THE CHINESE UNIVERSITY OF HONG KONG

FINAL YEAR PROJECT REPORT (TERM 2)

Smart Boxing Gloves

QXU1801

Faculty of Engineering

Department of Computer Science and Engineering

Supervisor: Xu Qiang

Author: Sung Bo Kim (1155074863)

April 17, 2019

Abstract

Faculty of Engineering

Department of Computer Science

BSc in Computer Science and Engineering

Smart Boxing Gloves

by Sung Bo Kim

Wearable device refers to all kind of electronic devices that can compute some programs while being attached to human body parts. These wearable devices are small and light so that users will be able to freely and conveniently use them to assist their daily life. With the help of Information Communication Technology (ICT), wearable devices are used in various fields such as fitness, health care, military, and medical purposes. Also, as the income level of people increases, wearable devices that can assist people in improving exercise performances are developed to meet the needs of people who want to enhance their leisure life. With the increasing interest, our project is aimed to develop a ‘Smart Boxing Gloves’ that can assist boxers’ training performance. The wearable device ‘Smart Boxing Gloves’ will be able to enhance boxers’ training by providing statistical information about their training and keeping track of their progress.

Acknowledgment

I would like to appreciate Professor Xu Qiang for giving me an excellent opportunity to explore the field of wearable devices, guiding with great instructions, and encouraging us with kindness. Throughout the project 'Smart Boxing Gloves,' I had much chance to have research and gained much knowledge about sensors and their mechanisms which I was not familiar with. It is also thankful that the project set the stage for me to become a better mobile application developer. I would also like to thank my colleague Jonathan Leonanda for putting effort to finish the project successfully.

Table of Contents

Table of Figures

Chapter 1. Introduction

1.1 Project Motivation

The human body in nature is capable of performing intensive physical exercise. However, it is hard for each individual to aware of their capability and examines the effectiveness of their performance. Fortunately, it has become easier for athletes to examine their performances as wearable technology evolves. In this project, we have developed smart boxing gloves which work along with a mobile application to record and show boxers' performances statistically. With the help of statistically measured data, we aim to assist users by letting them assure about their performances by referencing performance recorded on the mobile application.

1.2 Background

1.2.1 Wearable Technology

In this section, we will be discussing different types of wearable technologies that are developed to enhance athletes' performances in order to analyze the current wearable technology market.

1.2.1.1 The Kinexon Tracking System

The Kinexon Tracking System is a wearable device used to track basketball players' movement by measuring their jump capability and speed. In order to activate the system, all the players are required to wear uniforms with a small sensor device attached. The main purpose of the system is to record useful information that can be used for league broadcasting by finding out Top Players of the game in terms of jump and speed by computing statistics gained throughout the game.

1.2.1.2 Xampion

Xampion is a wearable device that is used to record soccer players' performances by measuring the number of ball contacts, speed of strikes and etcetera. These data are collected by a sensor embedded inside soccer players' shoes, and after soccer training, coaches will be able to check each players' performance instantly.

1.2.1.3 StrikeTec

StrikeTec is a wearable device that is used to measure boxers' performance. StrikeTec mainly aims to measure speed, power, endurance, punch accuracy and punch type. In order to keep track of these data, users are required to embed a small device on their wrist. Throughout training sessions, users will be able to check statistics about their punches including speed, power, punch type, and average speed and power instantly. After training sessions, users will be able to check their progress in the form of a graph. These histories will contain detailed information about all of the users' punches thrown during training sessions and it will be used to analyze the strengths/ weakness of users.

1.2.2 Important Aspects of Boxing

Boxing is a type of martial arts where two athletes fight each other in a bounded ring by only using their upper bodies with gloves on. In order to have a detailed understanding of the project, basic knowledge about boxing is also required. In this section, we will be introducing important aspects of boxing that are required to be considered.

1.2.2.1 Quickness

Being quicker than your opponent is one of the crucial traits one should possess to win boxing games because quicker movements will allow boxers to avoid opponents' punches while swiftly attacking the opponent. In order to develop boxers' quickness, we have implemented a feature that calculates the speed of boxers' punches.

1.2.2.2 Accuracy

There are four main types of punches used in boxing: Straight, Jab, Hook and Uppercut. Although the variation is small, different types of punches are thrown in a series to produce maximum impact. Therefore, good boxers should be able to throw different types of punches accurately to maximize the output. In order to assist boxers to throw accurate punches, we have implemented a punch classifying feature so that the users will be able to see if they are throwing accurate punches or not.

1.2.2.3 Power

Throwing a powerful punch is advantageous for boxers to win games. This is because it is important to make opponents defeat before using up all the energy and the only way to make opponents defeat is by making the opponents to consume all their energy by getting beaten by powerful punches. Therefore, we also keep track of boxers' punch power and let users monitor their progress in terms of punch power.

Chapter 2. Literature Review

2.1 Limitation of StrikeTec

StrikeTec is one of the most known and popular wearable devices for boxing training. The official webpage of StrikeTec says that in order to classify punch types, StrikeTec implemented an algorithm that compares a user's fist movement with an ideal form of movement that is used for a reference. Although the algorithm is undisclosed to users, it is clear that the algorithm itself has a limitation: The accuracy of the punch classification system is unknown. Since different types of punches are classified manually by comparing the movement with the reference movement, it will not be able to classify punches accurately since all boxers have different punching style and physical structure.

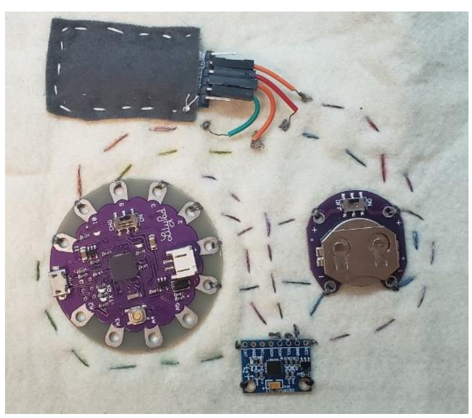

Figure 3.1 Hardware Architecture

Chapter 3. Hardware Development

3.1 Hardware Architecture

Figure 3.1 describes the overall architecture of the hardware. The hardware device of Smart Boxing Gloves mainly consists of two sensors with a microcontroller. In this project, MPU6050 is used for accelerometer, HC-05 for Bluetooth module and Arduino Lilypad USB for the microcontroller. Detailed usage of these sensors is described in the following section.

3.1.1 Microcontroller (Arduino Lilypad USB)

Arduino Lilypad USB is a microcontroller that is specialized for wearable devices because it can be easily sewn into fabrics. Another advantage of Arduino Lilypad USB is that the weight of the board is negligible, therefore embedding the Lilypad board into a boxing glove would minimize the inconvenience to boxers when throwing punches. The operation voltage of the Lilypad USB is 3.3V, and there are enough analog pins to implement our project.

3.1.2 Accelerometer (MPU6050)

In order to determine the speed and force of boxers' punches, it is required to measure the acceleration of punches. MPU6050, which contains both accelerometer and gyroscope, measures acceleration of an object and rate of rotation along 3-axis.

