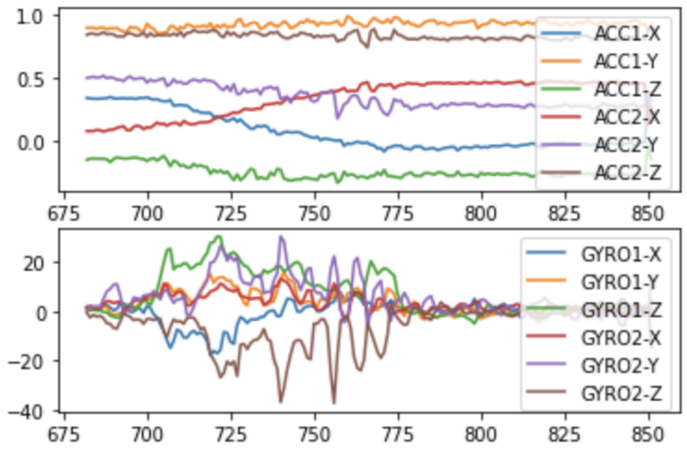

(a) Performed finger-counterclockwise gesture.



(b) Accelerometer and gyroscope raw data signals during the finger-counterclockwise gesture.

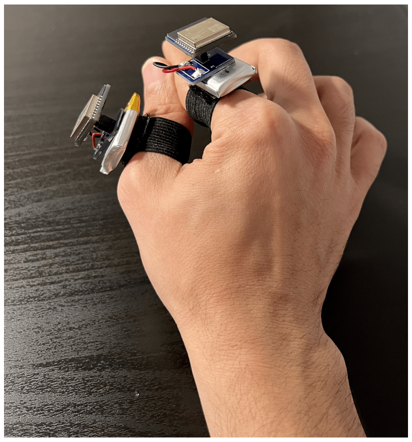
Figure 3.16: Finger-counterclockwise signal demonstration.


(a) Performed zoom-in gesture.

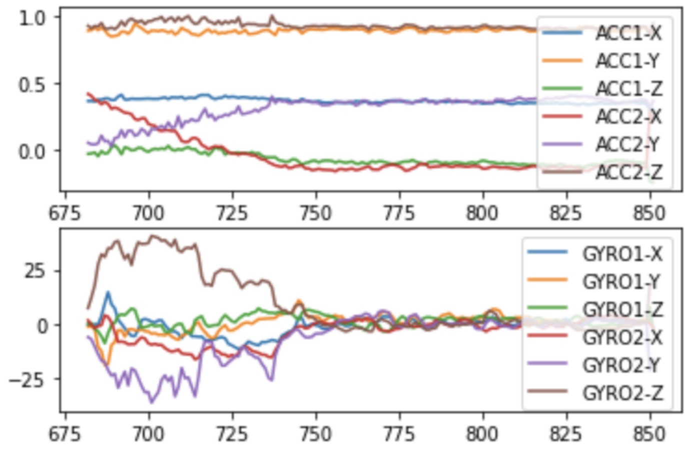


(b) Accelerometer and gyroscope raw data signals during the zoom-in gesture.

Figure 3.17: Zoom-in signal demonstration.

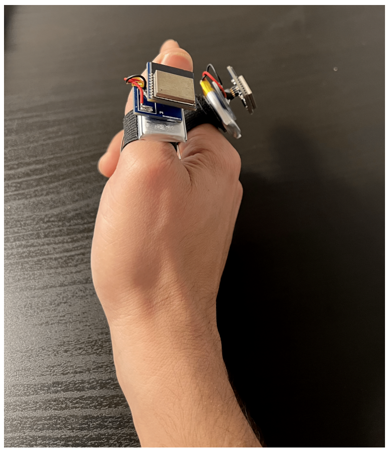


(a) Performed zoom-out gesture.

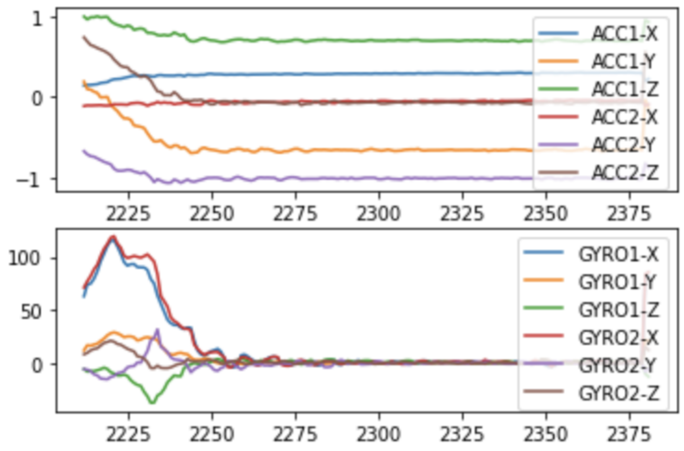


(b) Accelerometer and gyroscope raw data signals during the zoom-out gesture.

Figure 3.18: Zoom-out signal demonstration.



(a) Performed two-finger 90-degree clockwise gesture.



(b) Accelerometer and gyroscope raw data signals during the two-finger 90-degree clockwise gesture.

Figure 3.19: Two-finger 90-degree clockwise signal demonstration.

Chapter 4

Feature Selection and Machine Learning Algorithms

4.1 Introduction

Feature selection is an essential concept in machine learning model development. The most suitable set of features should be selected out of all the available features in order to develop an efficient machine learning model. This chapter introduces the standard feature selection methods utilized in finger and hand gesture recognition applications in Section 4.2. The feature selection algorithms utilized in this project are described in detail in Section 4.3. Then a review of extracted features is presented in Section 4.4. Finally, the machine learning algorithms used in this project are presented in Section 4.5.

4.2 Review of Feature Selection Methods

Feature selection aims to select d features from a given set of D measurements, $d < D$, without significantly impairing the recognition ability [61]. The importance of feature selection methods becomes apparent when a dataset contains many features. Having a high degree of dimensionality can significantly lengthen the training time of a machine learning model, which can lead to an overfitted model that is also complex. Feature selection is a solution to overcome these challenges. Features selection improves machine learning classifier scalability, prediction, and generalization performance. Additionally, it reduces computational complexity, storage requirements, and costs [62].

In our case, the raw data from accelerometers and gyroscopes are preprocessed before being used as a training dataset. In the case of one ring, a single gesture generates six different signals (three signals from the three-axis acceleration sensor and three signals from the three-axis gyroscope). Therefore, two rings generate twelve different signals. A signal's characteristics can be represented by calculating multiple statistical features for each signal. This process results in extracting a combination of gesture attribute from the raw data. Notably, the feature vector ultimately contains weak or redundant information about the signal and potent features that provide valuable information about the signal. Feature selection is performed to identify those solid features and discard the others. This reduces the feature vector size and prevents overfitting and high computational costs.

Feature selection methods are categorized into three main categories, namely:

- Filter Methods
- Wrapper Methods
- Embedded and Hybrid Methods

Before introducing the feature selection methods which are chosen in this project, we present a review on the main categories of feature selection methods.

4.2.1 Filter Methods

In feature selection based on filter methods, the model begins with all features and uses defined performance measures to select the best features subset [63]. Some of the standard filtering methods include:

- Information gain
- Correlation

- Fast correlation-based filter (FCBF)
- Chi-square
- Inconsistency criterion
- Minimum redundanc, maximum relevance (mRmR)
- Fisher score
- Feature weighting K-means
- Spectral feature selection (SPEC)
- Relief algorithm
- Linear Discriminant Analysis (LDA)
- Wilcoxon Mann Whitney test

The main disadvantage of filtering methods is ignoring the performance of the machine learning classification system employing the set of selected features. Although due to their general nature, filter methods are suitable for all situations, the mentioned disadvantage causes them to yield less promising results compared to other feature selection typically approaches.

4.2.2 Wrapper Methods

Wrapper methods evaluate feature subsets based on the quality of the performance on a machine learning algorithm for classification or clustering (e.g. SVM or NB), which is regarded as a black box evaluator. Due to their dependency on the modelling machine learning algorithm, wrappers are slower to find sufficiently good subsets compared to filter methods. Additionally, feature subsets selected using wrapper methods are biased toward the machine learning algorithm used for evaluation. The selected features result in better performance for the chosen classifier but worse results for another, mainly when their properties significantly differ. Therefore, it is necessary to use both an independent validation sample and another learning algorithm after finding the final subset to obtain a reliable general error estimate [64].

4.2.3 Embedded and Hybrid Methods

Hybrid feature selection methods were proposed to combine the best properties of filters and wrappers. The first step in hybrid feature selection strategies is to reduce the feature space dimension space using a filter method, possibly obtaining several candidate subsets. Afterward, the best candidate subset is determined using a wrapper [64]. Hybrid methods are computationally more efficient than the wrapper method and more accurate than filter and wrapper methods. However, they are unsuitable for high dimensional data. Many hybrid feature selection methods have been proposed recently. For example, several hybrid methods have been proposed based on the recursive feature elimination (RFE) method [65].

4.2.4 Search Strategies Utilized in Feature Selection

Search strategies are the core of feature selection algorithms, whether filter, wrapper or hybrid type. The utilized search strategies can significantly affect the performance of the feature selection process. To achieve optimum results, a search strategy should utilize local search capabilities and perform computationally efficiently [62]. Figure 4.1 illustrates the main stages in the feature selection process. Determination of a starting point and a direction for the search is the first step of the feature selection process. In general, the three types of search directions are forward search, backward search, and random search. In forward searching, new features are added recursively in each iteration, starting from an empty set. On the other hand, a backward elimination search does the opposite: starting with all features and iteratively removing features until the desired subset is reached. Another variant utilizes a random search method, which iteratively adds and removes features to build a subset of features.

The second stage in the feature selection process is choosing a search strategy. There

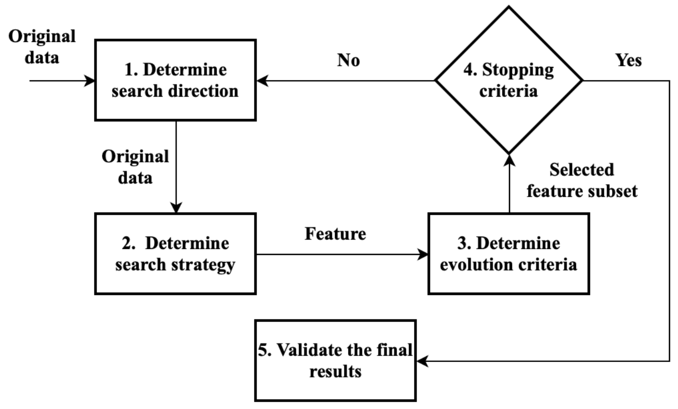

Original data
1. Determine search direction
No
4. Stopping criteria
Yes
Original data
Selected feature subset
2. Determine search strategy
Feature
3. Determine evolution criteria
5. Validate the final results

Figure 4.1: Stages in feature selection process.

are three types of search strategies: exponential, sequential, and randomized. Exponential search strategies are exhaustive and require 2^N combinations of feature selection for N features. Randomized search strategies have been introduced to overcome the drawbacks of exponential and sequential strategies. A sequential search adds or removes features sequentially from an empty set or a complete set. Table 4.1 summarizes the most common search strategies in the field of feature selection based on [64].

