Performance Analysis of Different Fall Detecting Algorithms with Different Combinations of Sensors.

By

Rashed Mahmood Jhalak
16121040
Fariya Rahman Tonni
16121030
Ishaq Ibne Ibrahim
16121028
MD. Fazle Rabbi
16321119

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering

Electrical and Electronic Engineering
Brac University
January 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I/We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rashed Mahmood Jhalak 16121040	**Fariya Rahman Tonni** 16121030
Ishaq Ibne Ibrahim 16121028	**MD. Fazle Rabbi** 16321119

Approval

The thesis/project titled "Performance Analysis of Different Fall Detecting Algorithms with Different Combinations of Sensors" submitted by

1. Rashed Mahmood Jhalak (16121040)
2. Fariya Rahman Tonni (16121030)
3. Ishaq Ibne Ibrahim (16121028)
4. MD. Fazle Rabbi (16321119)

of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 13th of January, 2021.

Examining Committee:

Supervisor:
(Member)

____
Mohammed Belal Hossain Bhuian, PhD
Associate Professor, Department of Electrical and Electronic Engineering
Brac University

Program Coordinator:
(Member)

____
Mohammed Belal Hossain Bhuian, PhD
Associate Professor, Department of Electrical and Electronic Engineering
Brac University

Departmental Head:
(Chair)

____
Md. Mosaddequr Rahman, PhD
Professor and Chairperson, Department of Electrical and Electronic Engineering
Brac University

Abstract

Fall is one of the major reasons for the death of elderly people. Fall detection systems with different sensors based on different algorithms are now quite well admired. In this paper we analyzed the performance of different algorithms that can be used to detect fall. We used four different types of machine learning algorithms for this project. At first, we have created our own data with accelerometer and gyroscope separately and simultaneously. Then we used this data on each algorithm and found the accuracy rate. After that we added Magnetometer and compared the new result with the previous results and the threshold difference among these algorithms.

Our final result is which algorithm has the highest rate to detect fall comparing all the sensors individually and all together and we found SVM algorithm with using accelerometer and gyroscope together gives the highest accuracy of about 97%.

Keywords: Algorithm, Accelerometer, Gyroscope, Magnetometer, Threshold, Performance Analysis

Dedication

DEDICATED TO OUR PARENTS AND OUR SUPERVISOR, DR. MOHAMMED BELAL HOSSAIN BHUIAN, WHO GAVE HIS WISE INSTRUCTION TO COMPLETE THE PAPER.

Acknowledgement

In the first place, we would like to convey our gratefulness to almighty Allah for giving us the capability to complete the thesis paper.

Secondly, we are indebted to our supervisor Dr. Mohammed Belal Hossain Bhuian, Associate Professor, Department of Electrical and Electronic Engineering, Brac University, for guiding us with his excellence. After that we appreciate the support of all the faculties and staffs of Electrical and Electronic Engineering Department of Brac University.

Finally, we are filled with gratitude to our parents for the care, support and sacrifice. Because of them we reached the position where we are in now.

Table of Contents

List of Tables

List of Figures

List of Acronyms

MEMS	Microelectromechanical Systems
AdaBoost	Adaptive Boosting
SVM	Support Vector Machines
WEKA	Waikato Environment for Knowledge Analysis
CDC	Centers for Disease Control and Prevention
DTG	Dynamically Tuned Gyroscope
RTG	Ring Laser Gyroscope

Chapter 1

1.1 Introduction

In all over the world with the blessings of science and technology the life expectancy has increased a lot. In 2020 it is 72.63 years which is 0.24% more than 2019 [1]. Every year the rate is increasing. With the increasing life expectancy, the number of elderly people is increasing. According to United Nations, in 2020, the population is approximately 727 million who are aged 65 years or over. The percentage of older persons in world population in 2020 is 9.3 which will be increased by 16.0% by the year of 2050 [2].

In Bangladesh a small developing country of South Asia has life expectancy of 72.72 years in 2020 which was 38.55 years in 1950. In last 70 years it has increased by 34.17 years. If it increases with the same rate in next 30 years the life expectancy will be 79.24 years [3].

People become less active, less responsive when they grow old. With the growing age of this large amount of population the possibility of facing any kind of accidents or health problems grows simultaneously. They face different kinds of accidents and get injured. Fall is one of the major reasons for elderly people to get injured badly. 30% of the older people aged 65 years or more who are living in the community and 50% of those who are living in nursing homes or in personal residential care facilities fall every year [8].

Due to fall incidents a lot of people die every year. From 2007 to 2016 the gross rate of death of elder people increased 31% (3.0% per year) [5]. Fall is behind of 70% accidental death who are over 75 years old. Statistics also show that when a person fall it harms more if there is delay in assistance and treatment rather than the direct hit of fall [13].

Monitoring adult people are necessary for this reason. But 24 hours manually monitoring is quite impossible because family members need to go outside for livelihood. It is very expensive for most of the family to provide 24 hours nursing facilities. So, there is a need of such system that can effectively monitor elderly people and in need can notify the family members or relatives near them. Fall detection is one of the solutions to reduce this kind of accidents.

1.2 Definition of fall

If a person loses balance and collapse from a higher to a lower place is called fall. Different researchers have different definition for it. A suitable definition of a fall is "Unintentionally coming to the ground or some lower level and other than as a consequence of sustaining a violent blow, loss of consciousness, sudden onset of paralysis as in stroke or an epileptic seizure." [6]. Fall is an unintentional occurrence; it happens when a person's center of gravity is displaced for any reason.

1.2.1 Classification of falls

There are some risk factors that indicates whether the chances of fall of a person is less or more. Including -

1. Intrinsic risk factors

In this factor they are included who are 65 years old or more who has less mobility, persons with incurable diseases, sight problems, bone frailty, who are addicted leads inactive life etc.

2. Extrinsic risk factors

These factors are environmental factors. For example, slipping floors, stairs, damaged roads, clouded places, poor lighting etc. [4]

Different researchers have divided fall in different ways on the basis of their research works. Four major categories of falls are [9]:

1. Falls related to extrinsic factors (55%)

2. Falls related to intrinsic factors (39%)

3. Falls from a non-bipedal stance (8%) (someone is not on his/her two feet)

4. Unclassified falls (7%)

There is a lot of way a person can fall. Based on the scenario of fall occurrences, falls can be divided into four types:

1. Fall from sleeping (bed).

2. Fall from sitting (chair).

3. Fall from walking or standing on the floor.

4. Fall from standing on supports such as ladder, tool.

Fall from standing on supports happens less for elderly people because most of them are retired or don't do this type of work at this age. The other three types of falls occur more for them. According to Yu the elderly and patient are mainly at risk by the first three classes of falls. From all of the falls mentioned above the chances of the head is in free fall is very high. That is why it can be very threatening for elderly people [7].