3.1.3 Bluetooth Module (HC-05)

Bluetooth module is one of the essential parts of wearable devices because it allows wireless communication between hardware and software. HC-05, which uses Bluetooth V2.0 with enhanced data rate 3Mbps, is chosen to be used because it is capable of sending data needed for the project.

3.2 Hardware Flow

Figure 3.2 describes the overall flow of the hardware of Smart Boxing Gloves. Both accelerometer and Bluetooth module are connected to the microcontroller, and the data gathered from the accelerometer is passed to the microcontroller, and then Bluetooth module will send the data from the hardware to the Android device through Bluetooth connection. In this section, a detailed process of data gathering, and data processing will be described.

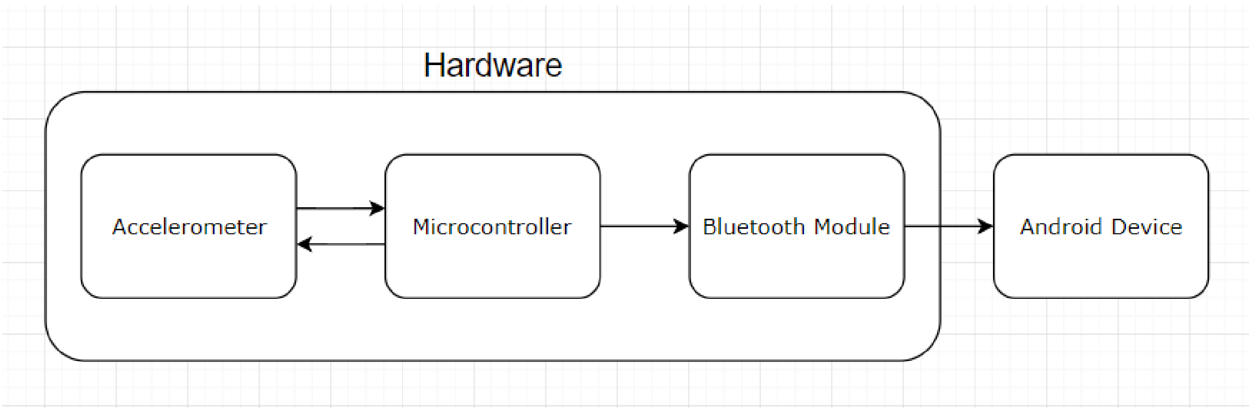

Figure 3.2 Hardware Flow

3.2.1 Data Gathering

There are two types of information needed for our system: 3-axis acceleration and time taken to for each punch. First of all, the accelerometer will be in charge of gathering 3-axis acceleration and constantly sends the data to the microcontroller.

3.2.2 Data Processing

The data the microcontroller receives from the accelerometer is in a raw form. Therefore, it is required for the microcontroller to process the data to get the g-force exercised by the earth. Every time the microcontroller receives data from the accelerometer, it will keep track of peak acceleration in order to recognize a punch. While keeping track of the peak acceleration, the microcontroller will measure the time taken to throw a punch at the same time.

3.2.3 Data Sending

The processed data will be sent to the Android device through the Bluetooth module. Since we need 3-axis acceleration and time taken to throw a punch, the Bluetooth module will send four float number (three acceleration values and one time stamp) with two decimal places concatenated with a comma. The data is 20bytes in total, and these 20 bytes of data will be sent to the android device every time a user throws a punch. The Bluetooth communication will be explained in detail in Johnathan Leonanda’s report.

Chapter 4. App Development

4.1 Software Architecture

Figure 4.1 describes the overall flow of the mobile application of Smart Boxing Gloves. Training mode, target mode, history mode will be the three main functionalities provided by the application to enhance boxers' training performance.