4.3 Selected Feature Selection Method

When the relationship between the features and the target variable is complex and cannot be adequately captured with a simple heuristic, wrapper methods are commonly utilized. A wrapper method that utilizes a specific machine learning algorithm, for example, may be able to capture interactions and non-linear relationships between features in addition to providing more accurate feature selection than a filter method. Wrapper methods can also be useful if the data has a high degree of dimensionality or a large number of features. The advantages of using the wrapper method, as well as the high dimensionality of our one-ring and two-ring feature vectors, led us to choose wrapper methods over filter methods in this study.

In [66], the authors present a wireless activity recognition system for patient monitoring using a body-worn accelerometer. Feature selection is performed using Relief-F

Table 4.1: Search Strategies for Feature Selection

Algoritm Group	Algoritm Name
Exponential	•Exhaustive search •Branch-and-bound
Sequential	•Greedy forward selection or backward elimination •Best-first search •Linear forward selection •Sequential Forward Selection (SFS) •Sequential Backward Selection (SBS) •Sequential Forward Floating Selection (SFFS) •Sequential Backward Floating Selection (SBFS) •Beam search (and beam stack search) •Race search
Randomized	•Random generation •Simulated annealing •Evolutionary computation algorithms (e.g. genetic, ant colony optimization) •Random hill-climbing •Las Vegas Algorithm •Tabu search •Scatter search

and sequential forward floating search (SFFS). Compared to Relief-F, SFFS selected almost half as many features and provided higher accuracy. In [67], human movement data were classified using features selected by a feature selection methodology. A comparison was conducted between four commonly used methods for selecting and classifying human movement features based on IMU data sets. Random Forest recursive feature elimination (RF-RFE) was also employed to rank features in order of importance. RF, SVM, KNN, and NB are the four classification algorithms used in this study. Considering the proven advantages of employing these feature selection methods to reduce dimensionality and select the appropriate feature in a large dataset, we decided to follow the recommendations of [66, 67] by using RF-RFE and KNN-SFFS as our choice of feature selection methods.

4.3.1 RF-RFE

RF-RFE is a wrapper feature selection method based on a random forest machine learning model and recursive feature elimination search strategy. Based on [65], the RF-RFE algorithm pseudocode can be summarized as presented in Algorithm 1.

Algorithm 1 RF-RFE Algorithm Pseudocode

```
begin
    Read the training dataset T
    Read the set of available features: $F = \{f_1, f_2, ..., f_n\}$
    Construct the set of remaining features: $R \leftarrow F$
    while $R \neq \emptyset$ do
        Train a RF model based on $R$
        Evaluate feature importance and determine the least important feature ($f^- \in R$)
        $R = R - f^-$
    end
    return Feature Ranking List
end
```

4.3.2 KNN-SFFS

KNN-SFFS is a wrapper method based on the KNN machine learning model and SFFS search strategy. The SFFS search algorithm improves the performance of the simpler SFS algorithm. An additional exclusion or inclusion step is added to the floating algorithm so that features that have been included can be removed in order to sample a more significant number of possible feature subset combinations. Note that this step is conditional and only occurs if the resulting feature subset is assessed as being better by the criterion function after being removed [61].

The pseudocode of the SFFS algorithm can be summarized as presented in Algorithm 2. It is noted that $J(X_k)$ represents the feature selection criterion function that is evaluated based on the performance of the utilized machine learning model trained by X_k. Moreover, Y is the set of all the features, $Y = \{y_1, y_2, ..., y_D\}$, and $X_k = \{x_j | j = 1, 2, ..., k; x_j \in Y\}$ where $k = (0, 1, 2, ..., d)$ [61].

Algorithm 2 Sequential Forward Floating Selection (SFFS) Algorithm Pseudocode

```
begin
    Start with an empty set of features: $X_0 = \emptyset$, $k = 0$.
    while $k < d$ (desired number of features) do
        Step 1 (Inclusion):
        $x_{k+1} = arg\max J(X_k + x)$ where $x \in Y - X_k$
        $X_{k+1} = X_k + x_{k+1}$
        $k = k + 1$
        go to step 2
        Step 2 (Conditional exclusion):
        if $x_k = arg\min J(X_k)$ then
            go to step 1
        else
            while $k > 2$ do
                $x_r = arg\min J(X_k)$
                if $J(X_k) \geq J(X_k - x_r)$ then
                    go to step 1
                else
                    $k = k - 1$
                    $X_k^{'} = X_{k+1} - X_r$
                    $X_k = X_k^{'}$
                end
            end
        end
    end
    return $X_k$
end
```

4.4 Feature Extraction

Various features have been utilized in sensor-based finger and hand gesture recognition. These features can be categorized into three main classes, namely:

- Time domain features
- Frequency domain features

- Discrete domain features

These different classes of features are discussed in detail in the following.

4.4.1 Time Domain Features

The signals recorded with different types of sensors, i.e. IMU-based sensors such as accelerometers and gyroscopes, are time domain signals. Therefore numerous time domain function can be mathematically defined and calculated for these signals and utilized as features for gesture recognition. These time domain features are introduces in the upcoming subsections.

Envelope metrics

The basic following time domain metrics, which are called envelope metrics, despite their simplicity and low computional costs, have been proven to be effective and valuable metrics in activity or gesture recognition [49]:

- Min
- Max
- Median

Statistical metrics

Mean, variance and standard deviation are the common statistical metrics utilized in gesture recognition applications, which are formulated respectivwely as follows [68]:

$$\mu = \sum_{i=1}^{N} x_i. \tag{3}$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)^2. \tag{4}$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2}. \tag{5}$$

Correlation coefficient

Another metric known as correlation coefficient [69] or normalized covariance [43] is also utilized as a feature in gesture recognition. This metric is formulated as follows:

$$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}, \tag{6}$$

in which

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y). \tag{7}$$

Root mean square (RMS)

As another simple metric, root mean square (RMS) of a signal shows how the signal amplitudes change over time. It has been a common time domain feature in gesture recognition and is formulated as follows [70]:

$$x_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}. \tag{8}$$

Zero-Crossing Rate

Zero crossing rate (ZCR) is the number of times a waveform crosses the zero axis. It is calculated by counting how often the following conditions are met [71]:

$$\begin{aligned} \{X(t) < 0 \text{ and } X(t+1) > 0\} \text{ or } \{X(t) < 0 \text{ and } X(t+1) > 0\} \\ |X(t) - X(t+1)| \geq \epsilon. \end{aligned} \tag{9}$$

To avoid counting zero crossings because of noise, the ϵ threshold is defined.

Skewness and Kurtosis

Skewness is described as the degree of asymmetry of a signal and Kurtosis is described as the degree of spikiness of a signal [72]. These metrics are mathematically calculated as below [73]:

$$SKEW = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu}{\sigma}\right)^3. \tag{10}$$

$$KURT = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu}{\sigma}\right)^4. \tag{11}$$

Windowed Time Domain Features

Some studies in the field of sensor-based gesture recognition suggest that the recorded signal during a user's movement be divided into W time windows without overlap. Then some time domain statistical metrics are introduced for these windowed signals. Some of thesse metrics are as follows [66]:

Mean Trend:
The signal is divided into W time windows with no overlap. Then the mean of signal during each of windows is calculated and subtracted from the mean of the succeeding window. The Mean Trend metrics is then calculated as follows:

$$\mu T = \sum_{i=2}^{W} |\mu_i - \mu_{i-1}|. \tag{12}$$

Variance Trend:
the variance of signal during each of windows is calculated and subtracted from the variance of the succeeding window. The Variance Trend metrics is then calculated as follows:

$$\sigma T^2 = \sum_{i=2}^{W} |\sigma_i^2 - \sigma_{i-1}^2|. \tag{13}$$

Windowed Mean Difference:
The mean value of the signal during each window is subtracted from its overall mean. Then

the Windowed Mean Difference metric is calculated as below:

$$\mu D = \sum_{i=1}^{W} |\mu_i - \mu|. \tag{14}$$

Windowed Variance Difference:
The variance of the signal during each window is subtracted from its overall variance. Then the Windowed Mean Difference metric is calculated as below:

$$\sigma D^2 = \sum_{i=1}^{W} |\sigma_i^2 - \sigma^2|. \tag{15}$$

4.4.2 Frequency Domain Features

Frequency-domain metrics have been widely utilized to identify repetitive characteristics of a sensor signals. Fast Fourier Transforms (FFT) and Fast Time Frequency Transforms (FTFT) are common methods for performing frequency domain analysis of a time signal with a specific length or window. Other frequency-based representations have been used in addition to the FFT and its spectral representation, e.g. Wavelet transforms. Using wavelet transforms, a time-domain signal can be decomposed into a set of weighted orthonormal vector bases or coefficients. In comparison to more established FFT methods, these transforms offer computational advantages. Spectral Energy and Entropy are two famous frequency domain features [74].

4.4.3 Discrete domain features

Dynamic time warping (DTW)

The dynamic time warping (DTW) metric is used to measure the similarity of two sequences, which may vary in length and thereby correspond to different time bases. Particularly in the context of gesture recognition, DTW can be applied to finding similarity metrics between signals. It has been shown that IMU-based gesture recognition accuracy is improved through dynamic time warping by categorizing movements by comparing various temporal sequences that may change in speed [15].

4.5 An Analysis of KNN, SVM, and NB Algorithms

4.5.1 KNN

The nearest neighbors are a simple classifier that, in its simplest form, assigns the label of its nearest neighbors training labels to an observation x. As a result, we only need a distance measure $d(x, x')$ between observations to implement the nearest neighbors method. Figure 4.2 shows a simple one nearest neighbor classifier, which uses one of three labels depending on whether the query point x is closest to the star, diamond or triangle. Although the one-nearest-neighbour method is straightforward, it can be problematic when the data itself is noisy. In this case, a majority vote can be used to decide the class membership of x by pooling some neighbors (k-nearest neighbors). Based on the pseudocode of the KNN algorithm described in [75], the pseudocode of finger gesture recognition utilized in this project is presented in Algorithm 3.

