Information Engineering and Technology Faculty
German University in Cairo

Intelligent Padel Tennis Sport Analytics based on Accelerometer/Gyro Sensors

Bachelor Thesis

Author:	Omar Hossam Yahia
Supervisors:	Dr. Tallal Osama Elshabrawy
Submission Date:	19 May, 2024

Information Engineering and Technology Faculty
German University in Cairo

Intelligent Padel Tennis Sport Analytics based on Accelerometer/Gyro Sensors

Bachelor Thesis

Author:	Omar Hossam Yahia
Supervisors:	Dr. Tallal Osama Elshabrawy
Submission Date:	19 May, 2024

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

Omar Hossam Yahia
19 May, 2024

Acknowledgments

Above all, I am grateful to God for all of his blessings in my life, particularly for the opportunity to pursue my education with success. I want to sincerely thank my supervisor, Prof. Dr. Tallal El-Shabrawy, for all of his help and his willingness to spend time with me and offer advice. I also want to thank all of my friends who supported me during the process; your help, together with all of your remarks and recommendations, made the final product better. Lastly, I want to express my sincere gratitude to my family for their constant support in helping me get to where I am now.

Abstract

This thesis explores the application of wearable sensor technology to enhance performance analysis in padel tennis. Utilizing the Seeed Studio Xiao NRF52840 microcontroller board, the study aimed to develop a wearable sensor system capable of tracking player shots during matches. Advanced data analysis methodologies were employed to generate insights for improving player performance. The research evaluated the accuracy of the developed stroke prediction model for three key padel strokes: forehand, backhand, and smash. Results indicate high accuracy rates of 96% for forehand, 94% for backhand, and 96% for smash strokes. Additionally, Principal Component Analysis (PCA) was utilized to visualize both training and testing motion data, providing insights into the effectiveness of the predictive model. The findings suggest significant potential for wearable sensor technology in enhancing performance analysis in padel tennis. Further research directions include incorporating ball tracking and shot outcome detection, increasing database size and diversity, and exploring opportunities for real-time feedback during matches and practice sessions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, technology has become increasingly integrated into sports, greatly impacting how athletes train and analyze performance. Wearable devices with sensors have been particularly influential, allowing athletes to monitor and improve their performance like never before. This thesis explores the application of wearable sensors, specifically the Seeed Studio Xiao NRF52840 microcontroller board, in enhancing the analysis of padel tennis performance. Padel tennis, a rapidly growing sport with unique movement patterns and shot techniques, presents an interesting arena for the implementation of advanced data analysis methodologies to gain deeper insights into player performance.

1.1 Internet of Things (IoT)

The Internet of Things (IoT) refers to the network of interconnected devices that collect, exchange, and analyze data over the Internet. These devices range from everyday objects like refrigerators and thermostats to specialized gadgets like wearable devices designed for specific applications such as sports and healthcare. IoT has revolutionized various industries by enabling seamless connectivity and real-time data exchange. In healthcare, IoT devices monitor patients' vital signs and provide remote diagnostics. In agriculture, smart sensors help optimize irrigation and crop yield. In smart cities, IoT technologies improve infrastructure management and enhance public services. Wearable devices play a crucial role within the IoT framework, utilizing its technologies to gather and transmit data on various metrics like movement, heart rate, and location. This data is then analyzed and interpreted to provide valuable information into user behavior, health metrics, and performance in specific activities like sports.

1.1.1 IoT in sports

IoT in sports [1] involves the integration of interconnected devices, sensors, and technologies to collect real-time data from athletes, equipment, and various aspects of sporting events. IoT applications in sports are Training and Simulation, Health and Fitness Tracking, Referee Assistance, and more.

Applications of IoT in sports

Training and Simulation:
IoT devices can be used to create realistic training environments for athletes. Virtual reality (VR) and augmented reality (AR) technologies combined with IoT sensors can simulate game scenarios, allowing athletes to practice and strategize in a realistic setting without the need for physical opponents.

Health and Fitness Tracking:
IoT devices like smartwatches and fitness bands can monitor athletes' health and fitness levels both on and off the field, providing valuable insights into their overall well-being and helping them maintain peak performance.

Referee Assistance:
IoT-based tracking systems can assist referees in making accurate decisions during games. For instance, goal-line technology in soccer uses sensors to determine whether the ball has crossed the goal line. These systems can help reduce human error and provide referees with additional information to make more informed decisions.

1.2 Wearable Sensors

Wearable technology has become increasingly popular in sports. These devices [2], which can be worn on the body or integrated into clothing, offer athletes valuable information about their performance. The development of wearable devices has been impressive. Starting from basic trackers that count steps or calories to advanced smartwatches and specialized wearables that can monitor heart rate, track GPS routes, and even analyze movement patterns. Wearable devices have transformed the way athletes train and compete. By providing real-time data on physical activity, and movement, these devices enable athletes to personalize their training routines, identify areas for improvement, and optimize their performance. In the context of padel tennis, wearable devices can be particularly beneficial. They can capture detailed information about player movements, strokes, and strategies, offering players valuable information to refine their techniques and enhance their performance.

The integration between IoT and wearable sensors in sports helps athletes and coaches communicate and share data easily. Wearable sensors, like those in clothing or gear, keep track of how athletes move and their physical condition. This information, sent through IoT networks, gives coaches and athletes valuable insights into performance and training progress. With this integration, athletes can better understand their strengths and weaknesses, improving their skills in sports like padel tennis.

1.3 Objective

The main objective is to create a wearable sensor system that tracks player shots during padel tennis matches. By analyzing the data collected from these sensors, the aim is to provide insights that can improve player performance. This integration of wearable sensors and data analytics can help athletes and coaches make informed decisions and customize training strategies for better results.

1.4 Thesis Outline

In order to fulfill the aim of the project, this thesis is structured as follows:

Chapter One
Introduction: Provides an introduction to the thesis topic, discussing the application of IoT and wearable sensors in sports performance monitoring. It outlines the objectives of the thesis in utilizing these technologies to enhance sports analytics.

Chapter Two
Background: The background section provides essential insights into sensor technologies like accelerometers and gyroscopes, crucial for motion data capture, Bluetooth Low Energy (BLE), the Seeed Studio XIAO nRF52840 microcontroller board, fundamental concepts of machine learning, and statistical techniques including Root Mean Square Error (RMSE) and Principal Component Analysis (PCA).

Chapter Three
Related Work: This chapter reviews existing research on using accelerometer and gyroscope data to classify different types of strokes in sports. It highlights relevant studies and their methodologies in analyzing player movements.

Chapter Four
Methodology: The methodology section outlines the hardware setup, including the rechargeable battery and the motion-sensing glove. It covers software implementation with Arduino and Bluetooth communication. Key processes discussed include motion detection, feature extraction, clustering analysis, data storage, stroke prediction, and Android app integration.