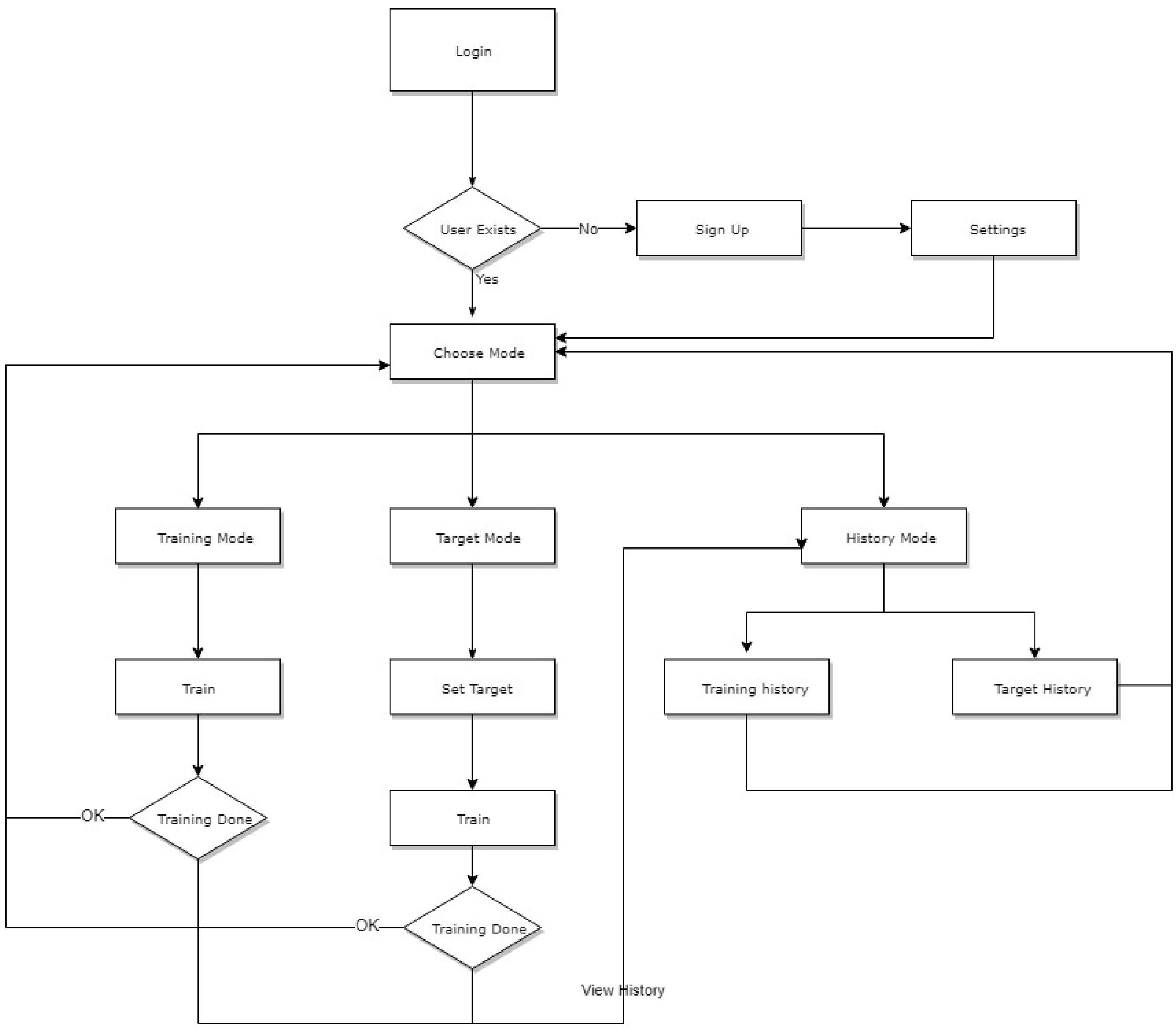

Figure 4.1 Software Architecture

4.1.1 Training Mode

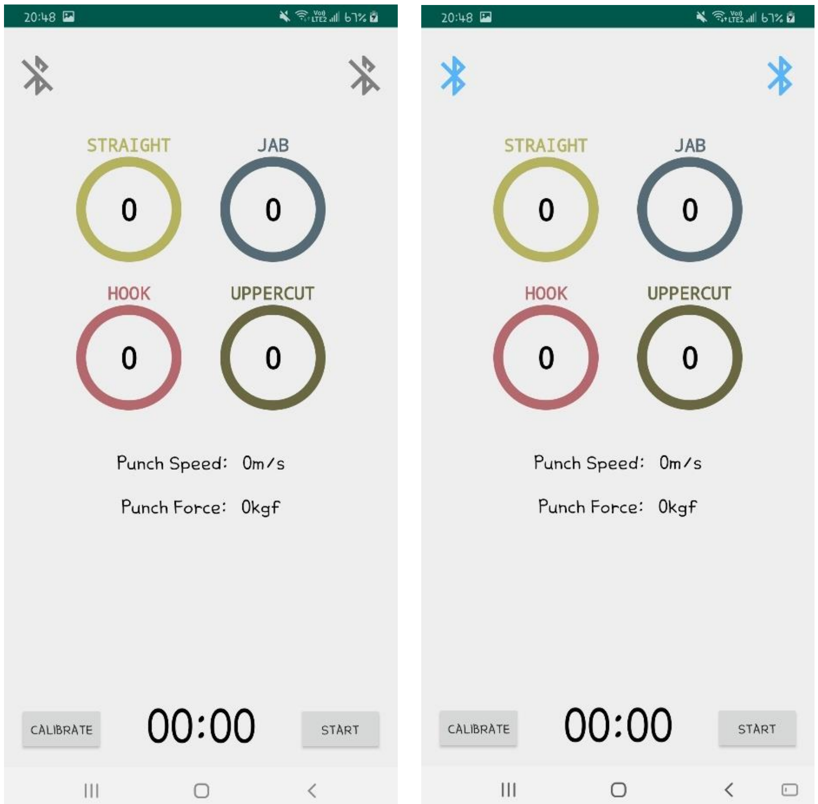

Figure 4.1.1.1 UI of Training Mode

Figure 4.1.1.1 illustrates the overall UI of the Training Mode. Training Mode is designed to keep track of boxers' performance in real time. First of all, users are required to pair each of the gloves manually by using the Bluetooth button on the top. The left picture of Figure 4.1.1.1 illustrates the condition where no Bluetooth is connected. The right picture of Figure 4.1.1.1 shows that the change of Bluetooth icons when both sides of the gloves are connected. When the connection is lost during the training mode, the icons will be changed back to disabled Bluetooth connection icon so that users can be notified with the Bluetooth connection status. To start the activity, users need to calibrate their gloves in position for more accurate calculation. Then they are required to start a timer and can throw as many punches as they would like to. In this mode, there are mainly three types of data that are provided in real time: Type, speed, and force of the punch. When a punch is thrown, the application will automatically classify the type of punch and will notify the user by blinking the corresponding field as illustrated in Figure 4.1.1.2.

Figure 4.1.1.2 Change of UI when Straight Punch is Thrown

The blinking feature is designed to let users interpret with the data more easily because it is very hard for users to interpret the data when the data is presented in the form of text especially when the information changes very rapidly. When the training is done, the user is required to stop the timer and will be automatically directed to a summary page which is illustrated in Figure 4.1.1.3.

Figure 4.1.1.3 Summary Page of Training Mode

The summary page will contain information including duration of the session, the number of each punches thrown during the session, average speed and average force of the boxers' punches. In this summary page, the user can either choose to see history or go back to the main page of the application.

4.1.2 Target Mode

Figure 4.1.2.1 UI of Target Mode

Figure 4.1.2.1 illustrates the overall UI of Target Mode. Target Mode is designed to let users set their goal and helps to reach their goal within a certain period of time. Before entering the target mode, users are required to choose the designated duration of the training session and number of punches they would like to throw within the period of time as illustrated in Figure 4.1.2.2.

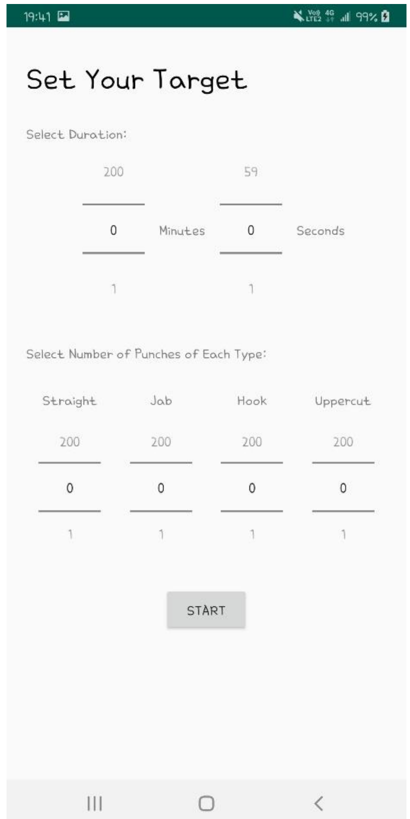

Figure 4.1.2.2 Target Setting Page

After setting the goal, users will enter target mode where they have to calibrate the gloves first and press the start button of a stopwatch to start the target mode. When a user presses the start button, the stopwatch will start counting down the timer. In target mode, instead of blinking different field for punch classification, we have implemented a progress bar to show how much a user has achieved their goal as illustrated in Figure 4.1.2.4.

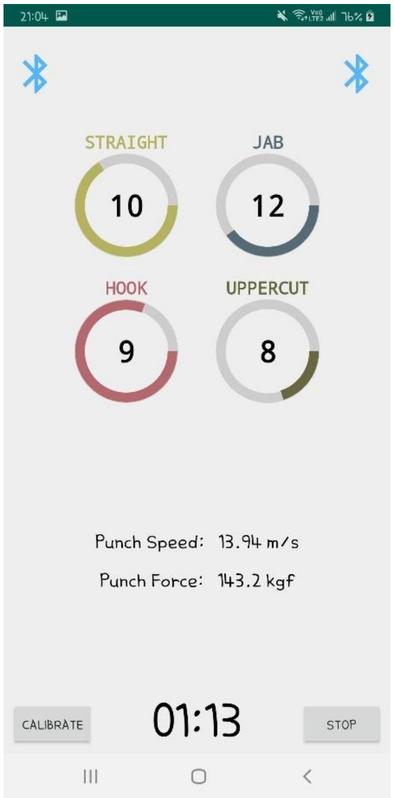

Figure 4.1.2.4 UI During Target Mode

It will also contain real-time information including punch force, punch speed and the number of punches. When the time is up, the application will inform the end of the session to the user by popup dialog as illustrated in Figure 3.1.2.5. The popup dialog will also play an alarm to notify the user about the end of the session easily.

