WEARABLE SYSTEMS FOR PERFORMANCE ASSESSMENT IN VOLLEYBALL

A Thesis Submitted to
the Graduate School of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Mechanical Engineering

by
Muhammed Emin Özdemir

July 2022
İZMİR

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Assist. Prof. Dr. Kerem Altun, for his guidance, support and inspiration through the thesis. His patience and encouragement are greatly appreciated. I have been fortunate to have Dr. Altun as my advisor.

My labmates Irmak Ege and Mehmet Ogün Sarp, to whom I am thankful for helping me with the many difficulties I encountered in this study.

Next, I would like to thank my friends, Muhammet Fatih Polat and Mustafa Kemal Kayahan for their encouragements, collaboration and contributions.

I am also beholden to Levent Matmeriçli, Ömer Tok and Altay Demirci from Altekma Volleyball for assisting me with their valuable technical knowledge and supports.

I am also grateful to Çağrı Aydın for helping me to organize data collection sessions and players of men's and women's institutional volleyball teams from Izmir Institute of Technology, who took time out of their lives and participated in the experiments in this thesis.

Finally, I would like to express my very profound gratitude to my lovely sister and parents for providing me with unfailing support and continuous encouragement through the process of researching and writing this thesis.

ABSTRACT

WEARABLE SYSTEMS FOR PERFORMANCE ASSESSMENT IN VOLLEYBALL

Nowadays, wearable sensors are used for many applications such as healthcare, animation, sports, to name but a few. In this study, they are used to recognize volleyball activities such as digs, blocks, serves and spikes. These activities are normally followed by statisticians on the field, their presences and frequencies are noted by them to be recorded at the match report. This study focuses on automating this procedure and identifying/recognizing them using wearable sensors. Five Xsens MTw Awinda sensors are used to collect data from 10 volleyball players (5 women and 5 men) who are between 19-21 ages and have 3-12 years of experience as an active player in volleyball. In this thesis, optimum number of sensors and their locations, effects of combinations of different features such as minimum, maximum values, means and variances of the raw data, impacts of combinations of different sub sensors such as accelerometer, gyroscope and magnetometer on the 4-class&10-class classification average accuracies are investigated. Two classification algorithms are applied with two different cross validation methods: For both cross validation methods, LDA (Linear Discriminant Analysis) produced better average accuracies than KNN (K Nearest Neighbor) where k value is taken as 5. The average accuracies for 4-class and 10-class classifications are respectively 99.56% and 89.56%. However, these results are respectively 92.39% and 66.08% for KNN (k=5).

Keywords: *Activity Recognition; Volleyball Activity Recognition; Volleyball Activity Classification; Wearable Sensors; Inertial Sensing; Sensor Network Design*

ÖZET

VOLEYBOLDA PERFORMANS DEĞERLENDİRMESİ İÇİN GİYİLEBİLİR SİSTEMLER

Günümüzde giyilebilir algılayıcılar sağlık, animasyon, spor gibi pek çok uygulamada kullanılmaktadır. Bu çalışmada, manşet, blok, servis ve smaç gibi voleybol aktivitelerini tanımak için kullanılmıştır. Bu faaliyetler normalde sahadaki istatistikçiler tarafından takip edilir, hangilerinin kaçar adet icra edildiği onlar tarafından not edilerek maç raporuna kaydedilir. Bu çalışma, istatistikçiler tarafından gerçekleştirilen bu işin otomatikleştirilmesi ve bunları giyilebilir algılayıcılar ile tanımaya/tanımlamaya odaklanmaktadır. Beş Xsens MTw Awinda algılayıcısı, yaşları 19-21, aktif voleybolculuk deneyimleri 3-12 yıl arasında değişmekte olan 10 voleybolcudan (5 kadın ve 5 erkek) veri toplamak amacıyla kullanılmaktadır. Bu tezde, en uygun algılayıcı sayısı ve konumları, ham verilerin minimum, maksimum değerleri, ortalamaları ve varyansları gibi farklı istatistiksel özelliklerin bileşimlerinin etkileri, ivmeölçer, jiroskop ve manyetometre gibi farklı alt algılayıcıların bileşimlerinin 4'lü ve 10'lu sınıflandırmaların ortalama doğruluklarına etkileri incelenmektedir. İki farklı çapraz doğrulama yöntemi, iki farklı sınıflandırma yaklaşımı ile uygulanmaktadır: Her iki çapraz doğrulama yöntemi için, doğrusal ayırma analizi (DAA), en yakın komşular yaklaşımından (komşu sayısı 5 olarak alındığında) daha iyi ortalama doğruluklar üretmiştir. 4'lü ve 10'lu sınıflandırmaların ortalama doğrulukları sırasıyla %99.56 ve %89.56'dır. Diğer taraftan, bu sonuçlar en yakın komşular yaklaşımında (komşu sayısı 5 olarak alındığında), sırasıyla %92.39 ve %66.08 olmaktadır.

Anahtar Kelimeler: *Etkinlik Tanıma, Voleybolda Aktivite/Faaliyet/Hareket Tanıma, Voleybolda Aktivite/Faaliyet/Hareket Sınıflandırma, Giyilebilir Algılayıcılar, Eylemsizlik Algılama, Algılayıcı Ağ Tasarımı*

I dedicated this thesis to my dear parents Lütfiye and Harun Özdemir

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Latest technological improvements have conducted to the development of cost-effective, non-invasive, small sensors, appropriate for collecting sport performance measures throughout the training or competition. Sensors are used for several purposes such as technical assessment, match analysis, capacity analysis and activity classification. They are applied in numerous sports varying from team to individual sports, from cyclic to winter and outdoor sports. At the same time, they are utilized for motor capacity assessments such as jumping and overload trainings. There are several device fixing methods to investigate these sports. In some cases, sensors are fixed on the players using straps, belts, tapes or they are worn as suits or harness. Some applications include attachment of the sensors onto the sport equipment like baseball, racquet, punching bag, skate chassis or skis [1].

This study focuses on classification and recognition of some activities which are also performed during a volleyball match. However, the data collected in this study includes distinct activities which last nearly five seconds. Using five IMUs positioned at different locations of the player's body, activities and sub activities produced by players are classified among dig, block, serve and spike categories.

At this introduction part, first, the literature survey about the motion capture systems, wearable sensors and their usage objectives are mentioned. Secondly, aim of the thesis is shared. Lastly, the boundaries and scope of this study are explained.

1.1. Literature Survey

Along with the developing technology, sports branches where competition is intense, seek ways to make use of the innovations, particularly in key areas such as performance enhancement and injury prevention. In this regard, motion capture technologies, which are the most used systems to ensure individual and/or team development, are examined under six different headings [2] as follows:

1) Optoelectronic motion capture systems
2) Electromagnetic motion capture systems
3) Inertial measurement unit motion capture systems

4) Motion capture systems based on image processing
5) Motion capture systems based on acoustic (sound) measurement
6) Exoskeleton and robotic motion capture systems

In this study, inertial measurement unit (IMU) motion capture system is used considering its advantages of easy installation, short analysis time, easy to use for field applications, high sensitivity and reasonable price advantages.

IMU motion capture systems, which are in the class of wearable sensors, are used for various purposes in volleyball:

1) Activity recognition
2) Limb movement tracking
3) Player's skills evaluation
4) Estimation of the player's jump count and jump height

Predominantly, the injuries in volleyball and beach volleyball result from overuse. Due to the large loads that take place for instance during a serve jump; shoulders, knees and the lower back tend to get injured. A major cause of impairment and reduced performance for professional players come from these resulting overuse injuries. Kautz et al. [3], studied to construct an automatic monitoring system based on wearable sensors and Deep Convolutional Neural Network. The researchers placed only one sensor unit (Bosch BMA280 tri-axis acceleration sensors) on the player's dominant-hand wrist to determine the presence and number of activities such as serve, set, short attack, spike, block and dig during beach volleyball training. They obtained 83.2% classification accuracy. A similar study was conducted by Haider et al. [4] to examine the effect of sensor placement on the player. To do that, they used XSENS MTw Awinda sensors. Accelerometer, gyroscope, magnetometer and barometer sensors were activated during the study. They conducted the experiments in three different modes. First, they placed one IMU only on the dominant hand. Then, secondly, only one IMU was located on the non-dominant hand and thirdly, they were mounted on the both hands. They concluded that using fusion approach, on average, dominant hand's best averaged unweighted average recall (UAR) (%) is slightly higher than the non-dominant hand's result. Usage of super-bagging method provided them a UAR of 84.19%. Roggen et al. studied to classify 64 beach volleyball serves from a forearm gyroscope sensor. The serves were recognized with only one false positive and 20 false negatives [5].

The issue of what kind of consequences the serve in volleyball can lead to due to excessive use of the ligaments and limbs in the shoulder area was considered in different studies. Rawadesh et al. aimed to track and discriminate shoulder motion gestures to help prevent shoulder over-use injuries. To do that, they attached an IMU on the upper arm of the player which is developed by themselves and contains an ITG-3200 MEMS triple-axis gyroscope (by InvenSense), an ADXL345 triple-axis accelerometer (by Analog Devices) and an HMC5883L triple-axis magnetometer (by Honeywell) sensors. They succeeded to attain a 94% accuracy to count correctly the number of throws and hits that performed by the subjects [6] [7].

Wang et al. [8] studied to evaluate the spikes of the volleyball players and classify the players as elite, sub-elite and amateur players. They used a wearable sensing device (WSD) developed by themselves which uses an accelerometer and a gyroscope. The sensor unit was positioned on the wrist and their player categorization method reaches an average accuracy of 94%.

In order to avoid injuries due to overload and improve player performance, it is also important to measure the jump height and frequency of the players. With this object in mind, Jarning et al. [9] examined and found that using a tri-axial accelerometer neither peak vertical acceleration (PVA) nor peak resultant acceleration (PRA) is enough to estimate jump frequency in volleyball.

Charlton et al. inspected to validate a small commercially available inertial measurement unit named as VERT (Mayfonk Athletic, Florida, USA). Its principal measurement outputs are jump count, jump height and landing impacts. It was inserted into an elastic waistband and demonstrated higher precision (0.995 – 1.000) when it is compared to the video analysis [10]. Skazalski et al. conducted a similar research with VERT (Model #JEM, Mayfonk Athletic, Fort Lauderdale, FL, USA). It was installed slimly inferior and lateral to the subject’s umbilicus. According to results, VERT accurately counts 99.3% of the 3637 jumps performed. It had a 9.7 cm of MDC (Minimum Detectable Change) and an average of 5.5 cm (95% CI 4.5-6.5) of overestimated jump height [11]. As previous studies assessed the VERT’s jump count and jump height measurement accuracies, Damji et al. focused on the validity of the VERT’s landing impact values. As a conclusion, they resolved that its landing impact values are usually inadequate with respect to the Shimmer (a research-grade accelerometer) [12]. Souse et al. also worked analysis and classification of the volleyball player's jumps using some machine learning models and sensors attached to the wrist and waist of the player [13]

Leeuw et al. strived to obtain player-specific relationship between overuse complaints, training load and wellness indicators. G-VERT was operated to get the number and heights of the players' jumps. It was demonstrated that G-VERT reports the jumps above 15 cm from the ground and detects these jumps with 99% accuracy. They deduced that to avoid overuse injuries, the monitoring should occur on a daily basis [14].

Tore et al. designed a glove which consists of 2 TekScan FlexiForce445N sensors and 1 TekScan FlexiForce A401 sensor to recognize the pressure applied by the hand to the ball during the serve [15]. Another study [16] employed a wrist-worn gyroscope to recognize and classify the beach volleyball serve types and they concluded that a sample rate higher than 300 Hz is needed.

Alonso et al. [17] attempted to control exercise effort and fatigue levels and analyze the player movements. They utilized Suunto t6d Black Smoke wrist-worn sensor to monitor the heart rate, Zephyr BioHarness belt-shaped biometric sensing device to measure the 3-axis body acceleration and Imote2 to prove the communication and sensing capability to the player. The study follows the training zones and estimates post-exercise oxygen consumption and maximal oxygen consumption. It identifies the jumps using KNN algorithm and classifies them with 93% true positives and 100% true negatives. In a similar research [18], the player's performance during the match was monitered using the heart rate measurement.

Holatka et al. [19] concentrates on classification of the sets of the volleyball players, judging the technical qualities and suggesting improvements like a coach. For this intent, MYO sensor armband unit (by Thalmiclabs) which contains an IMU and an EMG, was employed. In their survey, the sequence selection demonstrates optimal results for 54.4% of the samples and 26.6% of the selected sequences display minor displacements.

Among the studies reviewed, the most relevant and related research is the one which was conducted by Haider et al. [4]. In their research, they studied on classification of volleyball activities using two sensors on the dominant and non-dominant hand of the volleyball players. Eight volunteers participated at their work. The study consists of ten different volleyball activity classes and they are block, forearm pass, left hand pass, overhead pass, right arm pass/right hand pass/one hand pass/one hand touch, serve, smash, tip over net, underhand serve and noise.