Chapter Five
Results and Analysis: This chapter examines the outcomes of the padel stroke prediction model. It assesses the model's accuracy in stroke prediction. Moreover, Principal Component Analysis (PCA) is utilized to visualize motion data, providing insights into the model's performance in capturing motion patterns.

Chapter Six
Conclusion and Future Work: The conclusion summarizes the key findings of the research, discusses its significance, and suggests paths for future work.

Chapter 2

Background

This chapter provides a foundational understanding of key components in sports technology and analytics. It starts by introducing various sensors like accelerometers and gyroscopes, which capture player movements. It then explores Bluetooth Low Energy (BLE) technology, crucial for low-power communication between devices. The Seeed Studio Xiao NRF52840 microcontroller is highlighted for its role in data aggregation and real-time analysis. The chapter also introduces machine learning, explaining its applications in sports analytics. Finally, it covers statistical measures like Root Mean Square Error (RMSE) and dimensionality reduction techniques like Principal Component Analysis (PCA).

2.1 Types of Sensors

In sports, different sensors are used to collect a variety of data about athletes' movements, body mechanics, and physical responses. Among these sensors are accelerometers, gyroscopes, and inertial measurement units (IMUs), each playing specific roles in analyzing sports performance [3].

2.1.1 Accelerometer

An accelerometer is a sensor that detects changes in motion, specifically acceleration and provides information about the direction and magnitude of the change. This sensor is widely used in various devices, including smartphones for screen rotation and wearable sports devices. Accelerometers operate by measuring changes in motion using internal components that respond to movement. Within the sensor, these components move in response to changes in acceleration, and their movement is measured and translated into data. This data indicates the acceleration and direction of movement along three axes: X, Y, and Z.

2.1.2 Gyroscope

A gyroscope is a sensor that measures or maintains orientation and angular velocity. In simpler terms, it helps determine how an object is rotating or moving around its axis. Gyroscopes are commonly used in various devices like smartphones for screen orientation and play a crucial role in wearable sports gadgets like the nRF52840. Gyroscopes operate based on the principle of conservation of angular momentum. Inside the sensor, there's a spinning wheel or a vibrating element that resists orientation changes. When the device rotates or moves, this resistance is measured and translated into data that indicates the rate and direction of rotation or movement. Gyroscopes can measure rotational movement along different axes (three axes: X, Y, Z).

2.1.3 Inertial Measurement Units (IMUs)

Inertial Measurement Units (IMUs) combine accelerometers and gyroscopes into a single sensor package to measure both linear acceleration and angular velocity simultaneously. By combining accelerometers and gyroscopes, the IMU provides a comprehensive understanding of player movements. This technology allows for a deeper understanding of an athlete's biomechanics, enabling coaches and sports scientists to analyze movement patterns with greater precision. They are extensively utilized in sports science research, athlete monitoring systems, and wearable technology applications to enhance training methodologies and maximize athletic potential.

2.2 Bluetooth Low Energy (BLE)

BLE stands for Bluetooth Low Energy. It's a wireless communication technology designed for short-range communication with low power consumption. BLE is commonly used in various applications such as fitness trackers, smartwatches, medical devices, and home automation systems. It's an integral part of the Internet of Things (IoT) ecosystem, enabling devices to communicate with each other or other central devices while consuming minimal power. Bluetooth Low Energy (BLE) utilizes Universally Unique Identifiers (UUIDs) to uniquely identify various components within its communication protocol.

2.2.1 UUID

A Universally Unique Identifier (UUID) is a standardized identifier that consists of 32 hexadecimal digits grouped into five sections separated by hyphens, such as "b8b4e50d-f686-4757-918a-fcd2292782de". It is a 128-bit value commonly used as an identifier for services, and characteristics in a BLE system to ensure uniqueness and avoid conflicts.
UUIDs in BLE serve as unique identifiers for different components:

Service UUIDs:
Each service offered by a BLE device is identified by a UUID. These UUIDs define the type of service being offered, such as heart rate monitoring, battery level monitoring, etc. They help BLE devices understand what capabilities are provided by a particular device.

Characteristic UUIDs:
Characteristics within a service are also identified by UUIDs. Characteristics represent specific data points or functions within a service. For example, a heart rate service might have characteristics for measuring heart rate and for enabling notifications when the heart rate changes.

2.3 Seeed Studio XIAO nRF52840

Figure 2.1 represents the Seeed Studio Xiao NRF52480 which is a microcontroller board packed with features, designed for various applications including wearables and IoT projects. Powered by the Nordic nRF52840 System on Chip (SoC), it offers capabilities that make it an adaptable solution for innovative projects. The Seeed Studio Xiao NRF52480 offers several key features[4]:

-Powerful wireless capabilities: Bluetooth 5.0 with an onboard antenna.
-Ultra-Low Power: Standby power consumption is less than 5A.
-Battery charging chip: Supports lithium battery charge and discharge management.
-Onboard 6-axis LSM6DS3TR.
-Ultra Small Size: 21 x 17.5mm.

In the padel tennis, the Seeed Studio Xiao NRF52480 serves as the central component. It collects data from the integrated Accelerometer and Gyroscope, processes it in real-time, and sends it for analysis. This data helps in classifying different types of strokes and provides helpful guidance to assist players in enhancing their performance.

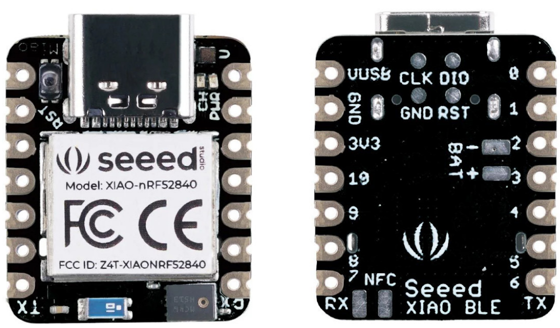

Figure 2.1: Seeed Studio XIAO nRF52840
[4]

2.4 Machine Learning

Machine learning [5] is a subfield of artificial intelligence (AI) that focuses on developing algorithms capable of learning from data to make predictions or decisions without being explicitly programmed. It includes a variety of techniques, including supervised learning, and unsupervised learning. Machine learning algorithms learn patterns and relationships from large datasets, allowing them to generalize and make predictions on new, unseen data. Applications of machine learning span across various domains, including finance, healthcare, marketing, and sports analytics.

2.4.1 Machine Learning Approaches

Supervised Learning

In supervised learning, the algorithm learns from labeled data, where each input example is associated with a corresponding target or label. The goal is to learn a mapping from inputs to outputs, enabling the algorithm to make predictions on new, unseen data.

Unsupervised learning

Unsupervised learning involves training algorithms on unlabeled data. The objective is to discover hidden patterns or structures within the data, such as clustering similar data points together or reducing the dimensionality of the data.