1.2.2 Consequences of falls

When one falls the person has no control over his body and for that reason, he faces injuries. As elderly people's physical strength is poor, they cannot avoid circumstances of unwanted fall. Fall is the first reason for death who are over the age of 75 years, second reason of death for them who are between 65 and 75, and sixth for the people who are 65 years old or over. According to CDC (Centers for Disease Control and Prevention) every year 36 million elderly people falls and from them more than 32,000 dies [11]. From falling there can be head injury which can damage brain tissue badly. In 70% of incidents fall leads to the upper and lower limb injuries including fractures (60%), superficial injuries (21%) and open wounds (8%) [10]. Sometimes injuries are so bad that they loss mobility and this causes other health issues.

Fall can affect a person psychologically. The persons who have experienced fall once in a lifetime has a great fear of it. From a research it is found that in two years 219 out of 487 elderly had to undergo in a fall accident and one-third of them has developed a fear of falling after the incident [12]. It also creates the doubt of living independently. Most of the people has the tendency of falling 2nd time who have fall once.

When older people fall and their injuries are fatal most of the time. The cost of treatment is very high. Falls lead to a total healthcare cost of 474.4 million which represents 21% of total healthcare expenses due to injuries [10]. It can cause economical imbalance for those who are dependent on others mainly for elderly people. It also affects the financial condition of their families who bears the expenses of their treatment.

1.3 Fall Detection

If a person falls and a system or device can recognize the activity that is fall detection. Different scientific way has been applied for taking care of elderly and monitor them with some device or system. Fall detection system or device is one of them. For elderly people to live healthy and independently different researcher have found different ways to detect fall. Some studies showed hardware devices and some showed software systems. They used different sensors and found specific amount of accuracy.

Method of automatic fall detection can be divided into two sections. Instrumenting surroundings is the first section like fitting cameras in the rooms which can detect the person's movement. Second section is all about wearable devices with different sensors like gyroscope, accelerometer, magnetometer etc. [14].

Researchers have worked on this topic and found different results applying different methods. They used sensors like accelerometer, gyroscope, magnetometer, cardio tachometer etc. Machine learning algorithm are also very popular to detect falls.

Accelerometer is very well known for detecting falls. Most of the researchers uses tri-axial accelerometer and threshold algorithm. Using dynamic threshold-based method on smartphone Otanasap et al. [16] made a system that can detect fall 97.40% accurately. Depending on support Vector Machine (SVM) algorithm, Zhang et al. [15] made a fall detector by using one waist-worn accelerometer. It detected 96.7% falls correctly.

Tapia et al. [17] introduced a real time algorithm which will detect fall and sometimes can detect the intensity. He used five tri-axial wireless accelerometer and a wireless heart rate monitor. The accuracy was 94.6% using subject dependent training and 56.3% using subject independent training.

Bourke et al. [18] presented a threshold-based fall detection algorithm using a bi-axial gyroscope sensor and found 100% accuracy. Another researcher worked with accelerometer, gyroscope and magnetometer from a smartphone, which detected falls applying machine learning algorithms and brought 98% of accuracy [19].

1.4 Aims and Methodology of this thesis

The aim of this thesis is to analyze the performance of different fall detecting algorithms for different sensors. Accelerometer, gyroscope and magnetometer are the sensors we have worked on. For detecting fall we chose 4 different machine learning algorithms and they are Support Vector Machine (SVM), AdaBoost M1, Random Forest and Naïve Bayes.

To create our own sample data, we used smartphone. Data was taken from each sensor separately and also simultaneously. We took six types of data including falling, going downstairs, going upstairs, sitting down, standing up and walking. After that we processed the data as the output was giving by the phone was noisy. Next, we put that data in each algorithm with each sensor and also with combined sensor. After that we analyzed all the accuracy of each algorithm with the combination of sensors. Our final result is which algorithm has the highest rate to detect fall comparing all the sensors individually and all together.

1.5 Thesis Structure

Our thesis work has been divided into several chapters. Each and every chapter discussed in this way that it helps to achieve the goal of our research work. In chapter 1 we described how elderly people are living longer life and the necessity of a system to monitor them. After that the definition of fall with the classification and consequences, fall detection with related work has been included. Chapter 2 is demonstrating accelerometer, gyroscope, magnetometer and how it works in smartphones. Chapter 3 is providing the design of the experiment, possible errors and adjustments, data collection with dataset formation. Chapter 4 is describing the algorithms of machine learning and how they work with related flow chart. Chapter 5 is showing the main experiment and result of our research work. Here accuracy test has shown by both individual and combined dataset with each algorithm. Chapter ends with the final result with accuracy comparison. Chapter 6 concludes the study of the thesis and discusses the future work with strong remarks.

Chapter 2

2.1.1 Accelerometer

Accelerometer is a device that can measure different types of acceleration forces or vibrations. It was first invented by George Atwood in 1783. The first accelerometer is widely recognized as Atwood machine [20].

Accelerometer is an electromechanical device. It can detect any type of movement of the device with which it is connected. Within a specific frequency it calculates acceleration several times and summarize this as a count over a pre specified time period [21]. Accelerometer identifies either dynamic forces or static forces. Movements, vibrations, displacements etc. are the example of dynamic force. Gravity can be an example of static force.

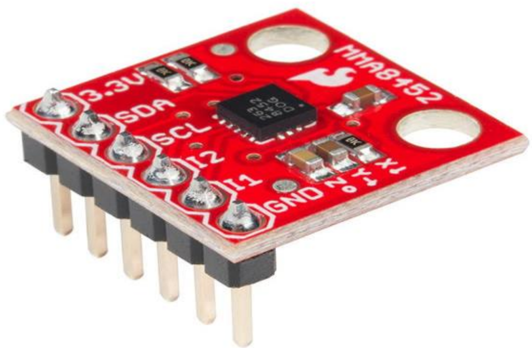

Figure 1: Accelerometer

Nowadays accelerometer is mostly used to identify physical activity in research work. Accelerometer can detect the amount of acceleration by sensing the change in motion and can also find the orientation if the device is not moving. It can determine the orientation for pitch and roll [22].