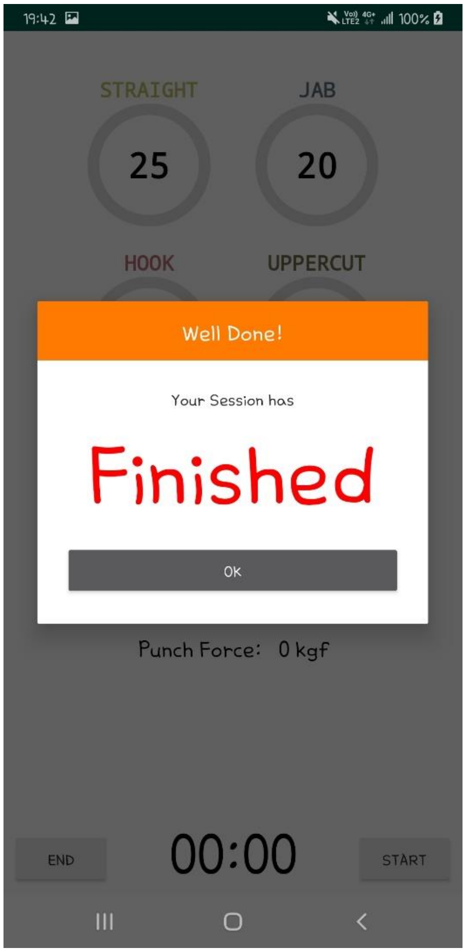

Figure 3.1.2.5 Popup for Session Expiration of Target Mode

The summary page for the target mode will illustrate how far the user reached to achieve the goal with progress bars. The page will include data such as percentage of completion for each punch type, average speed, average force and average completion as illustrated in Figure 3.1.2.6.

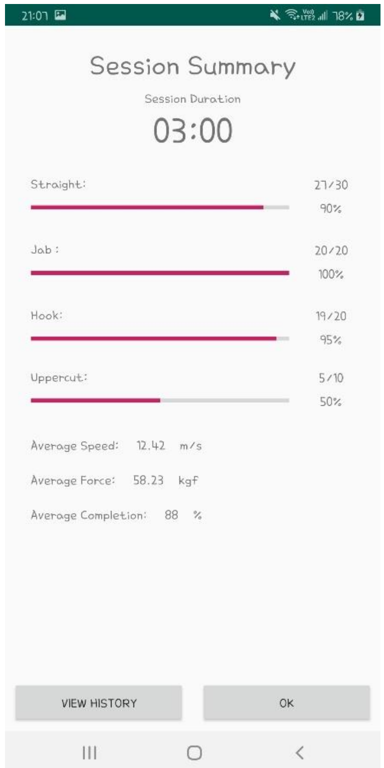

Figure 4.1.2.6 Summary Page of Target Mode

4.1.3 History Mode

History Mode is designed to show users about their training history with our Smart Boxing Gloves. The history of training sessions is categorized by month, so the users can choose a specific month to see the month's history. At the very top of the page, there are two spinners which allow users to choose between target history and training history and specific month they would like to see their history.

4.1.3.1 Training Mode History

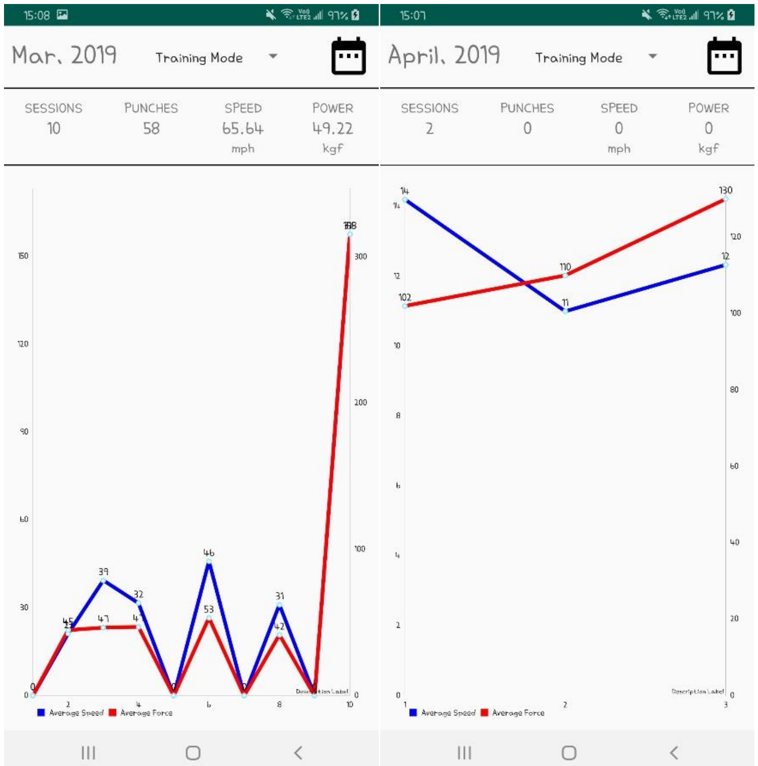

Figure 4.1.3.2 Training History

Training history mode displays a graph which describes users' progress in terms of average speed and force of punches recorded for each session as illustrated in Figure 4.1.3.2. The left y-axis is a scale for punch speed, and the right y-axis is a scale for punch force. By implementing the graph, users can easily see changes in their speed and force throughout the month, and by plotting the average speed and force together in one graph, users can interpret the relationship between their punch force and speed.

4.1.3.2 Target Mode History

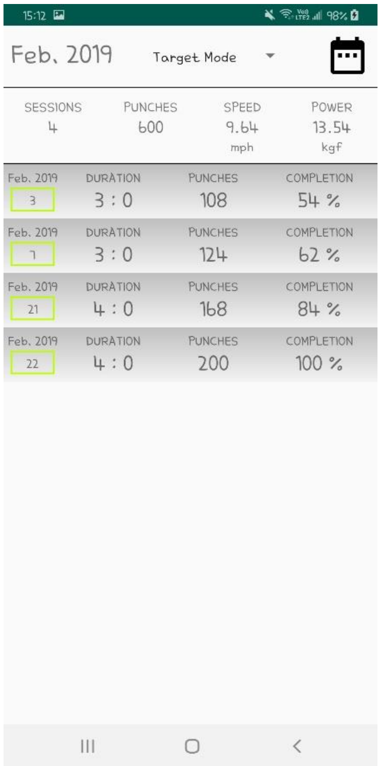

Figure 4.1.3.2 Target History

Target history mode list out all the training sessions of the specified month by including information such as the number of punches thrown during the session and average completion, duration and date of the session as illustrated in Figure 4.1.3.2. These sessions are listed by using recycler view, which is optimized to display a large amount of view in the form of a list, to optimize the memory usage. The ultimate goal of the target mode history is to let users to easily see their progress by comparing their completion among different sessions.