2.4.2 Machine Learning Algorithms

Supervised Learning Algorithms:

Linear Regression:
Linear regression is a simple yet powerful supervised learning algorithm used for predicting continuous values. It works by finding the best-fitting line through the data points, minimizing the difference between the predicted and actual values.

Logistic Regression:
Logistic regression is used for classification tasks. It models the probability of a binary outcome based on one or more predictor variables using a logistic function.

Support Vector Machines (SVM):
SVM is a powerful supervised learning algorithm used for classification and regression tasks. It finds the hyperplane that best separates classes in feature space while maximizing the margin between them.

Decision Trees:
Decision trees are supervised learning algorithms that recursively split the data based on feature values, creating a tree-like structure where each leaf node represents a class or value. They are intuitive and easy to interpret, making them popular for both classification and regression tasks.

Random Forest:
Random forests are ensemble learning methods that construct multiple decision trees and output the mode of the classes (for classification) or the average prediction (for regression). By averaging over multiple trees, random forests reduce overfitting and improve generalization.

Unsupervised Learning Algorithms:

K-Means Clustering:
K-means is a popular clustering algorithm that partitions data into k clusters based on similarity. It iteratively assigns data points to the nearest cluster centroid and updates the centroids until convergence, aiming to minimize the within-cluster variance.

Hierarchical Clustering:
Hierarchical clustering builds a hierarchy of clusters by recursively merging or splitting clusters based on their similarity. It doesn't require specifying the number of clusters beforehand and provides a hierarchical visualization of the clustering process.

Principal Component Analysis (PCA):
PCA is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving most of the variability in the data. It achieves this by finding orthogonal axes (principal components) that capture the maximum variance.

2.5 Machine Learning in Sports

Machine learning has been increasingly utilized in sports to gain insights, improve performance, enhance decision-making processes, and even prevent injuries. One significant application is in performance analysis, where machine learning algorithms process large amounts of data from sensors, video footage, and other sources to analyze player movements, tactics, and strategies. This analysis can help coaches and athletes identify strengths, weaknesses, and areas for improvement. Additionally, machine learning models can predict player performance, match outcomes, and even optimize game strategies based on historical data and real-time inputs. Overall, machine learning has emerged as a powerful tool in sports, offering opportunities to enhance performance, prevent injuries, and elevate the overall sporting experience for athletes, and coaches.

2.5.1 Applications of Machine Learning in Sports

Machine learning (ML) has found numerous applications in the world of sports [6], revolutionizing various aspects of training, performance analysis, Injury prevention, and more.

Examples:

Performance Analysis:
ML algorithms can analyze vast amounts of data from sensors, video feeds, and other sources to provide insights into athletes' performance. This analysis can include techniques like biomechanical analysis of movements, tracking player positions, and identifying patterns in gameplay.

Injury Prevention:
ML models can analyze data from wearable sensors to detect patterns that may indicate injury risk. By monitoring factors like player movements, heart rate, and muscle fatigue, ML algorithms can help coaches and medical staff identify when athletes are at risk of injury and adjust training routines accordingly.

Player Recruitment and Scouting:
ML algorithms can analyze player statistics and performance data to identify potential talent. By identifying patterns in player performance, ML can help scouts and coaches make more informed decisions about which players to recruit.

Game Strategy and Tactics:
ML can analyze historical data on team performance, opponent behavior, and game conditions to help coaches develop effective game strategies. This can include predicting opponent tactics, identifying optimal lineups, and suggesting in-game adjustments based on real-time data.

In-Game Performance Monitoring:
ML algorithms can process real-time data from sensors worn by athletes to provide coaches with insights into players' physical condition and performance during games. This information can help coaches make strategic decisions such as when to substitute players or adjust tactics.

2.6 Clustering

Clustering [7] is an unsupervised machine learning technique that involves partitioning a dataset into groups, or clusters as shown in figure 2.2, where data points within the same cluster share similarities based on predefined criteria or distance metrics. The goal of clustering is to identify inherent structures in the data, grouping similar objects together and distinguishing dissimilar ones. In simpler terms, clustering aims to discover patterns and structures in data without any prior knowledge of group labels, making it a valuable tool for exploratory data analysis, data visualization, and knowledge discovery in various fields such as data science.

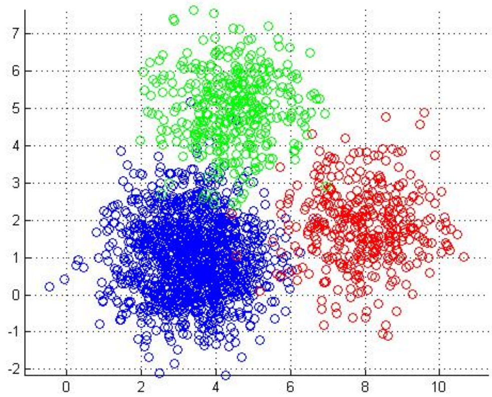

Figure 2.2: Clustering
[8]

2.6.1 K-means Clustering

K-means clustering is a partitioning method in unsupervised machine learning that aims to divide a dataset into K clusters, where K is a predefined number. The algorithm iteratively assigns each data point to the nearest centroid (cluster center) and then recalculates the centroids based on the mean of the data points assigned to each cluster. The process continues until the centroids no longer change significantly or a specified number of iterations is reached. Mathematically, it can be defined as follows:

1. Initialization: Choose k initial cluster centroids randomly from the data points. These centroids represent the centers of the initial clusters.

2. Assignment Step: For each data point, calculate the distance between the point and each centroid. Assign the data point to the cluster whose centroid is closest to it. This can be achieved using a distance metric such as Euclidean distance.

$$d(x_i, c_j) = \sqrt{\sum_{l=1}^{d}(x_{il} - c_{jl})^2}$$

3. Update Step: After all data points have been assigned to clusters, recalculate the centroids of the clusters based on the mean of the data points assigned to each cluster.

4. Iteration: Repeat the assignment and update steps until convergence, i.e., until the centroids no longer change significantly or a maximum number of iterations is reached.