1.2. Aim of the Thesis

The primary purpose of this thesis is to classify and recognize four selected fundamental volleyball activities which are dig, block, serve and spike. This study focuses on their recognition and classification when they are sampled as distinct activities. The study conducted by Haider et al. [4] includes only two sensors located at the wrists. In this thesis, alternatively, five sensors are used and they are positioned at different locations of the body compared to the aforementioned study. At the same time, it is aimed to evaluate various sensor combinations and find the optimum number and locations on the player's body because of the cost, installation effort and player's comfort. Effects of sub sensors such as accelerometer, gyroscope and magnetometer and impacts of features specifically taking minimums, maximums, means and variances of the raw data are also examined throughout this study. Minimizing the computational time by choosing the lowest and best k number for KNN algorithm is also investigated.

1.3. Scope of the Thesis

The scope of this study is limited to four selected fundamental volleyball activities which are dig, block, serve and spike. The variation of these activities are also considered as sub activities and in total there are ten of them. They are middle dig, left dig, right dig, middle block, left block, right block, serve, middle spike, left spike and right spike. Six different parameters such as gender, number of sensors, number and type of features, number and type of sub sensors, type of classification algorithms and type of validation methods have been considered. Each volunteer was asked to realize 12 digs, 12 blocks, 10 serves and 12 spikes. These were splitted into different categories to evaluate the constructed algorithm's 4-class and 10-class classification performance. Five sensors are used in this study and they are located at the left leg, right leg, waist, left arm and right arm. Only measurements coming from these sensor points are analyzed. Other locations of the body are not in the scope of this thesis. KNN and LDA classifiers are used within this study.

It should be noted that automatically tagging/identifying all the attempts realized by the players during a volleyball match is not within the scope of this research. However, this thesis can be considered as a one step closer for this objective.

CHAPTER 2

THEORY

The concentration of this thesis is to classify several volleyball activities. This aim can be achieved with human beings, too. However, developed technology gives opportunities to accomplish this plan using machine learning approach. The operation of programming computers to improve a performance measure using example data or experience is named as machine learning. Machine learning is investigated under three headings as supervised, unsupervised and reinforcement learning. In this study, as labeled data is utilized and the target is to make classifications, the focus is on the supervised machine learning applications.

2.1. What Is Machine Learning?

Machine learning is the process of programming computers to optimize a performance criterion based on example data or experience. There is a model described with a few parameters and the process of running a computer program to optimize the model's parameters based on the training data or experience, is defined as learning. The model could be predictive to make future predictions or descriptive to learn from data or both.

As the main task of machine learning is to make inference from a sample, it uses the statistics theory to construct mathematical models. The computer science has two roles: First, to solve the optimization problem, we need efficient algorithms in training. Besides, the big amount of data that we often have should be stored and processed. Secondly, after a model has been trained, its representation and algorithmic solution must also be efficient to make inference. In some cases, the learning or inference algorithm's efficiency, i.e. its space and time complexity, might be as significant as its predictive accuracy. [20]

There are three main types of learning. These are supervised, unsupervised and reinforcement learning methods:

2.1.1 Supervised Learning

Supervised learning aims to learn a pattern from the input to an output with correct values supplied by a supervisor [20]. Think yourself as a volleyball player in the pitch who is being observed by the coach to see how you serve and whether you do it correctly or not. This is similar to how a supervised learning method works, in which a model can learn from the information provided as a labeled dataset. A labeled dataset is the one in which each data is supported by an answer, a solution or a label. This labelling helps the model to learn and facilitate the problem's solution. [21]

So, a labeled dataset which contains sports types, for instance, can help the model whether the type of the sport is football, tennis, cycling or any other sport. If a model is trained by this approach, any time a new sport type is presented to the model; it can compare that sport type with the previously labeled dataset to predict the proper label.

Supervised learning deals with two types of problems which are classification and regression problems. Classification problems need the model to guess a discrete value and to tell the input data belongs to which specific class or group. This means, in a training dataset of vehicle images, each photo is pre-labeled as plane, boat or car. Then, the algorithm is tested to see how well it can properly classify new plane and car photos. Contrary, regression problems are concerned with continuous data. One application case, linear regression, should be familiar for algebra class attendees: what is the expected value of the y variable for a given x value? Similarly, an algorithm that aims to predict the price of a car based on its year of manufacture, brand, segment and model might be considered as a more realistic machine learning problem. [21]

As a result, if there is a set of available reference points to train the model, supervised learning might be considered as the best choice. However, the data to train the algorithms are not always available. [22]

2.1.2 Unsupervised Learning

As the name implies, there is no supervisor in unsupervised learning and all we simply have is input data. The goal is to identify the regularities in the input. The input space has a structure that causes some patterns to appear more frequently than others, and we want to recognize what occurs in general and what does not. This is known as density estimation in statistics. [20]

2.1.3 Reinforcement Learning

In certain applications, the system's output is a series of actions. For this scenario, a single action is negligible; what matters is the policy which is the series of correct actions to achieve the aim. In any intermediate state, there is not any optimal action; if an action is component of a good policy, it is good. In this scenario, the machine learning algorithm should be able to evaluate policy goodness and learn from previous good action series to build a policy. This type of learning approach is named as reinforcement learning. [20]. Table 1 given below sums up all these learning methods.

Table 1: Comparison table for supervised, unsupervised and reinforcement learning methods [23]

Criteria	**Supervised ML**	**Unsupervised ML**	**Reinforcement ML**
Definition	Learns by using labelled data	Trained using unlabelled data without any guaidance	Works on interacting with the environment
Type of data	Labelled data	Unlabelled data	No-predefined data
Type of problems	Regression and classification	Association and Clustering	Exploitation or Exploration
Supervision	Extra supervision	No supervision	No supervision
Algorithms	Linear Regression, Logistic Regression, SVM, KNN etc.	K-Means, C-Means, Apriori	Q-Learning, SARSA
Aim	Calculate outcomes	Discover underlaying patterns	Learn a series of action
Application	Risk Evaluation, Forecast Sales	Recommendation System, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

2.2. Machine Learning Methods

In this study, two different data analysis methods are used. Both of KNN and LDA classifiers are simple and fundamental methods. KNN classifier works based on votes coming from the selected number of nearest neighbors. LDA classifier tries to find the optimum solution for the minimum in-class variance and maximum between-class variance and classifies the data based on the obtained linear boundaries.

2.2.1 Data Analysis Methods

Collected data is analyzed with two different classification methods and validated with two different cross validation methods. KNN classifier is one of the easiest and fundamental method to make classifications. LDA is known as a dimensionality reduction technique and a classifier which is the second algorithm used to classify the obtained data. K-fold and leave one subject out cross validation methods are also commonly applied in data analysis studies to have an idea about the dataset whether it is biased or not.

2.2.1.1 K-Nearest Neighbor Method

K-Nearest Neighbors (KNN) is one of the simplest supervised machine learning algorithm mostly used for classification. This method needs the storage of the entire data. Then, when it comes to classify a new observation, the KNN method utilizes the similarity rule (the distance between the previous data and the new observation to be classified). The new observation is allocated to the most related class with respect to the majority of the votes coming from its k closest neighbors. The distance between an observation's neighbors is calculated based on the selected distance function. Most of the time, as default, it is Euclidean distance. However, Manhattan, Minkowski or Hamming distances are also possible as other distance options. Furthermore, when utilizing KNN algorithm and assigning a new sample to a class, the computation steps become more difficult as the number of existing samples in the dataset increases [24].

K value is determined based on the accuracy results obtained for 10-folds cross-validation tests. For the situation when 5 sensors were activated, the activities and sub activities are classified for different k values varying from 1 to 49. Only the odd values of the k are examined within this range. The highest accuracy for 4 activities classification obtained as 0.9391 for k=13. On the other hand, the highest accuracy for 10 sub activities classification obtained as 0.6608 for k=5. As 10 sub activities classification is more difficult than the 4 activities classification and also, for ease of calculation, simplicity and low computational cost; k value was taken as 5 for all along the study.

2.2.1.2. Linear Discriminant Analysis (LDA) Method

Linear discriminant analysis is mainly used for two different purposes. Principally, it is used for dimensionality reduction. If there is a high-dimensional data, LDA helps to find the most important features among all. It assists in transforming the data so that the classes become as distinguishable as possible. To make the classes as distinct as achievable, LDA tries to find the best feature options where between-class variation is maximized and within-class variations are minimized. When this optimum line is constructed (for a 2-class application), for every new point, LDA gives the class membership probabilities of these new points [25].

On the other hand, LDA is also used for classification purposes. In this study, linear discriminant analysis approach is used for the second purpose which is the classification of different actions and sub actions. There are two main assumptions made to use LDA as a classifier. The first one is assuming the class conditional distributions as Gaussian. The second assumption is that these Gaussians have the same covariance matrix. Let's have a look at the equations used for this classification.

Simple probabilistic models that simulate the class conditional distribution of the data $P\left(X \middle| y=k\right)$ for each class k can be used to develop LDA. Bayes' rule can then be used to make predictions for each training sample $x \in \Re^d$, as follows:

$$P(y=k|x) = \frac{P(x|y=k)P(y=k)}{P(x)} = \frac{P(x|y=k)P(y=k)}{\sum_l P(x|y=l)P(y=l)} \tag{2.1}$$

and the class k is chosen which maximizes this posterior probability.

For linear discriminant analysis, $P(x|y)$ is modeled as a multivariate Gaussian distribution with density:

$$P(x|y=k) = \frac{1}{(2\pi)^{d/2}\left|\Sigma_k\right|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k)\right) \tag{2.2}$$

where d is the number of features.

In accordance with the model above, the log of the posterior is:

$$\log P(y=k|x) = \log P(x|y=k) + \log P(y=k) + Cst \tag{2.3}$$

$$\log P(y=k|x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k) + \log P(y=k) + Cst \quad (2.4)$$

where the constant term Cst represents the denominator $P(x)$, in addition to other constant terms from the Gaussian. The class that maximizes this log-posterior is the predicted class.

The Gaussians for each class are assumed to have the same covariance matrix ($\Sigma_k = \Sigma$ for all k) in LDA, which is a particular case of QDA. The log posterior is reduced to:

$$\log P(y=k|x) = -\frac{1}{2}(x-\mu_k)^t \Sigma^{-1}(x-\mu_k) + \log P(y=k) + Cst \quad (2.5)$$

The log-posterior of LDA can also be represented as [26]:

$$\log P(y=k|x) = w_k^t x + w_{k0} + Cst \quad (2.6)$$

where

$$w_k = \Sigma^{-1}\mu_k \quad (2.7)$$

and

$$w_{k0} = -\frac{1}{2}\mu_k^t \Sigma^{-1}\mu_k + \log P(y=k) \quad (2.8)$$

2.2.2. Data Validation Methods

Two different validation method is utilized for this study. These are k-folds and leave-one-subject-out cross validation methods. In k-folds, the dataset is divided into k parts/folds randomly. For every trial, one fold is used for test and all the rest is used for training step. At the end of k trials, the average accuracy is calculated. Similarly, in leave-one-subject-out cross validation approach, the dataset is divided into the number of subjects. For every trial, differently from k-folds method, the dataset is divided non-randomly. At every step, one subject's data is conserved for test and the rest is used for training. In the end, the average accuracy is computed based on every part of the dataset.

2.2.2.1. K-Fold Cross Validation Method

In classification problems, we first divide our data set into train and test sets. Then, on the train data set, we build a model, and on the test data set, we test the predictions. However, there may be some problems with the train/test separation. It is possible that we were unable to separate the data sets randomly. We might have created a model based only on men or women of a certain age or from a specific region. This will cause overfitting problem. Cross validation helps us to resolve this problem.

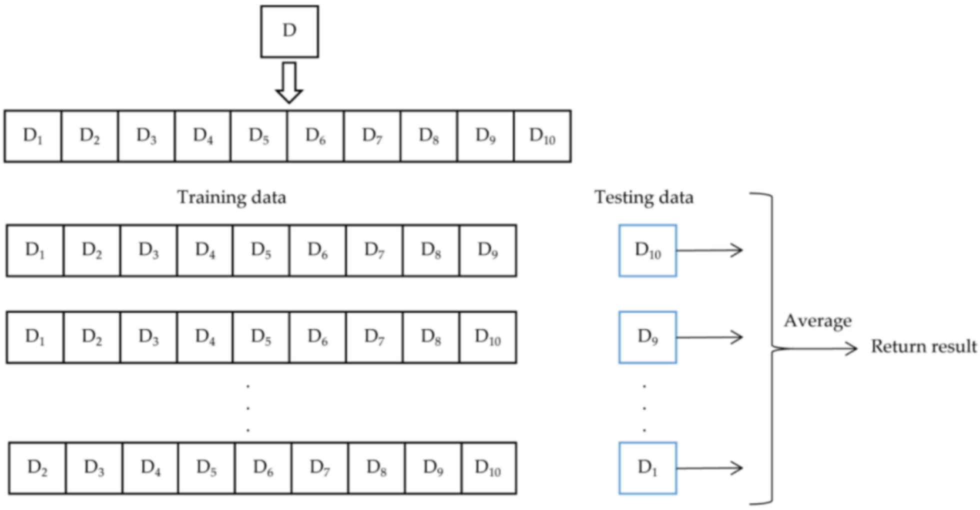


Figure 1: 10-folds cross validation [27]

In activity recognition problems, K-fold cross validation is considered as an accurate approach to select the model. In K-fold cross validation, the dataset is divided into k different subsets. As it is demonstrated in Figure 1 too, in 10-folds cross validation, 9 subsets are taken to train the model and 1 subset is taken to test the model. 10 experiments is done and the average of 10 experiments' results reveals the performance of the constructed model. As it is an easy-to-apply method, it is a widely used technique.