4.1.4 SignUp / Login System

The signup system of the application mainly consists of three parts: Facebook Login, Google Login, and email-based login as illustrated in Figure 4.1.4. Users will be directly signed up and logged into the application if they use Facebook or Google accounts to log in. User also may input a valid email address and password to get signed up for the application.

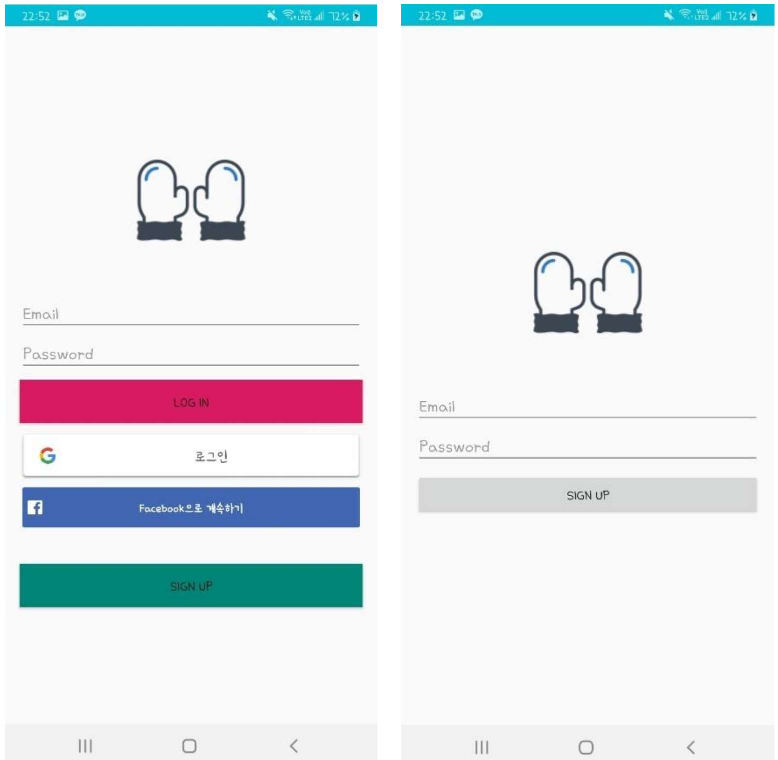

Figure 4.1.4 Login / Signup System

4.1.5 User Information Settings

Once users register for the application, they are redirected to settings page as illustrated in Figure 4.1.5.1 where they can input their height, weight, arm span, stance, and sex. These fields are required for the user to input to provide more accurate speed and force calculation. Users can always edit the personal information through the option menu as illustrated in Figure 4.1.5.2.

Figure 4.1.5.1 Personal Information Settings

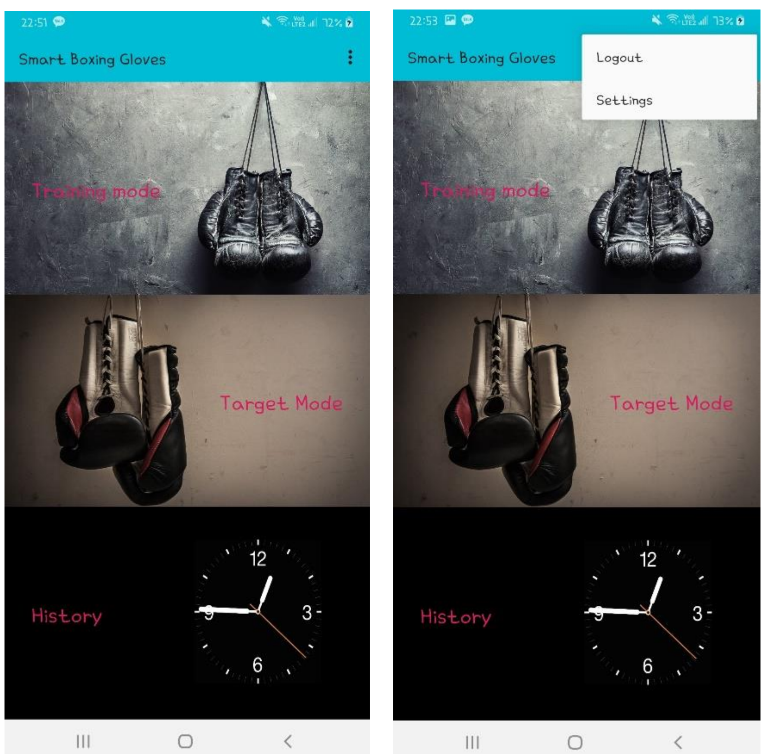

Figure 4.1.5.2 Option Menu for Settings

4.2 Methodology

4.2.1 Punch Classification

Punch Classification, which is a functionality that analyzes the movement of boxers' fist and classifies the punch into a specific type, is one of the most important parts of our project. In order to provide more accurate and precise punch classification to users, we have implemented an instanced-based learner machine learning model which classifies instances by making use of distance.

4.2.1.1 Weka

In order to obtain a machine learning model for punch classification, we implemented a Waikato Environment for Knowledge Analysis (Weka) machine learning model. Weka is a data mining software used for machine learning developed at Waikato of University New Zealand. It provides different machine learning algorithm such as clustering, classification, and regression. In order to use Weka for machine learning, it is required to provide dataset in Attribute-Relation file format as input and Weka will produce a machine learning model as user specified.

4.2.1.2 Instance-based learner

K nearest neighbor algorithm and K* are machine learning algorithms that belong to instance-based learner. The instance-based learner is designated to classify instances by referring to a dataset which consists of already-classified example data. The algorithm measures the similarity between the new data and reference dataset and based on the similarity measured, it predicts the class of the new instances. Since the new instance is classified into the class with the highest similarity, the instance-based learner is also called as “winner-take-all” or “memory-based” methodology.

4.2.1.3 Algorithms

We have used two different algorithms to classify punch types after precisely investigating the result of machine learning.

4.2.1.3.1 K Nearest Neighbor Algorithm

K nearest neighbor algorithm, which is one of the instance-based learning models, finds k number of nearest neighbor nodes based on Euclidian distance and classify the new instance depending on the most dominant class that neighbor nodes belong to. For easier understanding, Figure 4.2.1.3 illustrates the K nearest neighbor algorithm. In Figure 4.2.1.3, when k is chosen to be 3, the nearest three neighbors consist of two red class nodes and one green class node which means the new instance will be classified into red class. On the other hand, if k is chosen to be 5, the nearest five neighbors consist of 2 red class nodes and three green class nodes, and thus classified into green class.

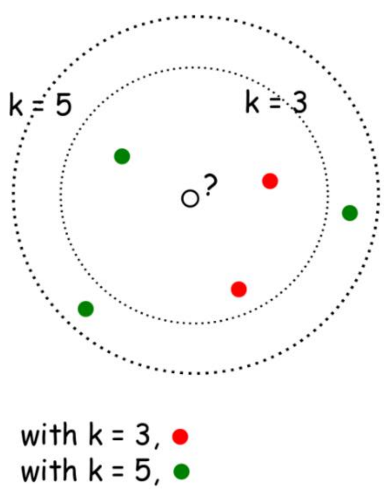

Figure 4.2.1.3 K Nearest Neighbor Algorithm

4.2.1.3.2 K-star Algorithm

Another algorithm implemented for our application is called K* algorithm. The algorithm finds a neighbor node that is nearest to the target node. The difference between the K nearest neighbor algorithm and K* algorithm is that K* uses entropy-based distance as a method measuring the distance between the new instance and trained instances and a new instance.