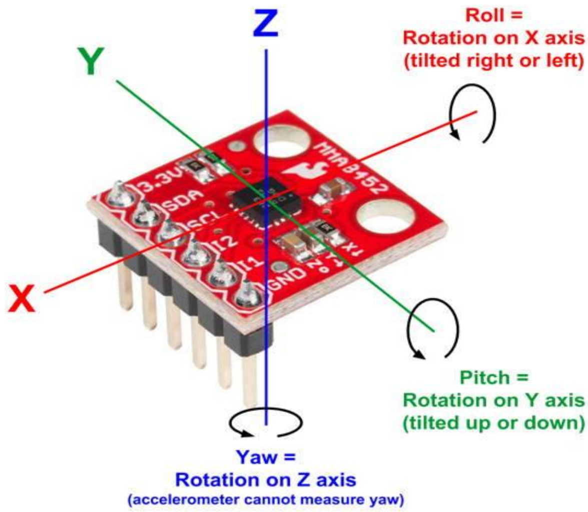

Figure 2: Orientation of 3 axes Accelerometer [22]

In fig 3 the orientation of a 3 axes accelerometer is shown. The rotation on the y axis is called pitch and rotation on the x axis is called roll. Pitch means the body tilted up or down and roll indicates the body tilted right or left [22]. Rotation on the z axis is called yaw. Accelerometer cannot detect yaw because of gravity.

2.1.2 Work Principle of Accelerometer in Mobile Phones

At present most of the smartphones has some features related to accelerometer. It measures linear acceleration of the smartphones. It regulates the screen orientation from landscape to portrait and vice versa for the users. Smartphones has built in 3-axes accelerometer which detects movements like shaking, tilting etc. Most of the cases smartphones uses MEMS based accelerometer.

In a smartphone acceleration sensor has two part. One of them has a signal processing chip and the other one is a micromechanical comb structure. The comb forms a capacitor and its capacity is determined with the distance between the microstructure. Microscopic crystal structures are moveable. They can change their structure or position on the basis of acceleration. The integrated electronic circuit can identify the result in change in capacitance as a transformed measured value and we get an output as a voltage signal.

2.2.1 Gyroscope

It was in the year 1852 when the French physicist Jean-Bernard-Leon Foucault came up with a device to exhibit the rotation of the Earth. He named it the gyroscope. A gyroscope can be broadly defined as a solid body capable of rotating at high angular velocity about an instantaneous axis which always passes through a fixed point. The fixed point may be the center of gravity of the solid or it may be any other point [23]. It is used to understand the orientation of a body from the reference point alongside measuring the angular velocity.

As shown in Fig 8 is nearly frictionless rings called gimbals which essentially isolates the rotor in the center from external torque.

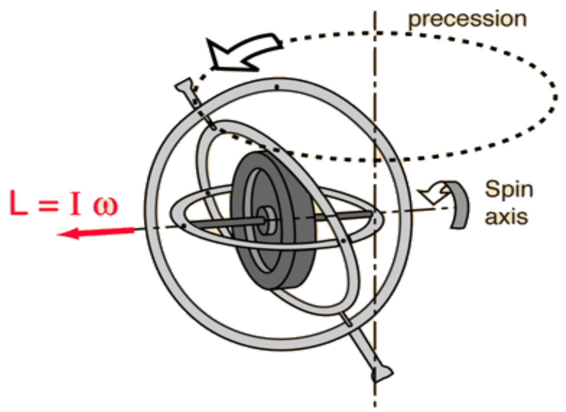

Figure 3: Gyroscope precession

If the gyroscope is tilted the gimbals will try to reorient the rotor to conserve its spin axis due to the external torque caused by Earth’s gravity. The magnitude and direction of angular momentum is thus maintained by the rotor [24].

Gyroscope thus is an excellent device to understand or study human motion. It can be used to differentiate between the regular daily activities and an instant of unwanted fall. This device is immensely used in the day-to-day life and varies according to the application.

Rotary gyroscope uses gimbals and rotor, a vibrating structure gyroscope use multiple mode of vibration to determine rotation about several axes. The optical gyroscope use light or laser beam to work out the physical orientation.

The possible potential of MEMS gyroscope outruns to that of any other types of gyroscope due to its small and compact structure. It works with the help of arms and central mass those detect small change in orientation in the three-dimensional space. These changes are the outputs that we get of the independent rotation about the three axes [24].

2.2.2 Gyroscope in Mobile phones

Steve Jobs was the pioneer to introduce gyroscopes in consumer electronics like mobile phones. In mobile phones gyroscopes are normally integrated with sensors like accelerometer, magnetometer for better and accurate performance. It is a vibration gyroscope sensor coupled with the MEMS technology used in smartphones.

Why MEMS gyroscope in mobile phones

Other options could be the dynamically tuned gyroscope (DTG) which is very similar to a mechanical gyroscope or the Ring Laser Gyroscope (RLG).

Advantages of MEMS gyroscope

The production cost of MEMS gyroscope is lot cheaper compared to its alternatives like the ring laser gyroscope or RLG. MEMS gyroscope has been under constant development over past years and are available in the form of chip. This also means they are reliable, accurate and unlike the DTG contains no moving components. However, another feature of the MEMS gyroscope is the compact structure which enables its widespread use in consumer goods like smartphones because it can be easily integrated in electric circuits.

The MEMS gyroscope is going through immense development as it is used for daily purposes as well as to serve military purposes. Also, in the controlling purpose of airplanes, rockets, etc. [25]

We used Samsung J7 pro smart phone for the gyroscopic data acquisition. Though the embedded gyroscope in mobile phones is lot cheaper compared to other use but they are highly accurate to serve our purpose of fall detection.

Working principle of mobile MEMS gyroscope

A symmetrical double-T structured crystal vibrates inside the sensor. The structure has a sensing arm in the middle and drive arms on both sides. When the mobile is moving the drive, arms moves in such a way that they balance one another and the sensing arms movement generate electric signals which can be classified to understand the movement [26].

Figure 4: Gyroscope data output from mobile

Data outputs from the gyroscope

Looking from above the horizontal rotation of mobile is known as the yaw. From upfront the vertical rotation gives the pitch value and the horizontal rotation gives the roll value.

Apart for sensing the angular velocity thus a gyroscope is very helpful to study the motion of an object. In our case monitoring the human daily activities.

For determining the angles from the raw gyroscope data, a simple integration of each of the independent outputs of yaw, pitch and roll over given time interval gives the result.

The data can be inaccurate due to several causes. The active elements in the MEMS gyroscope produce white noise which adds error in the output data. Though the deviation in precision caused by white noise is minimal. The bias offset error can cause significant change in output data. The sensor works best in definite temperature range and for instance if the mobile is too hot due to long usage hours, the output data can be inaccurate.