2.2.2.2. Leave-One-Subject-Out Cross Validation Method

Leave-One-Subject-Out (LOSO), which is also known as subject-independent cross-validation, is one of the most effective methods to assess realistic performance and can be used to stay away from probable overfitting. As LOSO takes into account inter-subject variability and previously unseen data is used to test; the accuracy decreases. LOSO cross validation is conducted for the classification algorithm where activity data for all volunteers except one is used. Then, the classifier is evaluated on the data for only the volunteer that is not included in the training dataset. Later on, the procedure is repeated for all volunteers and the average accuracy is calculated [28].

CHAPTER 3

METHODOLOGY

The experiments were conducted at the Izmir Institute of Technology Sports Hall. There were five male and five female volleyball players throughout the study to avoid gender biased results. Five sensors were used to be positioned at five different locations of the player's body. Each sensor was located at the same location for each trial. Every player realized four activities and ten sub activities. The collected data is exported using Xsens MT Manager 4.6 software and analyzed with Python version 3.9.7.

3.1. Experimental Setup

Inertial sensors are self-sustained devices which utilize direct measurements to give dynamic motion information. Accelerometers offer linear or angular velocity rate information, gyroscopes give angular rate information and magnetometers measure magnetic field strength.

Inertial sensors have been employed to navigate airplanes, land vehicles, boats and robots, to analyze the shocks and vibrations encountered in the automotive industry and telesurgy for several decades. Within the past few years, improvements of the accuracy of the inertial sensors made them good enough in terms of size, weight and cost for navigation and guidance applications [29] [30].

During the data collection procedure from the volunteers, MTw Awinda Development kit was used, manufactured by Xsens Technologies B. V. [31]. The kit contains:

- 5 MTw's
- 1 Awinda station
    - 1 USB cable
    - 1 power cable
- 6 Velcro body straps

3.1.1. Overview and Description of MTw Sensors

MTw’s are matchbox-sized inertial measurement units containing 3D linear accelerometers, 3D rate gyroscopes, 3D magnetometers and a barometer. Using these subcomponents, MTw’s supply respectively 3D acceleration, 3D angular velocity, 3D earth magnetic field and atmospheric pressure.

Figure 2: Front side of the MTw (motion tracker) [31]

The Velcro patch which is located at the back of the casing, helps attaching easily the sensors to the Velcro body straps.

Figure 3: Rear side of the MTw (motion tracker) where MTw product code (MTW2 – 3A7G6) and serial number (device id) are visible for the user to simply identify each MTW’s [31]

A LiPo battery helps the MTw to be in operation for up to 6.5 hours and after being one hour docked in the Awinda station, the battery becomes fully charged.

In this MTw, the gyroscopes measure the angular velocities within the range of ± 2000 deg/s, the accelerometers measure the accelerations in the range of ± 160 m/s^2 and the magnetometers measure the magnetic field between the range of ± 1.9 Gauss. Since the devices were calibrated at the factory by Xsens and the MTw provides calibrated 3D linear acceleration, angular velocity and (earth) magnetic field; the experiments were started without any extra calibration.

Figure 4: The object coordinate system [31]

All along the data collection procedure, participants have been instrumented of five MTw motion trackers. Two of them were worn on the arms, two of them were worn on the lower legs and one of them was worn on the waist of the volunteer, as depicted in Figure 5. The locations of the sensors have been determined by educated guesses.

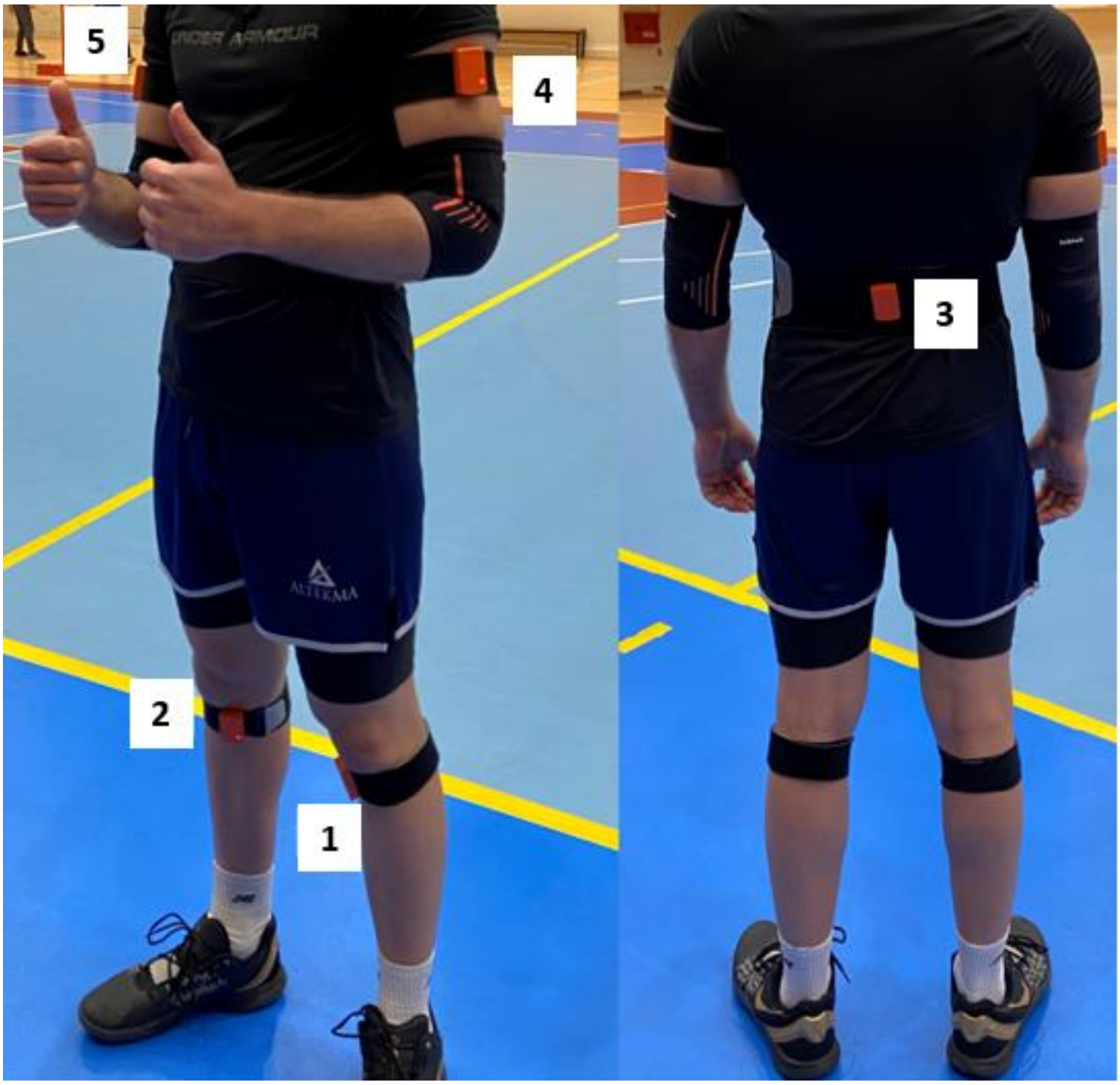


Figure 5: Positioning of the MTw sensors on the body volunteer's body

Each device was placed on the same area for all trials.

- Left Leg (LL): Sensor 1 (Device ID - SN: 00B44D29)
- Right Leg (RL): Sensor 2 (Device ID - SN: 00B45854)
- Waist (W): Sensor 3 (Device ID - SN: 00B46E84)
- Left Arm (LA): Sensor 4 (Device ID - SN: 00B46E88)
- Right Arm (RA): Sensor 5 (Device ID - SN: 00B46E90)

The 4 activities and 10 sub activities that are classified using these body-worn miniature inertial measurement units are listed respectively as follows:

- 4 classes: dig (A1), block (A2), serve (A3) and spike (A4)
- 10 classes: middle dig (MD/SA1), left dig (LD/SA2), right dig (RD/SA3), middle block (MB/SA4), left block (LB/SA5), right block (RB/SA6), serve (SRV/SA7), left spike (LSP/SA8), middle spike (MSP/SA9) and right spike (RSP/SA10).

3.2. Experimental Methodology

Each activity and sub activity listed above are performed by 10 different players (5 female, 5 male, between the ages 19 and 21). The player profiles are given in Table 2. The players are asked to perform the activities according to explained procedure.

Table 2: Players that performed the experiments and their profiles

Player No	Gender	Age (years)	Height (cm)	*Weight (kg)	**Experience Level (year)
P1	Female	19	175	57	10
P2	Male	19	201	95	4
P3	Male	20	180	84	3
P4	Male	21	185	75	12
P5	Female	21	176	73	12
P6	Female	21	175	58	11
P7	Female	19	169	53	8
P8	Female	19	178	63	4
P9	Male	20	185	80	5
P10	Male	20	190	87	3

*Shows the current weight of the player as 15/05/2022, not at the experiment day. **Signifies the total number of the active years of the players in volleyball.

The activities are performed at the Izmir Institute of Technology Sports Hall. Sensor units are set to 40 Hz update rate to acquire data. All the players have the right hand as the dominant hand.

3.2.1. Data Collection Procedure

1) The informed consent form was given to the players who wanted to attend the research project.
2) The data collection steps started with the players who had accepted/signed the form.
3) MT Manager 4.6 software provided by XSENS was started to collect the data.
4) Sensors were placed on the players as illustrated in the Figure 5.
5) An orientation (heading) reset, which is also known as “bore sighting”, was applied to all sensors.
6) The order of the activities was determined randomly for each player using a 4-digit non-repetitive random number generator. The numbers 1, 2, 3 and 4 define respectively the activities dig, block, serve and spike. For instance, if the random list indicated 3, 1, 2, 4; it means that, first the player served 10 float serves, secondly made 4 middle digs, 4 left-footed digs, 4 right-footed digs respectively. Thirdly, the player made 4 middle blocks, 4 left-footed blocks, 4 right-footed blocks respectively. Lastly, he/she made 4 spikes from left front zone, 4 spikes from middle front zone and 4 spikes from right front zone. In total, every player realized 12 digs, 12 blocks, 10 serves and 12 spikes. So, in the end, every player had 46 sub activities.
7) The subjects were asked to perform the activities in explained manner:
   - During the middle dig, they put the feet parallel to each other and did not step forward with right or left foot. For the left-footed/right-footed digs, they put their foot away to the left/right, moved the ball forward and went back to the initial position.
   - During the middle block, they jumped directly towards the ball at the middle front zone. For the left/right block, they moved towards the ball from the middle front zone to the left/right front zone.

- During the serve, they did not walk or jump; they just realized a float serve.
- For the spikes, the players could run or jump as they want but they were asked to stop after falling the ground to avoid the irrelevant movements

3.3. Implementation

The activities and sub activities are recorded using Xsens MTw Awinda sensors and MT Manager 4.6. Using MT Manager, the activities and sub activities are exported as txt files. The exported txt files are stored in a different folder for every player. These folders include four different activity folders where there are digs, blocks, serves and spikes. In every activity folder, there are sub activities saved as txt files. They are read by code written in Python programming language version 3.9.7 and related information is extracted from these txt files to be analyzed.

The data files accessing approaches and classification codes are shared and tried to be explained within this thesis to help people who want to conduct a similar study by themselves.

3.3.1. Data Files Accessing

Before constructing the set of features, every sensor's readings are saved as mtb files (mt binary logfiles) using MT Manager 4.6. Then, they are opened one-by-one, exported using ASCII Exporter (*.txt) and saved under another new folder entitled as "04_All_Volunteers_Data_to_Analyse" which includes some other folders as well. To clarify this procedure, let's see it with an example:

First, we start with the first trial of the player one. It is middle dig one. As there are five sensors, there are five sensor records exported in a folder as shown in Figure 6.

Name	Date modified	Type	Size
2022-02-21-Player1_Dig_Middle1-000-00h46_00B44D29.txt	21.02.2022 00:46	Text Document	22 KB
2022-02-21-Player1_Dig_Middle1-000-00h46_00B46E84.txt	21.02.2022 00:46	Text Document	22 KB
2022-02-21-Player1_Dig_Middle1-000-00h46_00B46E88.txt	21.02.2022 00:46	Text Document	22 KB
2022-02-21-Player1_Dig_Middle1-000-00h46_00B46E90.txt	21.02.2022 00:46	Text Document	22 KB
2022-02-21-Player1_Dig_Middle1-000-00h46_00B45854.txt	21.02.2022 00:46	Text Document	22 KB

Figure 6: Exported dig txt files of the player 1 coming from five different sensors

As it is the middle dig one, the folder is named as "01_Player1_Dig_01_Middle1". In total there are twelve dig trials of the player one. All of them are saved under their own folder. As you can see in Figure 7, there are four middle digs, four left digs and four right digs realized by player one.

