IDENTIFYING PEOPLE BASED ON PRESSURE BOARD MEASUREMENTS

by

SAMARTH LAKHATARIYA

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Janne Lindqvist

And approved by

____

____

____

New Brunswick, New Jersey

MAY, 2015

ABSTRACT OF THE THESIS

IDENTIFYING PEOPLE BASED ON PRESSURE BOARD MEASUREMENTS

By SAMARTH LAKHATARIYA

Thesis Director:

Janne Lindqvist

Identifying people is important for various applications and context. In this thesis, we examine the potential to identify people based on how they step. Towards this end, we designed and implemented a system utilizing the Wii Balance Board. When a user steps on the board, we can obtain weight distribution and center of pressure. These can be used to train our model and based on it, we can identify a user. We tested our system using dataset obtained from 19 volunteers. The presented approach has accuracy of 97%. This indicates the approach could have potential and warrants further work.

Acknowledgements

I would like to express my gratitude to my advisor Dr. Janne Lindqvist for his immense guidance and support in the research. The depth of knowledge that he poses in the field of human-computer interaction and security engineering is incomparable. You have been a tremendous mentor for me. I would like to sincerely thank you for encouraging my research. I could not have imagined having a better advisor and mentor for my research.

I would also like to thank the rest of my thesis committee : Dr. Yanyong Zhang and Dr. Saman Zonouz for serving as my committee, providing encouragement and insightful comments/critiques.

My special thanks to my fellow lab mates of Human-Computer Interaction group for their support in my research. For all the discussions we had, for knowledge we shared and cool projects we worked on. It was fun working with you guys.

Thank you to all my friends and all my Professors, to Rutgers University for giving me the opportunity to pursue my masters and providing me with state-of-the-art facilities.

Last, but not least, I am grateful towards my parents and my sister for their love and support. I am grateful to have you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

For decades, many diverse methods have been developed in the field of personal identification for the purpose of securing the companies, government bodies, e-commerce applications, etc. The purpose of these systems development is to ensure that certain highly secure information are only accessed by a legitimate user. There are several well known and widely used approaches to authentication, including: 1) PIN or password based access system, 2) Smart card or key based access system and 3) Biometric based access system. In pin or password based access time, a user enters the pin to access certain information. It is popular with the computers and mobile devices. In the smart card or key based system, a user swipes the card to get access to the secure location. The biometric system has the advantage that the user is not required to remember the password or carry a card. Biometrics measure a person's unique physical or behavioral characteristics for identification. Therefore among all security system, biometric is more secure and convenient authentication tool [1]. Biometric systems are classified into two types viz, physical based biometric authentication; includes fingerprints, hand geometry, retina, iris, face recognition etc. and behaviour based characteristics; include signature, voice, gait etc. A feasible biometric system should meet certain criterias such as specific recognition accuracy, robust to various attacks, harmless to users, acceptable by intended users, etc. [2].

Fingerprints recognition have been used for quite a long time. A fingerprint is recognition of pattern of ridges and valleys on the surface of fingertip. It has very high matching accuracy and low cost. But fingerprint recognition system is unsuitable for certain reasons such as genetic factors, occupational reasons such as cuts or bruises, environmental, etc. Hand geometry is the measurement and analysis of shape of hand.

The technique is easy to use and cheap. But it has been shown that geometry of hand is not very distinctive and it cannot be scaled up for large organization [2]. Retina based system analyses the blood vessels layer situated behind the eye. It is considered to be more secure biometric system since it is not easy to duplicate or change the retinal vasculature [2]. Retina based recognition was not warmly acceptable because it wouldn't work with glasses, requires a conscious effort and due to some medical conditions such as hypertension [1].

Iris is the colored ring tissue surrounding pupil. This iris texture carries distinctive information which can be used for biometric based predicition. The performance of iris based recognition is potentially higher than average template matching [1]. Also, iris does not require close contact and it is promising for developing system to be used in large scale recognition. It has a problem with ease of use, is intrusive, can be obscured by objects such as eyelid or eyelashes, requires proper illumination, etc. Face recognition has also gained considerable attraction but it requires more understanding to use the system. Also changing illumination, hair-style, occlusion, etc. affect automatic identification. Signature and voice based recognition could also be considered for identification but these features require user's cooperation.

Gait is a peculiar walking style and is a spatial temporal biometric. It can be used in low security application because it is supposed to be not very distinctive. Also it requires video-sequence footage of a person. It requires high computation, input intensive and a low resolution from a distance might not provide accurate result [2].

Nakajima started footprint based recognition. There were ten participants and he showed 85% of recognition rate [3]. In [4], Jung uses mat-type pressure sensor and one-foot approach for person recognition. They could reduce recognition error rate from 48.5% to 36.0%. In [5], Jung could achieve 80% recognition rate using only COP trajectory for eleven people. The work in this area demonstrates that there is a lot of possibility to explore inexpensive way of person identification.

1.1 Motivation

The motivation is to develop a novel system for person identification. There are various disadvantages of current biometric based identification system such as high computation cost, require high-end technologies, not user-friendly etc. So how can we develop a simple system for person identification based on person's walking behavior. The application for such development could be as a doormat for automatic unlocking of the door when an owner steps on it.

1.2 Contribution

We developed a system using Wii Balance Board (WBB). When a user steps on the balance board we can obtain weight and center of pressure (COP) trajectory.

Everyone has a unique way of walking, so identifying the pattern of walking (here it relates to standing on WBB), we can detect the user. We only have two features: weight and center of pressure and using the limited data we are trying to identify a person. We are using SVM, kNN and Random Forest as our multi class classifiers for identification. The accuracy of these classifiers are evaluated using nine-fold stratified cross validation method. We tested our system with datasets obtained from 19 participants. We can obtain the accuracy as high as 97% for Random Forest.

The system can be easily implemented and its economical. It does not require high-end equipments such as high resolution camera, etc. In our study, a user just have to step on the board which is much more convenient than other biometric based systems which requires user's cooperation and are intrusive. The application of such study can be for medical diagnosis, automatic profile setting in WBB, person recognition in home applications, etc.

1.3 Outline

Chapter 2 describes the previous work done in this field. Chapter 3 explains the hardware in detail. Chapter 4 and 5 discusses method and design of the study. Results and future work is discussed in Chapter 6 and Chapter 7. Conclusion in Chapter 8.

Chapter 2

Related Work

In this chapter, we discuss the related work in the field of footprint recognition. Nakajima introduced the footprint based recognition system in 2000. The chapter also explains about the information that we can obtain from footprint data and the performance achieved so far in this field of biometrics for person identification. It points out to the fact that it is much more convenient than the other biometric systems as discussed in the introduction.

Nakajima et al. started person recognition method using footprints. In [3], they used pressure sensing mat to collect footprint image. They used BIGMAT to acquire pressure distribution of footprints. The geometric information such as directional and position information is used from the obtained footprint of each leg. Later normalization of the data is done and provided to the prediction algorithm. They achieved 85% accuracy for ten male participants, collecting eleven samples from each of the participants. The paper also mentioned that the footprint based recognition system would work for personal recognition in a small group and human friendly environment because a person would already be inside a house when the footprint is obtained.

In [4], Jung used one-step footprint data for person recognition. They mentioned that human gait is insufficient for prediction because there might be change in walking velocity of a person and large amount of stable walking data is required for this method. The paper proposed a new method for recognizing. They used quantized based directionally aligned COP trajectory. The output of trajectory extraction is provided to left-to-right type Hidden Markov Model (HMM) which acts as a recognizer. The output of the HMM is then provided to Levenberg-Marquardt (LM) method to overcome two restrictions: same length and walking speed of left and right foot. LM is a combination of Gauss-Newton method and gradient descent algorithm which is used for curve-fitting. Based on their model, it showed 64% recognition rate for eight men footprint data. They also discussed about three important facts for person recognition using footprint: walking behavior is different for different people, walking behavior could be different in same person and left foot motion and right foot motion are not similar. Hence it is required to use both the feet data for prediction.

In [6], Shijia et al. introduced footstep induced floor vibration to identify people. They used geophones for sensing floor vibration. The system senses the floor vibration and detects the footsteps signal. In the study, they defined step event (SE) as floor vibration signal induced by a footstep. The idea is that these SEs are different for different people and its same for same person and hence can be used for identifying a person. From the SEs obtained, they performed step extraction and feature extraction. They modeled a hierarchical classifier for identification which included step level and trace level. In step level, they used SEs of different people and C-Support Vector Classifier (C-SVC) with the radial base kernel for prediction. In trace level, they eliminate same SEs and SEs which has low confidence level, which improved the accuracy. 80% of the data obtained is used for training and remaining for testing. By using the above prediction method they achieved 83% accuracy when identifying all traces and if trace level classification is used, accuracy is improved to 96% for five people. They also discussed about challenges using the system in real scenarios and methods to improve them. If number of registered users increase than the accuracy would drop down because user might fall into same footstep category and also computational complexity on SVM's would increase. To handle this situation, they are planning to separate level of features and use other localization information like stride width, stride length etc. The other challenges would be same person wearing different shoes. To overcome this scenario, the system should store information of a person with different shoe type as same person and also use different behavior patterns such as stride length etc.

In [7], Chakraborty et al. tried to explore the possibilities of gathering useful information by using single point pressure sensors for recognizing a person. Also they investigated optimum location of sensor(s) and number of sensors required for prediction. Pressure sensors were attached on the shoes insole. They experimented with three sensors and finally decided that two sensors are sufficient. They chose two different locations for two sensors: A-location has sensors at toes and under heel and B-location has sensors inside wide front part of foot and under the heel. They used Artificial Neural Network (ANN) of type feed forward network trained by error back-propagation as their classifier. They extracted two feature sets based on certain characteristics. For feature set-1 and for A-sensor location, it achieved 88.8% accuracy while B-location achieves 74.2% accuracy for five people. When feature set-2 is used, A-location gives accuracy of 78.3% while B-location gives 76% accuracy for five people. The other applications of such work might be classifying among walking or jogging or stepping up/down the stairs, calorie burnt, faults in walking or balancing problems in older people, etc.

In the paper, the Smart Floor : A mechanism for natural user identification and tracking [8], Orr and Abowd discussed about the Smart Floor system that they created, collection and testing of large footstep dataset. They created hardware with load cells fitted under steel plate and a data acquisition tool. They obtained ground reaction force (GRF) which is the output reaction force of the load cells exerted on a person at the same time when the body exerts a contact force. Both the forces are equal and opposite in nature. In modeling, they included ten features from the load profile and performed normalization. The normalized data is then provided to nearest neighbor recognizer. There were fifteen participants and collected 1680 footsteps out of which half were used for training and half for testing. They obtained 93% identification result. Also footwear is negligible on recognition accuracy and the system can be deployed into various household locations such as house entrances, kitchen, etc.

In [9], Pappas et al. designed a novel gait phase detection architecture. The system detected four different gait phases such as stance, heel off, swing and heel strike. They used three sensors inside the shoe sole and a miniature gyroscope attached to the shoe sole. The sensors measures the applied pressure and the gyroscope is used to measure rotational velocity. The system designed offers the reliability of 99% in detecting walking, standing ,sitting etc. Hence the system can be used in day-to-day activities.