The error can be mitigated by following a number of steps

1. For the bias offset error, the gyroscope can easily be calibrated as adjusting the reference points to zero.

2. The Allan variance technique which deals with clustering the data from MEMS sensor output and frequency domain analysis of power spectral density to act as a filter can be used to increase the precision [27].

2.3.1 Magnetometer

Magnetometer is a devise that measures magnetic field of a particular location or the change of magnetic dipole of a particular device. It can be used in various as an independent device like a compass or it can be used in a device to take measurement of its magnetic dipole changes.

In fall detection magnetometer can be used in various ways to take measurements. In this experiment we used magnetometer from mobile phones. Most of the mobile phones have a sensor called hall effect sensor [28], this sensor is located inside a chip that detects voltage changes inside the sensor and feedbacks accordingly. Hall effect sensor is most commonly used sensor and its accuracy is more than 99% [29] this is why we selected smart phone's magnetometer because it is reliable, accurate, easy to obtain, widely used.

2.3.2 How Magnetometer in mobile phones works

In modern smart phones there is a sensor named hall effect sensor that is used as magnetometer. This hall effect sensor works on a principle named hall effect principle. When a conductor or semiconductor with current flowing in one direction is introduced perpendicular to a magnetic field a voltage could be measured at right angles to the current path. Using this principle, hall effect sensor can measure magnetic field. In hall effect sensor a thin conductor or a semiconductor has a current applied along it. When it comes to the presence of a magnetic field, it produces a voltage across the particular metal. The voltage that it creates is called hall voltage. [30]

The hall voltage is represented by the expression V_H . The mathematical expression for hall voltage is

$$V_H = \frac{I\,B}{q\,n\,d}$$

Where,

I – Current flowing through the Sensor

B – Magnetic Field Strength

q – Charge

n – number of charge carriers per unit volume

d – Thickness of the Sensor

Chapter 3

Methodology and Data Collection

3.1 Introduction

In conducting the experiment, the specific scenario and outcome we expect to get determines the working principle. However, it is really important to fix the structure we need to follow to achieve the intended goal.

3.2 Design of the experiment

The design for the experiment is very simple. We used the smartphones built in sensor module to fetch the required data. After analyzing several mobile apps from play store, we decided to use Androsensor a free app that seemed reliable and precise enough to serve the purpose. The mobile was held above thigh by one hand where we normally have the pockets. The data were collected from young to middle aged people as it was impractical to collect falling data from old people what might cause them severe injury.

Figure 5: Experiment design

All raw data that we need to detect the fall is the accelerometer, gyroscope and magnetometer sensor outputs. Before using this data, we need to adjust the readings and account for issues that may cause deviation of the obtained data from the actual data.

3.3 Data Collection

3.3.1 Sensor placement

The sensor needs to be in the best place where it can perfectly classify the fall and also comfortable for the user. We assume that the user needs to have the sensor attached to him or her for most part of the day and is able to walk on their own. Six positions in the body, the head, chest, waist, wrist, thigh and ankle are most suitable to detect fall using machine learning [31]. Using six different machine learning algorithms Özdemir and Turan et al. [32] found that the waist to be the most accurate place with an accuracy of 98.42% as it is much less prone to high acceleration compared to other parts.

Considering the long usage hours and presumably aged user, strapping a sensor in the user waist can be really tiring. Then again as we are using the sensor from the mobile it is most easy and tightly attached if carried in the user pocket. So, we choose the thigh to be the sensor position which has the second highest accuracy of 97.89% [32].

3.3.2 Considering human to human differences in physical activities and attributes

For the data collection a group of 7 people, 5 males and 2 females were involved. The physical attributes were different from one another. The age, gender, weight and height are as per given in the table below.

Table 1: Data collection of a group of 7 people

Gender	**Age**	**Weight**	**Height**
Male 1	23 years	65 kg	5 feet 4 inches
Male 2	25 years	82 kg	5 feet 3 inch
Male 3	27 years	72 kg	5 feet 8 inch
Male 4	43 years	54 kg	5 feet 11 inch
Male 5	37 years	59 kg	5 feet 7 inch
Female 1	32 years	54 kg	5 feet 1 inch
Female 2	26 years	67 kg	5 feet 3 inch

3.4.3 Dataset Formation

We collected data for possible ways of falling instances and on the other hand got data for the daily activities. We collected 150 datasets from the seven participants that can be used as the raw data in the machine learning algorithms.

The dataset had both similarities and differences which will be fruitful to differentiate fall from the daily activities.

Chapter 4

Algorithms and Machine Learning

4.1 Introduction

In this experiment there are two ways to detect fall with the data sets we have. One is with threshold technique and another is machine learning technique. Threshold technique is simpler and does not have complex computational work. It detects fall but lacks behind when it comes to sophisticated human movement because the rate of false positive increases. One the other hand, machine learning technique is more complicated has more computational works but it can detect fall even when it comes to complex human movement. In machine Learning technique we can train an algorithm to our way to detect fall and have less false positives. As the machine train itself the more data it gets the smarter it becomes and less errors we get. In this chapter we are going to see four different types of algorithms we used in our experiments:

1. SVM (Support Vector Machines)
2. Naïve Bayes
3. AdaBoost M1
4. Random Forest

4.2 SVM (Support Vector Machines)

Support Vector machines commonly known as SVM is relatively new type of algorithm. It actually works on two major classifications. One is Multivariable linear regression and classification regression. In SVM algorithm data sets are classified in two different type where some are true and rest are false. It tries to create a margin between these two classifications.

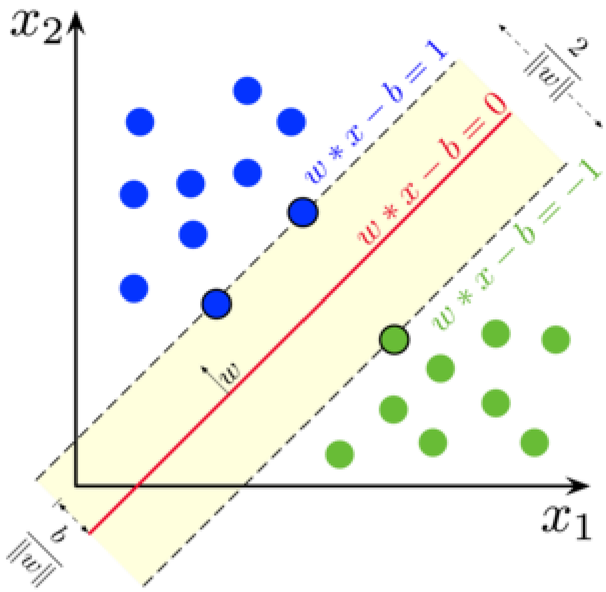

Figure 6: Linear SVM classification

SVM can be used to solve various real-world problems:

- SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and trunsductive settings [33]. Some methods for shallow semantic parsing are based on support vector machines. [34]
- Classification of Images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true for image segmentations systems, including those using a modified version SVM that uses the privileged approach as suggested. [35] [36]
- Classification of satellite data like SAR data using supervised SVM. [37]
- Hand-written characters can be recognized using SVM. [38][39]
- The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds

classified correctly. Permutation test based on [40] [41] Support-vector machine weights have also been used to interpret SVM models in the past. [42] Postdoc interpretation of support-vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.[43] [44]

4.2.1 How SVM Works

SVM algorithm has various steps from start of the machine to output. In SVM algorithm at first a machine is set to a certain condition where it starts. Then we input data in the algorithm that means algorithm acquires data. Which falls into data acquisition part. Then comes to the step where SVM needs to process the acquired data. After that it divides the data in to two parts one is radial basis kernel function where algorithm works on its kernel side to process the data another is super parameter initial value. Combining these two factors it reaches a general conclusion and gives a primary model. Once a simple primary model is created it start reputing input output parameters to check whether these is any error or not. After getting an initial error it responses accordingly by checking how much error it is getting. If the error is high it changes initial parameters and tries to find the new error. Once it reaches a minimum value of error set by the user it shows the final model output.

4.2.2 Flow Chart of SVM

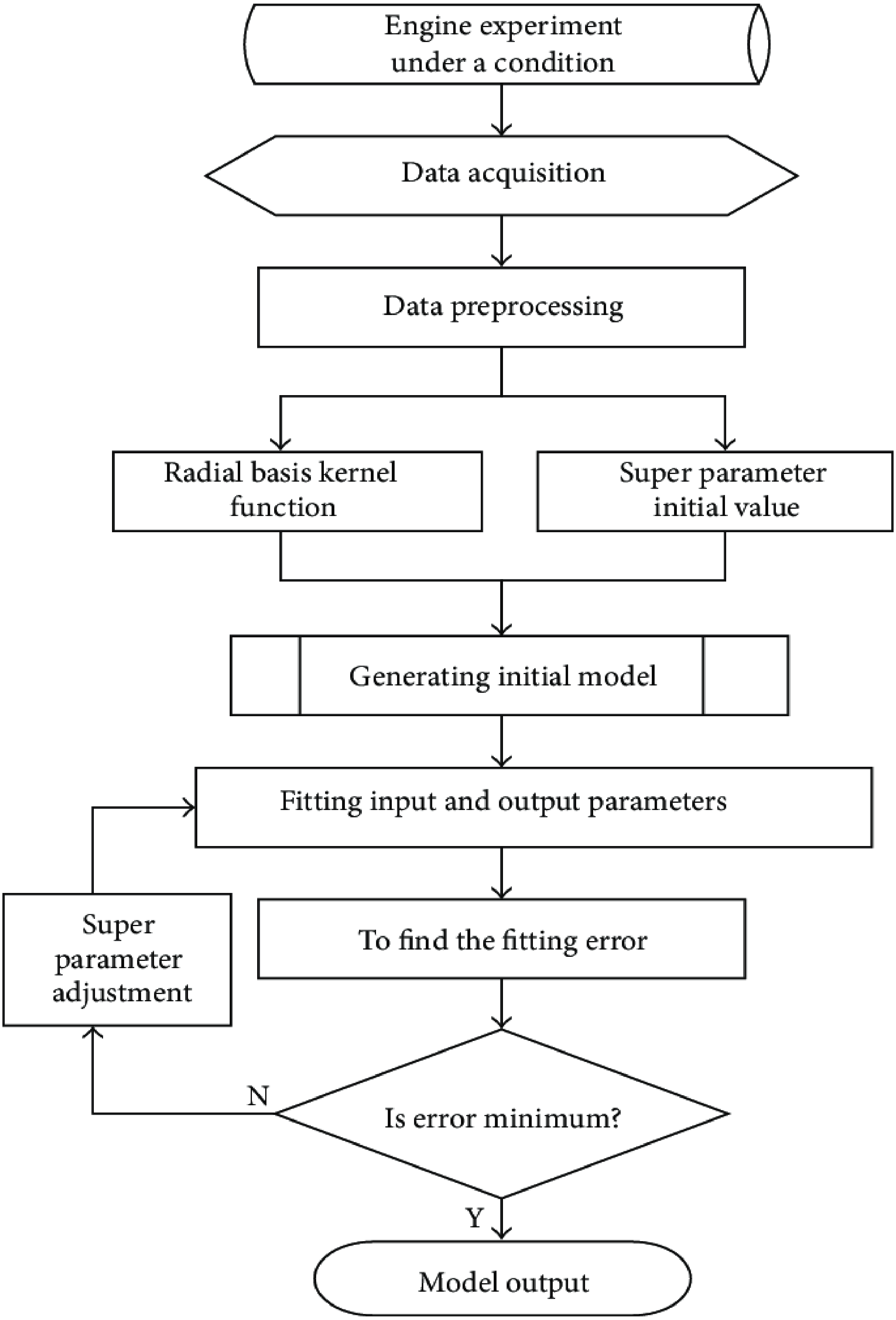

Figure 7: SVM Flowchart

4.3 Naïve Bayes

Naïve Bayes is based on Bayes' theorem. Naïve Bayes is a normal and simple way to create a classifier. This Classifier is not based on a single algorithm but a group of algorithms that follows a single principle. The main principle is to use Bayes' theorem to classify a problem and its solution based on its instances and features. It only requires a small amount of training data set to estimate a parameter necessary for classification which is a great advantage if we compare other algorithms with it. [45]

4.3.1 How Naïve Bayes Works

Unlike SVM, Naïve Bayes works a little differently. Support vector machine algorithm works on parameters where Naïve Bayes works on attributes. This algorithm works on a single attribute with all the instances at once. Then adds up other attributes to give the final results. At first it selects a particular attribute to begin with. Then it collects all the instances in that attribute. After collecting, it finds a probability to classify all the instances and reaches a primary conclusion. If it does not find other instances, it goes to next attribute and processes data accordingly. Finally adding all the attributes, it makes a final probability to classify all the attributes at a single time. In this process, this algorithm train itself to give a possible output with minimum error.