Name	Date modified	Type
01_Player1_Dig_01_Middle1	24.07.2022 18:31	File folder
01_Player1_Dig_02_Middle2	24.07.2022 18:21	File folder
01_Player1_Dig_03_Middle3	24.07.2022 18:21	File folder
01_Player1_Dig_04_Middle4	24.07.2022 18:21	File folder
01_Player1_Dig_05_Left1	24.07.2022 18:21	File folder
01_Player1_Dig_06_Left2	24.07.2022 18:21	File folder
01_Player1_Dig_07_Left3	24.07.2022 18:21	File folder
01_Player1_Dig_08_Left4	24.07.2022 18:21	File folder
01_Player1_Dig_09_Right1	24.07.2022 18:21	File folder
01_Player1_Dig_10_Right2	24.07.2022 18:21	File folder
01_Player1_Dig_11_Right3	24.07.2022 18:21	File folder
01_Player1_Dig_12_Right4	24.07.2022 18:21	File folder

Figure 7: Dig folders of the player1

As all the twelve trials belong to dig activity, they are stored as "01_Player1_Exported_01_Digs" as it is pointed in the Figure 8.

Name	Date modified	Type
01_Player1_Exported_01_Digs	24.07.2022 18:30	File folder
01_Player1_Exported_02_Blocks	24.07.2022 18:21	File folder
01_Player1_Exported_03_Serves	24.07.2022 18:21	File folder
01_Player1_Exported_04_Spikes	24.07.2022 18:21	File folder

Figure 8: Exported activity folders of the player1

Also, as these trials belong to player one, "01_Player1_Exported_01_Digs" folder is maintained under "01_Player1_Exported_Documents" folder as Figure 9 implies. As in total there are ten players/volunteers, it shows all the players folders under "04_All_Volunteers_Data_to_Analyse" folder.

Name	Date modified	Type
01_Player1_Exported_Documents	24.07.2022 18:23	File folder
02_Player2_Exported_Documents	24.07.2022 18:21	File folder
03_Player3_Exported_Documents	24.07.2022 18:21	File folder
04_Player4_Exported_Documents	24.07.2022 18:22	File folder
05_Player5_Exported_Documents	24.07.2022 18:22	File folder
06_Player6_Exported_Documents	24.07.2022 18:22	File folder
07_Player7_Exported_Documents	24.07.2022 18:22	File folder
08_Player8_Exported_Documents	24.07.2022 18:22	File folder
09_Player9_Exported_Documents	24.07.2022 18:22	File folder
10_Player10_Exported_Documents	24.07.2022 18:22	File folder

Figure 9: Exported player folders of all volunteers

First part of the file access code is shared below:

```
import pandas as pd
import numpy as np
import os
import re

root_path = r"C:\Users\Eminzd\Documents\04_All_Volunteers_Data_to_Analyse"
os.chdir(root_path)
main_folder = os.listdir(root_path)
entire_dataset = np.empty((0,184))
for t, subject in enumerate(main_folder):
    p = 0
    activity = os.listdir(f"{subject}")
    activity_tensor = np.array([])
    print(subject)
    for k, act in enumerate(activity):
        subactivity = os.listdir(f"./{subject}/{act}")
        subactivity_tensor = np.array([])
        print(act)
        for j, sub in enumerate(subactivity):
            p = p + 1
            h = 0
            if p in range(1, 5):
                h = h + 1
            elif p in range(5, 9):
                h = h + 2
            elif p in range(9, 13):
                h = h + 3
            elif p in range(13, 17):
                h = h + 4
            elif p in range(17, 21):
                h = h + 5
            elif p in range(21, 25):
                h = h + 6
            elif p in range(25, 35):
                h = h + 7
            elif p in range(35, 39):
                h = h + 8
            elif p in range(39, 43):
                h = h + 9
            else:
                h = h + 10
            txt_file_list = os.listdir(f"./{subject}/{act}/{sub}")
            df_tensor = np.array([])
            print(sub)
            features = np.empty((0, 184))
            for i, txt_file in enumerate(txt_file_list):
                print(txt_file)
```

As displayed in the code section above, first, required libraries and modules are imported. They are pandas and numpy libraries, os (operating system) and re (regular expression) modules. Pandas and numpy libraries are used for multi-dimensional array operations. Os module is required to communicate with the operating system to realize path manipulation, file opening operations etc. Re module is used for regular expressions. Main folder is “04_All_Volunteers_Data_to_Analyse” folder as root path defines. In this folder, there are t+1 subjects, for each subject, there are k+1 activities, h sub activities and p total number of trials realized by every subject.

3.3.2. Dataset Construction

Second part of the file access code is shared below:

```
        feature = np.empty((0, 36))
        with open(os.path.join(f"./{subject}/{act}/{sub}", txt_file), 'r') as file:
            for row, line in enumerate(file):
                if row == 4:
                    column_namelist = line.split("\t")
                    remove_namelist = ['SampleTimeFine', 'Year', 'Month', 'Day', 'Second']
                    for removed_item in remove_namelist:
                        column_namelist.remove(removed_item)
                    df = pd.DataFrame(columns=column_namelist)
                if row > 4:
                    line.strip("\t")
                    newline = re.sub(r'\t\t\t\t\t\t', '\t', line)
                    newline = newline.split('\t')
                    df = df.append(pd.Series(newline, index=column_namelist), ignore_index=True)
                    df = df.drop(columns=["PacketCounter"])
            df = df.astype(float).to_numpy()
            df = df[:,:9]
            print("df shape", df.shape)
            feature = np.append(feature, np.min(df, axis=0))
            feature = np.append(feature, np.max(df, axis=0))
            feature = np.append(feature, np.mean(df, axis=0))
            feature = np.append(feature, np.var(df, axis=0))
            features = np.append(features, feature)
    features = np.append(features, t+1)
    features = np.append(features, k+1)
    features = np.append(features, h)
    features = np.append(features, p)

    entire_dataset = np.concatenate((entire_dataset, features.reshape(1, 184)), axis=0)
    print("Dataset shape:", entire_dataset.shape)

np.savetxt("Action_Recognition_10Player_Dataset_5sensors_Together.csv", entire_dataset, delimiter=",")
```

When the txt file is opened at line 2 as shown in the code section above, first 4 lines are skipped as the information at these four lines is not used. At the fifth row, unused five columns named as “SampleTimeFine, Year, Month, Day, Second” respectively have been removed/deleted. Then, for the rows greater than four, empty spaces are removed; “PacketCounter” column is dropped and first nine columns with all lines have been saved as df.

From each txt file, tri-axial accelerometer, tri-axial gyroscope and tri-axial magnetometer information have been extracted. Then, minimum values of 9 columns, maximum values of 9 columns, means of 9 columns and variances of the 9 columns have been calculated respectively. After that, 36 columns have been obtained from sensor 1.

This procedure was repeated for every sensor with a for loop. All the sensors have the same weight. There is not any sensor priorities. In the end, a total of 180 columns (features) was obtained.

Sensor1												Sensor2	Sensor3	Sensor4	Sensor5	Player Number	Activity Number	Sub Activity Number	Number of Player's Total Action
Min									...Max...	...Mean...	...Var...								
Acc_X	Acc_Y	Acc_Z	Gyr_X	Gyr_Y	Gyr_Z	Mag_X	Mag_Y	Mag_Z											
1	2	3	4	5	6	7	8	9	10…18	19…27	28…36	37…72	73…108	109…144	145..180	181	182	183	184

Figure 10: Dataset construction steps (numbers at the bottom identifies the column numbers)

Additionally, there are label's at the last for columns. 181st column identifies the player number (varies between 1-10). 182nd column signifies the activity number (varies between 1-4). 183rd column expresses the sub activity number (changes from 1 to 10) and the last column which is the 184th column of the dataset depicts the number of player's total action (As there are 12 digs, 12 blocks, 10 serves and 12 spikes realized by players, this column's values change from 1 up to 46).

As a last step, as there are 10 players and 46 trials for each player, constructed full dataset includes 460 lines and 184 columns. The full dataset is saved as "Action_Recognition_10Player_Dataset_5sensors_Together.csv"

3.3.3. Data Analysis

After construction of the dataset, the next step is the data analysis part. First part of the data analysis code is given in the section below:

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

os.chdir(r"C:\Users\Eminzd\Documents")
# Define dataset from .csv files and print size of dataset
dataset = np.genfromtxt('Action_Recognition_10Player_Dataset_5sensors_Together.csv', delimiter=',')
print("Shape of dataset: ", dataset.shape)
```

At this part, multiple cell property of the Visual Studio Code is used. Scikit-learn library is also added at this part to use its built-in functions. At the first cell, previously saved csv file is accessed from “C:\Users\Eminzd\Documents” path. Using the saved csv, dataset variable is assigned.

At the second cell, different sensor combinations are constructed by selecting related columns. All of the thirty-one combinations have been created and saved as different variables at this part.

```
# For one sensor:
dataset1 = dataset[:,np.r_[:36,180:184]]
dataset2 = dataset[:,np.r_[36:72,180:184]]
dataset3 = dataset[:,np.r_[72:108,180:184]]
dataset4 = dataset[:,np.r_[108:144,180:184]]
dataset5 = dataset[:,np.r_[144:180,180:184]]

# For two sensors:
dataset12 = dataset[:,np.r_[:72,180:184]]
dataset13 = dataset[:,np.r_[:36,72:108,180:184]]
dataset14 = dataset[:,np.r_[:36,108:144,180:184]]
dataset15 = dataset[:,np.r_[:36,144:180,180:184]]
dataset23 = dataset[:,np.r_[36:108,180:184]]
dataset24 = dataset[:,np.r_[36:72,108:144,180:184]]
dataset25 = dataset[:,np.r_[36:72,144:180,180:184]]
dataset34 = dataset[:,np.r_[72:144,180:184]]
dataset35 = dataset[:,np.r_[72:108,144:180,180:184]]
dataset45 = dataset[:,np.r_[108:180,180:184]]

# For three sensors:
dataset123 = dataset[:,np.r_[:108,180:184]]
dataset124 = dataset[:,np.r_[:72,108:144,180:184]]
dataset125 = dataset[:,np.r_[:72,144:180,180:184]]
dataset134 = dataset[:,np.r_[:36,72:144,180:184]]
dataset135 = dataset[:,np.r_[:36,72:108,144:180,180:184]]
dataset145 = dataset[:,np.r_[:36,108:180,180:184]]
dataset234 = dataset[:,np.r_[36:144,180:184]]
dataset235 = dataset[:,np.r_[36:108,144:180,180:184]]
dataset245 = dataset[:,np.r_[36:72,108:180,180:184]]
dataset345 = dataset[:,np.r_[72:180,180:184]]

# For four sensors:
dataset1234 = dataset[:,np.r_[:144,180:184]]
dataset1235 = dataset[:,np.r_[:108,144:180,180:184]]
dataset1245 = dataset[:,np.r_[:72,108:180,180:184]]
dataset1345 = dataset[:,np.r_[:36,72:180,180:184]]
dataset2345 = dataset[:,np.r_[36:180,180:184]]

# For five sensors:
dataset12345 = dataset

accuracy_of_sensors = []

sensors =[dataset1,dataset2,dataset3,dataset4,dataset5,dataset12,dataset13,dataset14,dataset15,dataset23,dataset24,
dataset25,dataset34,dataset35,dataset45,dataset123,dataset124,dataset125,dataset134,dataset135,dataset145,
dataset234,dataset235,dataset245,dataset345,dataset1234,dataset1235,dataset1245,dataset1345,dataset2345,
dataset12345]

sensors_names = ["1","2","3","4","5","12","13","14","15","23","24","25","34","35","45","123","124",
"125","134","135","145","234","235","245","345","1234","1235","1245","1345","2345","12345"]
```

At the third cell, as described below, last 2nd and 3rd columns are accessed to extract activity and sub activity number and to make the 4-class and 10-class classifications. For all cases, k value is taken as five for KNN classifier. For every player, using leave-one-subject-out cross validation, one player's data is used as test data and the data of the other players is used as train data. The classifier is chosen as KNN or LDA. The train data is fitted to the classifiers and using the trained classifiers, the test data is fed to predict the activity and sub activity classes. The confusion matrix and accuracy score is stored for every trial. Then, the average accuracies are calculated for 4-class and 10-class classifications using LOSO CV. As a last step, all the average accuracies are used and relevant graphs are created which are shared at chapter 4.