In [10], Anuradha et al. used a wearable sensor system for human identification based on gait analysis. They used two wireless sensor nodes and received gait signals from it. These signals are then segmented and feature extraction is performed. Linear Discrimination Analysis (LDA) was used to select the best features and the output was provided to k Nearest Neighbor classifier with k = 1. They obtained 84 % accuracy for four participants. They also found features such as maximum , minimum values and RMS values were best features for their analysis.

In [11], Miyoshi et al. proposed a novel approach of person identification using capacitor microphone for recording footsteps. The features were extracted from the signals and provided to two classifiers k-Nearest Neighbor(kNN) and Gaussian Mixture Models (GMMs) for comparison. They collected 720 footstep data from twelve subjects and three types of footwear. They obtained 79.9 % accuracy in kNN and 92.8 % in GMMs. Also GMMs was more than 90 % accurate for all types of footwear.

In [12], Middleton et al. were able to obtain 80% accuracy by using gait information. They used three informations such as stride length, single step period (also known as stride cadence) and heel toe ratio. The system consisted of components such as coax cable type sensors which is analogous to computer keyboard, a large sensor mat, interfacing of hardware using PIC microcontrollers and USB cable and analyzing tool. The mat had a simple design with four isolated grids and two layers of sensor to avoid ghosting problem. The mat was interfaced with three PIC microcontrollers and a USB cable. Three informations: stride length, gait period and heel toe ratio was extracted and analysis was done using Euclidean distance and confusion matrix. There were fifteen participants who were asked to walk twelve times and in each case two complete gait cycle or four footfalls were captured. Out of fifteen participants twelve were predicted correctly giving accuracy of 80%. The paper also mentioned that if just one feature: heel to toe ratio is used, 60% accuracy can be obtained.

In summary, various biometric system have their advantages as well as disadvantages. Face recognition falls short in the resolution, occlusion, lighting problem, etc. Voice recognition has a problem with noisy environment. Some of the biometrics have closeness problem. There is a shortage in the gait recognition method due to insufficient video resolution, inflexible, low recognition rate, expensive instruments for constructing system. We plan to use footprint based person recognition system. The system is developed based on footprint recognition but the recognition rate is one factor for very few commercial use.

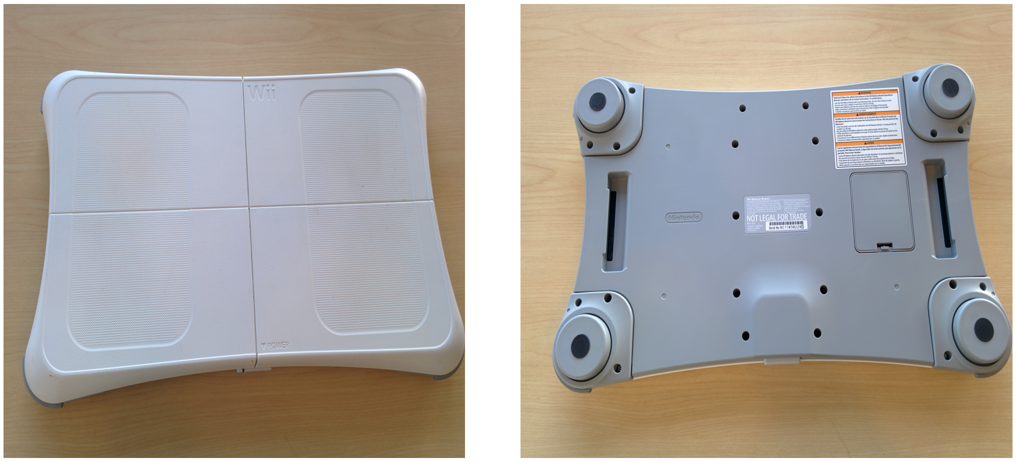
Chapter 3

Hardware

In this chapter, we describe Wii Balance Board in general, its usage and hardware interface technology. In the next section, we would study about the accuracy of WBB.

3.1 Wii Balance Board (WBB)

The WBB is an accessory for the Wii and Wii U video game consoles developed by Nintendo as shown in the Figure 3.1. It looks similar to weighing scale and it uses Bluetooth technology for communication with Wii. The dimension of WBB is approximately 23 X 43 X 5.3 cm and it weighs around 7.7 lb. It has four pressure sensors, located at the corners of the board, which measures the center of balance of a user. The load sensors used in the design are of different kind and are known as strain gauges. The advantage of strain gauge is it does not have any moving part and it is simple in design [13]. The design of WBB is robust and can withstand greater than 300 kg (660 lb) of weight.



(a) Wii Balance Board front side. The design is similar to bathroom scales.

(b) Wii Balance Board back side. Four sensors are provided at four corners of the board.

Figure 3.1: Wii Balance Board front and back view.

Figure 3.1a shows the front part of the Wii balance board. It is similar to bathroom scales and has two foot like design embossed in it. We can also see a power button in the center of the WBB. In the back part of WBB as shown in Figure 3.1b, we can see four sensors at four corners and a battery component which also has a reset pin. It runs on four AA batteries which can power the board for 60 hours.

In [13], Jones and Thiruvathukal has described about all the accessory developed by Nintendo. Their design ideals were simple household items. For WBB the design resembles a set of bathroom scales. They also talked about the evaluation of the WBB. They took great care of the economy, not to exceed the cost but at the same time it should be efficient.

WBB is developed as a motion sensitive game controller but with a different goal in mind. It considered health and fitness management as major factors and included various games of training, aerobics, balance and yoga exercise. It can be used for many different things such as walking, running, jumping, even flapping arms, tilting, jogging etc. during gaming. It would display animated balance board character on start screen and we could measure body mass index, weight etc. It is a different kind of gaming device and experience where the developers wants the user to feel the presence in the room at the same time enjoy the game instead of the immersive gaming where he forgets the environment around him.

Besides the usage in gaming industry, WBB has also found its application in the health sector. WBB is used for various purposes such as evaluation of static posturography of a patient [14], in virtual rehabilitation system providing exercises for rehabilitation of postural instability and balance disorders [15], in various gaming applications for balance disorder people which can also provide the feedback mechanism [16], in augmented reality providing virtual environment for walking in a synthetic world [17]. It has its various advantages because of its portability and robustness.

The WBB has its extension controller permanently connected through which it exposes data. The data is sent at the rate of sixty signals per second. It sends two types of information: the data itself and the calibration information. The WBB reports its 8 bytes of data readable at 0xa4008 and 24 bytes of calibration data readable at 0xa40024 to 0xa4003a [18]. Therefore each sensor returns 2 bytes of data and 6 bytes of calibration data.

3.2 Why WBB?

Bartlett et al. [19] measured the force and center of pressure accuracy of the WBB. The motivation behind their work was to compare the uncertainty metrics and reliability of WBB with laboratory-grade force plates across various conditions and provide calibration value. Also before their work, there was no standard information about accuracy and reliability of WBB force and COP even though it was being used in wide range of applications. They also discussed about the wear of WBB. It is highly robust and did not show any significant impact in wear over 4 years. Their analysis shows that uncertainty of force measurement is $\pm 9.1\ N$ and COP location has uncertainty of $\pm 4.1\ mm$. Also it could detect the difference in postural sway of greater than 10 mm, which is sufficient to distinguish between healthy and impaired person. In other study by Bartlett et al. [20], they found that internal calibration was within 1.1% of the experimentally determined values. And the combined weight measurements has uncertainty of ±1.4 kg. Now, if we consider a person of weight 60 kg then the force exerted by him would be in the range of 600 N so the uncertainty of ±9.1 N is almost negligible. Also in our study, we are using relative data since we are using the same balance board for identification of a person using multi class classifier. Therefore this error would not affect the dataset.

In summary, we described Wii Balance Board and its wide range of use, the idea behind its construction and usage, interfacing it to the Wii using Bluetooth connection. We also mentioned about the accuracy of WBB.

Chapter 4

Method

The chapter describes in detail about the method we used to collect the data. The first section is about the participants and general guidelines. The second section describes about the apparatus used, how to integrate Wii to Linux, followed by the procedure in the third section.

4.1 Participants

19 people voluntarily took part in our experiment. They received an explanation of the study. Out of 19 participants, 8 participants were females and 11 were males between the age group of 18-30 years. The demographic information about the participants is listed in the Table 4.1.

	Number of Participants	Average Weight (in kg)
Male	11	78.93
Female	8	65.45
Total Participants	19	73.26

Table 4.1: Demographic information of participants.

4.2 Apparatus

We are using WBB to collect the data of participants. The application is setup to work on ubuntu 14.04 LTS 64-bit OS platform. According to Matt Cutts [21], we can configure WBB to communicate with Linux. It uses Bluetooth technology which is inbuilt in WBB board to communicate. The experiment was conducted in the lab in our department. WBB was put into flat surface for proper measurements and working condition. According to [18], each sensor returns a 16-bit number as data and three 16-bit numbers as calibration data. These three calibration data corresponds to sensor reading for 0 kg, 17 kg and 34 kg. It is then interpolated to obtain further information.

4.3 Procedure

In this section, we would discuss about the procedure that we followed to collect the data. The WBB can be switched-on by pressing the reset button. Once it is connected with Linux system, it can be used to obtain data. We followed following steps for gathering information:

1. The participants were asked to step-on WBB in their normal walking posture. For that, we asked them to walk from a distance and then in their normal gait, step on the balance board.

2. The readings were collected for $4.5\,s$ with a sleep time of $15\,ms$ in between two readings.

3. We can obtain the raw pressure sensor values. There are four pressure sensors so we would obtain four values every 15 ms.

4. The weight on each sensors can be obtained by interpolating the pressure sensor values obtained in Step 2. Calculation of weight is explained in the next chapter. Summation of individual weight of each sensors would give total weight.

5. Center of Pressure was also calculated according to Equation 5.1 and Equation 5.2. The next chapter shows the calculation for COP.

6. Repeat steps 1-5 for nine times.

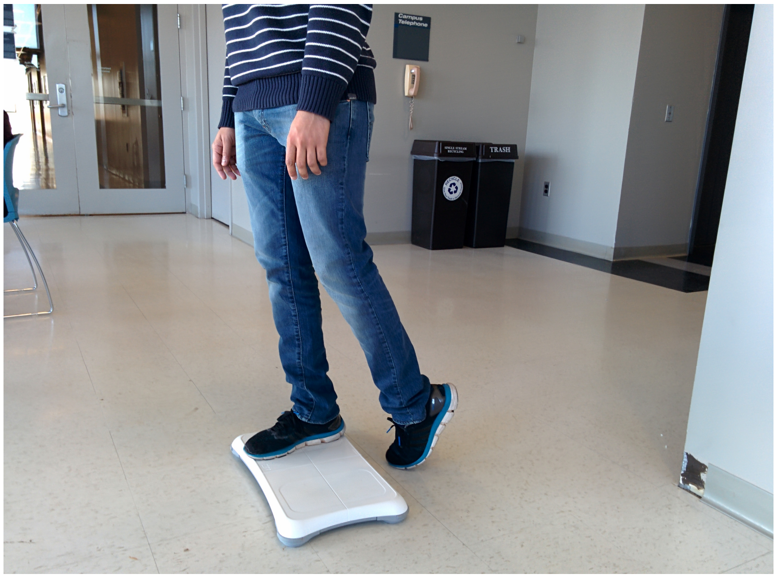


Figure 4.1: A volunteer stepping on the WBB during one of our study.