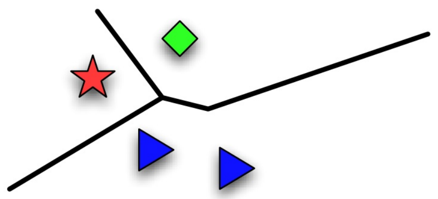


Figure 4.2: A sample 1 nearest neighbor classifier [75].

4.5.2 SVM

In this subsection we introduce the mathematical formulation of SVM method based on [75]. A hyperplane in a space $\mathcal{H}$ is described by:

$$\{x \in \mathcal{H} | \langle w, x \rangle + b = 0\}, \tag{16}$$

Algorithm 3 K-Nearest Neighbor Classification Pseudocode

```
begin
    Read train signals X, their corresponding finger gesture labels Y and input test signals x.
    for $i = 1$ to $m$ do
        Compute distance $d(x_i, x)$
    end
    Compute set $I$ containing indices for the $k$ smallest distance $d(x_i, x)$.
    return majority label of $\{y_i$ where $i \in I\}$
end
```

where $w \in \mathcal{H}$ and $b \in \mathbb{R}$. Using such a hyperplane, we can naturally divide $\mathcal{H}$ into two half-spaces: $\{x \in \mathcal{H} | \langle w, x \rangle + b \geq 0\}$ and $\{x \in \mathcal{H} | \langle w, x \rangle + b < 0\}$, and thus use it as the decision boundary of a binary classifier. Suppose we are given this set of training data for a binary classification task: $\{(x_1, y_1), ..., (x_m, y_m)\}$ with $x_i \in \mathcal{H}$ and $y_i \in \{\pm\}$. It is our goal to find a linear decision boundary parameterized by (w, b) such that $\langle w, x_i \rangle + b \geq 0\}$ whenever $y_i = +1$ and $\langle w, x \rangle + b < 0\}$ whenever $y_i = -1$. The scaling of w is fixed so that

$$\min_{i=1,\ldots,m} |\langle w, x_i \rangle + b| = 1. \tag{17}$$

In this case, the margin simply becomes $1/\|w\|$, as shown in Figure 4.3.

Therefore, we can formulate the problem of maximizing the margin as follows:

$$\max_{w,b} \frac{1}{\|w\|}$$
$$\text{s.t.} \;\; y_i(\langle w, x_i \rangle + b) \geq 1 \;\; \text{for all} \;\; i, \tag{18}$$

or equivalently

$$\min_{w,b} \frac{1}{2} \|w\|^2$$
$$\text{s.t.} \;\; y_i(\langle w, x_i \rangle + b) \geq 1 \;\; \text{for all} \;\; i, \tag{19}$$

this convex optimization problem has a quadratic objective function and linear constraints.

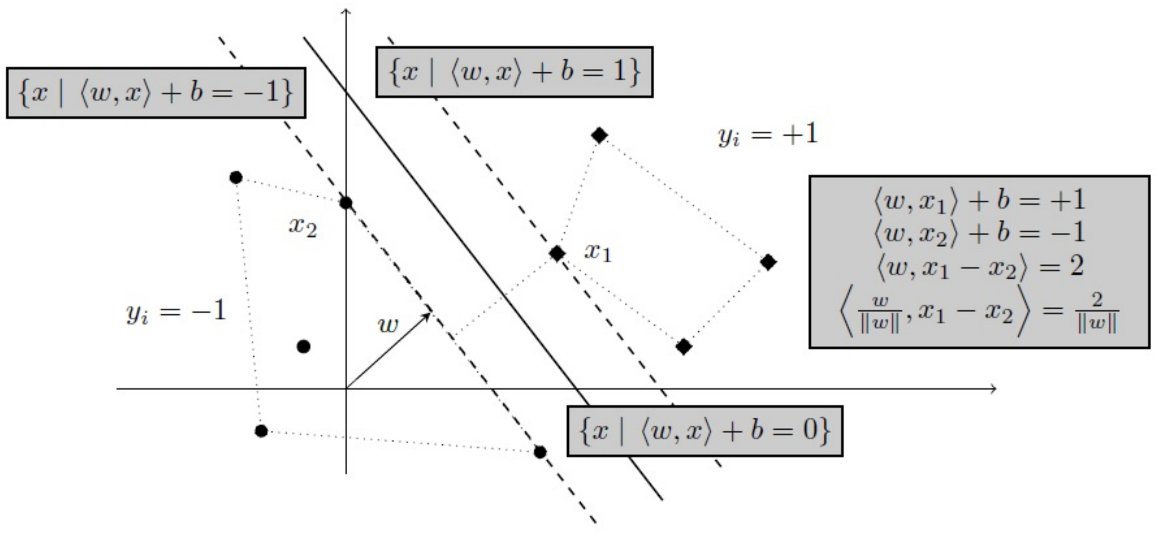

$\{x \mid \langle w,x\rangle + b = -1\}$
$\{x \mid \langle w,x\rangle + b = 1\}$
$y_i = +1$
$\langle w,x_1\rangle + b = +1$
$\langle w,x_2\rangle + b = -1$
$\langle w,x_1 - x_2\rangle = 2$
$\left\langle \frac{w}{\|w\|}, x_1 - x_2 \right\rangle = \frac{2}{\|w\|}$
x_2
x_1
$y_i = -1$
w
$\{x \mid \langle w,x\rangle + b = 0\}$

Figure 4.3: A sample linearly binary classication problem of separating the diamonds from the circles [75].

The training data were assumed to be linearly separable in the derivation of (4.5.2) – there is a hyperplane that correctly determines the classification of the training data. This type of classifier is called a hard margin classifier. This optimization problem cannot be solved if the data are not linearly separable. This situation can be mitigated by introducing non-negative slack variables ξ_i as follows:

$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \tag{20}$$

making ξ_i large enough will satisfy the constraints for any w and b and the optimization problem is modified as follows:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i$$
$$\text{s.t. } y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \text{ for all } i$$
$$\xi_i \geq 0, \tag{21}$$

where $C > 0$ is a penalty parameter. The Lagrangian for this optimization problem can be written as:

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\alpha_i(1 - \xi_i - y_i(\langle w, x_i\rangle + b)) - \sum_{i=1}^{m}\beta_i\xi_i, \quad (22)$$

where α_i and β_i are non-negative Lagrange multipliers. Taking gradients with respect to w, b and ξ and setting them to zero yields:

$$\begin{aligned} \nabla_w L &= w - \sum_{i=1}^{m}\alpha_i y_i \xi_i = 0 \\ \nabla_b L &= -\sum_{i=1}^{m}\alpha_i y_i = 0 \\ \nabla_{\xi_i} L &= \frac{C}{m} - \alpha_i - \beta_i = 0. \end{aligned} \quad (23)$$

By substituting (23) into the Lagrangian and doing some simplifications, the optimization problem can be formulated as:

$$\begin{aligned} \min_{\alpha} \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle &- \sum_{i=1}^{m}\alpha_i \\ \text{s.t.} \quad \sum_{i=1}^{m}\alpha_i y_i &= 0 \\ 0 \leq \alpha_i &\leq \frac{C}{m}. \end{aligned} \quad (24)$$

The pseudocode of finger gesture recognition utilized in this project is presented in Algorithm 4.

4.5.3 NB

NB classifier is one of the fundamental probabilistic classifiers used in machine learning [75]. Assuming that x is the input and that C is a multinomial variable that takes one of K states for the class code, Bayes' rule allows [76]:

Algorithm 4 SVM Classification Pseudocode

```
begin
    Load training signals $X$ and their corresponding labels $Y$
    Initialize $\alpha$
    while stopping criteria of the optimization problem is not met do
        for All $\{x_i, y_i\}, \{x_j, y_j\}$ do
            Optimize $\alpha_i$ and $\alpha_j$ (Equation (4.5.2))
        end
    end
    return optimum $\alpha_i$ and $\alpha_j$
end
```

$$P(C|x) = \frac{P(C)p(x|C)}{P(x)}. \tag{25}$$

Due to its analytical simplicity, the class-conditional densities are taken as normal densities:

$$p(x|C_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right]. \tag{26}$$

The parameters of these normal densities are calculated based on the training data as follows:

$$X = \begin{bmatrix} X_1^1 & X_2^1 & ... & X_d^1 \\ X_1^2 & X_2^2 & ... & X_d^2 \\ \vdots & & & \\ X_1^N & X_2^N & ... & X_d^N \end{bmatrix} \tag{27}$$

$$\mu = E(X) = [\mu_1, \mu_2, ..., \mu_d]^T \tag{28}$$

$$\sigma_{ij} = Cov(X_i, X_j) \tag{29}$$

$$\Sigma = Cov(X) = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2d} \\ \vdots & & & \\ \sigma_{d1} & \sigma_{d2} & \dots & \sigma_d^2 \end{bmatrix}, \tag{30}$$

where d is the number of features, and N is the size of the training set. Using the NB method, it is assumed that the inputs are independent and any possible dependency among them is ignored, i.e. $\sigma_{ij} = 0$ if $I \neq j$. Based on this assumption, we will have the following:

$$p(X|C) = \prod_{j=1}^{d} p(X_j|C). \tag{31}$$

The pseudocode of finger gesture recognition utilized in this project is presented in Algorithm 5, based on pseudocode of the NB algorithm described in [77].

Algorithm 5 Naive Bayes Algorithm Pseudocode

```
begin
    Read the training dataset
    Calculate the mean and standard deviation of the predictor features in each finger gesture class
    Read the testing dataset signals
    for $i = 1$ to d (where d is the number of features) do
        Calculate $P(x_i)$ (probability of predictor feature).
    end
    Calculate the likelihood of each finger gesture class.
    return The finger gesture with the greatest likelihood.
end
```

Chapter 5

Validation, Verification and Results

5.1 Introduction

This chapter presents the details of the training and testing process as well as the results of the recognition of multi-finger gestures using the proposed system. Section 5.2 describes the number of tests performed and stored to create the rich dataset that is needed for training our machine learning algorithm. Then in Section 5.3, the designated list of statistical features extracted from the raw data is exemplified. Following that, a review of the appropriate number of features for one-finger gestures and two-finger gestures is presented. Section 5.4 provides a definition of the evaluation metrics that are used to interpret our machine learning results, as well as an assessment of the accuracy of the machine learning classifier used in the one-finger and two-finger gesture cases.

5.2 Data Preparation

As only one user collected and evaluated the data, the proposed system is a single-user system. It is a common approach in machine learning research because it provides a controlled environment in which data and evaluations can be closely monitored and understood. An evaluation of one user is appropriate in our case as it allows us to have a detailed understanding of the user's behavior, preferences, and performance, which we can use to improve the system.

It should be noted that the proposed system can be easily extended to be a multi-user system by collecting and analyzing data from multiple users. This can be done by adding a user identification mechanism to the system and gathering data from multiple users. In addition, the results can be compared in order to evaluate the effectiveness of the system.