4.3.2 Naïve Bayes Algorithm Flow Chart

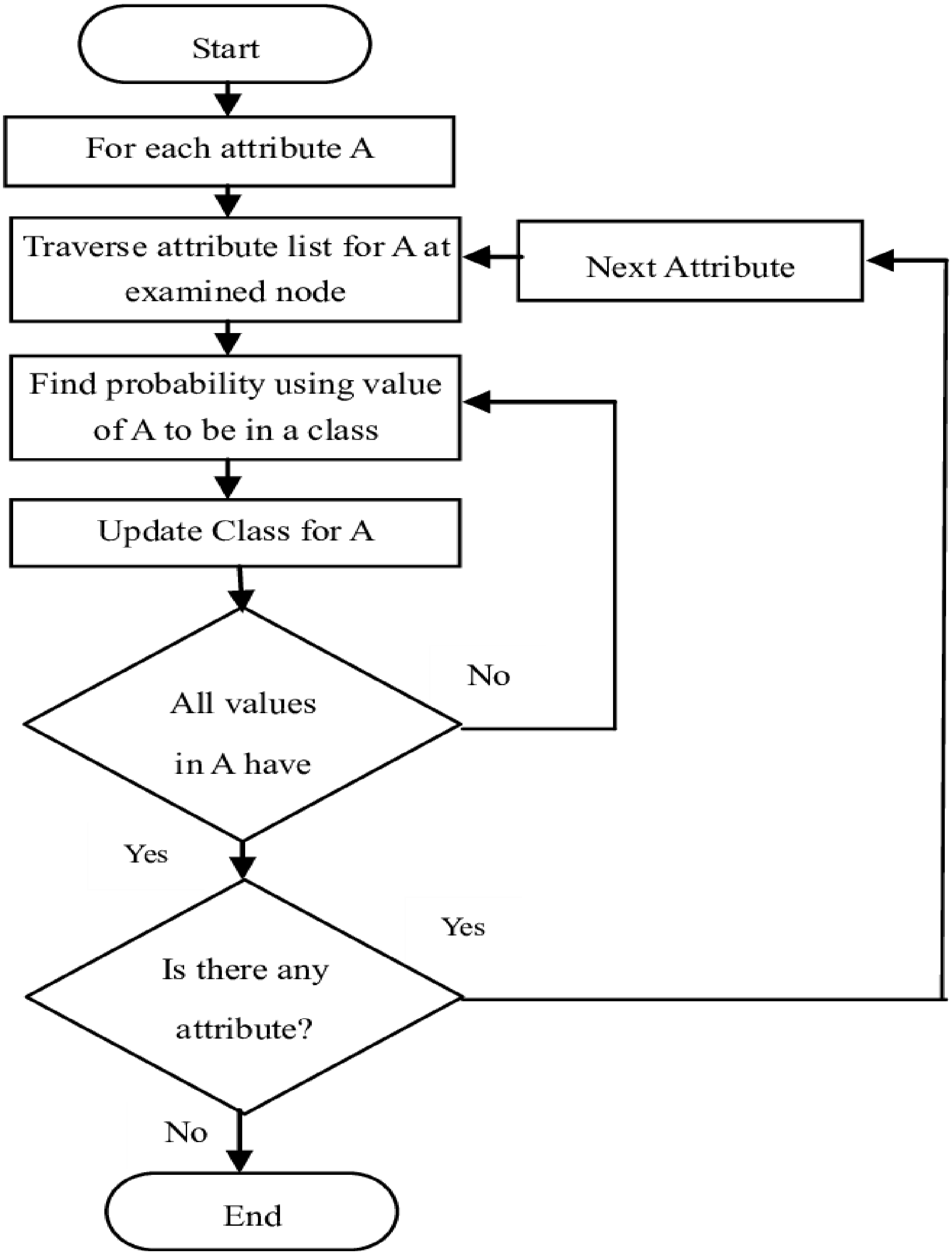

Figure 8: Naïve Bayes Flowchart

4.4 AdaBoost M1

AdaBoost is a meta-algorithm that can be used for boosts performance of the other algorithms. AdaBoost is also known as adaptive boosting. This algorithm can also work alone depends on what kind of work it is given to do. Every kind of algorithm has its own uniqueness and works best in particular cases. AdaBoost is a particular training method and it's a boosted classifier. [46]

4.4.1 How AdaBoost M1 Works

AdaBoost is a boosting algorithm. It takes the input data as samples. It takes an error value from the user and trains itself to distinguish the data samples with the error value. After giving it the data samples, it selects an optimal classifier to process the data. Once it comes to a point where it finds the suitable classifier it processes the data it was given. After that, it trains itself to minimize the error threshold. Until it reaches to a step where it can give the optimal results it repeats the whole process. [47]