Figure 4.1 shows one of the participants stepping on the Wii Balance Board in his normal walking style. The participant was asked to walk from a distance and considering it as a flat tile, step on the WBB. By doing this, we ensure that we obtain non-biased data and the participant should not consciously put the same leg on the WBB during each readings. This might bias our result. So asking a participant to walk from a near distance might be a good approach. We can collect the information of his entire gait motion with the help of WBB. But we can obtain only sensor values from WBB which is then interpolated to obtain weight and center of pressure trajectory.

So, at the end of step 5, we can obtain seven feature arrays namely Top Right (TR), Bottom Right (BR), Top Left (TL), Bottom Left (BL) raw sensor values, Weight distribution (WT), COP_x and COP_y values. Each of these seven features would be an array of 300 values (Since data is collected for 4.5 s with a delay of 15 ms). We ask participants to repeat the experiment nine times. Out of these nine repetition readings, seven were used for training the algorithm and two were used for testing. So in all, we collected 171 readings which consists of 300 values of each seven feature arrays.

In summary, we discussed about the method used to collect the data. Dataset consist of total of 171 samples obtained from 19 participants. The dataset is provided to classifiers to check their accuracy. We are using multi class C-Support Vector Classification, k-Nearest Neighbor and Random Forest as classifiers and performing cross validation to obtain the performance of classifiers.

Chapter 5

Design

In this chapter, first section explains architecture of the system. The next section describes the features used and interpolation of weight and center of pressure values from raw pressure sensor data. In third section, we talk about why standardization is important for our dataset followed by details about the recognizer we used. For our study, we used SVM, kNN and Random Forest as our classifiers. Later we discuss cross validation and suitable method for our study.

5.1 Flowchart

Figure 5.1 shows the basic flowchart explaining the data flow of the system. The first block is Wii Balance Board, the hardware that we use in our study. User is asked to step on WBB and the footstep of a user can be obtained from WBB as explained in the procedure section.

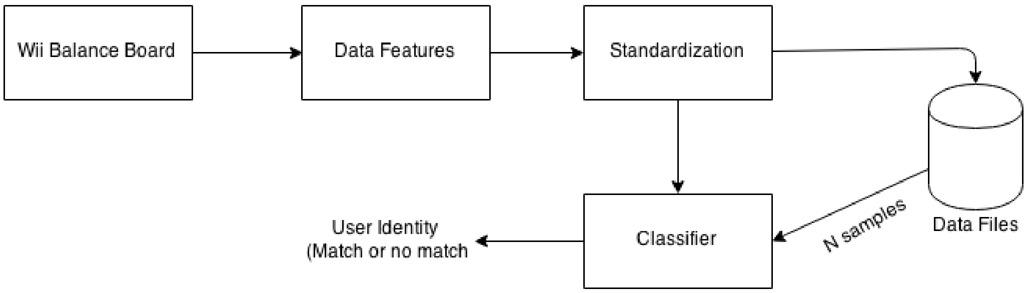


Figure 5.1: Flowchart of our system.

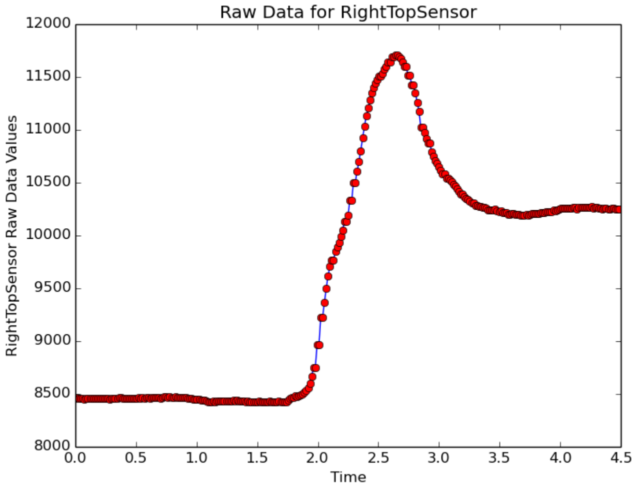
Data Features explains the features collected from WBB. The detail about the features is explained in the next section. We use these features to train the model.

The data obtained is standardized using a standardization tool from scikit. Standardization is an important aspect in machine learning. The output set would have the property of a standard normal distribution.

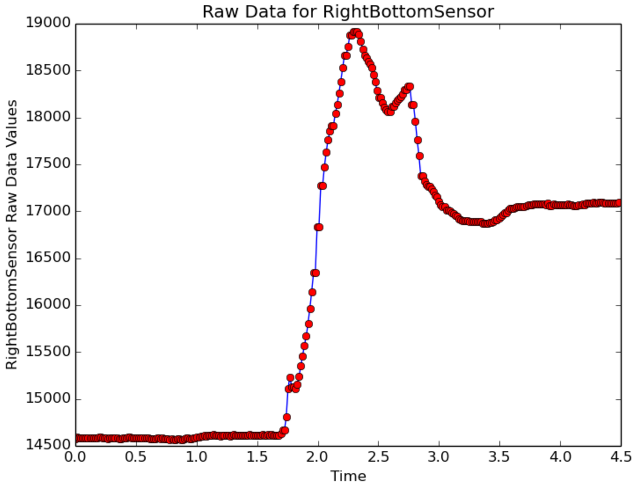
The standardized output is the input for Support Vector Machine (SVM) or k-Nearest Neighbor (kNN). Random Forest (RF) method does not require standardization since it is invariant to transformation of features. We are using SVM, kNN and Random Forest as our multi class classifier and comparing the output results of each of them for performance analysis. The classifier is trained using some training dataset obtained from dataset. Later, the classifier tries to match the input data known as testing data to the previously stored dataset known as training dataset. The classifier runs over through all the classes data previously provided as training set. It then tries to find the maximum probability matching class which matches the testing dataset. If the training dataset matches to the particular class, it displays the name of that class as an output match. In this way, a classifier identifies a user.

5.2 Features

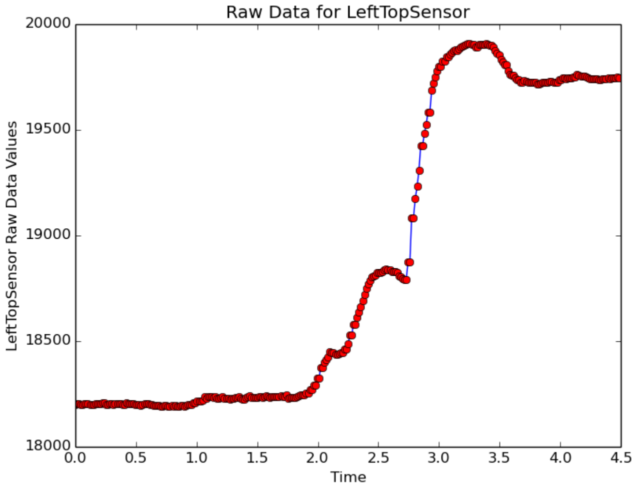
In our study, we can obtain seven feature arrays. Each feature arrays are of 300 values. We directly obtain four sensor arrays values namely Top Right (TR), Bottom Right (BR), Top Left (TL), and Bottom Left (BL) from WBB. In the Figure 5.2, we show the graph of feature arrays of all four sensors namely TR, BR, TL and BL. These four feature arrays along with weight array (WT) and COP_x & COP_y arrays are used as features in our dataset.



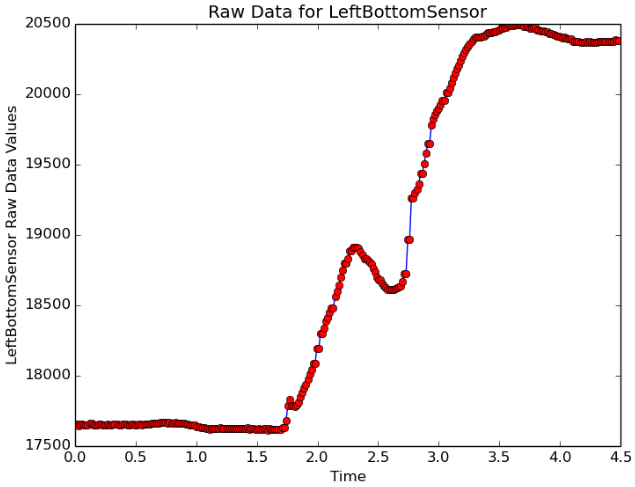
(a) Top Right (TR) Sensor Data obtained from WBB. The value increases when the weight is applied on the sensor and then when the user puts other leg, the value is decreased and stabilizes when he is completely on WBB.



(b) Bottom Right (BR) Sensor Data obtained from WBB. At t = 2 s, user starts keeping his heel. So the sensor value rises.



(c) Top Left (TL) Sensor Data obtained from WBB. The figure indicates that user has put his right foot and now he puts his left leg ball of the foot.



(d) Bottom Left (BL) Sensor Data obtained from WBB. At t= 3.7 s, he keeps the heel of the left foot since the bottom left sensor value is increased at the end.

Figure 5.2: Four sensor readings of WBB vs Time graph.

As shown in the Figure 5.2, we can see that whenever a user keeps his leg on top of any sensors, the sensor values rises since the weight is exerted on sensors. The values are the default readings of WBB and each sensor has different calibration values for 0 kg, 17 kg and 34 kg. So the values are in different range. We can also see from Figure 5.2 and time details in X-axis that user first put his right leg heel at t = 2 s since bottom right sensor values rises and reaches its peak as shown in Figure 5.2b and then he puts his right leg ball of the foot, as top right sensor value rises at time t=2.7 s as shown in Figure 5.2a. Later the user puts his left leg ball of the foot part first as shown in Figure 5.2c as top left sensor reaches its peak at t= 3.3 s and at the end when he puts his left leg heel portion then the bottom left sensor reaches its peak at t= 3.7 s as shown in Figure 5.2d.

Figure 5.3 shows the combined figure of Figure 5.2 after standardization. Here in this figure, it is more clear how these four pressure sensor values can be used to capture the walking behavior of a user. As seen from the Figure 5.3, it can also be said that user puts his more weight on right side than left side and also as explained previously how the user kept his legs. Such information would be useful to classifier because everyone has their unique style of walking and hence can predict a person based on this behavior.