The following are additional benefits of using a single-user system:

- Simplicity: It is much easier to collect, store and analyze data from a single user.
- Control: You have full control over the data collection process, which ensures the quality and representativeness of the data.
- Flexibility: You can collect data in a variety of settings, such as in a controlled environment like the lab, and explore different scenarios.
- Reproducibility: Your results can be easily reproduced by other researchers who can replicate your study with the same user.

In order to implement machine learning algorithms, it is imperative to obtain a rich dataset. In order to achieve this objective, each finger gesture was performed and recorded 2,000 times. Considering our target nine gestures which were introduced in Figure 3.12, a total of 18,000 data samples are collected and stored for machine learning training purposes. Every gesture is executed within a three-second time frame (i.e., the time required for each gesture to be performed by a finger). The time frame is considered sufficient, as most recent studies use sample sizes ranging between one and three seconds [78].

Test data, training data, and evaluation data are three different types of data that are used in machine learning (ML) to evaluate the performance of a model.

- Training data is used to train the ML model. The model uses this data to learn the relationship between input features and output labels. The model is trained by adjusting the parameters to minimize the error between the predicted output and the true output.

- Test data is used to evaluate the performance of the trained model. The model is presented with new data that it has not seen before and its performance is measured by comparing the predicted output with the true output. The test data helps to evaluate the generalization performance of the model, which is its ability to perform well on new unseen data.

- Evaluation data is used to evaluate the model's performance with various techniques, such as cross-validation, which are used to estimate the model's generalization performance. Evaluation data can be used to compare different models, or different configurations of the same model, to select the best one.

In our case, the data set was chosen by considering the problem's characteristics, the data's representativeness, and the amount of data. We used a random sampling method to split the data into a training and evaluation set. We used 80% of the data for training, and 20% of the data for evaluating the model. This process is done to ensure that the data is representative of the problem and to reduce the chances of overfitting. Also, a test dataset is collected by performing each gesture and collecting an unseen dataset in order to evaluate the trained machine learning model with different feature sets.

The capability of machine learning models to learn is directly dependent on the quality of the data and the features that can be derived from it during the pre-processing phase. In order to eliminate noise and prevent false gesture recognition, the collected data are pre-processed before being fed into our model. The following steps are carried out during the pre-processing phase:

(1) Handling missing values

(2) Normalization

(3) Statistical feature extraction

(4) Feature selection

Normalization is usually performed prior to the extraction of statistical features during data preprocessing. It is because normalizing the data helps to scale all features so that they are on a similar scale, which is important for many machine learning algorithms. In the absence of normalization, certain features can have a far greater impact on the final result than others, resulting in poor performance. Moreover, normalization can reduce the impact of outliers in the data, which can also enhance performance. Normalizing the data makes it easier to extract meaningful statistical features from the data. Consequently, the model is now ready to be trained using the data.

5.3 Feature Selection Results

Candidate Features

The raw data can be used to extract a few statistical measures that can then be used to construct the dataset. Our real-time processing and recognition requires features with a low processing time and mathematical complexity. This is one important consideration in selecting appropriate features for our study. In our study, we selected the following statistical characteristics: minimum (min), maximum (max), average (avg), standard deviation (sd), RMS, variance (var), covariance (cov), signal entropy (ent), zero-crossing (zcr), and sum. In previous studies, these features have proven to be effective in the extraction of features in the frequency and time domains, as shown in Table 5.1. Studies [37, 74, 66] have demonstrated that the essential characteristics of signals can be captured effectively by utilizing the variables min, max, avg, RMS, and sum. In finger gesture recognition applications [72, 79], ent, avg, var, cov, sd, min, max, zcr have been shown to be powerful factors in capturing the characteristics of signals. After selecting these specific features and extracting features from the raw data after being normalized, the output of the feature selection method has proven to be effective enough to facilitate the training of our machine learning algorithms.

Based on these feature choices for each accelerometer and gyroscope, the feature vector in the case of one ring includes 60 columns and, in the case of two rings, includes 120 columns. Each column represents a statistical feature extracted from the raw data. The new feature vector is a representation of each gesture and is used as input for our feature

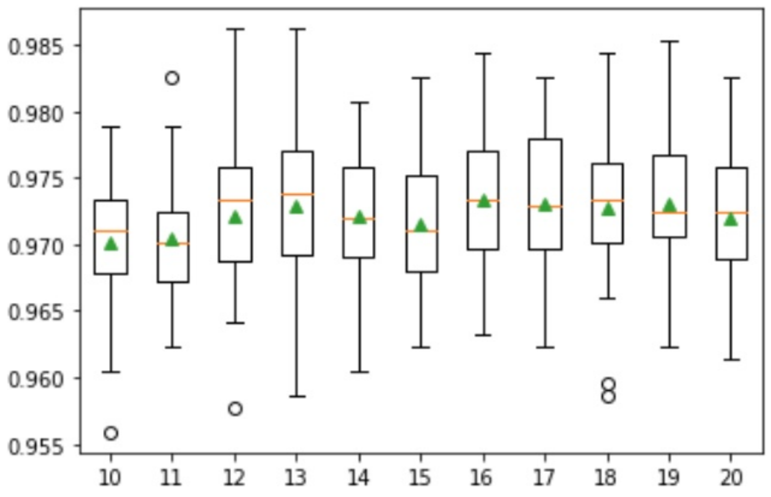
Table 5.1: Statistical features used in finger gesture recognition

Feature extraction in the frequency and time domains	Reference
min, max, avg, sd, signal magnitude area (SMA), and signal vector magnitude (SVM)	[74]
power spectral density (PSD), ent, and spectral energy	[79]
avg, max, min, 3quantiles, median, sd, mode, peak2rms, cov, and RMS	[49]
avg, var, sd, min, max, standard deviation auto-correlation, mean auto-covariance, skewness, mean crossing rate, standard deviation auto-covariance, mean auto-correlation, standard deviation auto-covariance, kurtosis	[72]
max, min, mean, RMS, sum	[37]
energy spectral, ent, mean, var, mean trend, and max	[66]

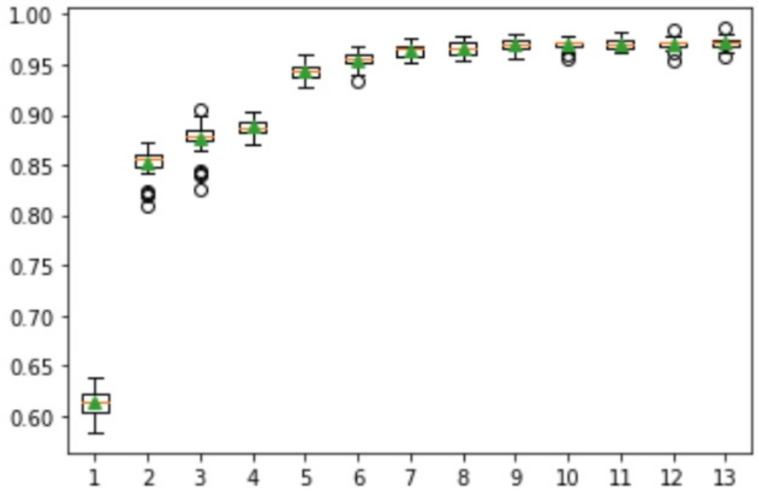
selection methods. All these calculations happen within the Raspberry Pi before being used as input for machine learning training.

Optimal Number of Features

In order to successfully utilize RFE-based feature selection, two factors must be considered: how many features to select and which algorithm to use. To configure RFE, we chose Random Forest as the machine learning algorithm core and Selected K Best as our configuration parameters to decide which feature to select. An illustration of the optimal number of features to be selected in cross-validation for higher accuracy and the minimal standard deviation is shown in Figure 5.1. According to our observation, the optimal number of features is between 10 and 20 for gestures performed with one finger and 10 for gestures made with two fingers.



(a) One-finger cross-validation function recommended feature quantity



(b) Two-finger cross-validation function recommended feature quantity

Figure 5.1: The recommended number of features.

5.4 Machine Learning Results

5.4.1 Machine Learning Evaluation Metrics

Various performance measures are derived from machine learning classification algorithms in order to assess their effectiveness in various contexts. The following performance indicators have been used:

- The confusion matrix summarizes the results of a classification problem. The correct and incorrect predictions are summarized with count values and broken down by each class.

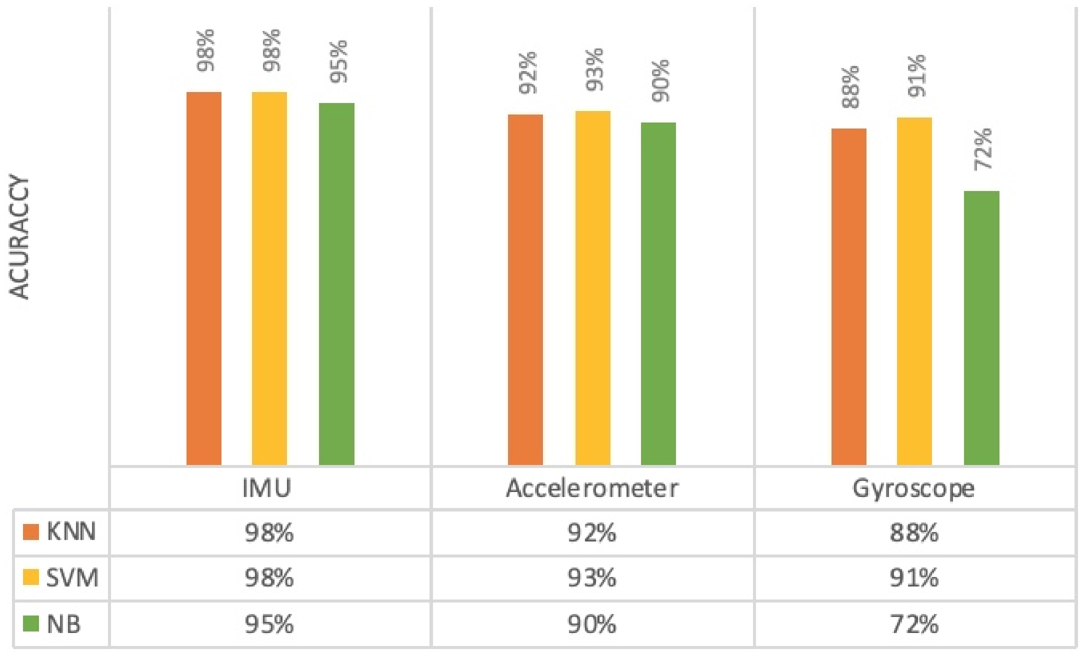
- The precision score is a way to evaluate the model's performance in regard to counting the number of true positives incorrectly out of all positive predictions made.
- The recall score measures the model's performance in measuring the number of true positives that are caused by incorrect measurement out of all the actual positive values.
- An accuracy score is used to measure the performance of the model by measuring the ratio between the sum of true positive and true negative predictions.
- The F1-score is the harmonic mean of the precision and recall score, and is used as a metric in situations in which either precision or recall scores could result in excessive false positives or false negatives.