The objective of K-means clustering is to minimize the within-cluster variance, often measured as the sum of squared distances from each data point to its assigned centroid.

$$J = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} ||\mathbf{x} - \mu_{\mathbf{i}}||^2$$

where:

$$\begin{aligned} k &: \text{Number of clusters} \\ C_i &: \text{Set of data points assigned to cluster } i \\ \mu_{\mathbf{i}} &: \text{Centroid of cluster } i \\ ||\mathbf{x} - \mu_{\mathbf{i}}||^2 &: \text{Squared Euclidean distance between data point } \mathbf{x} \text{ and centroid } \mu_{\mathbf{i}} \end{aligned}$$

2.7 Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) [9] is a statistical measure that quantifies the magnitude of errors between predicted and observed values in a dataset. It gives a clear measure of how close predictions are to actual values by averaging the prediction errors. RMSE is calculated by taking the square root of the mean of the squared differences between predicted and observed values. A lower RMSE indicates that the model's predictions are closer to the actual values, while a higher RMSE suggests that the model's predictions have larger errors compared to the actual values.
The equation for calculating the Root Mean Square Error (RMSE) is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

where:

$$\begin{aligned} n &: \text{Total number of observations or data points in the dataset} \\ y_i &: \text{Observed or actual value for the } i\text{th data point} \\ \hat{y}_i &: \text{Predicted value for the } i\text{th data point provided by the model} \end{aligned}$$

2.8 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [10] is a statistical technique used for dimensionality reduction in data analysis. The primary goal of PCA is to identify patterns in data by transforming the original variables into a new set of uncorrelated variables called principal components. These principal components are linear combinations of the original variables and are ordered in such a way that the first few components capture the maximum variance present in the data. PCA achieves dimensionality reduction by retaining only the most significant features while discarding redundant or less informative ones. This reduction simplifies the dataset, making it easier to visualize and understand while still preserving much of the original information. PCA is widely used in various fields, including data visualization, feature extraction, and data preprocessing for machine learning algorithms. It helps in identifying underlying structures and patterns in high-dimensional datasets, facilitating tasks such as clustering, and classification.

Chapter 3

Related Work

The relevant works will be discussed in this section, and how researchers used accelerometer and gyroscope data to classify different types of strokes.

In [11] this study introduces an automated method for analyzing tennis matches based on player strokes, using a single Wireless Inertial Measuring Unit (WIMU) attached to a player's forearm. The WIMU includes accelerometer, gyroscope, and magnetometer sensors. The stroke detection system uses a two-step process: first filtering non-stroke events from accelerometer data, then categorizing candidate strokes into serves, backhands, and forehands. Non-stroke events are distinguished through a trained binary classifier, differentiating between strokes and other arm movements or activities commonly encountered during matches. Candidate stroke classification involves training separate classifiers for serves, backhands, and forehands using data from each sensor type. Once noise is filtered from the data, temporal locations of candidate strokes are determined, and classifiers are applied to predict stroke types based on accelerometer, gyroscope, or magnetometer data. Significantly, accelerometer classifiers achieved a remarkable 79% accuracy, however, the gyroscopes performed the worst, the reason is that measurement of temporal orientation during a stroke varies significantly among players of different skill levels. Combining data from all three sensors through sensor fusion resulted in an impressive 90% stroke recognition accuracy, surpassing the results obtained from individual sensor classifications.

In [12] this study looks at how artificial intelligence can help analyze padel tennis strokes. To do this, researchers collected data on over 2000 strokes using small devices attached to players' wrists. They made sure to organize the data well and developed methods to accurately identify and separate each stroke. The strokes were categorized into 13 groups based on various techniques commonly used in padel tennis, such as different types of ground strokes, volleys, lobs, smashes, and serves. To classify these strokes effectively, the researchers tested six different computer algorithms: fully connected neural networks, 1D convolutional neural networks, decision trees, K-nearest neighbors, support vector machines (SVM), and Eigenvalue classification. Each algorithm was trained using two types of input data: the raw temporal series of strokes and feature-engineered data representing key characteristics of each stroke. Results show that temporal series input generally outperforms feature engineering. The 1D convolutional neural networks achieves the highest accuracy of 93.35%, followed closely by SVM. However, K-nearest neighbors and SVM present lower time complexity.

In [13] the process involves accurately detecting and classifying three common tennis strokes: forehand, backhand, and serve. Accelerometer and gyroscope data are observed, with noticeable peaks in accelerometer readings for each stroke. To detect strokes accurately, a two-point derivative of the acceleration curves is calculated to identify significant changes at the point of impact. By averaging the derivative across all three accelerometer axes, the differences in the stroke force for different players do not represent a vital influence for accurate stroke detection. A threshold parameter of 300 was found suitable for consistent detection across various players. The classification process utilizes gyroscope information along with accelerometer data. The algorithm identifies minimum and maximum values around the point of contact (when the tennis ball hits the racket) to determine stroke type. Classification criteria include the axis of maximum angular rate and additional conditions based on gyroscope readings. Evaluation of stroke classification accuracy was performed on a database containing recordings from seven tennis players in various conditions (indoors/outdoors, clay/hard surfaces). Results indicate relatively high classification accuracy, providing a platform for further improvement. Forehand stroke classification shows the lowest accuracy due to its frequent use and variability in player positioning.

In [14] this paper proposes an AI-assisted system for table tennis training, leveraging machine learning and real-time sensor data. The proposed system comprises SensorTile (The SensorTile is an Internet of Things module that integrates various sensors, such as accelerometers, gyroscopes, and magnetometers onto a small board), Data Acquisition, Feature Extraction, Machine Learning Algorithm, Classify Tennis Stroke, and Display Modules. Data is collected using SensorTile's MEMS sensors, capturing acceleration, gyroscope, and magnetic data. Each table tennis stroke is segmented into two-second intervals, resulting in 180-dimensional data points(60 for each). Various machine learning algorithms are tested, including Support Vector Machine (SVM), Neural Networks (NN), Decision Trees, Random Forest, and K-nearest Neighbor (k-NN).400 strokes of data are collected, with 70% used for training and 30% for testing. Neural networks achieve the highest accuracy of 100%, followed by Random Forest with 99.3%.

In [15] the paper aims to recognize and classify shot types and footwork movements in tennis. It defines five shot classes (forehand topspin, forehand slice, backhand topspin, backhand slice, and smash) and two-step classes (shot steps and side steps) relevant to gameplay. A wearable sensor system is designed to capture motion data from the tennis racket and the player's feet. Sensors are attached to the racket and shoes to gather data on shot type and footwork, respectively. Shot movements are classified using a Longest Common Subsequence (LCSS) algorithm applied to gyroscope data from the tennis racket. Foot movements are classified using gyroscope data from foot-mounted sensors. A motion estimation technique is utilized to detect step moments, and a support vector machine (SVM) is used for step classification based on features extracted from the sensor data. The system's performance is evaluated using data collected from amateur and expert tennis players. User-dependent and user-independent evaluations are conducted for shot and step classification, with precision values analyzed. Shot classification accuracy improves significantly with user-dependent training data, reaching up to 94%. Footstep detection shows a high accuracy of around 95%, with slight variations based on the amount of training data used.

In [16] this paper describes a comprehensive methodology and experimental setup for stroke recognition in table tennis utilizing wearable sensor technology. Key aspects involve data preprocessing, feature extraction, and classification using KNN and SVM algorithms. Normalization techniques, such as linear and zero mean normalization, are investigated to enhance recognition performance. Additionally, Principal Component Analysis (PCA) is used for dimensionality reduction. Data is collected from 15 participants executing five distinct table tennis strokes, resulting in a dataset of 750 strokes for analysis. The analysis shows that while the KNN algorithm demonstrates recognition rates of 94%, its practical value is limited by its dependency on training sample size and computational inefficiency. Conversely, the SVM algorithm achieves satisfactory classification and recognition rates, addressing the limitations observed with KNN.