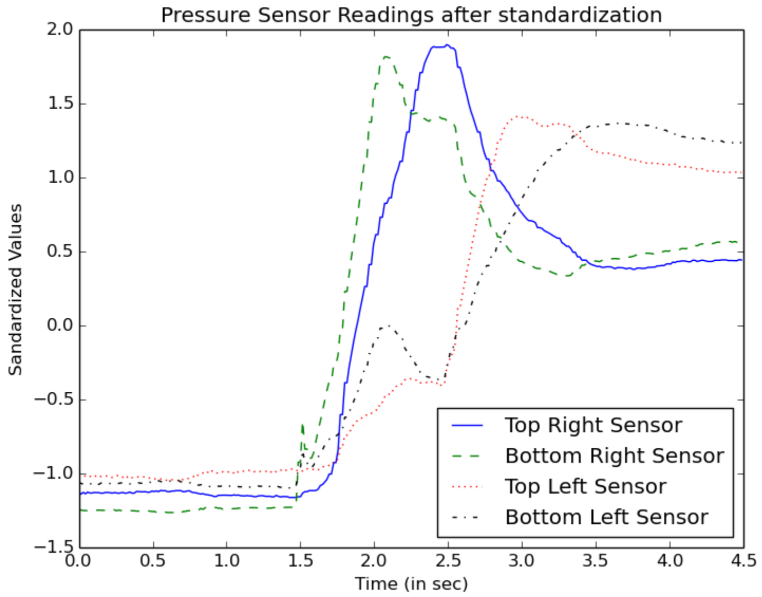


Figure 5.3: Pressure sensor values vs Time graph after standardization.

5.2.1 Weight Calculation

According to [18], the weight of each sensor can be obtained by interpolating between two calibration values where the reading value falls between. If the reading value, exceeds the calibration value then extrapolate the value to obtain weight. The total weight can be obtained by summation of weight of each sensors.

For example, according to [21], let one sensor value be :

$$right\_top \quad 3618 \;\; [2293, 4004, 5725]$$

The reading value is 3618 which falls between the $0kg$ calibration value 2293 and $17kg$ calibration value 4004 as shown in Figure 5.4. So the weight of *right_top* (w_{rt}) can be calculated by interpolating as follows :

$$w_{rt} = 17 * (3618 - 2293)/(4004 - 2293)$$

$$w_{rt} = 17 * 0.7744$$

$$w_{rt} = 13.2kg$$

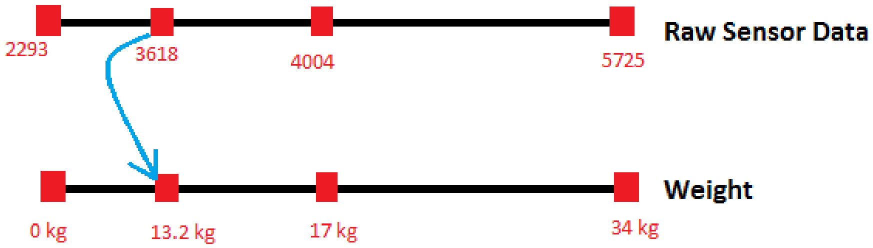


Figure 5.4: Interpolating weight from pressure sensor values.

Similarly, we can calculate for other sensors. The summation of weight obtained from all four sensors gives total weight. As shown in the Figure 5.5, the weight distribution is obtained. From the figure we can see that when the user starts stepping up on WBB, the overall weight increases and when the user is on the WBB, we can see that weight graphs tries to stabilize itself. The figure shows the total weight of a user i.e. summation weight of all four sensors and not the individual sensor weight.

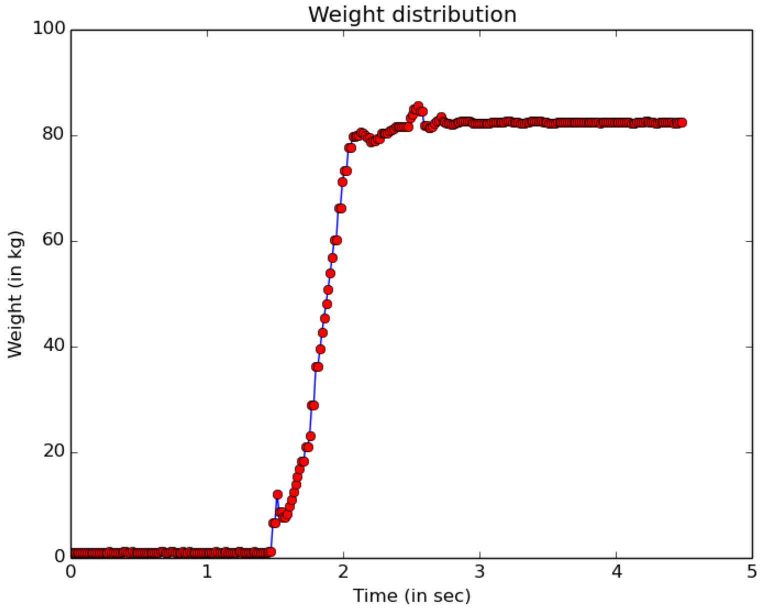


Figure 5.5: Weight vs Time graph. The value increases when user steps on and then stabilizes to the value equal to user's weight.

5.2.2 Center of Pressure (COP) Calculation

Center of Pressure (COP) is the point at which the ground reaction force vector is applied according to [22]. Center of Pressure for WBB can be calculated as follows [19]: Consider the center of the board as origin as shown in Figure 5.6. The length (L) of board is 433 mm and the breadth (W) of the board is 228 mm then

$$COP_x = \frac{L}{2}\frac{((TR+BR)-(TL+BL))}{TR+BR+TL+BL} \tag{5.1}$$

$$COP_y = \frac{W}{2}\frac{((TR+TL)-(BR+BL))}{TR+BR+TL+BL} \tag{5.2}$$

Here x indicates length among X-dimension, y is width among Y-dimension, TR = top right, TL = top left, BR = bottom right and BL = bottom left as shown in Figure 5.6.

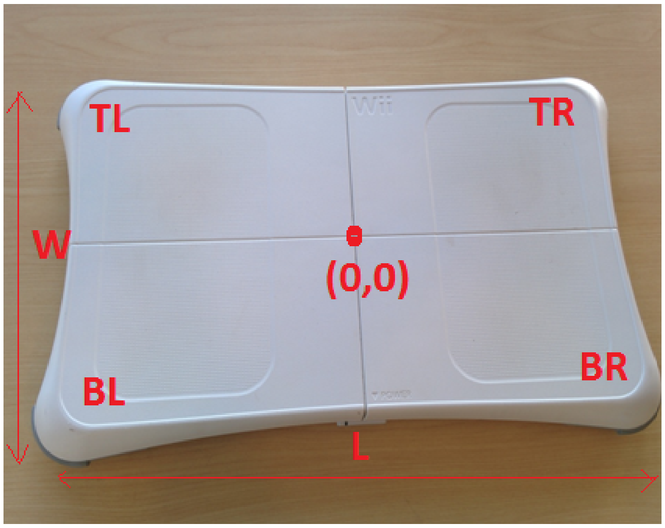


Figure 5.6: Wii Balance Board with the coordinate system.

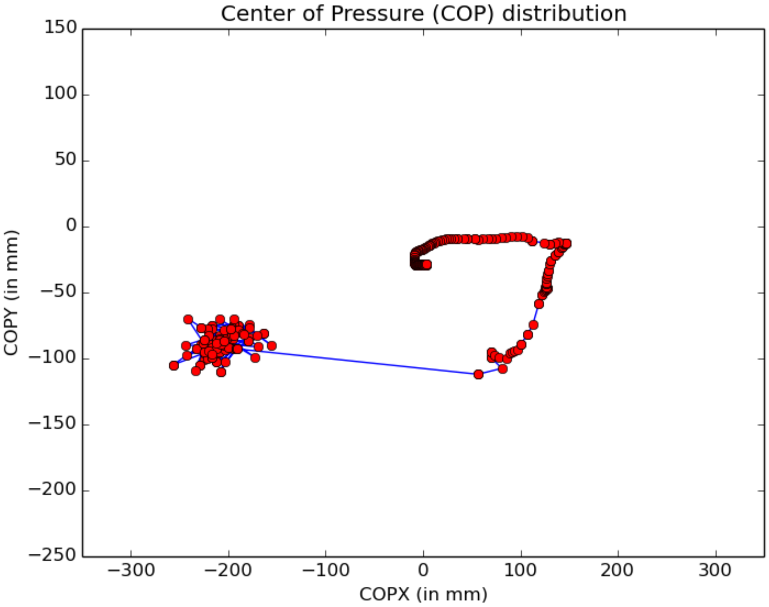


Figure 5.7: Center of Pressure Trajectory. The points in the left are default values due to calibration and the points in the right shows the COP trajectory of a user when he steps on WBB starting from approximate coordinates (50,-100) to (0,-50) where the value stabilizes.

Figure 5.7 shows the center of pressure trajectory. When a user is about to step-on the WBB, we do obtain some default values in the range of around -200 to -300 in

X-axis and -50 to -100 in Y-axis due to calibration and weight fluctuation in Wii which is in the left side of the graph. When user steps on WBB, the curve starts rising from -100 to 0 in Y-axis and 50 to 100 in X-axis. Now, when the user has put both of his legs, the graph stabilizes around the value of 0 in both axis. The graph can reveal useful information for classifiers as about how a user steps on WBB.

5.3 Why standardization?

Standardization of data is an important step in machine learning algorithm. It is because it makes each features in the data behave like normally distributed data with zero mean and unit variance. If we have a feature which has wide range of values then the feature distance will be governed by this particular feature. If the values are not normalized, then change in the higher valued feature would have an impact on entire prediction algorithm. The weight of that particular feature would be considered more than the other features and a minor change in that feature value would lead to unexpected recognition. As seen from the Figure 5.2 we can see that all four sensors are in different range and so does weight and COP. Also, we calculated the cross validation performance on data without using standardization method and found that the performance is 62% for 19 people using SVM linear kernel type. So, we would be using standardization approach in all our study.

5.4 Algorithm Implementation

In our study, we are using Support Vector Machine (SVM), k-Nearest Neighbor (kNN) and Random Forest (RF) as our recognizer and comparing the results of them. Our application is developed in python and we are using scikit machine learning library for SVM [23], kNN [24] and RF [25].

The seven feature arrays obtained from the experiment were concatenated in the following sequence respectively TR, BR, TL, BL, WT, COP_x and COP_y. The dataset is standardized using scikit standardization library [26]. Later, the dataset is provided to multi class classifier for prediction. RF does not require standardization.

5.4.1 SVM

Support Vector Machine (SVM) is a group of supervised learning algorithm. It can be used for both classification and regression. SVM is based on Vapnik - Chervonenkis (VC) dimensions introduced by Vladimir Vapnik and Alexey Chervonenkis. The goal of SVM is to achieve a large separation boundary among the classes. Therefore it is also known as Large Margin Classifier. SVM uses a nonlinear mapping to transform the original training data into high dimensional space. SVM construct hyperplanes in an high dimensional space. The hyperplane is decision boundary separating classes. During classification, we are given some data points belonging to different classes and for new datapoint we should find the class it belongs to. In this scenario, SVM select p-1 dimensional hyperplane for p dimension data points to find the maximum separation. Now, there would be many hyperplanes which might classify the data. The goal is to select the best hyperplane which has maximum separation between these classes. The complex computations in the dimensional space can be avoided by using a kernel function, which allows computations to be performed in the input space [27].