In this algorithm, the entropy distance will be measured by investigating the complexity of the new instance to transform into the reference node. The entropy of transforming instance a to instance b is measured by the following equation:

$$K^*(b|a) = K^*(x) = \frac{1}{2x_0} e^{-\frac{x}{x_0}} \, \mathrm{dx}$$

$$\text{where x} = |a - b| \text{ and } x_0 = parameter\ to\ be\ chosen$$

The distance of instance will be calculated by using the above equation for a specific attribute. Therefore, if an instance consists of more than one attribute, it will calculate the final distance by unifying all the distance values as described in the following equation:

$$\text{resulting distance} = \sqrt{{d_1}^2 + {d_2}^2 + {d_3}^2}$$

Finally, it will calculate the probability for an instance to be in a specific category and choose the one with the highest probability for the classification. Probability for instance x to be in class X is calculated by adding up all the probabilities of instance x to become instance y when y is set of all instances that belong to class X, and the equation is described as:

$$P^*(X|x) = \sum_{y \in X} K^*(y|a)$$

4.2.1.4 Data Preparation

In order to use the KNN and K* algorithm to predict the type of punch, we prepared the dataset in Attribute-Relation File Format. This dataset includes relation name, attribute field and datasets as described in Figure 4.2.1.4. For our system, we have decided to use three values (x-axis acceleration, Y-axis acceleration, and z-axis acceleration) to classify punches into four different categories classes (straight, jab, hook and uppercut). The dataset will be prepared in the form of {x-axis acceleration, y-axis acceleration, z-axis acceleration, class}.

```
%ARFF file for the punch data%
%class 0 = straight, 1 = hook, 2 = uppercut%

@relation punch

@attribute x_acceleration NUMERIC
@attribute y_acceleration NUMERIC
@attribute z_acceleration NUMERIC
@attribute class {0, 1, 2}

@DATA
4.38,2.78,2.42,0
6.28,4.27,2.63,0
7.65,7.14,3.82,0
7.79,3.06,-4.35,0
9.46,3.23,-5.96,0
6.43,4.11,3.55,0
6.75,4.3,-3.18,0
7.21,4.19,-6.31,0
5.44,2.88,2.93,0
6.57,2.79,4.38,0
5.61,3,3.37,0
5.79,2.72,3.18,0
5.83,1.45,3.52,0
5.98,3.22,3.96,0
7.07,2.86,3.95,0
8.35,1.29,-5.02,0
5.89,3.59,3.14,0
6.72,2.11,3.4,0
8.3,1.5,-6.72,0
6.23,2.81,3.34,0
```

Figure 4.2.1.4 Dataset Preparation for Classification

4.2.1.5 Classification

In order to effectively classify punch types, we have implemented the Weka model inside the Android code. The process of classifying punch is as follow:

1. Create a Sample data instance and set the features (attributes) using data read in from Bluetooth.

```
public class Sample {
    public double [] features;

    public Sample(double[] _features) {
        this.features = _features; // 3 features x_acceleration, y_acceleration, z_acceleration
    }

}
```

2. Load WEKA model

```
public void onClickButtonLoadModel(View _v) {

        AssetManager assetManager = getAssets();
        try {
            mClassifier = (Classifier) weka.core.SerializationHelper.read(assetManager.open("righthand.model"));

        } catch (IOException e) {
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

3. Predict punch type with WEKA model

```java
public void onClickButtonPredict(View _v) {

    // Attriutes list and Class list
    final Attribute x_acceleration = new Attribute("x_acceleration");
    final Attribute y_acceleration = new Attribute("y_acceleration");
    final Attribute z_acceleration = new Attribute("z_acceleration");
    final List<String> classes = new ArrayList<String>() {
        {
            add("straight");
            add("hook");
            add("uppercut");

        }
    };

    // Make data in form of {x-axis acceleration, y-axis acceleration, z-axis acceleration, class}
    // for prediction
    ArrayList<Attribute> attributeList = new ArrayList<Attribute>(2) {
        {
            add(x_acceleration);
            add(y_acceleration);
            add(z_acceleration);
            Attribute attributeClass = new Attribute("@@class@@", classes);
            add(attributeClass);
        }
    };

    // make a list of data to be predicted
    Instances dataUnpredicted = new Instances("TestInstances",
            attributeList, 1);
    dataUnpredicted.setClassIndex(dataUnpredicted.numAttributes() - 1);

    DenseInstance newInstance = new DenseInstance(dataUnpredicted.numAttributes()) {
        {
            setValue(x_acceleration, mSample.features[0]);
            setValue(y_acceleration, mSample.features[1]);
            setValue(z_acceleration, mSample.features[2]);
        }
    };
    newInstance.setDataset(dataUnpredicted);

    // predict punch type
    try {
        double result = mClassifier.classifyInstance(newInstance);
        String className = classes.get(new Double(result).intValue());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

4.2.2 Database System

The Smart Boxing Gloves uses Firebase real-time database as a primary data storage method. All the users registered for the application will be listed the Authentication part of the firebase as illustrated in Figure 4.2.2.1. Then all the data gathered from training will be saved in the real-time database so that they can be retrieved for history mode.

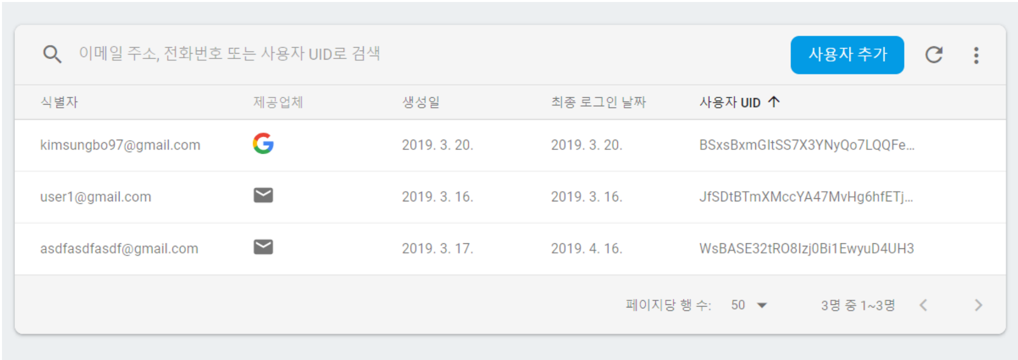

Figure 4.2.2 Firebase User Authentication

4.2.3 Speed Calculation

The speed calculation will be explained in Johnathan Leonanda’s report.

4.2.4 Power Calculation

The force calculation will be explained in Johnathan Leonanda’s report.