In [17] the researchers integrate an MPU-6050 sensor, which includes both an accelerometer and gyroscope, into a badminton racket. This sensor captures raw acceleration and gyroscope data, which is then processed and analyzed using MATLAB. The study identifies key badminton moves, including serve, return, smash, backhand, and forehand, which are essential for activity recognition. Feature selection focuses on identifying important features for activity recognition, such as mean, median, standard deviation, variance, RMS, peak values, and energy. The recognition of activities is achieved through various algorithms, including Root Mean Square (RMS) value calculation, K-Nearest Neighbors (K-NN) algorithm, and Support Vector Machines (SVM). Results from RMS analysis indicate limitations due to similarities in RMS values across different activities. However, K-NN classification yields an average accuracy of approximately 58%, while SVM achieves an accuracy of 88.89% on a dataset that includes smash, serve, and backhand activities.

In [18] the paper explores the application of wearable sensors in sports monitoring, particularly focusing on tennis and table tennis. The Arduino Nano 33 BLE Sense board is identified as a key hardware component due to its integration of various sensors, including the LSM9DS1 for 3D accelerometer, gyroscope, and magnetometer functionalities. This microcontroller board is equipped with Bluetooth 5 for wireless communication, requiring only a power source to operate. The software development process involves four main steps: programming the Arduino Nano 33 board for data collection, converting collected data to CSV files on a personal computer, training a neural network on Google Colaboratory using TensorFlow 2.0, and developing a program for the Arduino Nano 33 board to utilize the trained neural network. Data was collected for ten forehand and ten backhand strokes, with each stroke record containing 300 values representing accelerometer and gyroscope readings. The implemented neural network achieved 100% accuracy in forehand stroke detection but showed a slight decrease in accuracy to 96% for backhand strokes.

Chapter 4

Methodology

This chapter focuses on the hardware components and software implementation of the padel stroke prediction system. In terms of hardware, the system utilizes a rechargeable battery for power and a specially designed glove embedded with the chip for motion sensing. The software implementation involves configuring a microcontroller using Arduino, establishing Bluetooth communication with a server, and employing motion detection algorithms to identify strokes. Features extraction, clustering analysis, and stroke prediction are key software processes, along with integration with an Android application for real-time stroke visualization and analysis. The chapter details each component's functionality and role in the overall system architecture.

4.1 Hardware Components

4.1.1 Battery

A battery as shown in figure 4.1 is used as a power source for the microcontroller which is a 3.7V rechargeable battery with a capacity of 700 mAh. Its rechargeable nature makes it cost-effective and environmentally friendly, as it can be recharged and reused multiple times. The small size of this battery makes it ideal for integration into small electronic devices where space is limited, it can be easily installed into wearable devices.

Figure 4.1: Battery
[19]

4.1.2 Glove

Figure 4.2 represents a glove that is worn by players in matches or training, in which the chip is embedded.

Figure 4.2: Glove

4.2 Software Implementation

The software implementation is done using Arduino, which is used to configure the microcontroller unit, the nRF52840. It initializes the connection via Bluetooth between it and an external laptop, which acts as a server using Python.

4.2.1 Microcontroller Configuration

The configuration of the nRF52840 is done using Arduino. It is set up to be discoverable to other devices for Bluetooth connections and to send the raw data of the accelerometer and gyroscope. This is done by using the following steps:

Libraries

Using ArduinoBLE, LSM6DS3, and Wire libraries.

-ArduinoBLE:
The ArduinoBLE library facilitates Bluetooth Low Energy (BLE) communication, enabling wireless connectivity.
-LSM6DS3 Library:
Library for interfacing with the LSM6DS3 sensor module, which is a sensor module that combines a 3-axis accelerometer and a 3-axis gyroscope, making it an inertial measurement unit (IMU). The LSM6DS3 communicates with microcontrollers like the nRF52840 using communication protocols like I2C (Inter-Integrated Circuit).
-Wire Library:
The Wire library allows communication with I2C devices, which is a simple communication protocol used to connect microcontrollers to sensors.

BLE initialization [BLE.Begin()]:

This initializes the Bluetooth Low Energy (BLE) functionality on the Arduino.

Setting Up BLE Service and Characteristic:

The ArduinoBLE library operates on the concept of service and characteristic UUIDs in Bluetooth Low Energy (BLE) communication. A BLE service is typically created with a unique UUID, for example: (b8b4e50d-f686-4757-918a-fcd2292782de). Within this service, a characteristic with a unique UUID is defined, such as: (85212279-0d50-4ca0-9a7e-730bdebf8e74), intended for reading (BLERead) and notifying (BLENotify) sensor data. This characteristic should have a size sufficient to hold 6 float values, as per your requirement.

Start Advertising [BLE.advertise()]:

Initiates advertising, making the chip discoverable for other BLE devices.

Data Collection:

The code in figure 4.3 collects accelerometer and gyroscope data from the microcontroller, storing these readings as float values for each axis in the newData structure. In this implementation, sensor data is read every 200 milliseconds, ensuring a consistent sampling rate.

```
newData.accelX = myIMU.readFloatAccelX();
newData.accelY = myIMU.readFloatAccelY();
newData.accelZ = myIMU.readFloatAccelZ();
newData.gyroX = myIMU.readFloatGyroX();
newData.gyroY = myIMU.readFloatGyroY();
newData.gyroZ = myIMU.readFloatGyroZ();
```

Figure 4.3: Data retrieval

Bluetooth Data Transmission:

Once the sensor data is captured, it's transmitted wirelessly as a BLE notification. The method shown in figure 4.4 allows the transmission of data wirelessly over Bluetooth using a BLE characteristic. It is provided with a pointer to the data (dataPointer) and specify the size of the data (dataSize).

```
characteristic.writeValue(dataPointer, dataSize);
```

Figure 4.4: Data Transmission

4.2.2 Bluetooth Low Energy (BLE) Communication

The bleak library is utilized, which is a Bluetooth Low Energy (BLE) library for Python, to establish communication with a BLE-enabled device, the nRF52480. The MAC address of the chip is used to establish the connection. It uniquely defines the chip which is defined as (1E:54:94:E5:F0:83). Data is transmitted in binary format and contains raw sensor readings for accelerometer and gyroscope axes. The received binary data is unpacked using the struct module to extract individual sensor values, including accelerometer and gyroscope readings along the X, Y, and Z axes.

4.2.3 Motion Detection

The defined motions are forehand, backhand, and smash. Predefined time intervals are established for each motion type, indicating the duration of motion capture for forehand, backhand, and smash motions. These intervals are chosen based on the expected duration of each motion type and can be adjusted as needed for specific sports activities. At the beginning of each motion interval, an audible signal is emitted to indicate the start of motion capture. Each motion type (forehand, backhand, smash) is associated with a specific audible signal, allowing the system to determine the type of motion being performed. In addition to motion intervals, non-motion intervals are also included into the system. These non-motion intervals represent periods during which the player is not actively performing a forehand, backhand, or smash motion. Sensor data collected during each motion interval is labeled according to the corresponding motion type. Similarly, sensor data obtained during non-motion intervals is labeled accordingly to differentiate it from active motion data.