```
Class_columns = [-3,-2]
Class_numbers = ["4","10"]
k_range = [*range(5,7,2)]
for t, dataset in enumerate(sensors):
    knn_scores_per_k = []
    for k in k_range:
        knn_scores_per_n = []
        for l, n in enumerate(Class_columns):
            knn_scores_per_p= []
            for i in range(1, 11):
                X_train = dataset[np.where(dataset[:,-4] != i)] # Dataset of actions related to players other than playeri
                y_train = X_train[:,n] # 10 sub actions classification
                X_train = X_train[:,:-4]

                X_test = dataset[np.where(dataset[:,-4] == i)] # Dataset of actions related to playeri
                y_test = X_test[:,n] # 10 sub actions classification
                X_test = X_test[:,:-4]

                # Search for an optimal value of k for kNN
                #classifier = KNeighborsClassifier(n_neighbors=k, p=2, metric='euclidean')
                classifier = LinearDiscriminantAnalysis(solver='svd', shrinkage=None,
                priors=None, n_components=None, store_covariance=False, tol=0.0001, covariance_estimator=None)

                #Train the classifier based on training dataset
                classifier.fit(X_train, y_train)

                #Predict the test set results
                y_pred = classifier.predict(X_test)
                #print(y_pred)

                #Evaluate Model
                cm = confusion_matrix(y_test, y_pred)
                print(cm)

                #print(f1_score(y_test, y_pred))
                a = accuracy_score(y_test, y_pred)
                #print(accuracy_score(y_test, y_pred))

                knn_scores_per_p.append(a)

            mean_of_knn_scores_per_p = np.mean(knn_scores_per_p)

            knn_scores_per_k.append(mean_of_knn_scores_per_p)

    Class4_list = knn_scores_per_k[0::2]
    Class10_list = knn_scores_per_k[1::2]
```

```
print(f'K_based_LOOCV_with_10_players_for_sensor{sensors_names[t]}_to_classify_4_classes_results_averages = {Class4_list}')