Chapter 5. Implementation and Results

5.1 Punch Classification

The performance of the punch classification system is examined by using 10-fold cross-validation. 10-fold cross-validation test the machine learning model by separating the dataset into ten different groups and using the selected group as a test set while the remaining dataset is used as the training datasets. The result of testing will be determined by examining the confusion matrix which of its row represent actual classes and columns represent predicted classes. By interpreting the confusion matrix, it will be able to retrieve the following five factors: TP rate, FP rate, precision, recall, and F-measure. These five factors are the analysis criteria, that can be obtained from a confusion matrix, for machine learning models and the meaning of each term is as described below:

1. TP rate: TP rate represents the percentage of the items that are correctly classified within the pool.
2. FP rate: FP rate represents the percentage of the items that incorrectly classified within the pool.
3. Precision: precision refers to the system's ability to classify instances correctly and can be represented as

$$\text{precision} = \frac{TP}{TP + FP}$$

4. Recall: recall refers to the system's ability only to select relevant items. In other word, it can be referred to as the sensitivity of the model and can be determined using the following equation.

$$\text{Recall} = \frac{TP}{TP + FN}$$

5. F-measure: F-measure is used to test the accuracy of the system by combining precision and recall because they have a negative impact on each other. The F-measure can be calculated by using the following equation

$$\text{F} - \text{measure} = 2 \times \frac{precision \times recall}{precision + recall}$$

5.1.1 Lead Hand Punch Classification

For rear hand punch classification, we used 2940 data sets to build the WEKA model. More specifically, the data set consists of 1013 straight punches, 914 hooks, and 1013 uppercuts and the distribution of punch types is as described in Figure 5.1.1.

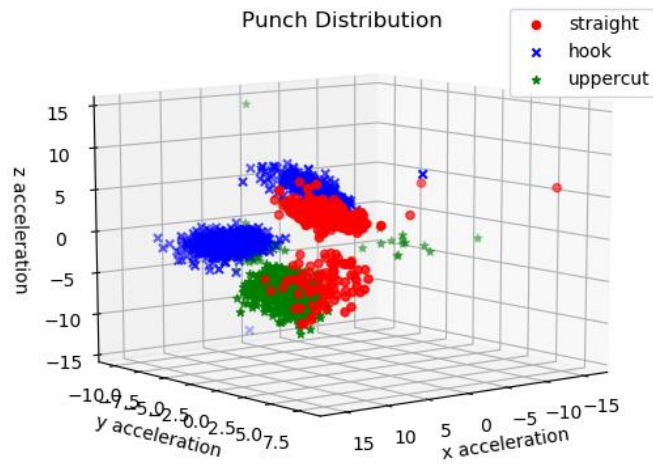

Figure 5.1.1.1 Acceleration Distribution of Different Punch Types for Lead Hand

The dataset is tested with both the K nearest neighbor algorithm and K* algorithm. When K nearest neighbor algorithm with K set to 7 is used to train the dataset, the model achieved up to 95.9524% of accuracy where K* algorithm resulted in 95.71% of accuracy. Therefore, for lead hand punch classification, the model that is trained with K nearest neighbor is used.

```
  a   b   c   <-- classified as
964  12  37 |   a = 0
 11 883  20 |   b = 1
  8  31 974 |   c = 2
```

Figure 5.1.1.3 Lead Hand Classifier Confusion Matrix

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.952	0.010	0.981	0.952	0.966	0.949	0.989	0.978	0
	0.966	0.021	0.954	0.966	0.960	0.942	0.992	0.980	1
	0.962	0.030	0.945	0.962	0.953	0.928	0.988	0.971	2
Weighted Avg.	0.960	0.020	0.960	0.960	0.960	0.939	0.989	0.976	

Figure 5.1.1.2 Lead Hand Classifier Accuracy by Class

In general, the model classified 2821 out of 2940 instances correctly. More specifically if you look at Figure 5.1.1.2 and 5.1.1.3, 95.2% of straight punches, 96.6% of hook and 96.2% uppercut are classified correctly. The F-measure, which represents the overall accuracy, tells us that the model is about 96% accurate in classifying the punch types.

5.1.2 Rear Hand Punch Classification

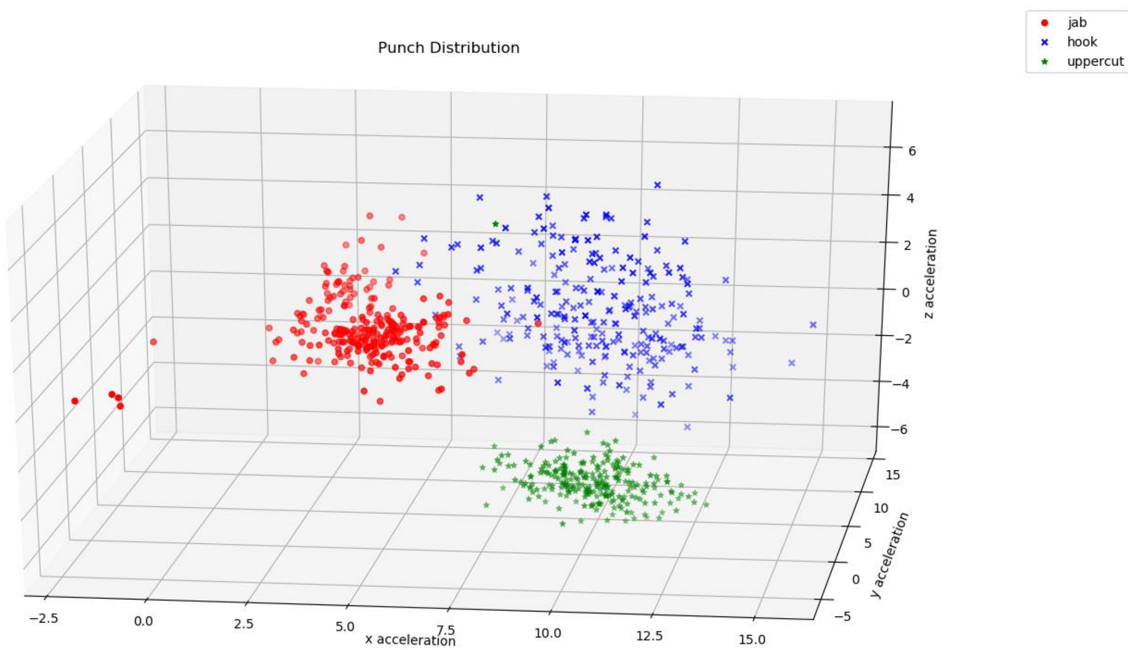

Figure 5.1.2.1 Acceleration Distribution of Different Punch Types for Rear Hand

For rear hand classification, the KNN algorithm with k=5 is used to build the classifier. For the dataset, 252 jabs, 239 hooks, and 247 uppercuts are used, and the acceleration distribution of different punch types is illustrated in Figure 5.1.2.1. Figure 5.1.2.2 shows that only one jab, four hooks, and one uppercut are incorrectly classified, and this may be caused by noise in the dataset.

```
=== Confusion Matrix ===

   a   b   c   <-- classified as
 251   1   0 |   a = 0
   4 235   0 |   b = 1
   0   1 246 |   c = 2
```