4.4.2 Flowchart of AdaBoost M1

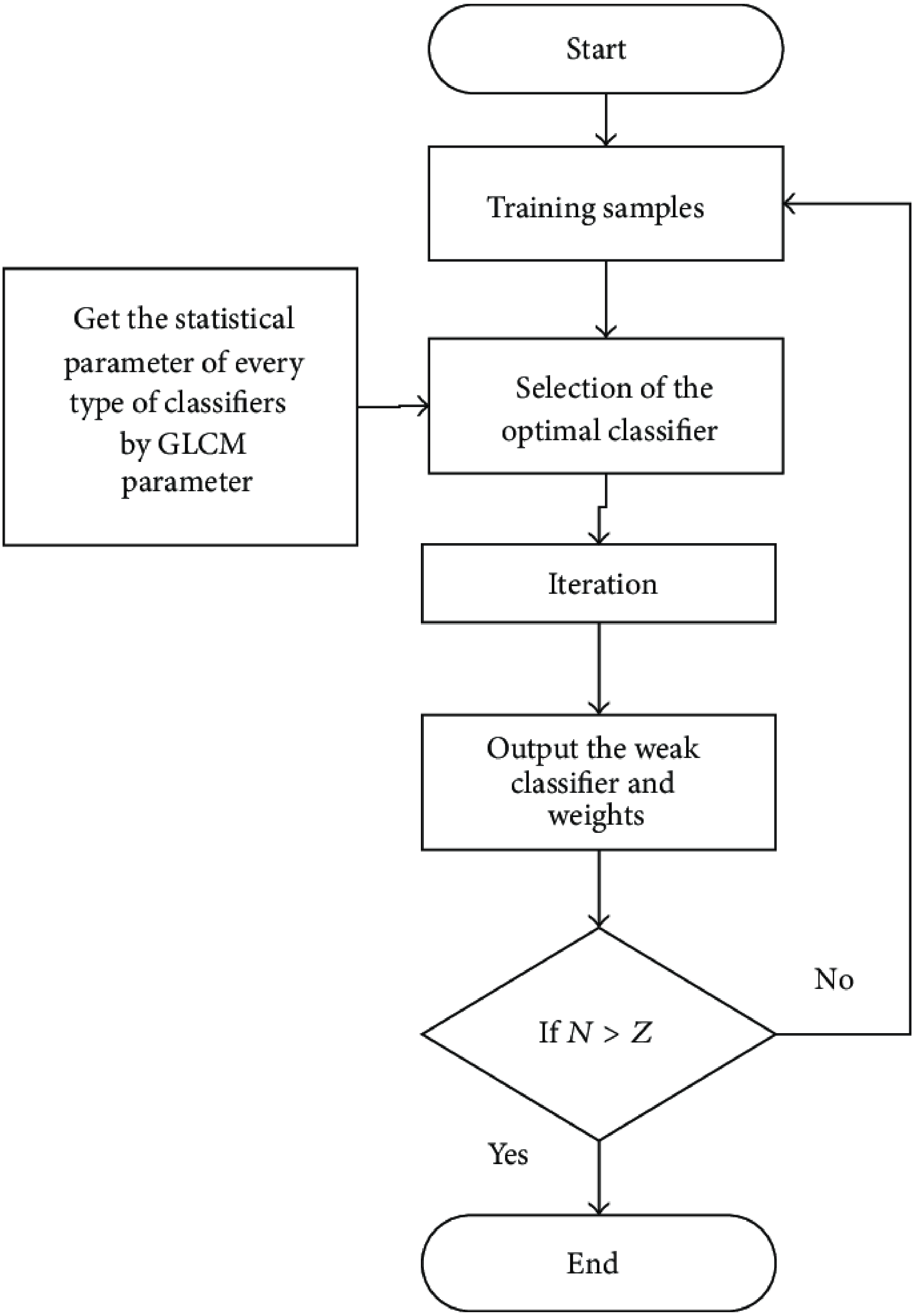

Figure 9: AdaBoost Flowchart

4.5 Random Forest

Random Forest also known as random decision forests is a machine learning method that works on classification and regression. It uses these two methods and also other tasks to construct decision trees and give output.

4.5.1 How Random Forrest Works

Decision trees are a popular method for various machine learning tasks. Tree learning "come[s] closest to meeting the requirements for serving as an off-the-shelf procedure for data mining", say Hastie et al., "because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. However, they are seldom accurate"

In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, i.e., have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

Forests are like the pulling together of decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree. Though not quite similar, forests give the effects of a K-fold cross validation.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. [48]

4.5.2 Random Forest Flowchart

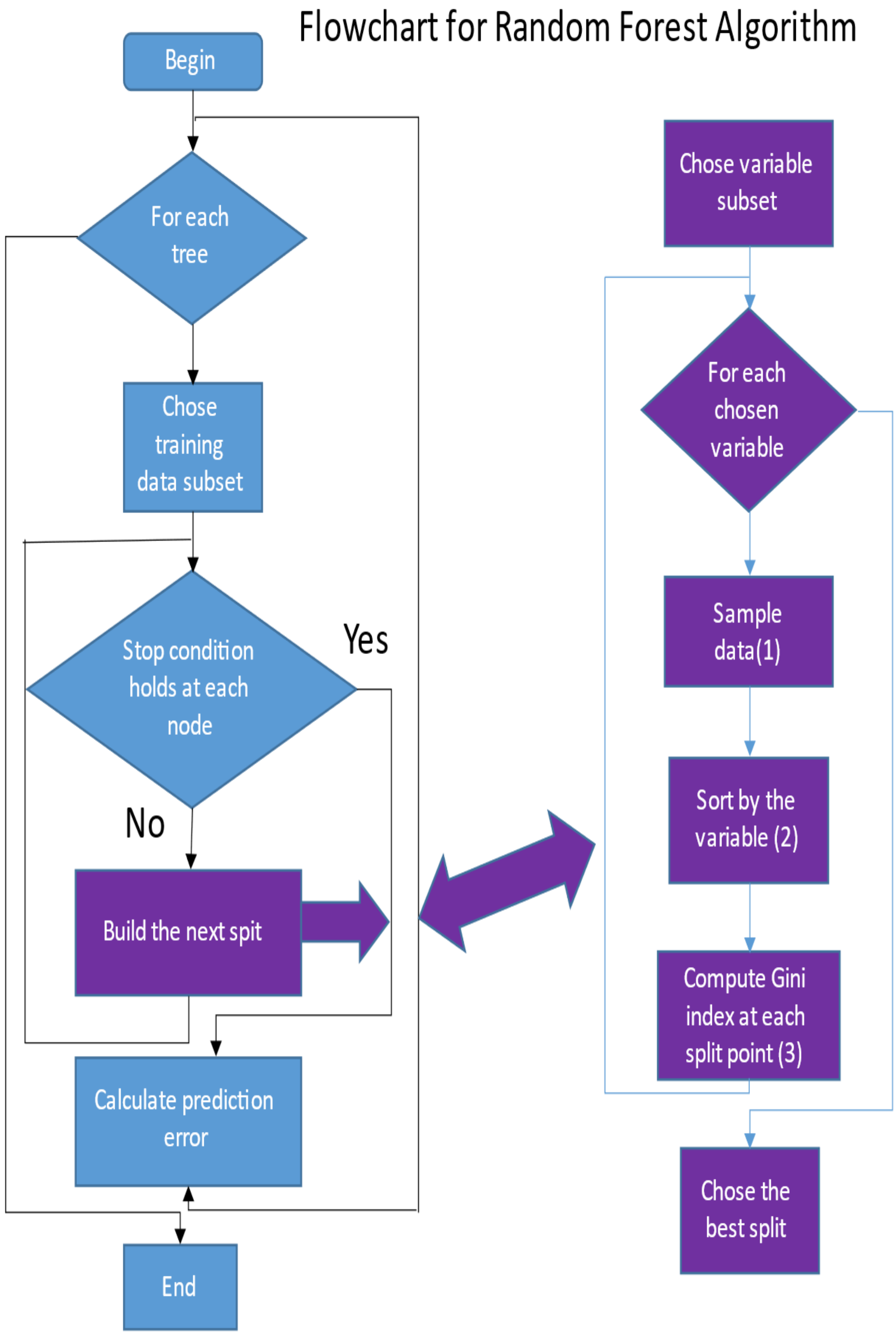

Figure 10: Random Forest Flowchart [49]

Chapter 5

Experiment & Results

We wanted to figure out if the machine can actually detect fall of a person from given data set and if it can how accurately can it detect fall from other physical postures. The thesis paper on Fall Detection and Activity Recognition with Machine Learning by Mitja Luštrek and Boštjan Kaluža we are following as reference [49] mentions 4 algorithms (AdaBoost M1 boosting, Naïve Bayes, SVM and Random Forest) to be most accurate. As for input, we collected first-hand data using mobile device from different physical postures, i.e., walking, running, falling down, going upstairs and so on. We used Accelerometer, Gyroscope and Magnetometer as data sensors to collect data regarding several body postures. We also included nominal values (True/False) and designated them to the sensor data so the machine may learn which instances of data to sort out. Then to test the accuracy of the aforementioned algorithms, we loaded the data onto WEKA (Waikato Environment for Knowledge Analysis) software and tested the data sets against the algorithms to find out which of them can detect fall properly among other postures.