print(f'K_based_LOOCV_with_10_players_for_sensor{sensors_names[t]}_to_classify_10_classes_results_averages = {Class10_list}')
```

As the covariance matrix is not computed, for data with a high number of features, singular value decomposition is advised to be utilized as a solver, rather than eigenvalue decomposition or least squares solution. It predicts the class labels for the test data.

Linear Discriminant Analysis (LDA) has 3 types of solvers which are singular value decomposition ('svd'), least squares ('lsqr') and eigenvalue ('eigen'). As singular value decomposition was recommended for data with lots of features, in this study, LDA classifier in Python uses singular value decomposition as solver. Shrinkage option is selected as none. Prior probabilities were undefined, as default, they are deduced from the training data. Number of components was not adjusted as it only affects the transform method and we do not use the transform method. "Store_covariance" parameter was defined as false. The class covariance matrices are not explicitly computed. "Covariance_estimator" parameter is selected as none.

The LDA classifier is fitted to the training data. Then, using the fitted classifier, class labels for samples in the test dataset are predicted. After that, the predicted labels are compared with the test class labels and confusion matrix is constructed and accuracy is computed.

When KNN classifier is activated and the code is run, it decides the classes of the test data based on the highest number of votes coming from the selected number of nearest neighbors.

All dataset construction and analysis steps were conducted using Python version 3.9.7. on a computer with Intel(R) Core(TM) i7-4710HQ CPU at 2.5 GHz and 16.0 GB of RAM, running Microsoft Windows 10 Pro operating system.

CHAPTER 4

RESULTS & DISCUSSION

This study has been conducted with five female and five male, in total ten, volleyball players. Each player performed twelve digs, twelve blocks, ten serves and twelve spikes. The results are obtained for different perspectives. They are compared in terms of gender, classification method, validation method, number of sensors, sub sensor types, feature types, number of activities and sub activities.

4.1. Experimental Results and Discussion

Figure 11 shows the inertial data graph of fourth player during the first dig which is a middle dig. This figure is taken from XSENS MT Manager 4.6. As mentioned in the section 3.1.1, sensor “00B44D29” indicates the sensor positioned on the left leg. Tri-axial acceleration, angular velocity and magnetic field information is displayed at this window.

Figure 11: Inertial data graph of the player 4’s first dig recorded by left leg sensor (sensor 1)

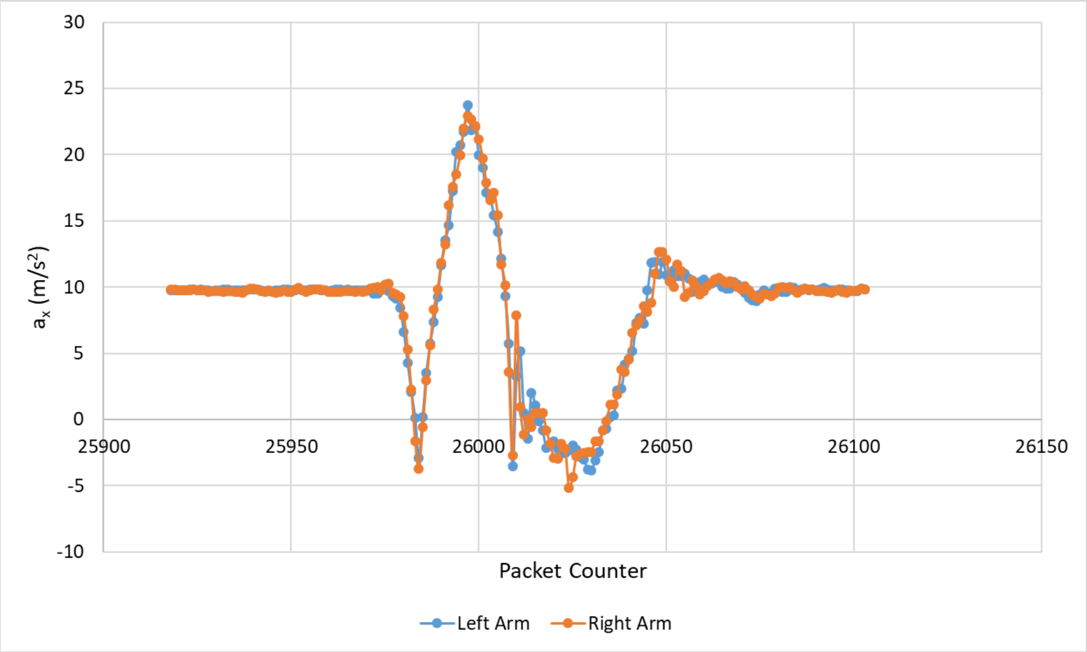


Figure 12: Inertial data graph of the player 4's first dig from left arm and right arm sensors

Figure 12 depicts the inertial data graph of fourth player during the first middle dig. As the activity is a middle dig, both left and right arms move nearly the same. The x axis indicates the axis on which the gravity acts along.

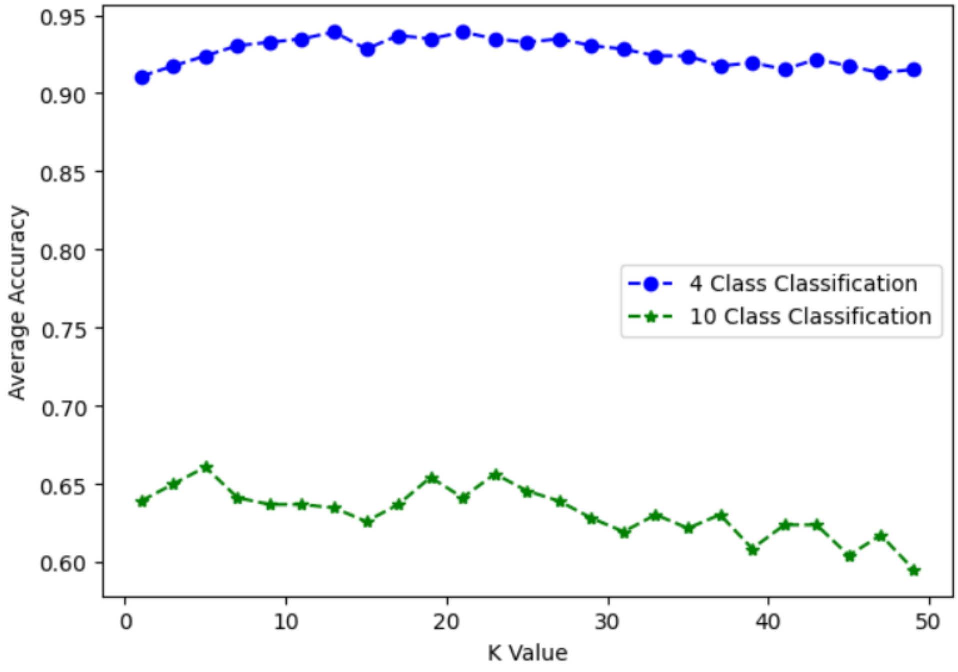
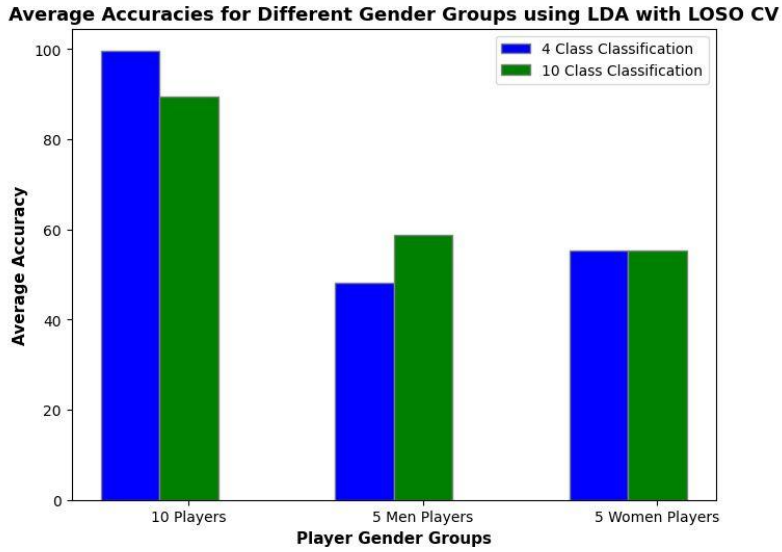


Figure 13: Change of average accuracies for different k values used in KNN method

As it is depicted in the Figure 13, average classification accuracies have been investigated through different k values. After examining odd k values from 1 to 49, it is seen that the highest accuracy to make 10-class classification comes for k equals to 5 value with 66.08% average accuracy. The highest average accuracy for 4-class classification is 93.91% for k equals to 13. As 10-class classification is much more difficult than the 4-class classification and also seeking the computational cost and time, after that step, for all KNN classifier application, k value is taken as 5.



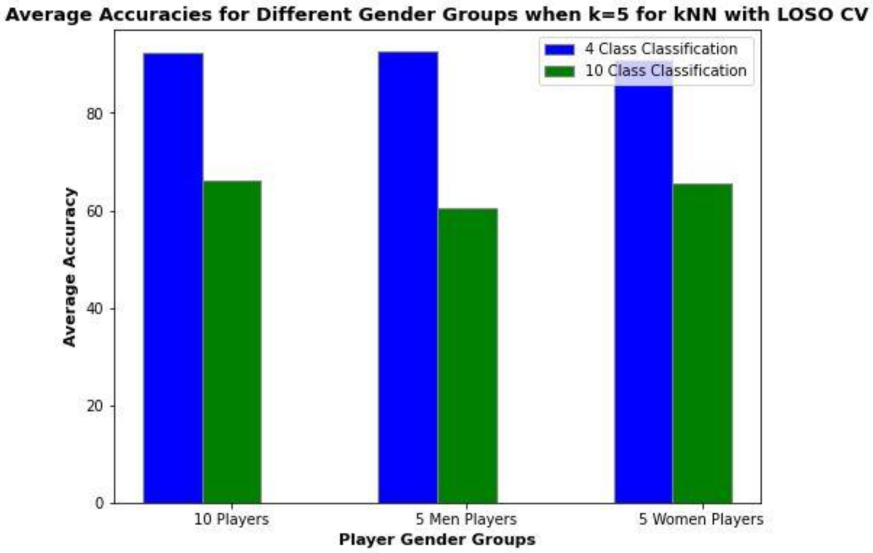
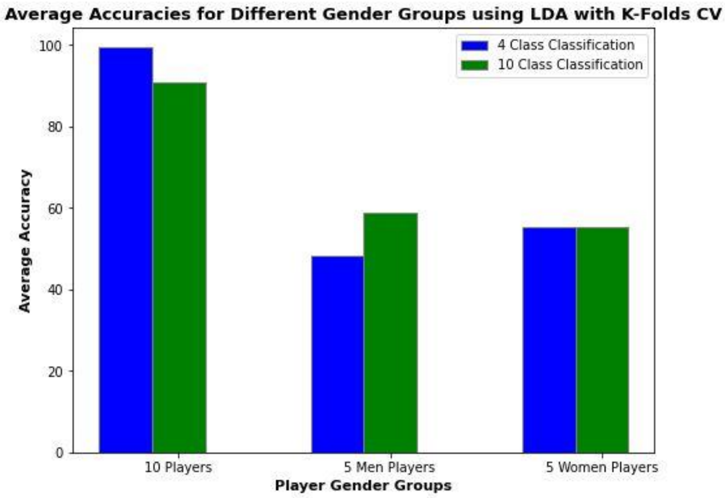


Figure 14: Average accuracies for different gender groups using LDA and using k=5 for KNN with LOSO CV

There were 5 male and 5 female volunteer players in the experiments. The average accuracies were evaluated for only 5 men players, for only 5 women players and for all 10 players. To do this, KNN and LDA classification algorithms were used. Cross validations were done with LOSO and K-fold. For both cross validation methods, average accuracies are higher with KNN for 5 men and 5 women player groups.

For 10 players group, the average accuracies reach to 99.34% for 4-class classification and 90.86% for 10-class classification using LDA and 10-folds cross validation methods. Similarly, for 10 players group, the average accuracies reach to 99.56% for 4-class classification and 89.56% for 10-class classification using LDA and LOSO cross validation methods.

For 10 players group, the average accuracies arrive at 92.39% for 4-class classification and 66.08% for 10-class classification using KNN and 10-folds cross validation methods. However, using LDA and 10-folds cross validation methods, for 10 players group, the average accuracies reach to much better values such as 99.34% for 4-class classification and 90.86% for 10-class classification.



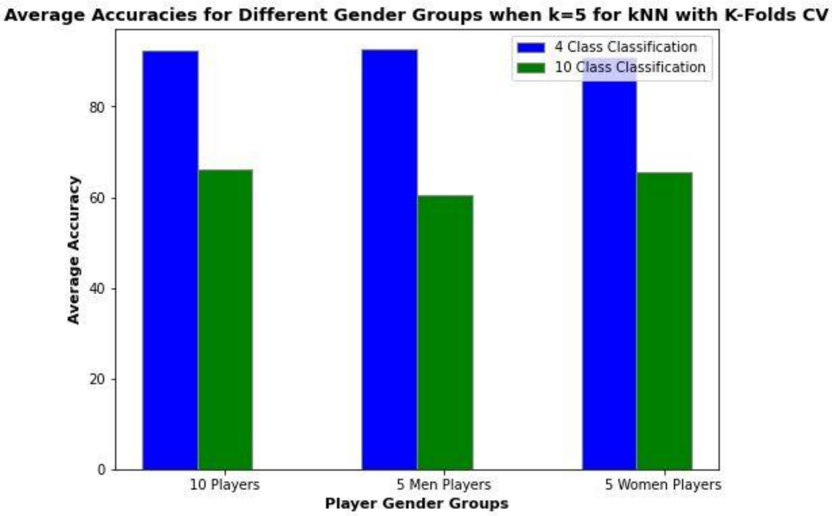


Figure 15: Average accuracies for different gender groups using LDA and using k=5 for KNN with K-fold CV

In Figure 14, gender-based player groups are compared using LDA and KNN with LOSO CV. In next figure, Figure 15, they are compared still using LDA and KNN but this time with K-fold cross validation. Looking at the both figure, it is obviously seen that, for LDA, when the number of player increases, from 5 men or 5 women to 10 players, both 4-class and 10-class average accuracies increase. However, in a counter intuitive way, 10-class classification average accuracy is larger than the 4-class one for LDA in both LOSO CV and K-fold CV. For KNN case, 4-class and 10-class classification average accuracies are almost the same; there is not a significant difference between gender-based groups.

Table 3: Confusion matrix of activities for 10 players using LDA with LOSO CV (99.34%)

		CLASSIFIED			
		DIG	BLOCK	SERVE	SPIKE
TRUE	DIG	120	0	0	0
	BLOCK	0	120	0	0
	SERVE	0	0	100	0
	SPIKE	0	3	0	117

Table 4: Confusion matrix of activities for 5 female players using LDA with LOSO CV (55.21%)

		CLASSIFIED			
		DIG	BLOCK	SERVE	SPIKE
TRUE	DIG	45	4	10	1
	BLOCK	4	33	20	3
	SERVE	2	4	31	13
	SPIKE	5	6	31	18

Table 5: Confusion matrix of activities for 5 male players using LDA with LOSO CV (48.26%)

		CLASSIFIED			
		DIG	BLOCK	SERVE	SPIKE
TRUE	DIG	46	6	7	1
	BLOCK	7	29	4	20
	SERVE	1	6	18	25
	SPIKE	12	18	12	18

Confusion matrices given in Table 3, 4 and 5 show the 4-class classification results using LDA with LOSO CV of ten players, five female players and five male players respectively. 4-class activities are classified truly with 99.34% average accuracy using LDA with LOSO CV when there are ten players. In Table 3, all digs, blocks and serves are classified correctly. However, 2.5% of the spikes are classified as blocks. When we look at Table 4, we find out that the average accuracy is 55.21%. Also, we notice that

40% of the blocks are classified as serves and 31 spikes out of 60 are misclassified as serves for five female players. Table 5 indicates that average accuracy for five male players using LDA is at 48.26%.

Table 6: Confusion matrix of activities for 10 players using KNN (k=5) with LOSO CV (92.39%)

		CLASSIFIED			
		DIG	BLOCK	SERVE	SPIKE
TRUE	DIG	120	0	0	0
	BLOCK	0	97	0	23
	SERVE	0	0	100	0
	SPIKE	0	12	0	108

Table 7: Confusion matrix of activities for 5 female players using KNN (k=5) with LOSO CV (90.86%)

		CLASSIFIED			
		DIG	BLOCK	SERVE	SPIKE
TRUE	DIG	60	0	0	0
	BLOCK	0	45	0	15
	SERVE	0	0	10	0
	SPIKE	0	6	0	54

Table 8: Confusion matrix of activities for 5 male players using KNN (k=5) with LOSO CV (92.60%)

		CLASSIFIED			
		DIG	BLOCK	SERVE	SPIKE
TRUE	DIG	60	0	0	0
	BLOCK	0	51	0	9
	SERVE	0	0	50	0
	SPIKE	0	8	0	52

When we come to Table 6, we see that 4-class classification accuracy using KNN is about 92.39% for ten players. All the digs and serves are truly classified. However, 23 blocks out of 120 are classified as spikes and %10 of the realized spikes are classified as blocks. When we pass from ten players to five female or five male player tables, we still notice that only blocks and spikes are confused. On the other hand, KNN classifies much more successfully five female or five male player scenarios. It shows that, with a limited number of data, KNN performs far better than the LDA.

Table 9: Confusion matrix of sub activities for 10 players using LDA with LOSO CV (90.86%)

		CLASSIFIED									
		MD	LD	RD	MB	LB	RB	SRV	LSP	MSP	RSP
TRUE	MD	39	0	1	0	0	0	0	0	0	0
	LD	0	36	4	0	0	0	0	0	0	0
	RD	0	5	35	0	0	0	0	0	0	0
	MB	0	0	0	40	0	0	0	0	0	0
	LB	0	0	0	0	40	0	0	0	0	0
	RB	0	0	0	2	0	36	0	0	0	2
	SRV	0	0	0	0	0	0	100	0	0	0
	LSP	0	0	0	0	0	2	0	27	8	3
	MSP	0	0	0	0	0	0	0	2	35	3
	RSP	0	0	0	0	1	0	0	4	5	30

Table 10: Confusion matrix of sub activities for 5 female players using LDA with LOSO CV (55.21%)

		CLASSIFIED									
		MD	LD	RD	MB	LB	RB	SRV	LSP	MSP	RSP
TRUE	MD	19	0	0	0	0	0	1	0	0	0
	LD	5	13	2	0	0	0	0	0	0	0
	RD	5	3	12	0	0	0	0	0	0	0
	MB	0	0	0	16	1	2	1	0	0	0
	LB	0	0	0	3	7	7	0	1	0	2
	RB	0	0	0	5	6	5	1	2	1	0
	SRV	3	1	1	9	0	0	29	1	1	5
	LSP	0	0	0	3	0	2	2	8	2	3
	MSP	1	0	0	1	0	0	0	4	8	6
	RSP	0	0	0	0	1	0	1	3	5	10

Table 11: Confusion matrix of sub activities for 5 male players using LDA with LOSO CV (58.69%)

		CLASSIFIED									
		MD	LD	RD	MB	LB	RB	SRV	LSP	MSP	RSP
TRUE	MD	14	6	0	0	0	0	0	0	0	0
	LD	6	11	3	0	0	0	0	0	0	0
	RD	5	8	6	0	0	0	1	0	0	0
	MB	0	0	1	15	0	0	0	0	2	2
	LB	1	2	0	2	11	0	0	0	2	2
	RB	0	1	2	2	0	15	0	0	0	0
	SRV	0	0	1	0	0	0	49	0	0	0
	LSP	1	0	0	0	4	2	0	6	6	1
	MSP	3	0	0	0	0	1	0	2	6	8
	RSP	2	0	0	3	1	2	0	4	6	2

Table 12: Confusion matrix of activities for 10 players using KNN (k=5) with LOSO CV (66.08%)

		CLASSIFIED									
		MD	LD	RD	MB	LB	RB	SRV	LSP	MSP	RSP