Figure 5.1.2.1 Rear Hand Classifier Confusion Matrix

Furthermore, Figure 5.1.2.3 illustrates that 99.2% of the instances are correctly classified while only 0.4% of them are incorrectly classified. By combining the TP rate and FP rate, it is concluded that the model is 99.2% accurate if you have a look at F-measure.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.996	0.008	0.984	0.996	0.990	0.985	0.997	0.994	0
	0.983	0.004	0.992	0.983	0.987	0.981	0.996	0.990	1
	0.996	0.000	1.000	0.996	0.998	0.997	0.998	0.998	2
Weighted Avg.	0.992	0.004	0.992	0.992	0.992	0.988	0.997	0.994	

Figure 5.1.2.2 Rear Hand Classifier Accuracy by Class

5.2 Application Performance

5.2.1 CPU Usage

In order to examine the application's performance, I have used Android 'dumpsys' to measure the CPU usage of the application. The examination is conducted in terms of CPU usage in two situations: training/target mode which uses Bluetooth communication and history which uses the process of retrieving data from firebase. Figure 5.2.1 illustrates the CPU usage during history mode, and it is observed that the CPU usage rapidly increases every time I changed the month of history or changed the mode of history. I have tested the CPU usage by changing the history view five times, and every time it needs to retrieve data from firebase, the device CPU usage increased a lot.

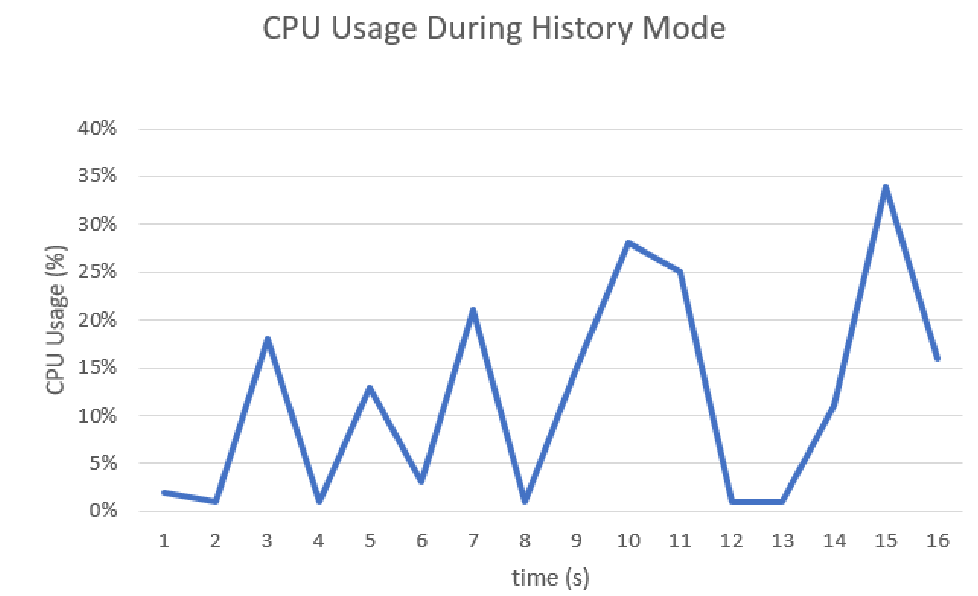

Figure 5.2.1 CPU Usage During History Mode

Figure 5.2.2 illustrates the CPU Usage during Training/Target mode which requires communication between Android and the hardware devices. The highly fluctuating part was caused by Bluetooth connection, calibration and data receiving processes. It can concluded that data transferring between Android application and hardware devices take up some CPU.

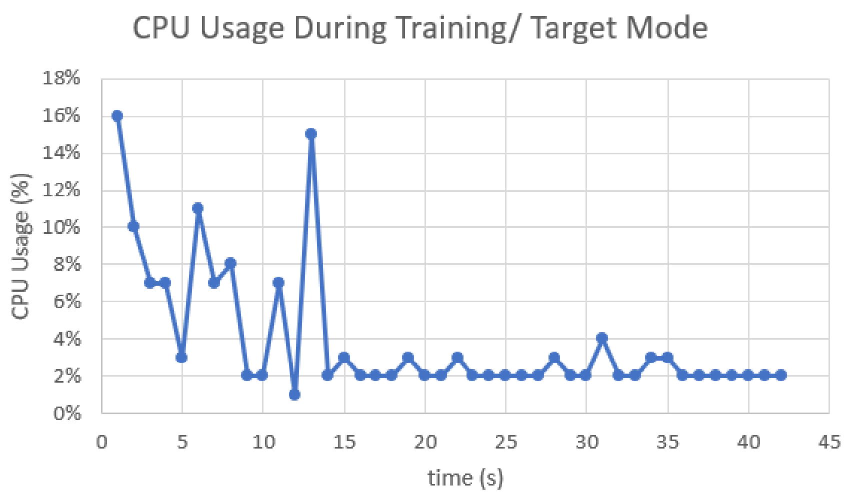

Figure 5.2.2CPU Usage During Training/Target Mode

Chapter 6. Conclusion

This report empirically explains the methodology used for boxing training enhancing wearable device ‘Smart Boxing Gloves.’ The final product satisfactorily measures punch force, speed, and type with the help of accelerometer and machine learning model. In the first term, we faced a problem regarding Bluetooth. Therefore, we have implemented another way of sending data through Bluetooth in this term. Instead of sending all acceleration data, we only send the peak acceleration of each punch in the final product. By doing so, we successfully reduced the amount of data that are transmitted through Bluetooth and as a result, the data-jamming problem is solved. The limitation we could not fix throughout the project is that the machine learning model we used for punch classification only consists of data gathered from one person. In other words, the model is only learned and trained with limited data. The performance of the system will be amplified if the dataset used for the training are gathered from various users. Therefore, as future work, we would like to implement a feature that asks users to input their punch information before start using the application. For example, we can implement a function that asks users to throw 5 of each punch types inside the personal information setting part. Then these data can be retrieved for updating the machine learning model by training again, and this will thus create a better classifier for punch classification.

Bibliography

"KINEXON Real-Time Tracking System Brings Performance Insights to ANGT Finals." *ADIDAS Next Generation Tournament*, Euroleague.net, 22 June 2018, www.adidasngt.com/u18/news/latest/i/8tnaulh7pnovw5mb/kinexon-real-time-tracking-system-brings-performance-insights-to-angt-finals.

Romeiro, Vera Digner. "The Sport Wearables of Tomorrow That Will Unleash Your Performance to Its Fullest Potential!" *Wearable Technologies*, 24 Jan. 2018, www.wearable-technologies.com/2018/01/the-sport-wearables-of-tomorrow-that-will-unleash-your-performance-to-its-fullest-potential/.

Mischke, Johanna. "Wearable Technology: The Latest Trend in Professional Sports." *Wearable Technologies*, 22 May 2018, www.wearable-technologies.com/2018/05/wearable-technology-the-latest-trend-in-professional-sports/.

"TRACK AND MEASURE YOUR PERFORMANCE." *StrikeTec*, striketec.com/the-sensors-2/.

Mccintock, Kahu, et al. "Pilot of Te Tomokanga: A Child and Adolescent Mental Health Service Evaluation Tool for an Indigenous Population." *The International Indigenous Policy Journal*, vol. 3, no. 1, 1995, doi:10.18584/iipj.2012.3.1.5.