4.2.4 Features Extraction

After collecting data during time intervals, features are extracted using data collected during each interval for the accelerometer and gyroscope data. The features calculated are:

-Mean: The average value along each axis (X, Y, Z).
-Maximum: The maximum value observed along each axis (X, Y, Z).
-Minimum: The minimum value observed along each axis (X, Y, Z).
-The average difference along each axis (X, Y, Z) between consecutive readings.

24 features are collected in total for each stroke type for both, the accelerometer and gyroscope, these features provide comprehensive information about the motion captured by the sensor data. They serve as quantitative descriptors for analyzing and understanding motion patterns.

4.2.5 Clustering Analysis

Clustering analysis is performed on the extracted features using the K-means algorithm. The feature matrix is constructed by arranging the extracted features of each motion event into a two-dimensional array. Each row of the feature matrix corresponds to a motion event, and each column represents a specific feature dimension. KMeans is configured to create 4 clusters, it effectively separates the motion intervals into distinct groups based on the similarity of their features. Following clustering, each motion interval is assigned a cluster label corresponding to the cluster it belongs to. These labels serve as identifiers for the different groups formed by the algorithm. Central to the clustering process are the centroids, which represent the mean feature values within each cluster. These centroids serve as representative points for their respective clusters, summarizing the characteristics of the motion intervals grouped within.

4.2.6 Data Storage

Data storage is achieved through the creation and population of a CSV file. The data stored in the CSV file includes essential information such as motion type, the centroid of each cluster, cluster labels, and extracted feature values for each motion instance.
As shown in figure 4.5 each stroke is labeled and has its features and the centroid of its cluster.

Motion	Centroid	Label	Mean_Ax	Mean_Ay	Mean_Az	Mean_Gx	Mean_Gy	Mean_Gz	Max_Ax	Max_Ay	Max_Az
non_moti	[	1	0.217613	-0.46283	0.888787	1.645	-1.835	-1.26	0.227408	-0.40309	0.909144
forehand	[	2	0.595715	-0.32949	0.786345	-18.5882	-40.1036	-9.8	1.596736	0.218624	1.161928
non_moti	[	1	0.191914	-0.48094	0.888778	-1.78267	-2.548	-1.60533	0.22936	-0.44701	0.986736
backhand	[	3	0.496589	-0.51547	0.787242	1.533001	-13.594	5.929001	1.586	0.650504	1.175592
non_moti	[	1	0.208083	-0.37576	0.935431	0.541333	2.244667	-5.61867	0.244976	-0.33818	1.00284
smash	[-	0	-0.10965	-0.1405	0.780263	11.403	-4.165	7.245	0.532408	0.559736	1.667008

Figure 4.5: Data Storage sample

4.2.7 Stroke Prediction

New sensor data is collected and accumulated at regular intervals. The system analyzes new sensor data by comparing it with centroids from the CSV file. The similarity between the new sensor data and centroids is quantified using Root Mean Square Error (RMSE). It calculates the average difference between the feature values of the observed sensor data and those of each centroid. A lower RMSE indicates a closer match between the observed motion and the characteristics of a particular motion category represented by a centroid. By computing the RMSE for each centroid, the system can identify the motion category that best aligns with the observed sensor readings.

4.2.8 Integration with Android Application

To facilitate real-time stroke prediction and analysis, an Android application was developed using Android Studio with Java. This application serves as a user interface, providing a seamless interaction experience for users while receiving stroke predictions from the Python-based prediction system via a TCP connection. Once the Python prediction system processes the sensor data and determines stroke predictions, it sends these predictions back to the Android application over the established TCP connection. The Android application features an intuitive user interface designed to display the strokes performed by the user in real-time. Through graphical representations or textual feedback, users can observe their motions as captured by the sensors integrated into their mobile devices. Additionally, the application dynamically updates to display stroke predictions received from the Python prediction system. In addition to real-time stroke visualization and stroke predictions, the Android application includes functionality to track and tally the number of strokes performed by the user for each motion category.

In this Android application, consisting of several activities, each component plays a distinct role in facilitating communication and displaying stroke count data collected from a TCP server.

MainActivity

The MainActivity serves as the entry point for the application, providing users with a simple interface featuring a single button labeled "Start." When this button is tapped, it triggers the opening of the MessageActivity, initiating the communication process with the server as shown in figure 4.6

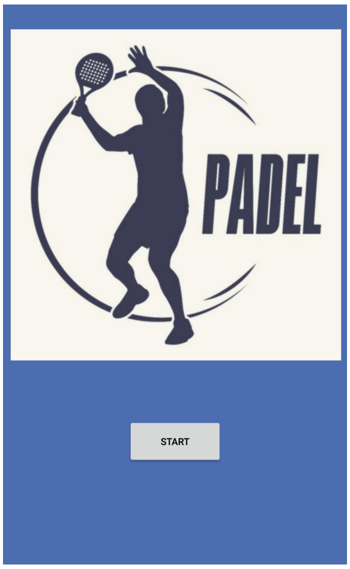

Figure 4.6: Home Page of the app

MessageActivity

In contrast, the MessageActivity is responsible for managing the TCP server functionality. Upon launch, it initializes the server and begins listening for incoming connections. As messages are received from connected clients, they are displayed in a designated text view within the activity's layout as shown in figure 4.7. Additionally, the activity includes a button labeled "End," which, when pressed, halts the server operation and transitions the user to the CountActivity, where the counts of various stroke types are displayed.

Figure 4.7: Page that displays the stroke type

CountActivity

Moving to the CountActivity, its primary role is to present users with a summary of the stroke counts gathered during the server operation. It retrieves this data from the previous activity and displays it in both text form and visually through a pie chart as shown in figure 4.8. Furthermore, the activity offers a button for users to return to the MainActivity, providing a seamless navigation experience within the application.

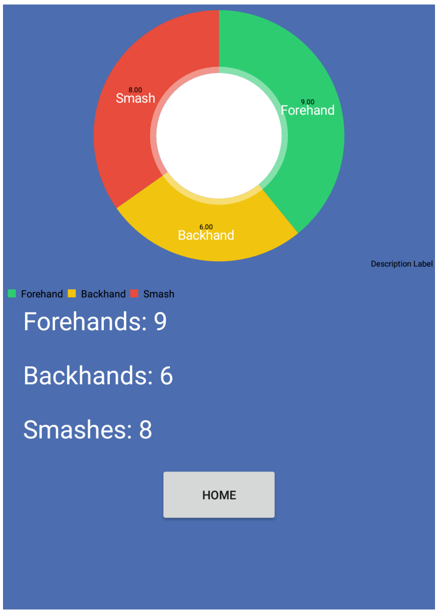

Figure 4.8: Page that displays the count of each stroke

TCPServer Class

Behind the scenes, the TCPServer class encapsulates the logic for handling TCP connections. It operates independently of the user interface, managing the receipt of stroke type messages from connected clients. Upon receiving each message, it updates the stroke count variables accordingly and communicates this information to the MessageActivity for display to the user. This separation of concerns ensures a clear and organized structure within the application, facilitating efficient communication between the user interface and underlying server functionality.