TRUE	MD	36	1	3	0	0	0	0	0	0	0
	LD	0	39	1	0	0	0	0	0	0	0
	RD	3	2	35	0	0	0	0	0	0	0
	MB	0	0	0	26	8	0	0	3	1	2
	LB	0	0	0	5	18	10	0	0	3	4
	RB	0	0	0	6	20	10	0	3	0	1
	SRV	0	0	0	0	0	0	100	0	0	0
	LSP	0	0	0	1	6	2	0	13	8	10
	MSP	0	0	0	0	3	0	0	17	9	11
	RSP	0	0	0	1	6	1	0	9	5	18

Table 13: Confusion matrix of activities for 5 female players using KNN (k=5) with LOSO CV (65.65%)

		CLASSIFIED									
		MD	LD	RD	MB	LB	RB	SRV	LSP	MSP	RSP
TRUE	MD	20	0	0	0	0	0	0	0	0	0
	LD	0	20	0	0	0	0	0	0	0	0
	RD	1	1	18	0	0	0	0	0	0	0
	MB	0	0	0	13	5	1	0	0	0	1
	LB	0	0	0	3	8	3	0	2	1	3
	RB	0	0	0	3	9	3	0	2	0	3
	SRV	0	0	0	0	0	0	50	0	0	0
	LSP	0	0	0	0	3	0	0	10	4	3
	MSP	0	0	0	0	2	0	0	10	4	4
	RSP	0	0	0	1	3	0	0	8	3	5

Table 14: Confusion matrix of activities for 5 male players using KNN (k=5) with LOSO CV (60.43%)

		CLASSIFIED									
		MD	LD	RD	MB	LB	RB	SRV	LSP	MSP	RSP
TRUE	MD	18	1	1	0	0	0	0	0	0	0
	LD	0	20	0	0	0	0	0	0	0	0
	RD	3	2	15	0	0	0	0	0	0	0
	MB	0	0	0	9	8	0	0	2	0	1
	LB	0	0	0	2	7	8	0	1	0	2
	RB	0	0	0	1	13	5	0	0	1	0
	SRV	0	0	0	0	0	0	50	0	0	0
	LSP	0	0	0	3	3	2	0	3	1	8
	MSP	0	0	0	1	0	0	0	9	4	6
	RSP	0	0	0	1	2	1	0	5	3	8

When we look at Table 9, we see that LDA classifies ten sub activities for ten players with 90.86% accuracy. Most confused sub activities are confused in their own activity classes. For instance, only one middle dig is classified as right dig out of 40 trials. %10 of the left digs are classified as right dig and 5 right digs out of 40 are categorized as left dig. Similarly, the spikes are classified with 76.67% accuracy. However, almost all of the misclassified spikes are within the spike class; only 2 left spikes are classified as right block and 1 right spike is classified as left block out of 120 spikes in total. On the contrary, when we pass to five female or five male tables, we see that accuracies drops to 55.21% and 58.69% for female and male players respectively similarly as it was in 4-class classification.

Table 12 sums up the confusion matrix of the 10-class classification accuracy for ten players using KNN. The average accuracy is 66.08%. The greatest part of the sub activities are confused within their own activities as in case of LDA 10-class classification. All of the 100 serves are correctly classified. There is a little confusion between classes and all of them are between blocks and spikes. When we go from ten players table to five female and five male tables, we see that the average accuracies change a little. Average accuracies of the five female players and five male players are 65.65% and 60.43% respectively. This demonstrates that, even though the number of sample decreases, KNN still performs almost at the same level as in case of ten players. In case of LDA, it is not possible; the accuracies drops dramatically from 90.86% to 55.21% and 58.69% respectively for five female and five male players.

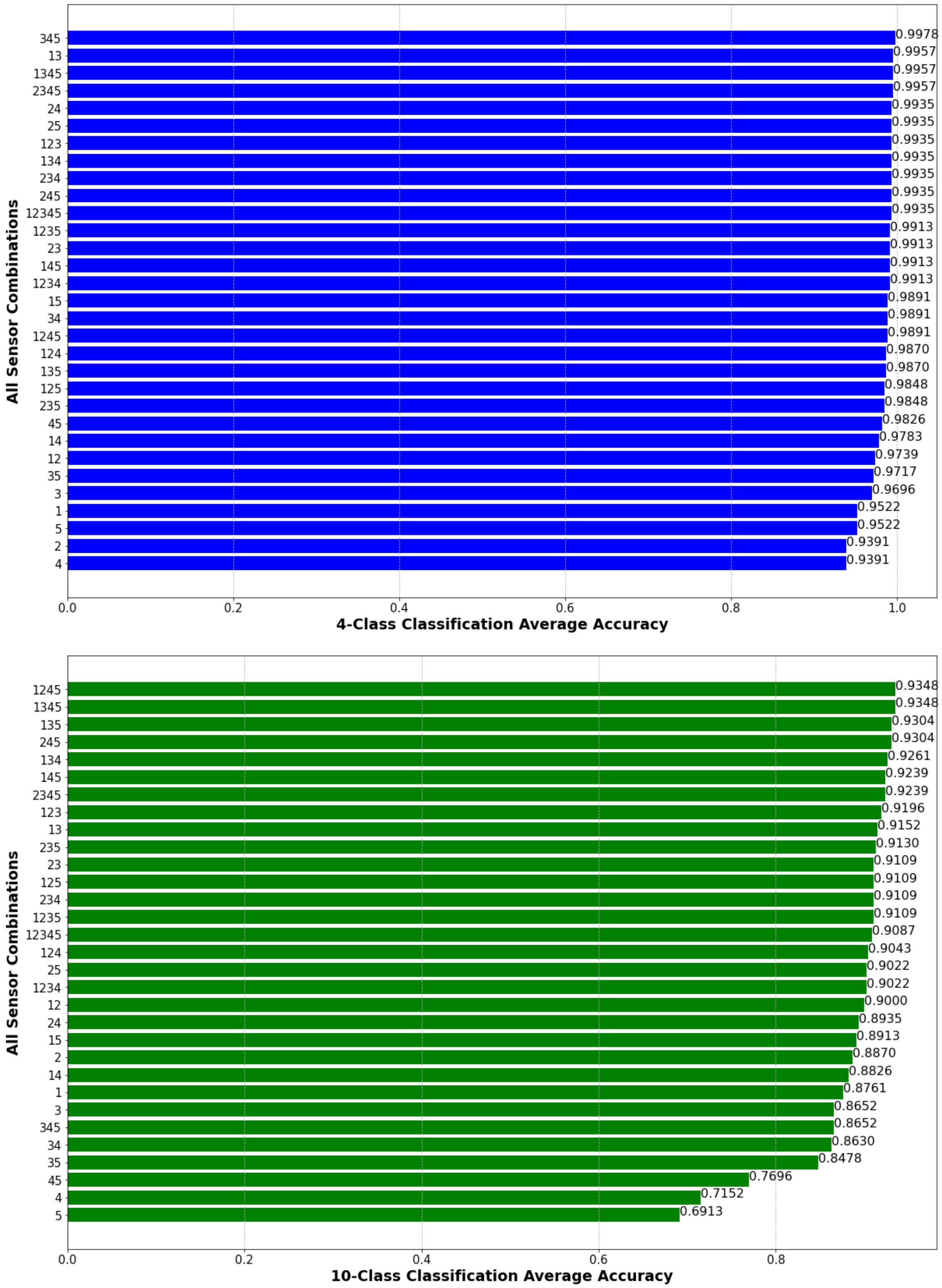


Figure 16: Average accuracies for all sensor combinations using LDA with LOSO CV

Usage of MTw sensors does not make the players feel comfortable during the game. Replacing them with smaller ones will certainly be a better choice.

For 4-class classification, instead of using 5 sensors, usage of sensors 345 (waist, left arm and right arm sensors) brings a slightly better average accuracy of 99.78% compared to the usage of all sensors which reaches an accuracy of 99.34%. Using five, four, three or two sensors at the same time brings average accuracies between 99.78-97.17%. In terms of economics, if one can only afford one sensor unit and focuses on the 4-class classification, placing it on the waist is the best option among one-sensor combinations and it brings a 96.95% average accuracy. The waist location is the best choice because it is the most stable and suitable position for a reference frame. However, there is not an extremely big difference between other options as well. Placing only just one sensor on the body, to the left leg, right arm, right leg or left arm results in 95.21%, 95.21%, 93.91% and 93.91% average accuracies respectively. The worst average accuracy for 4-class classification comes with the combination for which there is only one sensor at the left arm and it brings an average accuracy of 93.91%. Contrary, the best average accuracy for the 4-class classification comes from the 345 combination, which signifies the sensors on the waist, left arm and right arm. The result is 99.78%.

For 10-class classification, among one-sensor combinations, the best three locations are right leg, left leg and waist with accuracies 88.69%, 87.60% and 86.52%. In two-sensors situation, using sensors 13 (left leg and waist sensors) points an accuracy of 91.52%. When the sensor at the left leg is also activated, the accuracy advances from 91.52 to 93.04%. The best average accuracy for the 10-class classification comes from the 1245 combination, which means all the sensors except the one on the waist. This 4-sensor combination has an average accuracy of 93.47%.

Best combination among thirty-one sensors is sensors 12345 combination using KNN with LOSO CV to classify 10 sub activities. However, it rests at 66.08% accuracy. The best combination is sensors 1245 combination using LDA with LOSO CV as depicted in Figure 16 and the accuracy is 93.47% in this case. To classify the 10 sub activities, LDA brings 41% better accuracy compared to the KNN method.

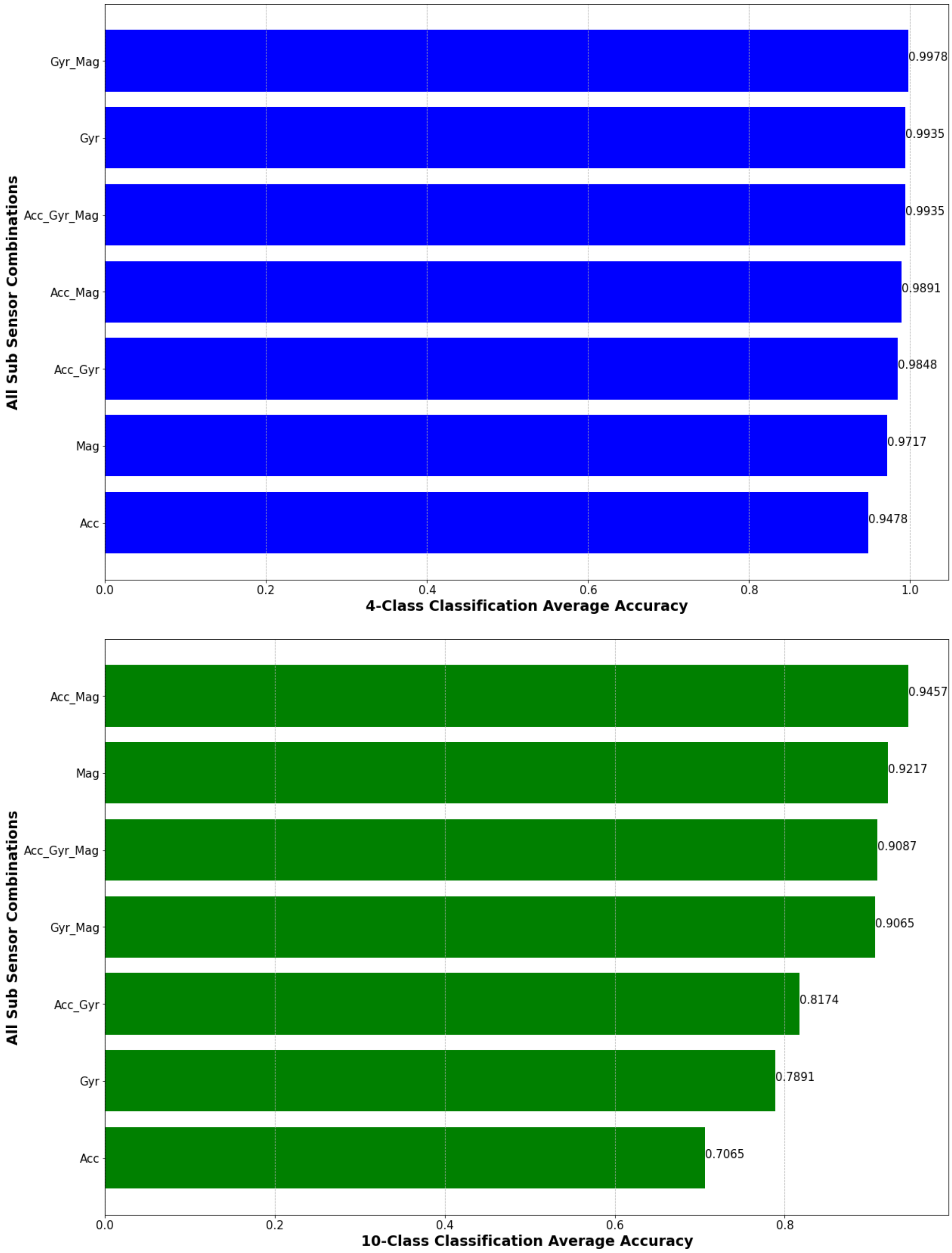


Figure 17: Average accuracies for all sub sensor combinations using LDA with LOSO CV

In Figure 17, sub sensor combinations are examined for both 4-class and 10-class classifications. Investigating different sub sensor combinations show that using only magnetometer information among one-sub sensor combinations brings an average accuracy result of 97.17% and 92.17% for 4-class and 10-class classifications respectively. As all the data has been collected at the same sports hall and mostly at the same zones of the volleyball court, these results coming only from magnetometers are highly likely biased. If training data was collected in another place of the city/country/world or whether the activities were realized along any other direction (i.e. from north to south) and the test data was collected in a place different than the training data, most probably, the magnetometer will not bring that much good classification results.

It is clearly seen that, there is not an important difference between sub sensor combinations to classify 4 activities. The highest average accuracies come from "Gyr_Mag" combination, the lowest average accuracies come from "Acc" combination and they are respectively 99.78% and 94.78%. On the other hand, for 10-class classification, "Acc_Mag" combination has the highest, "Acc" combination has the lowest average accuracies with respectively 94.56% and 70.65%.

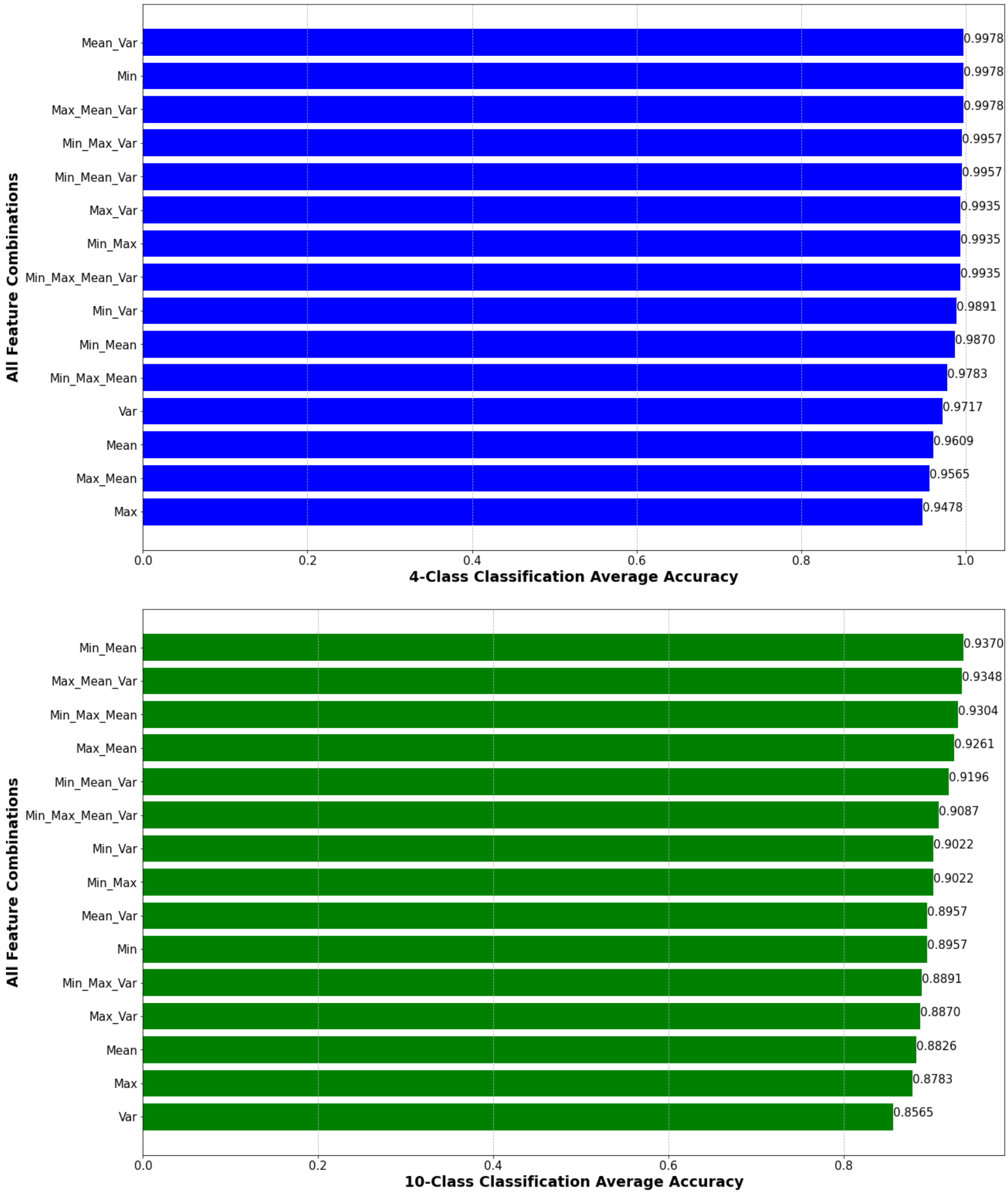


Figure 18: Average accuracies for all feature combinations using LDA with LOSO CV

In Figure 18, evaluation of feature combinations demonstrates that, for 4-class classification, the average accuracies for different feature combinations do not change very much. They varies from 99.78% to 94.78%. However, it is easily seen that, only taking minimums of the input data helps the 4-class classification to reach an average accuracy of 99.78%.

In 10-class classification also, taking only minimums of the input data brings higher average accuracy than calculating only mean, maximum or variance. The average accuracy value is 89.56% for 10-class classification. The best configuration in the 10-class classification is realized by “Min_Mean” combination which results in 93.69% average accuracy.

When only one feature is used, it is seen that taking only minimums does a great job. However, if the sensors are rotated and/or fixed at any other configuration, instead of taking minimums, maybe, taking maximums might be the best option. For this reason, taking only minimums might not be the best feature in all cases. This should be seriously taken into consideration. It is the same thing for taking maximums, too.

CHAPTER 5

CONCLUSION

This thesis has presented a study of how volleyball actions can be classified using KNN and LDA methods with the maximum average accuracies possible. Besides the usage of two different classification methods, two different cross validation methods were used. During this study, normally, activities of 17 different volunteers were recorded. However, only data of 12 volunteers was used. There were some problems like false sampling rate adjustment, missing file problem, not realizing the orientation reset step etc.

In total, 5 sensors were used and they were placed respectively from sensor 1 to sensor 5 in the following order: Left leg, right leg, waist, left arm and right arm. To classify four activities, instead of buying three sensors and getting an accuracy of 99.78%, if only one sensor will be placed on the player, among the five locations, the right place is the waist for 4-class classification. Positioning it at the waist brings an accuracy of 96.95%. Similarly, to classify ten sub activities, one can choose to use only one sensor at the right leg, which is the best place, and have an average accuracy of 88.69% rather than using four sensors at the 1245 combination and having an accuracy of 93.47%. However, if two-sensor combination is also affordable by someone, then locating one at the left leg and one at the waist produces the best average accuracies respectively 99.56% for 4-class and 91.52% for 10-class.

In overall, the classification performance of LDA is better than the KNN performance. However, when sub sensor combinations are examined, there is not a big difference between them for 4-class classification but having only "Mag" brings an average accuracy of 97.17% and 92.17% respectively for 4-class and 10-class classifications. So, instead of having an IMU with a tri-axial accelerometer, a tri-axial gyroscope and a tri-axial magnetometer; only the magnetometer sub sensor can bring the average accuracies mentioned at the previous sentence.