5.4.1.1 SVM 'linear' kernel

A standard SVM is a type of linear classification using dot product. The linear type kernel performs well if number of features are large compared to size of the data. In the SVM with kernel 'linear', we can tweak just one parameter which is known as C - the cost factor or penalty factor. C is also known as soft margin. The soft margin allows some examples to be ignored or placed on the wrong side of the margin. When C is very large the algorithms become very sensitive to outliers. If C is large, we would have overfitting situation. Also large value of C means high variance, low bias. On the other hand, if C is small, the algorithm won't be sensitive to outliers. Small value of C indicates low variance, high bias.

5.4.1.2 SVM 'rbf' kernel

In 1992, Vapnik et al. proposed a way to model more complicated relationships. Their proposed method replaced dot product with a nonlinear kernel function for example Radial Base Function (RBF), etc. RBF kernel is particularly used when we cannot find a linear solution to certain problems. By using RBF, we can map the data points into higher dimension space. By doing this, the separation can be easily achieved [28]. It almost fits all the data even the ones which has nonlinear relation between classes and attributes. There might be a problem of overfitting sometimes if not used correctly. This recognizer is similar to kNN but, here all the points have a vote. The weightage of each vote is determined by Gaussian in the following manner: The points which are nearer gets more vote than the points which are farther away.

For SVM kernel type 'rbf', we can tune two parameters: 1)C which is known as penalty factor as explained earlier and 2) gamma also known as kernel coefficient. Gamma parameter defines how far the influence of a single training example reaches. It controls the peaks of the points. Large value of gamma indicates that feature vary smoothly. It would have high bias and lower variance. But for small value of gamma it would have low bias, high variance. The best combination of C and gamma can be obtained by using GridSearch [29].

The RBF is the popular choice of kernel types in SVM. This is because of their finite responses and localization across the range of values.

5.4.2 kNN

K- Nearest Neighbor (kNN) is a non-parametric and instance based learning algorithm. kNN can be used for both classification as well as regression. The word non-parametric means that it does not make any assumptions on the underlying dataset. This is very important as in practical, real world data set, these theoretical assumption might not work. Instance based learning method or lazy algorithm is the one which does not make any generalization from the training point. All computations are deferred until classification. So they are very fast in training phase. Also it differs from SVM since it keeps all training data during testing phase instead of discarding non support vectors just like SVM. Since it keeps all the training data points, the testing phase is costly both in terms of time and memory [28].

kNN assumes that the training examples are vectors in a multidimensional feature space. The class labeled is associated with each of these vector. During the training phase, the algorithm consists of storing feature vectors and labels of the class. The principle behind k nearest neighbor method is to look for training samples which are near to the new data points and predict the class label based on the nearest neighbor. Here 'k' decides the number of neighbors to be selected near the new data point and is the influence for classification [30].

If k = 1, then the algorithm is nearest neighbor algorithm. In this case, the algorithm finds a point nearest to the new data point and labels the new data point with the same class as nearest point class. But it works only when the data points are not very large and the error rate would be almost near to twice the Bayes error rate. If for k = k, then the algorithm works on similar fashion but it tries to find the k nearest neighbor and performs majority voting. The k is generally selected to be odd so for k =7 if class A has five instances and class B has four instances near to the new training set then it assigns new training set as class A [28].

The value of k is data-oriented. So changing the position of few training points might lead to poor performance. The other consideration for the choice of k is that, the small value of k will have higher noise influence on the result. A large value of k makes computation cost more. Therefore the value of k should be chosen which is not very high and not very low. The value of k is optimized by taking many trials on training and validation set [31]. To improve kNN, instead of assigning one vote to all neighbors, weighted kNN can be used where weight of each point is calculated based on its distance.

5.4.3 RF

Random forest consists of a diverse set of decision trees. Its an ensemble technique used for both classification and regression, introduced by Breiman [32]. The trees are constructed by randomly selecting a subset of the training samples (bootstrap sample) for constructing each tree. Because of these bagging technique, random forest has low variance compared to a single deep decision tree but slight increase in bias is possible. Further, a random subset of the features set is selected at each split. These two things: random sampling of data and random selection of features adds randomization to RF and makes it better. At a given point each tree sees only part of the training sets and captures part of the information. Hence random forests are more robust to noise, fast, etc. [33].

5.5 Cross Validation

To evaluate the performance of our method, we are using cross-validation method [34]. The cross validation method is also used to avoid overfitting of the curve. The curve is said to be overfitted when the same dataset which is fed for training is used for testing purpose too. So to avoid such circumstances, certain data is used for training and certain data is used for testing (also known as holdouts) from a given dataset. Therefore cross validation is used. In general, the dataset is split into ‘k’ smaller sets. Estimation model is trained using ‘k-1’ of the folds as training data and the remaining part is validated to compute performance of a model.

5.5.1 Why Stratified k-Fold cross validation?

According to [35], Kohavi analysed different accuracy estimation methods using different datasets. It has been shown that ten-fold stratified cross validation is the better method to use for real world datasets. We therefore tested our dataset using various methods such as leave-one-out, stratified k-fold cross validation and bootstrap. Various parameters are changed for all these methods to find the best result. 171 samples are provided and SVM linear kernel is used as classifier. The Table 5.1 shows the result.

Methods	Parameters	Accuracy
Leave One Out	-	82%
Stratified K-Fold	n_fold = 9	82%
Bootstrap	train_size = 0.8	78%

Table 5.1: Accuracy results for different cross validation method and Bootstrap.

Table 5.1 shows that both Leave-One-Out and Stratified k-Fold both has same performance result. But leave-one-out method has high variance and its time consuming. On the other hand, k-fold stratified method is generally a better method both in terms of bias and variance. Also it is a standard method of evaluation.

Stratified k-fold is a variation of k-fold where mean response is equal in almost all folds. In other words, each fold contains approximately equal percentage of samples for each class [36]. We are using Scikit Stratified tool for estimating the classifier's accuracy. We varied number of folds of stratified k-fold for our dataset with SVM 'linear' kernel classifier. The result score is shown in the Figure 5.8. For k-fold = 9, higher accuracy can be obtained. Therefore we would be using nine-fold stratified method as our estimation tool.

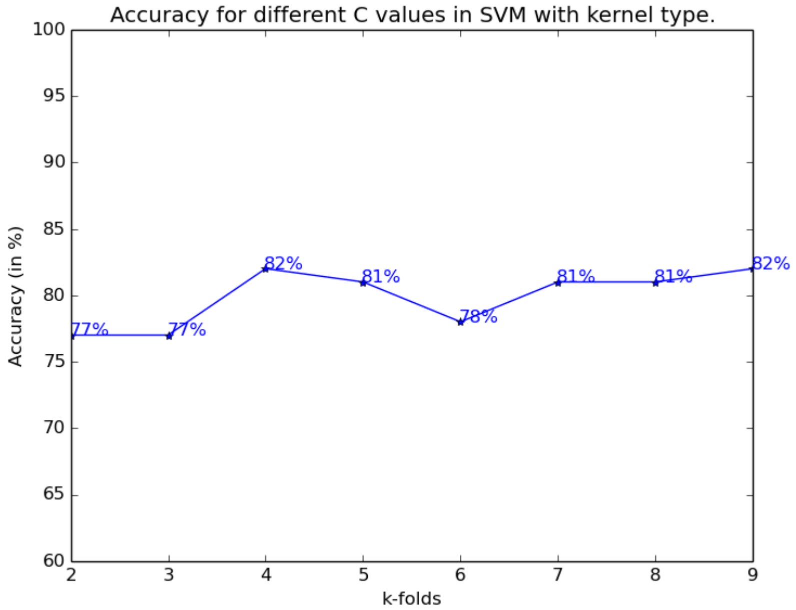


Figure 5.8: Accuracy of Stratified k Folds for different value of k. For k = 9, 82% accuracy is obtained.

In the summary, we saw basic architecture of our data collection model and testing tool. Later we explained about the features that we obtained from WBB and usage of these features in our research. SVM, kNN and RF are explained in brief. In the end, we discussed about why cross validation and k-fold stratified cross validation method to compare performance of classifiers by changing different parameters.

Chapter 6

Results

In this chapter, we provide the results of our study for different multi class classifiers such as SVM, kNN and RF. We tweak different parameters for each of these classifiers to obtain best performance estimation.

We are using nine-fold stratified cross validation method and scikit cross validation helper function to evaluate the performance of our prediction model. By using the scikit library, we obtain the scores for our classifiers. The scores list out whether the given data is predicted correctly. If the value is 1 or nearly 1, the data or rather testing set is predicted correctly. If 0 or nearly 0 then the recognizer failed to predict the test data. By taking the mean of the scores, we can calculate number of correct predictions divided to total number of observations. This is known as accuracy expressed in percentage.

6.1 SVM

SVM takes certain data as inputs known as training dataset and builds a model based on the classes assigned. Later, when the new dataset is provided it tries to assign new example into one category or other. We are using C-Support Vector Classification library of scikit as our recognizer. In the following section, different parameters that can be tuned for different kernel is explained along with the results obtained using nine-fold stratified cross-validation method.

6.1.1 SVM 'linear' kernel

SVM with kernel type linear has only one parameter which can be tuned, namely C. C is known as soft margin cost function which controls the influence of each individual support vector. The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. The linear type kernel performs well if number of features are large compared to the size of the data. Figure 6.1 shows accuracy obtained for all other parameters set to default and changing only C. 82% accuracy is obtained for C = 0.1 and higher values of C.

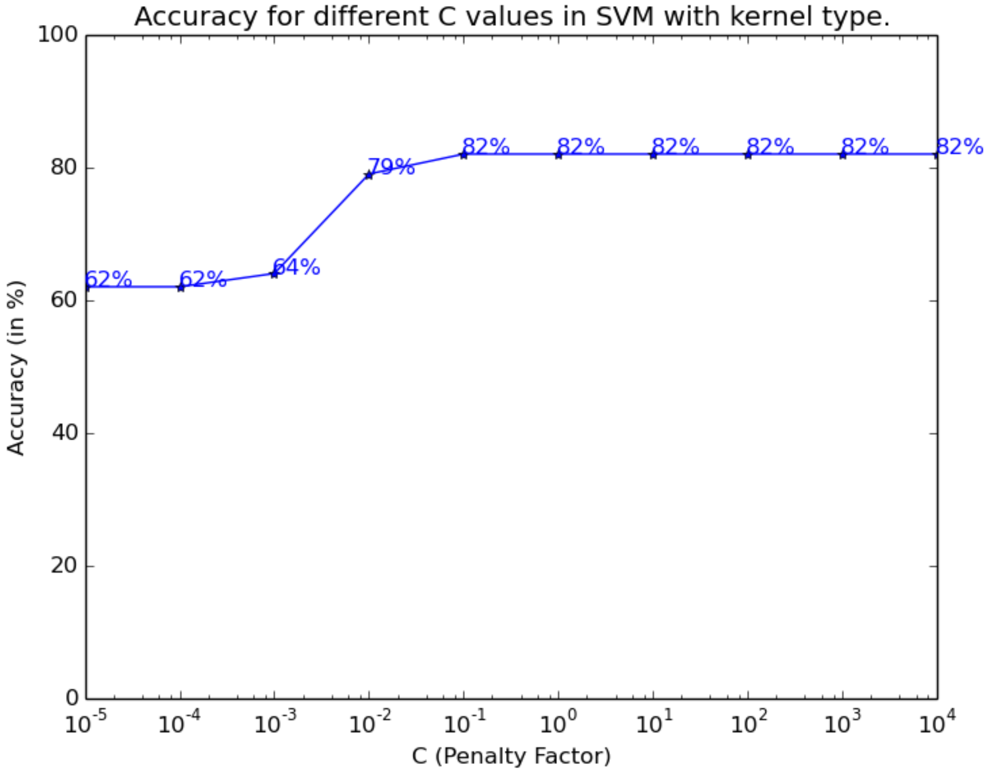


Figure 6.1: Accuracy for different C values of SVM 'linear' kernel when using seven features. Here C is a cost function or penalty factor. For C = 0.1 and higher, 82% accuracy is obtained.