As each machine learning algorithm is applied separately to the RFE and SFFS output dataset, six different states are obtained, and are further analyzed in the following section.

5.4.2 One-finger Gesture Recognition Results

Figure 5.2 depicts the difference in machine learning accuracy when the accelerometer and gyroscope have been used alone or together as an IMU unit. After applying three different machine learning algorithms to the three different datasets, we found that the IMU dataset's accuracy was improved by $6\% - 7\%$ over using the accelerometer dataset alone and $8\% - 13\%$ over using the gyroscope dataset alone. Therefore, combining both sensors in our design allows more freedom in detecting complex gestures and results in higher accuracy.

As a result of applying the IMU dataset before performing any feature selection, the result of machine learning evaluation metrics are shown in Table 5.3. Due to the high number of features that the machine learning model needs to be processed, the machine learning response time is within a few minutes, which is not acceptable for our finger gesture recognition system. Furthermore, the possibility that less valuable features or noise in the signal may result in overfitting and reducing the quality of the dataset leads us to use feature selection methods.



	IMU	Accelerometer	Gyroscope
🟧 KNN	98%	92%	88%
🟨 SVM	98%	93%	91%
🟩 NB	95%	90%	72%

Figure 5.2: A comparison of accelerometers and gyroscopes when used together, and when used separately

In order to improve accuracy and processing speed while reducing the risk of overfitting, two feature selection techniques were used. Two different sets of features are selected for each of our machine learning methods in order to train and test them. According to Figure 5.1 and based on the cross-validation function results, 10 to 20 would be an appropriate number of features for our design.

Four experiments were conducted to understand better the effect of the number of features on machine learning accuracy. Both RFE-10 and SFFS-10 stands for RFE and SFFS feature selection method when choosing ten feature, which is the initial number of features for each method. As a result of increasing the number of features to 20, namely RFE-20 and SFFS-20, the machine learning results are shown in Figure 5.3. It is evident that the RFE features selection method produces lower accuracy when using 10 rather than 20 features. However, there is a difference of approximately 1% between 10 and 20 features when

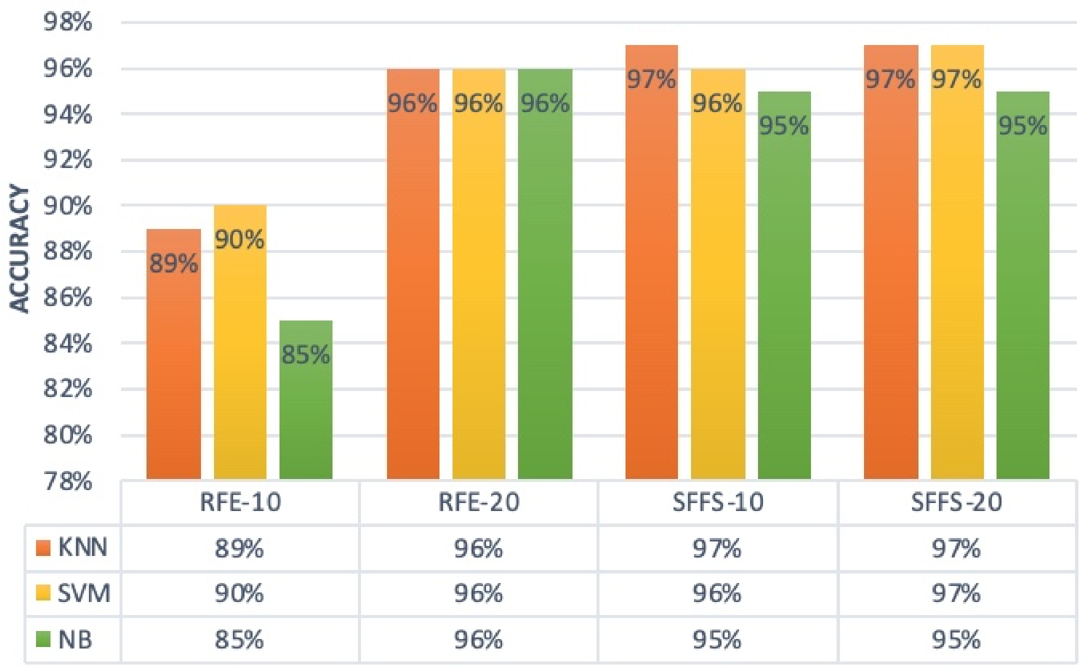
Table 5.2: The result of a feature selection process

Methods	Selected features
One-ring RFE-20	min-acc-x, min-acc-y, min-acc-z, min-gyro-z , max-acc-y, max-gyro-z, avg-acc-x, avg-acc-y, avg-acc-z, sd-gyro-y, sum-acc-x, sum-acc-y, sum-acc-z, sum-gyro-y, var-gyro-z, cov-gyro-xy, rms-acc-x, rms-acc-y, rms-acc-z, zcr-acc-y
One-ring RFE-10	min-acc-z, min-gyro-x, max-acc-y, max-gyro-y, avg-gyro-x, sd-gyro-y, var-gyro-x, sum-acc-z, rms-gyro-x, rms-gyro-y
Two-rings RFE-10	min-acc1-y, min-gyro2-z, max-gyro1-y, max-acc2-x, sd-acc1-x, sd-gyro2-y, sum-acc1-x, sum-gyro1-y, sum-acc2-x, ent-gyro1-x
One-ring SFFS-20	min-acc-x, min-acc-y, min-acc-z, min-gyro-z, max-acc-x, max-acc-y, max-acc-z, max-gyro-z, avg-gyro-z, avg-acc-x, avg-gyro-z, sd-gyro-y, sum-acc-y, sum-acc-z, var-gyro-z, cov-gyro-xy, ent-gyro-y, rms-gyro-x, rms-gyro-y, zcr-acc-x
One-ring SFFS-10	min-gyro-x, min-gyro-z, max-gyro-y, max-gyro-z, sd-gyro-y, sum-acc-x, sum-acc-y, sum-acc-z, cov-gyro-xy, rms-gyro-x
two-rings SFFS-10	min-acc1-x, min-acc1-y, min-acc2-z, max-gyro1-x, avg-gyro2-z, sum-acc1-x, sum-acc2-y, avg-acc2-y, rms-gyro2-x, var-acc1-x

using SFFS feature selection methods. We obtained a similar result with SFFS by using 10 features as opposed to RFE by using 20 features. Additionally, compared to applying machine learning to a dataset without feature selection, as shown in Figure 5.3, the number of features decreases dramatically from 60 to 10, and the final machine learning results vary by less than 2%. A slight difference in accuracy is acceptable to reduce the processing time to a few seconds and use one-sixth of the original feature vector numbers when using SFFS as our feature selection method.

Based on the application of feature selection methods, the selected features are shown in Table 5.2. These results were used to conduct the four experiments mentioned above. It is imperative to understand that RFE-20 and SFFS-10 for one-ring serve as inputs for evaluating our machine learning model. Additionally, RFE-10 and SFFS-10 for two rings are used in the following section for the evaluation of two two-ring gesture recognition.

Table 5.4 shows the evaluation metrics of the applied RFE feature selection with 20 features. Across three different machine learning algorithms, an average accuracy of 96% is achieved. As it can be seen from Table 5.5, a higher performance is achieved with the



	RFE-10	RFE-20	SFFS-10	SFFS-20
🟧 KNN	89%	96%	97%	97%
🟨 SVM	90%	96%	96%	97%
🟩 NB	85%	96%	95%	95%

Figure 5.3: The comparison between RFE and SFFS when using 10 or 20 features

SFFS feature selection method utilizing only 10 features. SFFS reached an accuracy of 97.1% with KNN classifiers, which outperforms the RFE in terms of accuracy with half the number of features. Accordingly, we recommend the SFFS method with its selected 10 features as the method to select features.

Table 5.3: One-finger machine learning results without feature selection methods

Evaluation Metrics	**KNN**	**SVM**	**NB**
Selected Features	60	60	60
Accuracy	0.979	0.982	0.95
F1-Score	0.974	0.978	0.935
Recall	0.974	0.979	0.934
Precision	0.975	0.978	0.949

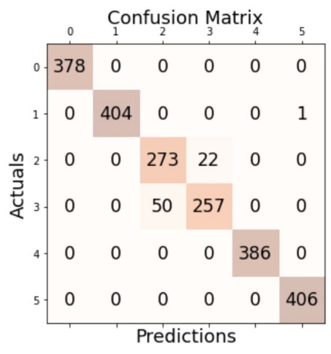
Table 5.4: One-finger machine learning results with RFE method

Evaluation Metrics	**KNN**	**SVM**	**NB**
Selected Features	20	20	20
Accuracy	0.968	0.967	0.964
F1-Score	0.960	0.962	0.957
Recall	0.960	0.961	0.957
Precision	0.961	0.965	0.958

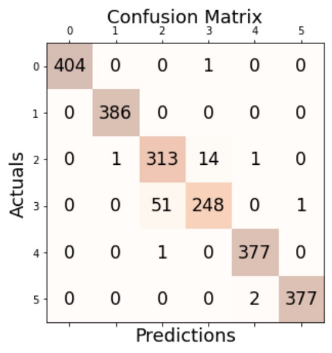
Table 5.5: One-finger machine learning results with SFFS method

Evaluation Metrics	**KNN**	**SVM**	**NB**
Selected Features	10	10	10
Accuracy	0.971	0.963	0.956
F1-Score	0.964	0.956	0.946
Recall	0.964	0.956	0.944
Precision	0.965	0.959	0.954

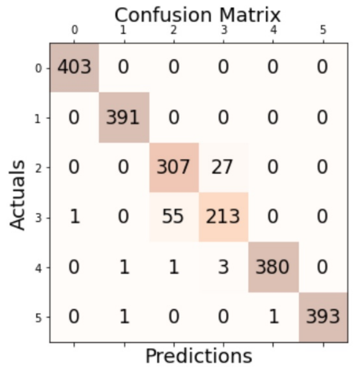
Additionally, Figure 5.4 illustrates the confusion matrices for KNN, SVM, and Naive Bayes classifiers, with 20 features for RFE and 10 features for SFFS.