After analyzing the effect of all feature combinations to the 4-class and 10-class classifications, it is seen that taking only the minimum values of the raw data produces a 99.78% average accuracy 'which is the highest for 4-class classification with "Max_Mean_Var" and "Mean_Var" combinations. Similarly, taking minimum offers a

89.56% average accuracy for 10-class classification where the highest average accuracy is 93.69% using "Min_Mean" combination. So, according to results obtained, using only "Min" feature is highly acceptable for 4-class and 10-class classifications. In addition to this, sensors' placements are also extremely important at this situation. If they are rotated and/or fixed at any other configuration, taking minimums might not be the best option as the sensors' axes and orientations change.

This thesis adds novelty to detection and classification of volleyball activities and sub activities by examining various sensor, sub sensor and feature combinations. However, there are some points to be corrected and developed in future studies. In this study, all the players have right hand as the dominant hand. Players who have left hand as dominant hand is absolutely required in the dataset to be able to make more accurate and strong comments about sensors placements. All the experiments have been realized at the Izmir Institute of Technology Sports Hall and mostly at the same zones of the volleyball court. To avoid biased results coming from magnetometers, it is very important to collect data from other sports hall which are located at other regions of the world and perform the experiments at different zones of the volleyball court as much as possible. Adding some new activities such as overhead and left/right hand passes should be considered too. There was only one type of serve which was overhand serve. Adding new serves such as underhand or jumping serves might be also taken into consideration.

REFERENCES

[1] Camomilla, Valentina, Elena Bergamini, Silvia Fantozzi, and Giuseppe Vannozzi. 2018. “Trends Supporting the In-Field Use of Wearable Inertial Sensors for Sport Performance Evaluation: A Systematic Review.” *Sensors (Switzerland)*. MDPI AG. doi:10.3390/s18030873.

[2] Çıklaçandır, Samet. “Biyomedikal Mühendisliğinde İnsan Hareketleri Yakalama Sistemleri.” Essay. In *Mühendislik Alanında Araştırma Makaleleri*, edited by Şenay Mihçin, 343–64. Ankara, TURKEY: Gece Kitaplığı, 2019.

[3] Kautz, Thomas, Benjamin H. Groh, Julius Hannink, Ulf Jensen, Holger Strubberg, and Bjoern M. Eskofier. 2017. “Activity Recognition in Beach Volleyball Using a Deep Convolutional Neural Network.” *Data Mining and Knowledge Discovery* 31 (6). Springer Science and Business Media LLC: 1678–1705. doi:10.1007/s10618-017-0495-0

[4] Haider, Fasih, Fahim A. Salim, Dees B.W. Postma, Robby van Delden, Dennis Reidsma, Bert Jan van Beijnum, and Saturnino Luz. 2020. “A Super-Bagging Method for Volleyball Action Recognition Using Wearable Sensors.” *Multimodal Technologies and Interaction* 4 (2). MDPI AG: 1–14. doi:10.3390/mti4020033.

[5] Roggen, Daniel, Luis Ponce Cuspinera, Guilherme Pombo, Falah Ali, and Long Van Nguyen-Dinh. 2015. “Limited-Memory Warping LCSS for Real-Time Low-Power Pattern Recognition in Wireless Nodes.” In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8965:151–67. Springer Verlag. doi:10.1007/978-3-319-15582-1_10.

[6] Rawashdeh, Samir A., Derek A. Rafeldt, and Timothy L. Uhl. 2016. “Wearable IMU for Shoulder Injury Prevention in Overhead Sports.” *Sensors (Switzerland)* 16 (11). MDPI AG. doi:10.3390/s16111847.

[7] Rawashdeh, Samir A., Derek A. Rafeldt, Timothy L. Uhl, and James E. Lumpp. 2015. “Wearable Motion Capture Unit for Shoulder Injury Prevention.” In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2015*. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/BSN.2015.7299417.

[8] Wang, Yufan, Yuliang Zhao, Rosa H.M. Chan, and Wen J. Li. 2018. “Volleyball Skill Assessment Using a Single Wearable Micro Inertial Measurement Unit at Wrist.” *IEEE Access* 6 (January). Institute of Electrical and Electronics Engineers Inc.: 13758–65. doi:10.1109/ACCESS.2018.2792220.

[9] Jarning, Jon M., Kam Ming Mok, Bjørge H. Hansen, and Roald Bahr. 2015. “Application of a Tri-Axial Accelerometer to Estimate Jump Frequency in Volleyball.” *Sports Biomechanics* 14 (1). Routledge: 95–105. doi:10.1080/14763141.2015.1027950.

[10] Charlton, Paula C., Claire Kenneally-Dabrowski, Jeremy Sheppard, and Wayne Spratford. 2017. “A Simple Method for Quantifying Jump Loads in Volleyball Athletes.” *Journal of Science and Medicine in Sport* 20 (3). Elsevier Ltd: 241–45. doi:10.1016/j.jsams.2016.07.007.

[11] Skazalski, C., R. Whiteley, C. Hansen, and R. Bahr. 2018. “A Valid and Reliable Method to Measure Jump-Specific Training and Competition Load in Elite Volleyball Players.” *Scandinavian Journal of Medicine and Science in Sports* 28 (5). Blackwell Munksgaard: 1578–85. doi:10.1111/sms.13052.

[12] Damji, Faraz, Kerry MacDonald, Michael A. Hunt, Jack Taunton, and Alex Scott. 2021. “Using the VERT Wearable Device to Monitor Jumping Loads in Elite Volleyball Athletes.” *PLoS ONE* 16 (1 January). Public Library of Science. doi:10.1371/journal.pone.0245299.

[13] De Sousa, Michael Ferreira, and Fernando Trinta. 2020. “VolleyJump 2.0: A Computational Approach to Analysis and Classification of Volleyball Jumps Based on Multiple Inertial Devices.” In *ACM International Conference*

Proceeding Series, 121–24. Association for Computing Machinery. doi:10.1145/3428658.3431759.

[14] Leeuw, Arie Willem de, Stephan van der Zwaard, Rick van Baar, and Arno Knobbe. 2022. "Personalized Machine Learning Approach to Injury Monitoring in Elite Volleyball Players." *European Journal of Sport Science* 22 (4). Taylor and Francis Ltd.: 511–20. doi:10.1080/17461391.2021.1887369.

[15] Di Tore, Pio Alfredo, and Gaetano Raiola. 2019. "Powerglove: Genesis of a Wearable Technology Aimed at Studying Volleyball Service." *Journal of Human Sport and Exercise* 14 (Proc1). University of Alicante: S77–83. doi:10.14198/jhse.2019.14.Proc1.09.

[16] Cuspinera, L. Ponce, Sakura Uetsuji, F. J.Ordonez Morales, and Daniel Roggen. 2016. "Beach Volleyball Serve Type Recognition." In *International Symposium on Wearable Computers, Digest of Papers*, 12-16-September-2016:44–45. IEEE Computer Society. doi:10.1145/2971763.2971781.

[17] Vales-Alonso, Javier, David Chaves-Dieguez, Pablo Lopez-Matencio, Juan J. Alcaraz, Francisco J. Parrado-Garcia, and F. Javier Gonzalez-Castano. 2015. "SAETA: A Smart Coaching Assistant for Professional Volleyball Training." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45 (8). Institute of Electrical and Electronics Engineers Inc.: 1138–50. doi:10.1109/TSMC.2015.2391258.

[18] Gielen, Jasper, Elise Mehuys, Kris Eyckmans, and Jean Marie Aerts. 2020. "Heart Rate and Activity Measured during Volleyball Competition Using Wearable Technology." In *IcSPORTS 2020 - Proceedings of the 8th International Conference on Sport Sciences Research and Technology Support*, 212–16. SciTePress. doi:10.5220/0010047802120216.

[19] Holatka, Ann Kathrin, Hirohiko Suwa, and Keiichi Yasumoto. 2019. "Volleyball Setting Technique Assessment Using a Single Point Sensor." In *2019 IEEE International Conference on Pervasive Computing and*

Communications Workshops, PerCom Workshops 2019, 567–72. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/PERCOMW.2019. 8730811.

[20] Alpaydin, Ethem. 2014. *Introduction to Machine Learning, Third Edition*. *The MIT Press*.

[21] YESHURUN ALEMAYEHU, Ankit Soral, Jerry Miller, and Elahe_Saed. "Supervised vs Unsupervised vs Reinforcement Learning: Intellipaat." *Intellipaat Blog*, June 24, 2022. https://intellipaat.com/blog/supervised-learning-vs-unsupervised-learning-vs-reinforcement-learning/.

[22] Salian, Isha. "SuperVize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?" *NVIDIA Blog*, August 20, 2019. https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning.

[23] Arora, Surbhi. "Supervised vs Unsupervised vs Reinforcement." *AITUDE,* January 29, 2020. https://www.aitude.com/supervised-vs-unsupervised-vs-reinforcement/.

[24] Attal, Ferhat, Samer Mohammed, Mariam Dedabrishvili, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. 2015. "Physical Human Activity Recognition Using Wearable Sensors." *Sensors (Switzerland)*. MDPI AG. doi:10.3390/s151229858.

[25] Ho, George. "Linear Discriminant Analysis for Starters." *George Ho*, December 30, 2017. https://www.georgeho.org/lda/.

[26] Pedregosa FABIANPEDREGOSA, Fabian, Vincent Michel, Olivier Grisel OLIVIERGRISEL, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Jake Vanderplas, et al. 2011. "Scikit-Learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot." *Journal of Machine Learning Research* 12: 2825–30. http://scikit-learn.sourceforge.net.

[27] He, Ziyang, Xiaoqing Zhang, Yangjie Cao, Zhi Liu, Bo Zhang, and Xiaoyan Wang. 2018. “Litenet: Lightweight Neural Network for Detecting Arrhythmias at Resource-Constrained Mobile Devices.” *Sensors (Switzerland)* 18 (4). MDPI AG. doi:10.3390/s18041229.

[28] Janidarmian, Majid, Atena Roshan Fekr, Katarzyna Radecka, and Zeljko Zilic. 2017. “A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition.” *Sensors (Switzerland)* 17 (3). MDPI AG. doi:10.3390/s17030529.

[29] Altun, Kerem. “Intelligent sensing for robot mapping and simultaneous human localization and activity recognition.” (2011).

[30] El-Sheimy, Naser, and Ahmed Youssef. 2020. “Inertial Sensors Technologies for Navigation Applications: State of the Art and Future Trends.” *Satellite Navigation*. Springer. doi:10.1186/s43020-019-0001-5.

[31] “MTW Hardware, MT Manager, Awinda Protocol - Xsens.” *XSENS*. Xsens Technologies B. V., May 3, 2018. https://www.xsens.com/hubfs/Downloads/Manuals/MTw_Awinda_User_Manual.pdf.