6.1.2 SVM 'rbf' kernel

Radial Basic Function (RBF) kernel is chosen to be first reasonable algorithm to test with. The kernel maps the nonlinear attributes into higher dimensional space [29]. The RBF kernel is not suitable when number of features are very large compared to the dataset. In that case, it is best to use linear kernel. The following Figure 6.2 shows the performance when changing different values of C and gamma. C is the penalty factor while gamma is known as coefficient of 'rbf'. Larger C values means high variance and low bias. While the small values of C means low variance and high bias. If gamma is large, we get high bias and low variance and vice versa for small values of gamma.

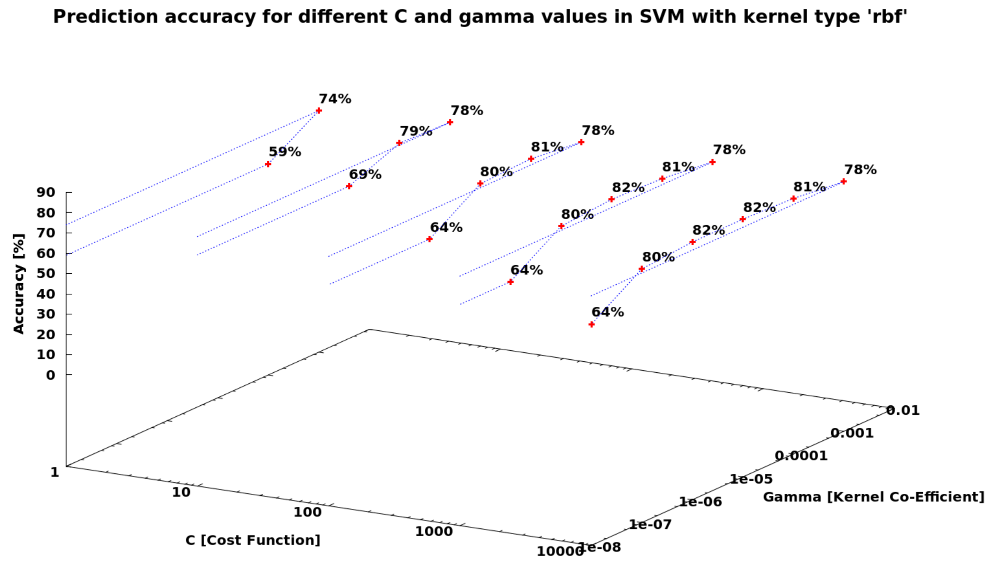


Figure 6.2: Accuracy for different C and gamma values of SVM 'rbf' kernel when using seven features. Here C is the cost function. Gamma is a kernel coefficient of 'rbf'. 82% accuracy can be obtained for C = 1000 and gamma = 0.00001.

As seen from the Figure 6.2, X-axis is represented by C, the cost function. Y-axis shows gamma values and Z-axis shows the accuracy obtained for various combination of C and gamma. When C = 1000 and gamma = 0.00001, 82% accuracy can be achieved. The accuracy of 80% is achieved when C = 10 and gamma = 0 and also for many other combinations of C and gamma, same accuracy can be achieved.

6.2 kNN

In our study, we are using kNN using scikit library and tuning various parameters to obtain the reasonable performance. We observed that for our datasets, if we change the parameter "algorithm" from 'ball_tree' , 'kd_tree' , 'brute' or 'auto', there is no change in the performance characteristics. The parameter "algorithm" decide which algorithm to use to compute nearest neighbor. But if the parameter known as "weight" is changed from 'uniform' to 'distance', there is a significant change in the performance. Here weight = 'uniform' means uniform equal weight for all the points in the neighbor while weight = 'distance' means weight points by inverse of the distance. By using this parameter close neighbors would have greater influence than the farther ones. For parameter "p", we noticed that it has some impact on the performance but not the major effect. Parameter "p" is known as Power metric. For p =1, it uses Manhattan distance but for default p=2, it uses Euclidean distance. Therefore in our study, we are using parameters weight= 'distance', p = 1, and algorithm = 'auto'. For different values of *n_neighbors*, following accuracy score can be obtained as shown in the Figure 6.3. For *n_neighbors* value greater than twenty, the performance was decreasing and hence not included in the figure 6.3.

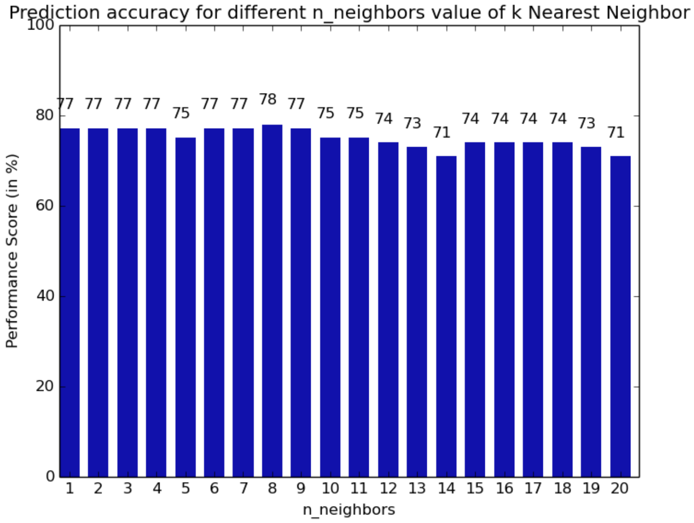


Figure 6.3: Accuracy for different *n_neighbors* value of k-Nearest Neighbor when seven features are used. 78% accuracy is achieved for $n\_neighbors = 8$.

6.3 RF

In our study, we are using random forest classifier from scikit library. Two parameters: $n\_estimators$, which is number of trees in the forest and $max\_features$, which is number of features to look for best split, can be adjusted. If $n\_estimators$ is more, the accuracy would be more but computation cost would be more. Better results can be obtained for $max\_features = \sqrt{n\_features}$ when used for classification according to scikit. Here $n\_features$ is equal to number of features provided to classifier. For all other parameters set to default and $max\_features$ = "sqrt", good performance can be obtained. So, we would be using $max\_features$ = "sqrt" for all other calculations.

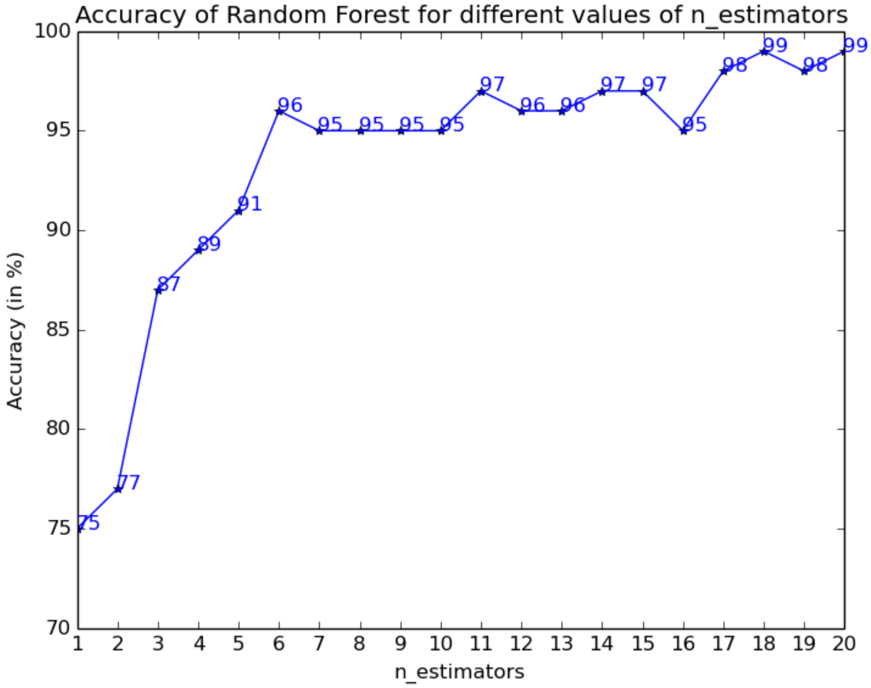


Figure 6.4: Accuracy for different $n\_estimators$ in Random Forest when seven features are used. Here $n\_estimators$ is equal to number of trees in the forest. Figure shows that for $max\_features$ = "sqrt" and $n\_estimators = 11$, 97% accuracy is obtained.

Figure 6.4 shows that for $max\_features$ = "sqrt" and $n\_estimators = 11$, 97 % accuracy can be obtained and for all values of $n\_estimators \geq 20$, higher accuracy between 98% and 99 % can be obtained.

6.4 Considering only 3 features

Previously, we saw the accuracy of the recognizer using all the seven feature arrays. In this section, we would consider only three feature arrays, namely weight (WT), COP_x and COP_y. We are concatenating these feature arrays in the sequence WT, COP_x and COP_y. We obtained nine samples from each participants and therefore our dataset consist of 171 samples. We will provide this dataset to standardization method and then selecting the recognizer, we will perform nine-fold stratified cross validation to evaluate the performance of each recognizer. We chose three features because there is a variation of weight among people. Also, the COP distribution i.e the way a user steps on the WBB would differ from person to person even for those with the same weight. In addition, these three features inherently contain all the data from the four sensor values. Therefore for the purpose of dimensionality reduction and since these three features are derived from the four sensor values, we can work with three features.

6.4.1 SVM ‘linear’ kernel

Here the default parameters for SVM scikit library are used. By tuning the values of C different accuracy can be obtained as shown in the Figure 6.5. But as compared to seven features, the accuracy is decreased from 82% to 74%.