Chapter 5

Results And Analysis

This chapter explores the outcomes and analysis of the padel stroke prediction model. It begins by evaluating the model's accuracy in predicting three strokes: forehand, backhand, and smash. Then, Principal Component Analysis (PCA) is used to visualize motion data, revealing how well the model captures motion patterns. This analysis helps understand the effectiveness of the model in enhancing padel performance.

5.1 Accuracy of stroke prediction

The accuracy of stroke prediction was determined by comparing the predicted motion generated by the developed model with the actual motions performed. A prediction was considered accurate if the executed motion matched the predicted motion during the stroke execution. Following the completion of 50 strokes for each type, the accuracy of the predictions was calculated based on this criterion.

-Forehand:
After executing 50 forehand strokes and comparing each predicted motion with the actual motion performed, it was observed that 48 out of 50 predicted motions were consistent with the performed motion, resulting in an accuracy rate of 96%.
-Backhand:
Similarly, After executing 50 backhand strokes and comparing each predicted motion with the actual motion performed, it was observed that 47 out of 50 predicted motions were consistent with the performed motion, resulting in an accuracy rate of 94%.
-Smash:
Finally, After executing 50 smash strokes and comparing each predicted motion with the actual motion performed. It was observed that 48 out of 50 predicted motions were consistent with the observed motions, resulting in an accuracy rate of 96%.

These results show that the model works well in classifying different padel shots using sensor data.

Stroke type	Accuracy
Forehand	96%
Backhand	94%
Smash	96%

Table 5.1: Accuracy of the stroke prediction

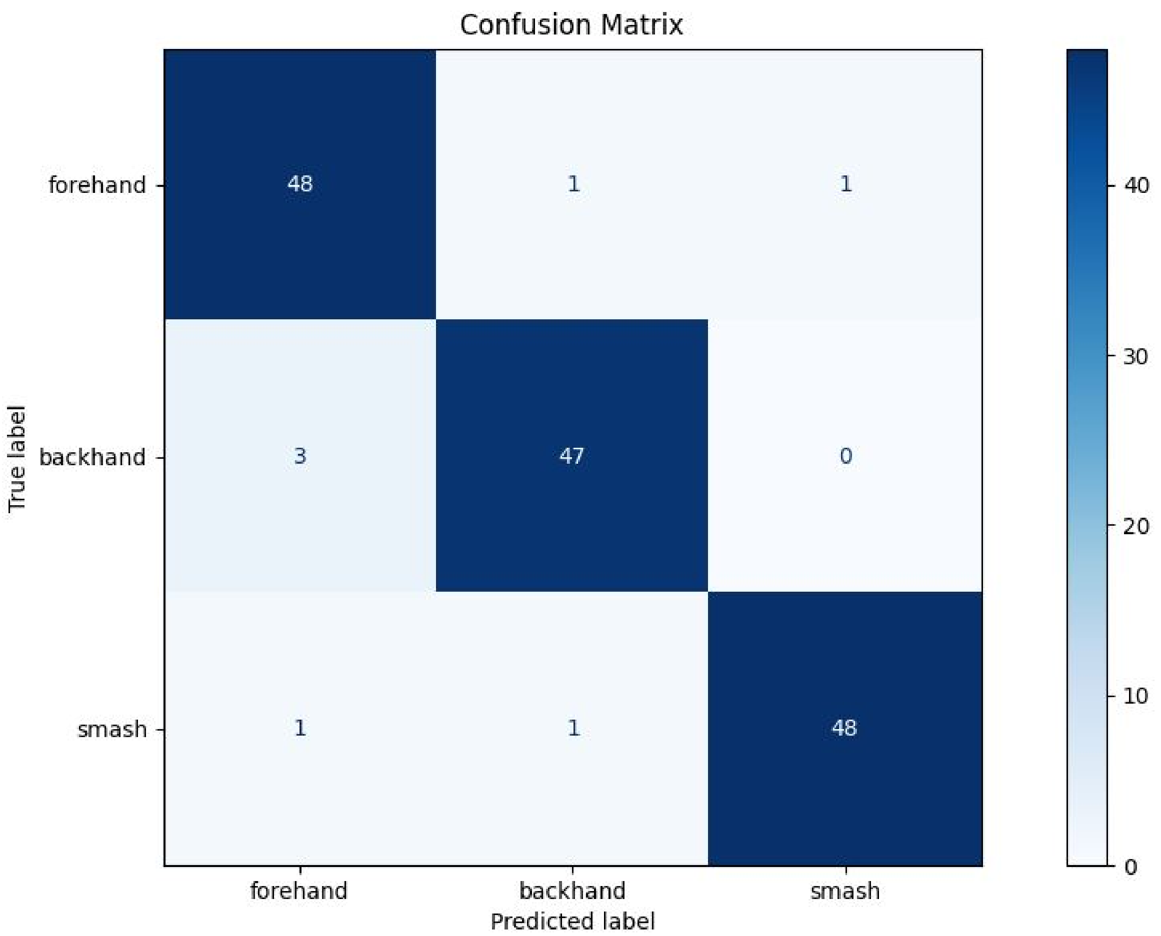

Figure 5.1: Confusion matrix of the stroke prediction

In the confusion matrix in figure 5.1, forehand strokes are only confused once each by smash and backhand strokes. Backhand strokes are confused three times by forehand strokes. Smash strokes are confused once each by forehand and backhand strokes. Despite a few errors, the model shows strong performance across different stroke types.

5.2 Visualizing Training and Testing Motion Data with PCA

In figure 5.2, Principal Component Analysis (PCA) is utilized to visualize both the original and predicted motion data. PCA as mentioned is a dimensionality reduction technique used to represent high-dimensional data in a lower-dimensional space while preserving the most important information. By applying PCA to the original motion data, the dimensionality of the feature space is reduced while retaining the key characteristics of the data. This allows us to visualize the data in a two-dimensional plot, with each point representing an instance of motion data. Similarly, PCA is applied to the predicted motion data. This enables us to project the predicted data onto the same lower-dimensional space defined by the original data. By visualizing both datasets in this reduced-dimensional space, the similarity between the trained and tested motion patterns can be evaluated. If the predicted data points cluster closely with the original data points of the same label, it indicates that the model has successfully captured the underlying patterns in the data. This visualization approach provides valuable insights into the performance of the predictive model and the accuracy of its predictions.