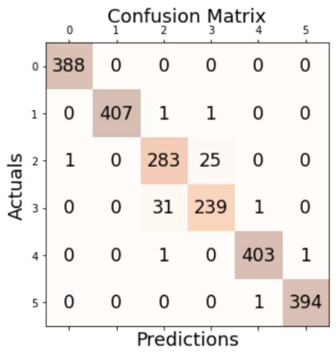
(a) KNN classifier with RFE method



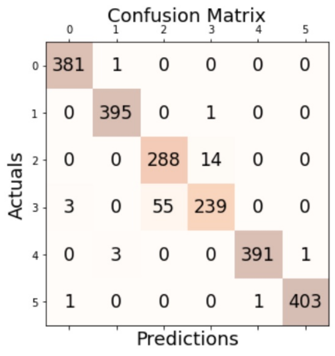
(b) SVM classifier with RFE method



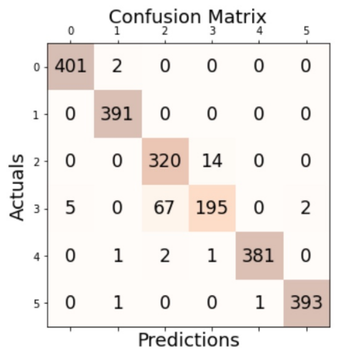
(c) NB classifier with RFE method



(d) KNN classifier with SFFS method



(e) SVM classifier with SFFS method



(f) NB classifier with SFFS method

Figure 5.4: One-ring gestures confusion matrices.

5.4.3 Two-finger Gesture Recognition Results

Before applying feature selection, the primary two-finger dataset includes 120 feature columns fed into three different machine learning algorithms. According to Figure 5.1, It could be argued that 10 features could be considered suitable for training our machine learning model to recognize two-finger gestures. Table 5.6 shows that different machine learning models are applied to the original dataset, and evaluation metrics are extracted and reported. Using appropriate features extracted from the data and gathering a rich dataset by performing each finger measurement 2000 times, our model achieved a primary test result of 99% for KNN after training.

Although we have achieved good results with this model, it is not acceptable to make timely predictions for newly performed two-finger gestures. Having to extract 120 statistical features from data, feed them into a model, and display the results in a real-time scenario is too time-consuming. To address this issue, we have chosen the most significant features and built our model based on them through data pre-processing and feature selection methods. Apart from reducing computation costs significantly, it also accelerates processing time while maintaining an acceptable level of accuracy. According to Table 5.7, the RFE feature selection method achieves an accuracy of 97.9%, when using the KNN machine learning algorithm for ten features which is similar to the SFFS result when using KNN with 97.5% accuracy.

Table 5.8, the detailed evaluation matrices of SFFS are shown. We selected the KNN classifier in this case as our preferred machine learning classifier because it outperformed the SVM and NB classifiers by a small margin. Figure 5.5 shows the confusion matrices corresponding to the KNN, SVM, and NB classifiers with two different feature selection methods. Note that the SFFS with 10 features is sufficient for our design, regardless of whether it is used for one-finger or two-finger gesture recognition.

Table 5.6: Two-finger machine learning results without feature selection methods

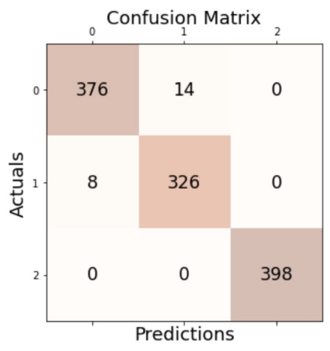
Evaluation Metrics	**KNN**	**SVM**	**NB**
Selected Features	120	120	120
Accuracy	0.999	0.996	0.986
F1-Score	0.999	0.996	0.985
Recall	0.999	0.996	0.912
Precision	0.999	0.996	0.986

Table 5.7: Two-finger machine learning results with RFE method

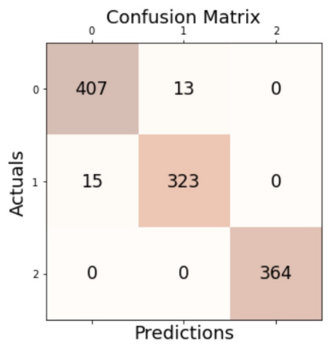
Evaluation Metrics	**KNN**	**SVM**	**NB**
Selected Features	10	10	10
Accuracy	0.979	0.975	0.960
F1-Score	0.980	0.975	0.960
Recall	0.980	0.975	0.960
Precision	0.980	0.975	0.960

Table 5.8: Two-finger machine learning results with SFFS method

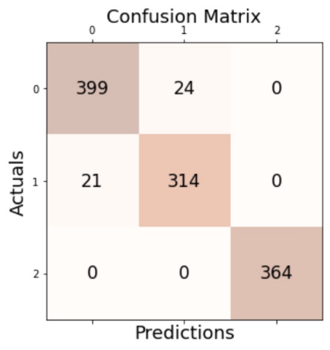
Evaluation Metrics	**KNN**	**SVM**	**NB**
Selected Features	10	10	10
Accuracy	0.970	0.964	0.959
F1-Score	0.970	0.964	0.959
Recall	0.969	0.964	0.960
Precision	0.970	0.964	0.958



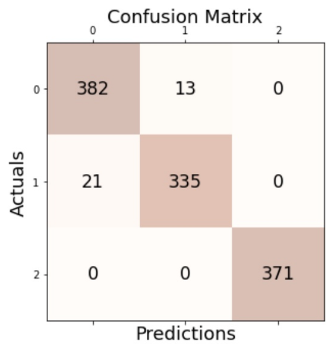
(a) KNN classifier with RFE method.



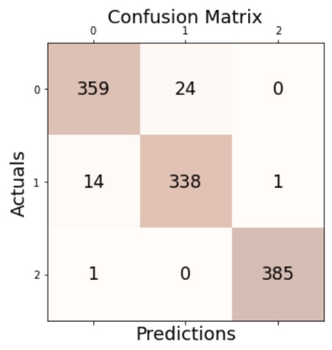
(b) SVM classifier with RFE method.



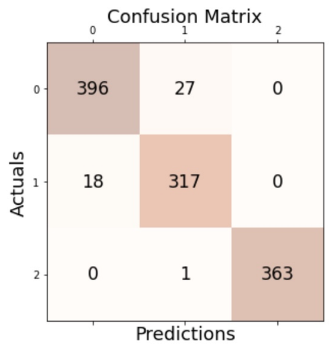
(c) NB classifier with RFE method.



(d) KNN classifier with SFFS method.



(e) SVM classifier with SFFS method.



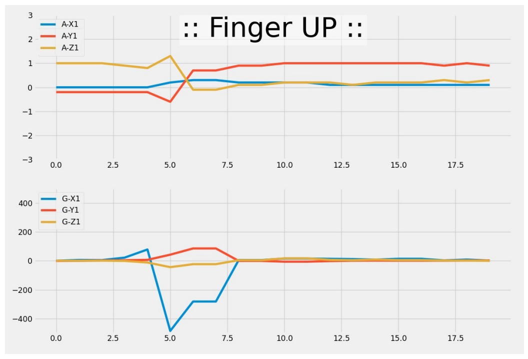
(f) NB classifier with SFFS method.

Figure 5.5: Two-ring gestures confusion matrices.

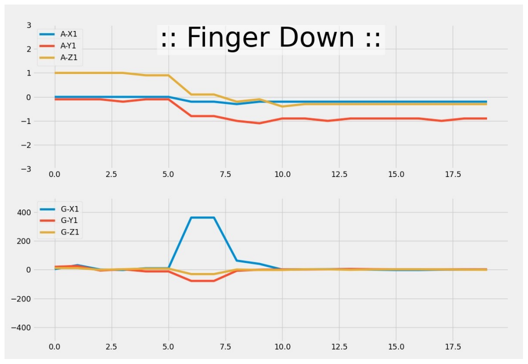
5.4.4 Classification Performance

We verified that the SFFS feature selection method outperforms the RFE method when detecting one-finger and two-finger gestures, using fewer features while keeping a similar accuracy. The applied modular approach in our study causes one or more measurement devices to be worn based on the number of finger motions to be examined. Using one ring is sufficient to perform one-finger gestures (Finger up, Finger down, Finger right, Finger left, Finger 90 deg clockwise, Finger 90 deg counterclockwise), while using two rings is restricted to two-finger gestures (Zoom in, Zoom out, Two fingers 90 deg clockwise).

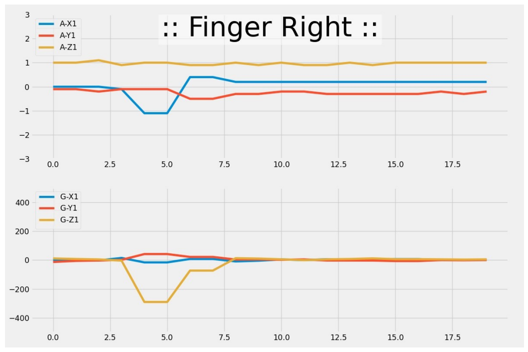
Figure 5.6 and 5.7 illustrate the machine learning classification performance while performing our pre-defined finger gestures. Each illustration shows the period before a gesture is performed, during which the gesture is in progress, and the final position of each finger after the gesture has been performed. In comparison to what we expected our actual signal to be like Figures 3.11 - 3.19, the detected gestures and the fluctuation of accelerometer and gyroscope data confirm our expectation and prove our choice of feature selection method and machine learning algorithm.



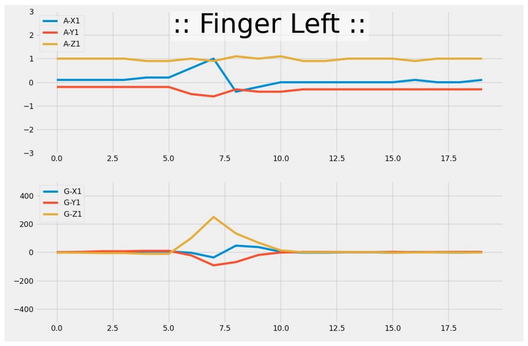
(a) Machine learning predicted movement for the finger-up gesture.



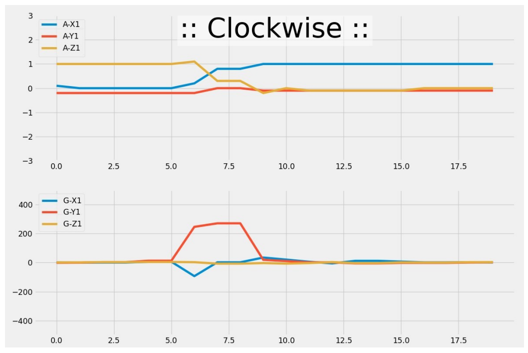
(b) Machine learning predicted movement for the finger-down gesture.