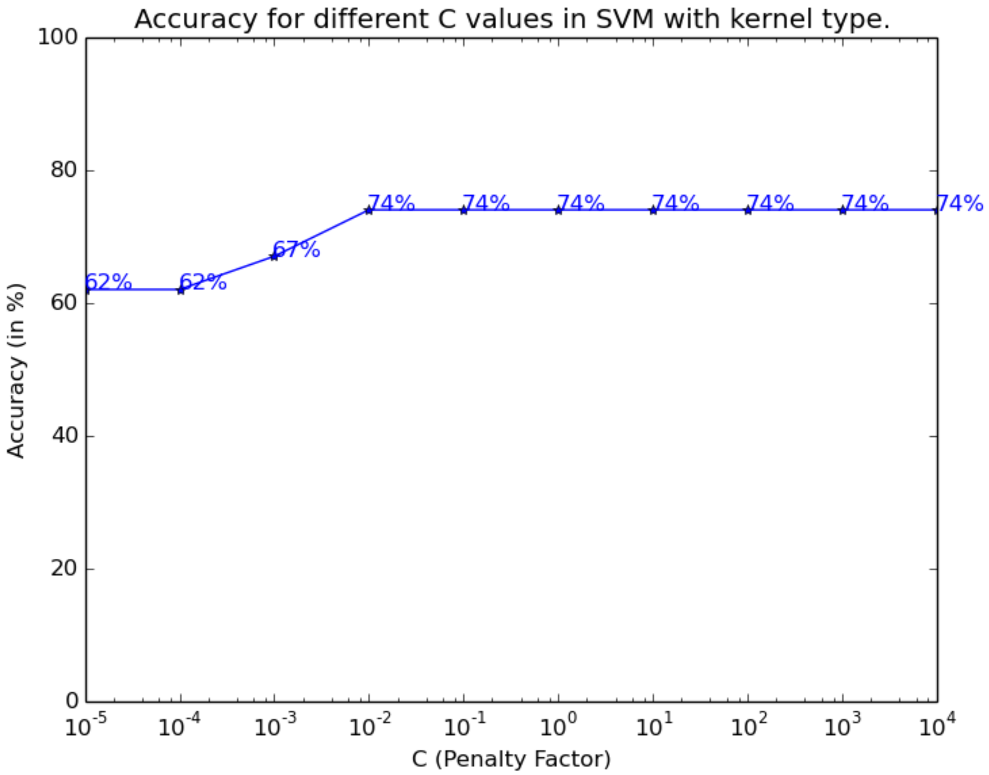


Figure 6.5: Accuracy for different C values of SVM ‘linear’ kernel when using three features. Here C is a cost function. 74 % accuracy is obtained for C = 0.01 and higher values.

6.4.2 SVM 'rbf' kernel

We are checking SVM with 'rbf' kernel for accuracy estimation and model selection. The following Figure 6.6 shows the performance when tweaking different values of C and gamma.

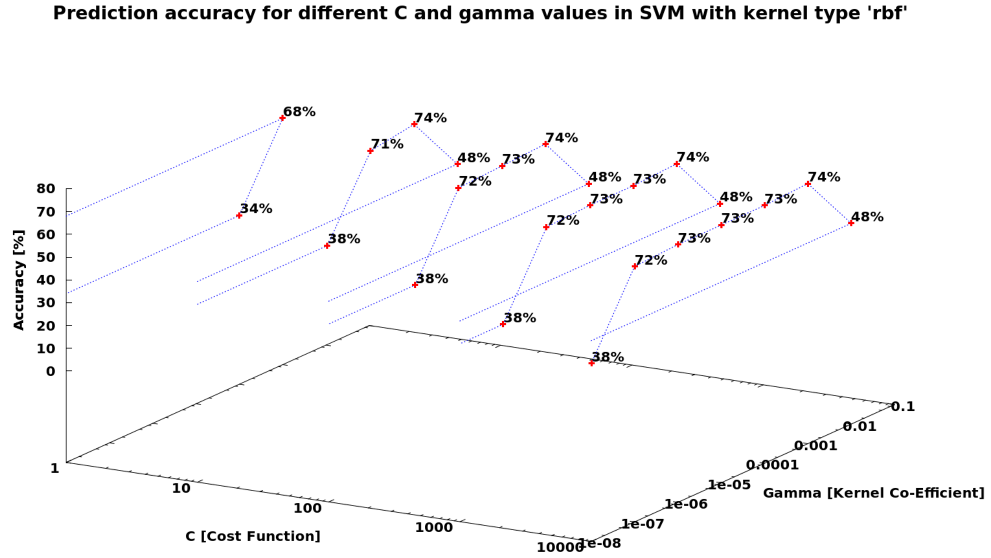


Figure 6.6: Accuracy for different C and gamma values of SVM 'rbf' kernel when using three features. Here C is the cost function. Gamma is a kernel coefficient of 'rbf'. 74% accuracy can be obtained for C =1000 and gamma = 0.001.

As seen from the Figure 6.6, accuracy of 74% can be achieved when C = 1000 and gamma = 0.001. Also the accuracy of 73% is achieved when C = 10 and gamma = 0 and for many other combinations of C and gamma.

6.4.3 kNN

For parameters: p = 1, algorithm = ‘auto’, weights = ‘distance’ and for different values of *n_neighbors*, the accuracy evaluation is as shown in the Figure 6.7. Figure 6.7 and Figure 6.3 shows similar accuracy but for different *n_neighbors* values.

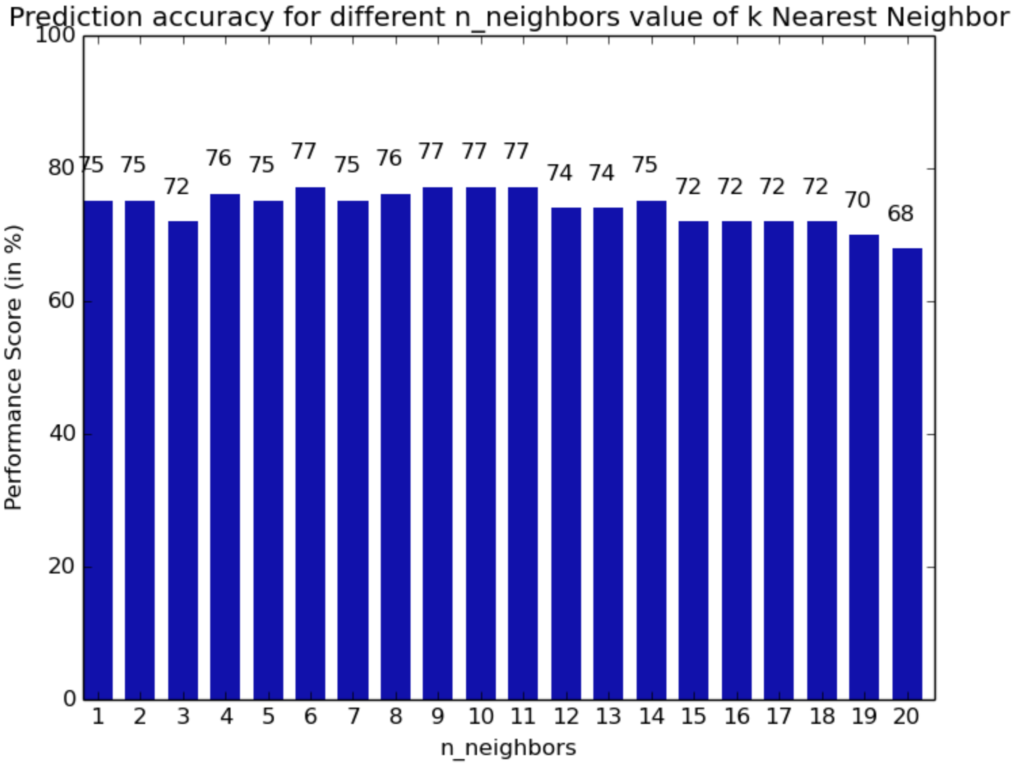


Figure 6.7: Accuracy for different *n_neighbors* value of k-Nearest Neighbor when three features are used. 77% accuracy is obtained for *n_neighbors* = 6.

As shown in the Figure 6.7, the accuracy of 77 % can be obtained for many values of *n_neighbors* = 6, 9, 10, 11. The accuracy for values of *n_neighbors* greater than twenty was decreasing hence its not included in the Figure.

6.4.4 RF

Figure 6.8 shows for $max\_features$ = "sqrt" and different values of $n\_estimators$, accuracy obtained for Random Forest when three features are used. Comparing Figure 6.4 and Figure 6.8, it shows that random forest predicts better accuracy even with three features.

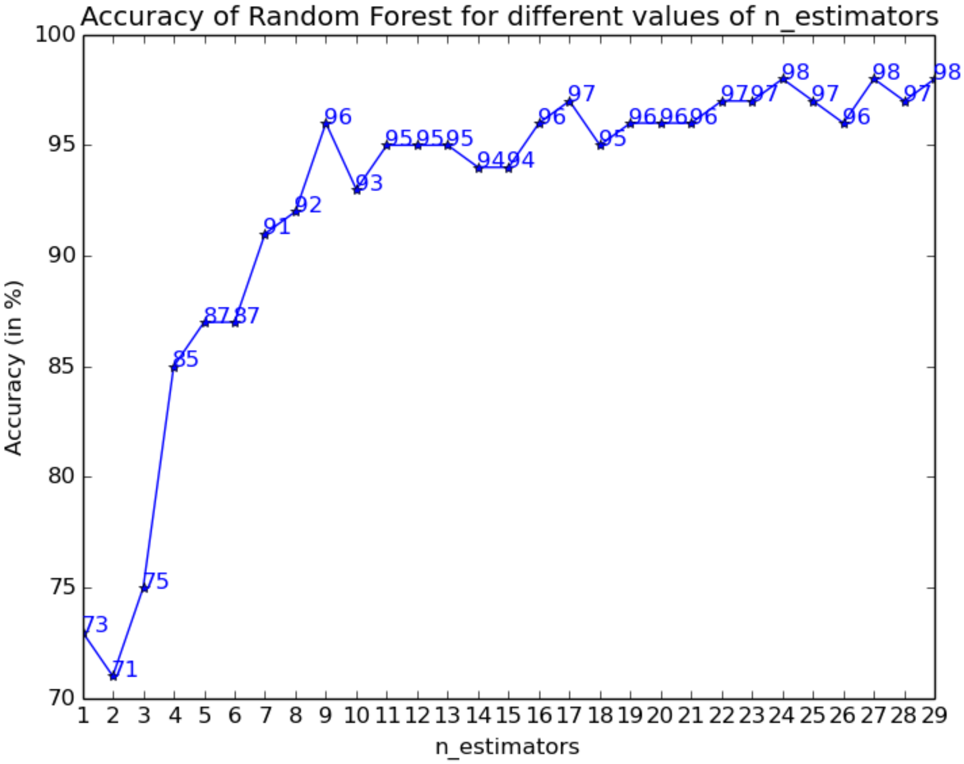


Figure 6.8: Accuracy for different $n\_estimators$ in Random Forest when three features are used. Here $n\_estimators$ is equal to number of trees in the forest. Figure shows that for $max\_features$ = "sqrt" and $n\_estimators = 9$, 96% is obtained.

For comparison, all the classifiers along with their best accuracy results is provided in Figure 6.9. Random Forest is having the best accuracy of 97 % for our dataset.