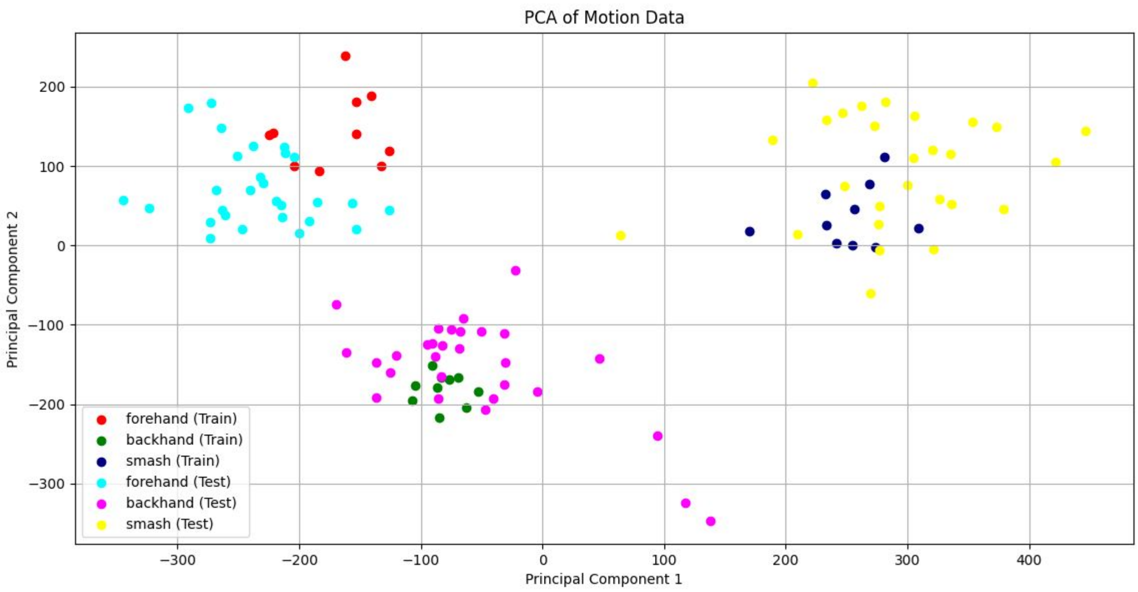

Figure 5.2: Principal Component Analysis (PCA)

Chapter 6

Conclusion And Future Work

6.1 Conclusion

This thesis aimed to utilize wearable sensor technology, particularly the Seeed Studio Xiao NRF52840 microcontroller board, to enhance padel tennis performance analysis. The primary goals were to create a wearable sensor system to track player shots during matches and to use advanced data analysis methods to generate information for improving player performance. The findings of this research hold significant implications for sports performance analysis. By integrating wearable sensor technology and advanced data analysis methodologies, athletes and coaches gain deeper information about performance. In conclusion, this thesis highlights the significant potential of wearable sensor technology in padel tennis performance analysis. By integrating data collection and analysis, athletes and coaches are equipped with valuable tools for performance optimization.

6.2 Future Work

There are several paths for future work to enhance the capabilities of the system:

Ball Tracking and Shot Outcome Detection:

Building on the existing stroke type detection, adding ball tracking and shot outcome detection would be a valuable next step. This enhancement would allow the system to analyze whether a shot is successful or missed, providing immediate feedback during matches or practice sessions. It could help players and coaches understand shot accuracy and effectiveness better, leading to improved performance and strategy refinement.

Increasing Database Size and Diversity:

Collecting data from different players in general while also specifically including left-handed players would indeed be an excellent approach to enhance the representativeness and diversity of the database. This combined strategy ensures that the database encompasses a wide range of player demographics, including both right-handed and left-handed individuals. By including data from players of various backgrounds, skill levels, and handedness, you can create a more comprehensive resource for understanding padel tennis performance across different groups.

Bibliography

[1] https://www.matellio.com/blog/iot-in-sports/.

[2] https://www.techtarget.com/searchmobilecomputing/definition/wearable-technology.

[3] https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu.

[4] https://www.cnx-software.com/2021/12/02/seeed-xiao-ble-tiny-nrf52840-bluetooth-5-0-imu-sensor-microphone/.

[5] https://www.geeksforgeeks.org/types-of-machine-learning/.

[6] https://appinventiv.com/blog/ai-in-sports/.

[7] https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/.

[8] https://kshitiz-sharan.medium.com/hierarchical-clustering-in-machine-learning-428e4f7c6094.

[9] https://www.aporia.com/learn/root-mean-square-error-rmse-the-cornerstone-for-evaluating-regression-models/.

[10] https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[11] Damien Connaghan, Phillip Kelly, Noel E. O'Connor, Mark Gaffney, Michael Walsh, and Cian O'Mathuna. Multi-sensor classification of tennis strokes. In *SENSORS, 2011 IEEE*, pages 1437–1440, 2011.

[12] Alejandro Tapia Córdoba Daniel Gutiérrez Reina Guillermo Cartes Domínguez, Evelia Franco Álvarez. A comparative study of machine learning and deep learning algorithms for padel tennis shot classification. 2023.

[13] Marko Kos and Iztok Kramberger. A wearable device and system for movement and biometric data acquisition for sports applications. *IEEE Access*, 5:6411–6420, 2017.

[14] Kevin Ma. Artificial intelligence aided training in ping pong sport education. In *2020 Second International Conference on Transdisciplinary AI (TransAI)*, pages 43–49, 2020.

[15] Lars Büthe, Ulf Blanke, Haralds Capkevics, and Gerhard Tröster. A wearable sensing system for timing analysis in tennis. pages 43–48, 2016.

[16] Huihui Wang, Lianfu Li, Hao Chen, Yi Li, Sen Qiu, and Raffaele Gravina. Motion recognition for smart sports based on wearable inertial sensors. In Lorenzo Mucchi, Matti Hämäläinen, Sara Jayousi, and Simone Morosi, editors, *Body Area Networks: Smart IoT and Big Data for Intelligent Health Management*, pages 114–124, Cham, 2019. Springer International Publishing.

[17] Md. Ariful Islam Anik, Mehedi Hassan, Hasan Mahmud, and Md. Kamrul Hasan. Activity recognition of a badminton game through accelerometer and gyroscope. In *2016 19th International Conference on Computer and Information Technology (ICCIT)*, pages 213–217, 2016.

[18] Kristian Dokic, Tomislav Mesic, and Marko Martinovic. Table tennis forehand and backhand stroke recognition based on neural network. In Mayank Singh, P. K. Gupta, Vipin Tyagi, Jan Flusser, Tuncer Ören, and Gianluca Valentino, editors, *Advances in Computing and Data Sciences*, pages 24–35, Singapore, 2020. Springer Singapore.

[19] https://li-polymer-battery.com/3-7v-rechargeable-li-polymer-battery-lp902040-700mah-with-pcm-and-wires/.