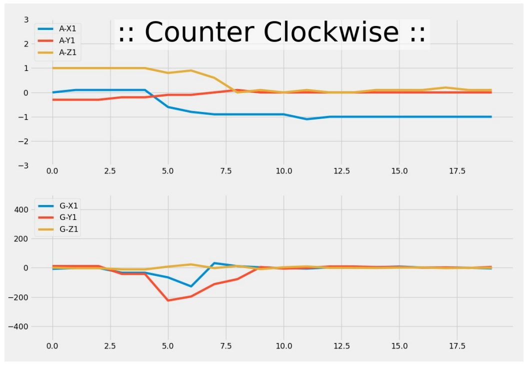
(c) Machine learning predicted movement for the finger-right gesture.



(d) Machine learning predicted movement for the finger-left gesture.

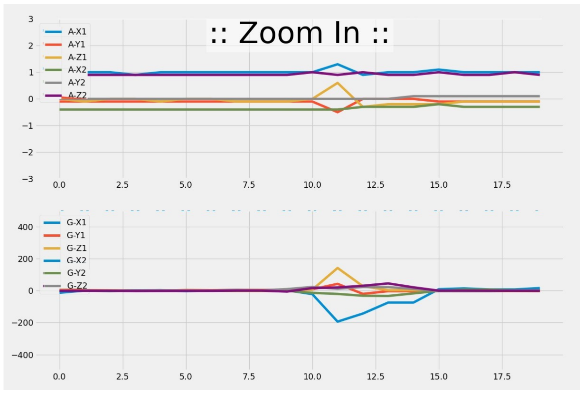


(e) Machine learning predicted movement for the finger-clockwise gesture.

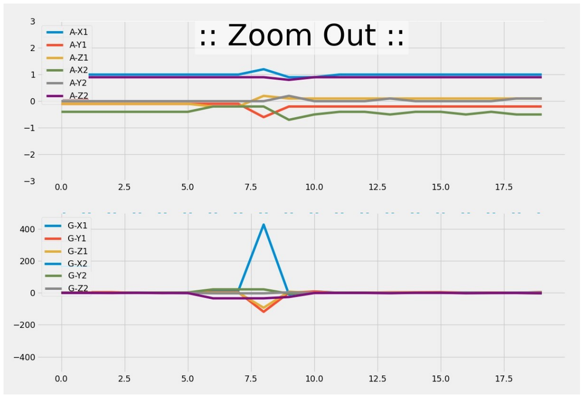


(f) Machine learning predicted movement for the finger-counterclockwise gesture.

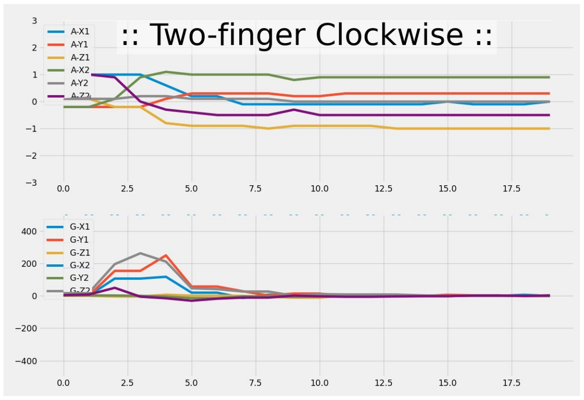
Figure 5.6: One-finger gesture recognition results.



(a) Machine learning predicted movement for the zoom-in gesture.



(b) Machine learning predicted movement for the zoom-out gesture.



(c) Machine learning predicted movement for the two-finger 90-degree clockwise rotation gesture.

Figure 5.7: Two-finger gesture recognition results.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

A wearable IMU-based smart ring prototype with an integrated BLE module is presented in this thesis. For the purpose of detecting finger gestures, a combination of accelerometers and gyroscope sensors have been integrated into the sensing system. This combination has proved to be more practical from the point of view of accuracy and detecting complex finger gestures than relying on only one sensor. Utilizing accelerometers and gyroscopes, the system can identify nine predefined finger gestures. The gestures can be used to control a robotic arm or interact with smart devices, such as smartphones and televisions. The raw data has been preprocessed on the Raspberry Pi, and the statistical features have been extracted and stored. Using RFE and SFFS methods, the dataset is used to determine which features are valuable and which are less important. Following this step, the results are labelled according to the finger gestures associated with them. Using machine learning data analysis, the recommended system was validated. We further investigate the detection of finger gestures by using three different machine learning algorithms, namely KNN, NB, and SVM. By utilizing the selected features from the SFFS and incorporating them into KNN as our machine learning classifier, we were able to significantly reduce the number of features required to achieve an acceptable level of accuracy and reduce the processing and response time of the system as well. Results indicate that the recommended system can accurately recognize one-finger gestures with 97.1% accuracy and two-finger gestures with 97% accuracy.

6.2 Limitations

One of the primary limitations of our proposed method is the hardware design. Even though BLE is more energy efficient than Bluetooth 4.0, it still consumes a substantial amount of energy and has low bandwidth compared to WiFi. It should be noted that when using BLE, communication between devices is also restricted to relatively short distances only. There is a limit to how much data the BLE-based devices can handle. Generally, they are only able to transmit up to 100 KBps for short periods of time. The high output current demanded by Bluetooth requires a DC/DC converter with a 3.3V output voltage and a maximum output current of more than 300mAh. This is why we are limited in choosing DC/DC converters. The battery used in our design is a 3.7V 350mAh Lithium Polymer battery. In proportion to the battery's current capacity, the battery's size increases as the battery's current capacity increases. Therefore, our battery options are limited due to the need to mount the board conveniently on the finger with a small battery. Our fabricated board is 20 mm $\times$ 20 mm, and the closest battery we could find was 20 mm $\times$ 30.5 mm. This led us to select a battery 10mm larger than the board size to provide the current needed to initiate the BLE data transmission.

6.3 Future Work

The results and development of this project can have many practical applications: virtual reality applications, sign language recognition, remote robot controllers, air sketches, and smart controllable devices by finger and hand gestures. As part of future work, the prototype will be reduced in size, and the current consumption will be decreased (resulting in a smaller battery size). Duo to the modular approach used in our design made it possible to add another sensor to gather more data for detecting more complex finger or hand gestures, such as an electromyography (EMG) sensor. The sensor could be employed on another ring or embedded in the next version of our design fabricated ring to gather the user's finger electrical signals generated by muscles.

We intend to enable our embedded WiFi connection as part of our future work. This will give access to the advantages of WiFi wireless data transmission and BLE. When compared to WiFi, BLE consumes less power. However, BLE does not support data transmission in real-time to a server. In contrast, WiFi technology is 20-30 times faster than Bluetooth. As a result, it is well suited for sharing large data files such as videos and photos. Compared to WiFi, Bluetooth provides more accurate proximity data; however, its maximum wireless range is 30 meters. Alternatively, WiFi is a better choice for open spaces.

Machine learning recognition functionality is another aspect of our plan for future improvements. The defined finger gestures can be expanded to include more complex multi-finger gestures as well as hand gestures for more complex movement recognition results such as sign language or air sketches. In addition, further improvements can be made to our choice of feature selection methods, machine learning classifiers, and modifications to the choice of statistical features in the time and frequency domains.

Ultimately, our goal is to move from the Raspberry Pi as the main processor to create software that can be installed in smart devices such as phones and watches to use their processor to handle the detection process and apply machine learning algorithms.

Bibliography

[1] S. Tan and J. Yang, “Wifinger: Leveraging commodity wifi for fine-grained finger gesture recognition,” in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA: Association for Computing Machinery, 2016, p. 201–210.

[2] K. Haratiannejadi, “Smart glove and hand gesture-based control interface for multi-rotor aerial vehicles,” M.S. thesis, Department of Electrical and Computer Engineering, Concordia University, Montréal, Québec, Canada, 2020.

[3] A. G. Perera, Y. Wei Law, and J. Chahl, “Uav-gesture: A dataset for uav control and gesture recognition,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[4] N. SaiChinmayi, C. Hasitha, B. Sravya, and V. Mittal, “Gesture signals processing for a silent spybot,” in *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2015, pp. 756–761.

[5] Y. Ma, Y. Liu, R. Jin, X. Yuan, R. Sekha, S. Wilson, and R. Vaidyanathan, “Hand gesture recognition with convolutional neural networks for the multimodal uav control,” in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2017, pp. 198–203.

[6] S.-Y. Shin, Y.-W. Kang, and Y.-G. Kim, “Hand gesture-based wearable human-drone interface for intuitive movement control,” in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–6.

[7] B.-F. Gheran, O.-C. Ungurean, and R.-D. Vatavu, “Toward smart rings as assistive devices for people with motor impairments: A position paper,” in *RoCHI*, 2018.

[8] K. Kudrinko, E. Flavin, X. Zhu, and Q. Li, “Wearable sensor-based sign language recognition: A comprehensive review,” *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 82–97, 2021.

[9] B. G. Lee and S. M. Lee, “Smart wearable hand device for sign language interpretation system with sensors fusion,” *IEEE Sensors Journal*, vol. 18, no. 3, pp. 1224–1232, 2018.

[10] T.-M. Tai, Y.-J. Jhang, Z.-W. Liao, K.-C. Teng, and W.-J. Hwang, “Sensor-based continuous hand gesture recognition by long short-term memory,” *IEEE Sensors Letters*, vol. 2, no. 3, pp. 1–4, 2018.

[11] R. Xu, S. Zhou, and W. J. Li, “Mems accelerometer based nonspecific-user hand gesture recognition,” *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1166–1173, 2011.

[12] I. H. Sarker, “Machine learning: Algorithms, real-world applications and research directions,” *SN Computer Science*, vol. 2, pp. 1–21, 2021.

[13] K. Taunk, S. De, S. Verma, and A. Swetapadma, “A brief review of nearest neighbor algorithm for learning and classification,” 05 2019, pp. 1255–1260.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[15] Y.-T. Wang and H.-P. Ma, “Real-time continuous gesture recognition with wireless wearable imu sensors,” in *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2018, pp. 1–6.

[16] M. Jena and S. Dehuri, “Decisiontree for classification and regression: A state-of-the art review,” *Informatica*, vol. 44, 12 2020.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,

M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.

[19] D. N. Kakade and J. S. Chitode, “Dynamic hand gesture recognition : A literature review,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 1, no. 9, 2012.

[20] C. Wang, Z. Liu, and S.-C. Chan, “Superpixel-based hand gesture recognition with kinect depth camera,” *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 29–39, 2014.

[21] G. Zhu, L. Zhang, P. Shen, and J. Song, “Multimodal gesture recognition using 3-d convolution and convolutional lstm,” *IEEE Access*, vol. 5, pp. 4517–4524, 2017.

[22] R. Xie, X. Sun, X. Xia, and J. Cao, “Similarity matching-based extensible hand gesture recognition,” *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3475–3483, 2015.