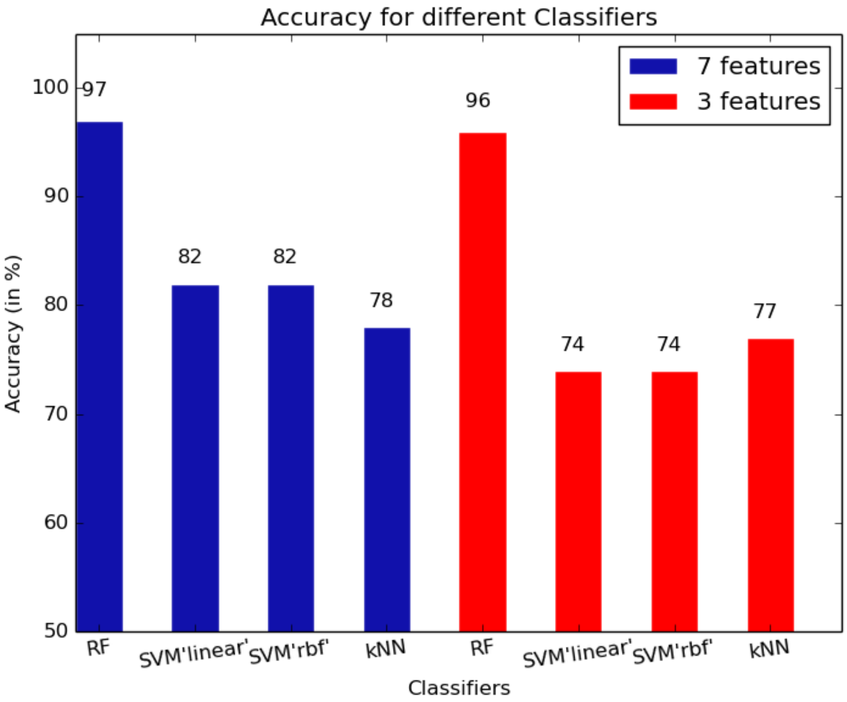


Figure 6.9: Accuracy of all classifiers. It shows that RF outsmarts other.

To summarize, we look over different classifiers and their prediction accuracy for our dataset of seven features and three features. We saw that Random Forest has 97% accuracy when we use seven features, followed by SVM kernel type ‘linear’. If we consider only three features, then also Random Forest has very high accuracy of 96% followed by kNN.

Chapter 7

Discussion

In this chapter, we discuss the results.

As shown in the Figure 6.1, 82% accuracy can be obtained for C = 0.1. C is the cost function which controls trade-off between achieving a low error rate to minimising norms of the weight. The optimized value of C for the given dataset would be C = 0.1 because for higher values of C, there is no change in the performance but it will overfit the curve. While smaller values of C, the performance decreases since it selects wider margin hyperplane. Therefore, allowing few misclassification which leads to low performance.

Figure 6.2 shows the performance of SVM 'rbf' kernel. For C = 1000 and gamma = 0.00001, 82% accuracy can be achieved which is equal to SVM 'linear' kernel. Also it can be seen that for many different combinations of C and gamma, same performance can be obtained. But C = 1000 and gamma = 0.00001 is optimized value with less bias and variance than the other combinations. The value of C and gamma can be obtained from Gridsearch.

Figure 6.3 shows performance of kNN for different values of $n\_neighbors$. The performance was not that good compared to SVM 'linear' kernel type. But with the help of general rule, $n\_neighbors$ value can be selected, which is equal to $\sqrt{N}$ where N is the number of classes. But for given dataset and $n\_neighbors = 8$, 78% accuracy is obtained. Also, increasing the value of $n\_neighbors$ beyond twenty, the performance started decreasing. The accuracy for values of $n\_neighbors$ greater than twenty is not shown in the Figure.

When using three features, the performance of SVM 'linear' kernel and SVM 'rbf' kernel decreased as shown in Figure 6.5 and in the Figure 6.6 respectively. On reducing the dimension, the classifier would have missed certain important training parameters which are required so that the classifier can improve their accuracy. Therefore the accuracy is lower compare to accurcy of SVM classifier when using seven features. But the accuracy for kNN, for three features is almost same as that of seven features. It can be argued that for more data points, the accuracy of kNN is not as good as with less data points. But further study might be needed to thoroughly understand this behavior.

Random Forest in the Figure 6.4 and Figure 6.8 gives the best performance for both seven features and three features. It is expected because it's an ensemble learning algorithm and suitable for unbalanced and missing data. Also, it is not sensitive to outliers. To be assured that the curve is not overfitting, 60 %, 70 %, 80 % of dataset is provided as training data to RF and for all these conditions, 97% accuracy is obtained but for different values of $n\_estimators$.

In RF, number of tree are constructed using random bootstrap samples of data and nodes are split using random subset of feature set. This may be counterintuitive but turns out to be better than other classifiers [32]. The system was also checked using Decision Tree and 88 % accuracy can be obtained. In RF, number of trees = 9 or 11 are used and $\sqrt{Number of features}$ is used, therefore, the performance of RF is better than Decision Tree.

Figure 6.9 shows performance of all classifiers used in seven features and three features. RF gives best result followed by SVM. And for less number of features, again RF gives best result but kNN outperforms SVM.

To summarize, we discuss the results obtained. From our study, we could obtain performance of 97% for 19 volunteers using Random Forest. The system design is simple and is user friendly.

Chapter 8

Conclusion

Person identification is crucial in various applications. The technology of human identification based on footprint has been emerging. In our study, we demonstrated the possibility to identify people based on their steps.

We designed the system to obtain the footprint of a person using Wii Balance Board. We can obtain weight and center of pressure trajectory when a user steps on the balance board. The challenge was to design a system, based on just these features, which can be useful for identifcation. SVM, kNN and RF classifiers are used as a recognizing tool and different parameters can be tuned to obtain the best result. GridSearch is used to find the best parameters and high accuracy of the classifiers. We obtained dataset from 19 volunteers and a total of 171 samples. Performance evaluation was done using nine-fold stratified cross validation. We obtained 97% accuracy using Random Forest. This indicates the potential scope of person identification through footstep using WBB.

The advantages of using WBB is because it is portable, robust and cheap. The system design using WBB is simple and does not require complex computational resources. Also it does not require any high-end devices such as high-resolution camera etc. The user has to just step on balance board which is much more convenient than other biometric based recognition. The direct application of our research can be to make Wii smart enough to identify a user and set his profile history. Beside this, it can be used as a doormat for household applications to prevent intruders.

References

[1] Simon Liu and M. Silverman. A practical guide to biometric security technology. *IT Professional*, 3(1):27–32, Jan 2001.

[2] A.K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, Jan 2004.

[3] K. Nakajima, Y. Mizukami, K. Tanaka, and T. Tamura. Footprint-based personal recognition. *Biomedical Engineering, IEEE Transactions on*, 47(11):1534–1537, Nov 2000.

[4] Jin-Woo Jung, Z. Bien, Sang-Wang Lee, and T. Sato. Dynamic-footprint based person identification using mat-type pressure sensor. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 3, pages 2937–2940 Vol.3, Sept 2003.

[5] Jin-Woo Jung, T. Sato, and Z. Bien. Dynamic footprint-based person recognition method using a hidden markov model and a neural network: Research articles. *Int'l J. Intelligent Systems*, 19:1127–1141 Vol.19, Nov. 2003.

[6] Shijia Pan, Ningning Wang, Yuqiu Qian, Irem Velibeyoglu, Hae Young Noh, and Pei Zhang. Indoor person identification through footstep induced structural vibration. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 81–86. ACM, 2015.

[7] G. Chakraborty, T. Dendou, D. Kikuchi, and K. Chiba. How much information could be revealed by analyzing data from pressure sensors attached to shoe insole? In *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, pages 1963–1967, May 2012.

[8] Robert J Orr and Gregory D Abowd. The smart floor: a mechanism for natural user identification and tracking. In *CHI'00 extended abstracts on Human factors in computing systems*, pages 275–276. ACM, 2000.

[9] I. Pappas, T. Keller, and M.R. Popovic. A Novel Gait Phase Detection System. In *Automatisierungstechnische Verfahren für die Medizin*, pages 69–70, Darmstadt, Germany, February 1999.

[10] Anuradha Annadhorai, Eric Guenterberg, Jaime Barnes, Kruthika Haraga, and Roozbeh Jafari. Human identification by gait analysis. In *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, page 11. ACM, 2008.

[11] Masato Miyoshi, Kentaro Mori, Yasunori Kashihara, Masafumi Nakao, Satoru Tsuge, and Minoru Fukumi. Personal identification method using footsteps. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 1615–1620. IEEE, 2011.

[12] L. Middleton, A.A. Buss, A. Bazin, and M.S. Nixon. A floor sensor system for gait recognition. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, pages 171–176, Oct 2005.

[13] Steven E Jones and George Kuriakose Thiruvathukal. *Codename revolution: The Nintendo Wii platform*. MIT Press, 2012.

[14] L. Pivnickova, V. Dolinay, and V. Vasek. Evaluation of static posturography via the wii balance board. In *Control Conference (ICCC), 2014 15th International Carpathian*, pages 437–441, May 2014.

[15] J.-A. Gil-Gomez, J.-A. Lozano, M. Alcaniz, and S.A. Perez. Nintendo wii balance board for balance disorders. In *Virtual Rehabilitation International Conference, 2009*, pages 213–213, June 2009.

[16] Rene Baranyi, Rainer Willinger, Nadja Lederer, Thomas Grechenig, and Wolfgang Schramm. Chances for serious games in rehabilitation of stroke patients on the example of utilizing the wii fit balance board. In *Serious Games and Applications for Health (SeGAH), 2013 IEEE 2nd International Conference on*, pages 1–7, May 2013.

[17] C. Da Cruz Teixeira, M. Kulberg, and J. Cavalcante de Oliveira. Virtual walking in a synthetic world through an low-cost interaction device. In *Virtual and Augmented Reality (SVR), 2013 XV Symposium on*, pages 212–215, May 2013.

[18] The wiibrew website: Wii balance board, 2012. Available: `http://wiibrew.org/wiki/Wii_Balance_Board`.

[19] Harrison L Bartlett, Lena H Ting, and Jeffrey T Bingham. Accuracy of force and center of pressure measures of the wii balance board. *Gait & posture*, 39(1):224–228, 2014.

[20] Harrison Bartlett, Jeff Bingham, and Lena H Ting. Validation and calibration of the wii balance board as an inexpensive force plate. *American Society of Biomechanics*, 1(2):3–4, 2012.

[21] Matt Cutts. Use a wii balance board with linux, 2009. Available: `https://www.mattcutts.com/blog/linux-wii-balanceboard`.

[22] Peter R Cavanagh. A technique for averaging center of pressure paths from a force platform. *Journal of biomechanics*, 11(10):487–491, 1978.

[23] Scikit Learn Developers. Svc. Available: `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`.

[24] Scikit Learn Developers. Kneighborsclassifier. Available: `http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`.

[25] Scikit Learn Developers. Random forest classifier. Available: `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`.

[26] Scikit Learn Developers. Preprocessing data. Available: `http://scikit-learn.org/stable/modules/preprocessing.html`.

[27] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[28] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 3rd edition, 2011.

[29] Steven M LaValle, Michael S Branicky, and Stephen R Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):673–692, 2004.

[30] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[31] Baoli Li, Shiwen Yu, and Qin Lu. An improved k-nearest neighbor algorithm for text categorization. *arXiv preprint cs/0306099*, 2003.

[32] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[33] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[34] Scikit Learn Developers. Cross-validation: Evaluating estimator performance. Available: `http://scikit-learn.org/stable/modules/cross_validation.html`.

[35] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[36] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. In LING LIU and M.TAMER ZSU, editors, *Encyclopedia of Database Systems*, pages 532–538. Springer US, 2009.