[23] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy, “Gesture pendant: a self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring,” *International Symposium on Wearable Computers, Digest of Papers*, 2000.

[24] P. Mistry and P. Maes, “Sixthsense: a wearable gestural interface,” in *ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation*, 2009, pp. 85–85.

[25] A. Butler, S. Izadi, and S. Hodges, “Sidesight: multi-” touch” interaction around small devices,” in *Proceedings of the 21st annual ACM Symposium on User Interface Software and Technology*, 2008, pp. 201–204.

[26] F.-T. Liu, Y.-T. Wang, and H.-P. Ma, “Gesture recognition with wearable 9-axis sensors,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[27] C. K. Mummadi, F. P. P. Leo, K. D. Verma, S. Kasireddy, P. M. Scholl, J. Kempfle, and K. V. Laerhoven, "Real-time and embedded detection of hand gestures with an imu-based glove," *Informatics*, vol. 5, no. 2, 2018. [Online]. Available: https://www.mdpi.com/2227-9709/5/2/28

[28] S. Nirjon, J. Gummeson, D. Gelb, and K.-H. Kim, "Typingring: A wearable ring platform for text input," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 2015, pp. 227–239.

[29] H.-S. Yeo, J. Lee, H.-i. Kim, A. Gupta, A. Bianchi, D. Vogel, H. Koike, W. Woo, and A. Quigley, "Wrist: Watch-ring interaction and sensing technique for wrist gestures and macro-micro pointing," in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '19. New York, NY, USA: Association for Computing Machinery, 2019.

[30] L. Jing, Y. Zhou, Z. Cheng, and T. Huang, "Magic ring: A finger-worn device for multiple appliances control using static finger gestures," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 5775–5790, 2012.

[31] M. Roshandel, A. Munjal, P. Moghadam, S. Tajik, and H. Ketabdar, "Multi-sensor based gesture recognition with a smart finger ring," in *Human-Computer Interaction, HCII 2014*. Springer International Publishing, 2014, p. 316–324.

[32] H. Wen, J. Ramos Rojas, and A. K. Dey, "Serendipity: Finger gesture recognition using an off-the-shelf smartwatch," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 3847–3851.

[33] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 9–14.

[34] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, “An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition,” *IEEE Sensors Journal*, vol. 15, no. 1, pp. 154–163, 2015.

[35] J.-S. Wang and F.-C. Chuang, “An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 2998–3007, 2011.

[36] T. Deselaers, D. Keysers, J. Hosang, and H. A. Rowley, “Gyropen: Gyroscopes for pen-input with mobile phones,” *IEEE Transactions on Human-Machine Systems*, vol. 45, 2015.

[37] M. Kurz, R. Gstoettner, and E. Sonnleitner, “Smart rings vs. smartwatches: Utilizing motion sensors for gesture recognition,” *Applied Sciences*, vol. 11, no. 5, 2021.

[38] W. M, K. D, and A. S, “Perisense: Ring-based multi-finger gesture interaction utilizing capacitive proximity sensing,” *IEEE Sensors Journal*, vol. 20, no. 14, 2020.

[39] seven02 design. Accessed 2022-Dec-2. [Online]. Available: https://www.seven02design.com/livescribe-pulse-smart-pen

[40] P. Bellitti, A. D. Angelis, M. Dionigi, E. Sardini, M. Serpelloni, A. Moschitta, and P. Carbone, “A wearable and wirelessly powered system for multiple finger tracking,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2542–2551, 2020.

[41] Y. Gu, C. Yu, Z. Li, W. Li, S. Xu, X. Wei, and Y. Shi, “Accurate and low-latency sensing of touch contact on any surface with finger-worn imu sensor,” *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019.

[42] S. A. Khomami and S. Shamekhi, “Persian sign language recognition using imu and surface emg sensors,” *IEEE Transactions on Instrumentation and Measurement*, vol. 168, p. 108471, 2021.

[43] H. Han and S. W. Yoon, “Gyroscope-based continuous human hand gesture recognition for multi-modal wearable input device for human machine interaction,” *IEEE Sensors Journal*, vol. 19, no. 11, 2019.

[44] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, “A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors,” *IEEE Sensors Journal*, vol. 16, 2016.

[45] M. Georgi, C. Amma, and T. Schultz, “Recognizing handand finger gestures with imu based motion and emg based muscle activity sensing,” in *International Conference on Bio-inspired Systems and SignalProcessing (BIOSIGNALS-2015)*, 2015, pp. 99–108.

[46] Y. Peng, X. Wang, L. Zhong, K. Pang, Y. Chen, M. Wang, and W. Liu, “A flexible dual-modal sensing system for synchronous pressure and inertial monitoring of finger movement,” *IEEE Sensors Journal*, vol. 21, no. 9, pp. 10 483–10 490, 2021.

[47] W. Wong, F. H. Juwono, and B. T. T. Khoo, “Multi-features capacitive hand gesture recognition sensor: A machine learning approach,” *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8441–8450, 2021.

[48] C. Zhang, A. Waghmare, P. Kundra, Y. Pu, S. Gilliland, T. Ploetz, T. E. Starner, O. T. Inan, and G. D. Abowd, “Fingersound: Recognizing unistroke thumb gestures using a ring,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, sep 2017.

[49] Y. Li, N. Yang, L. Li, L. Liu, and Y. Yang, “Finger gesture recognition using a smartwatch with integrated motion sensors,” *Web Intelligence*, vol. 16, pp. 123–129, 06 2018.

[50] J. V. Tu, “Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes,” *Journal of clinical epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996.

[51] P. Visconti, F. Gaetani, G. A. Zappatore, and P. Primiceri, “Technical features and functionalities of myo armband: An overview on related literature and advanced applications of myoelectric armbands mainly focused on arm prostheses,” *International Journal on Smart Sensing and Intelligent Systems*, vol. 11, no. 1, pp. 1–25, 2018.

[52] M. Caeiro-Rodríguez, I. Otero-González, F. A. Mikic-Fonte, and M. Llamas-Nistal, “A systematic review of commercial smart gloves: Current status and applications,” *Sensors*, vol. 21, no. 8, p. 2667, 2021.

[53] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, “Analysis of the accuracy and robustness of the leap motion controller,” *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[54] L. Jing, Y. Zhou, Z. Cheng, and J. Wang, “A recognition method for one-stroke finger gestures using a mems 3d accelerometer,” *IEICE Transactions on Information and Systems*, vol. E94.D, no. 5, pp. 1062–1072, 2011.

[55] L.-H. Jhang, C. Santiago, and C.-S. Chiu, “Multi-sensor based glove control of an industrial mobile robot arm,” in *2017 International Automatic Control Conference (CACS)*. IEEE, 2017, pp. 1–6.

[56] M. Borghetti, E. Sardini, and M. Serpelloni, “Sensorized glove for measuring hand finger flexion for rehabilitation purposes,” *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 12, pp. 3308–3314, 2013.

[57] N. Gupta, *Inside Bluetooth Low Energy*, 2nd ed. Artech House, 2016.

[58] (2021) Raspberry Pi Zero W. Raspberry Pi Foundation. Accessed 2022-Aug-1. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-zero-w/

[59] *MPU-9250 Product Specification*, InvenSense Inc., 2016, reversion 1.1.

[60] *ESP32-C3-WROOM-02 and ESP32-C3-WROOM-02U Datasheet*, Espressif Systems, 2021, version 1.0.

[61] P. Pudil, J. Novovičová, and J. Kittler, “Floating search methods in feature selection,” *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[62] B. Venkatesh and J. Anuradha, “A review of feature selection and its methods,” *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3–26, 2019.

[63] U. Stańczyk and L. C. Jain, *Feature Selection for Data and Pattern Recognition*. Springer, 2015.

[64] A. Jović, K. Brkić, and N. Bogunović, “A review of feature selection methods with applications,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1200–1205.

[65] H. Jeon and S. Oh, “Hybrid-recursive feature elimination for efficient feature selection,” *Applied Sciences*, vol. 10, no. 9, 2020.

[66] P. Gupta and T. Dallas, “Feature selection and activity recognition system using a single triaxial accelerometer,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1780–1786, 2014.

[67] M. A. O’Reilly, W. Johnston, C. Buckley, D. Whelan, and B. Caulfield, “The influence of feature selection methods on exercise classification with inertial measurement units,” in *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 2017, pp. 193–196.

[68] C. Altın and O. Er, “Comparison of different time and frequency domain feature extraction methods on elbow gesture’s emg,” *European Journal of Interdisciplinary Studies*, vol. 2, no. 3, p. 35–44, 2016.

[69] R. Xie and J. Cao, “Accelerometer-based hand gesture recognition by neural network and similarity matching,” *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4537–4545, 2016.

[70] F. Duan, X. Ren, and Y. Yang, "A gesture recognition system based on time domain features and linear discriminant analysis," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 200–208, 2021.

[71] A. A. Torres-García, O. Mendoza-Montoya, M. Molinas, J. M. Antelis, L. A. Moctezuma, and T. Hernández-Del-Toro, "Chapter 4 - pre-processing and feature extraction," in *Biosignal Processing and Classification Using Computational Learning and Intelligence*, A. A. Torres-García, C. A. Reyes-García, L. Villaseñor-Pineda, and O. Mendoza-Montoya, Eds. Academic Press, 2022, pp. 59–91.

[72] A. A. Badawi, A. Al-Kabbany, and H. Shaban, "Daily activity recognition using wearable sensors via machine learning and feature selection," in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 2018, pp. 75–79.

[73] M.-K. Liu, Y.-T. Lin, Z.-W. Qiu, C.-K. Kuo, and C.-K. Wu, "Hand gesture recognition by a mmg-based wearable device," *IEEE Sensors Journal*, vol. 20, no. 24, pp. 14 703–14 712, 2020.

[74] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, p. 645–662, 2010.

[75] A. Smola and S. Vishwanathan, *Introduction to Machine Learning*. Cambridge University Press, 2008.

[76] E. Alpaydın, *Introduction to Machine Learning*, 2nd ed. The MIT Press, 2010.

[77] M. F. A. Saputra, T. Widiyaningtyas, and A. P. Wibawa, "Illiteracy classification using k means-naïve bayes algorithm," *International Journal on Informatics Visualization (JOIV)*, vol. 2, no. 3, pp. 153–158, 2018.

[78] T. Sztyler and H. Stuckenschmidt, “On-body localization of wearable devices: An investigation of position-aware activity recognition,” in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.

[79] A. A. Sukor, A. Zakaria, and N. A. Rahim, “Activity recognition using accelerometer sensor and machine learning classifiers,” in *2018 IEEE 14th International Colloquium on Signal Processing & its Applications (CSPA)*. IEEE, 2018, pp. 233–238.