We put the data from each sensor against the algorithms and then put all the sensor data together to measure accuracy threshold. Each sensor datasets are discussed in detail in the following sub-chapters.

5.1 Accuracy Testing

The datasets of three sensors, Accelerometer, Gyroscope and Magnetometer were first individually put into WEKA and then we merged datasets of two sensors and finally added all the data from all three sensors to compare the accuracy threshold. In each case, we identified which parts of the dataset were to be considered as Fall data so the machine can accurately detect fall from the other postures. The comparison between the sensor dataset according to the algorithms used are being listed in the following section.

5.1.1 Accelerometer Dataset

5.1.1.1 Using AdaBoost M1 algorithm

Table 2: AdaBoost M1 algorithm on Accelerometer

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	82(92.1348%)	7(7.8652%)	0.086	0.2582	45.3648 %	85.4866 %
2	79	75(94.9367%)	4(5.0633%)	0.0777	0.2302	45.3676 %	80.6013 %
3	87	81(93.1034%)	6(6.8966%)	0.0788	0.2492	50.2234 %	91.2287 %
4	90	86(95.5556%)	4(4.4444%)	0.0596	0.2022	44.4671 %	80.5638 %
5	95	90(94.7368%)	5(5.2632%)	0.0768	0.2391	42.8306 %	81.5065 %
6	81	76(93.8272%)	5(6.1728%)	0.0969	0.2525	47.0412 %	80.1761 %
7	79	72(91.1392%)	7(8.8608%)	0.1007	0.2651	52.6903 %	87.5576 %
8	88	82(93.1818%)	6(6.8182%)	0.0751	0.2418	43.2276 %	83.8635 %
9	85	81(95.2941%)	4(4.7059%)	0.0854	0.2384	47.6397 %	81.3778 %
10	96	91(94.7917%)	5(5.2083%)	0.0756	0.2341	46.9731 %	84.4632 %
11	98	87(88.7755%)	11(11.2245%)	0.132	0.3061	83.6252 %	111.489 %
12	88	82(93.1818%)	6(6.8182%)	0.0993	0.2629	63.8847 %	96.7269 %
13	79	74(93.6709%)	5(6.3291%)	0.1017	0.2487	59.3286 %	87.0712 %
14	81	75(92.5926%)	6(7.4074%)	0.1183	0.2752	63.1957 %	91.8923 %

15	89	85(95.5056%)	4(4.4944%)	0.0805	0.2254	31.4365 %	63.6562 %
16	78	74(94.8718%)	4(5.1282%)	0.0994	0.2413	37.1615 %	66.7174 %
17	85	76(89.4118%)	9(10.5882%)	0.1012	0.2816	43.506 %	83.7781 %
18	89	82(92.1348%)	7(7.8652%)	0.1012	0.2742	53.3485 %	90.8027 %
19	75	70(93.3333%)	5(6.6667%)	0.066	0.2278	32.9935 %	73.5455 %
20	75	71(94.6667%)	4(5.3333%)	0.0871	0.2432	48.5888 %	83.1765 %
21	87	76(87.3563%)	11(12.6437%)	0.116	0.3004	66.092 %	103.6416 %
22	81	75(92.5926%)	6(7.4074%)	0.0865	0.2661	51.5975 %	94.2082 %
23	74	66(89.1892%)	8(10.8108%)	0.1148	0.302	63.2742 %	102.6455 %
24	83	78(93.9759%)	5(6.0241%)	0.0787	0.2318	39.0463 %	74.3988 %
25	72	64(88.8889%)	8(11.1111%)	0.12	0.2967	57.9288 %	94.0273 %

5.1.1.2 Using Naïve Bayes algorithm

Table 3: Naïve Bayes algorithm on accelerometer

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	77(86.5169%)	12(13.4831%)	0.1488	0.3283	78.4831 %	108.7022 %
2	79	75(94.9367%)	4(5.0633%)	0.0608	0.2076	35.4798 %	72.6802 %
3	87	82(94.2529%)	5(5.7471%)	0.0916	0.2409	58.3185 %	88.1861 %
4	90	85(94.4444 %)	5(5.5556%)	0.0623	0.2265	46.4816 %	90.2237 %

5	95	85(89.4737%)	10(10.5263%)	0.1161	0.2884	64.81 %	98.31 %
6	81	71(87.6543%)	10(12.3457%)	0.1348	0.3106	65.4527 %	98.6162 %
7	79	71(89.8734%)	8(10.1266%)	0.1237	0.2894	64.7053 %	95.6079 %
8	88	84(95.4545%)	4(4.5455%)	0.0674	0.2084	38.7865 %	72.2801 %
9	85	77(90.5882%)	8(9.4118%)	0.1227	0.2664	68.4407 %	90.9583 %
10	96	89(92.7083%)	7(7.2917%)	0.0973	0.2561	60.4883 %	92.4022 %
11	98	89(90.8163%)	9(9.1837 %)	0.1148	0.2683	72.7449 %	97.728 %
12	88	83(94.3182%)	5(5.6818%)	0.092	0.2392	59.1784 %	88.0035 %
13	79	73(92.4051%)	6(7.5949%)	0.1174	0.2669	68.5133 %	93.4407 %
14	81	75(92.5926%)	6(7.4074%)	0.0972	0.2586	51.9337 %	86.3449 %
15	89	84(94.382%)	5(5.618%)	0.0656	0.2191	25.6093 %	61.883 %
16	78	72(92.3077%)	6(7.6923%)	0.1179	0.2746	44.0855 %	75.9145 %
17	85	79(92.9412%)	6(7.0558%)	0.0998	0.2541	42.9188 %	75.5989 %
18	89	83(93.2584%)	6(6.7416%)	0.0776	0.2316	40.8954 %	76.6996 %
19	75	67(89.3333%)	8(10.6667%)	0.1153	0.2794	57.6555 %	90.1759 %
20	75	69(92%)	6(8%)	0.1141	0.2625	63.6724 %	89.7843 %
21	87	78(89.6552%)	9(10.3448%)	0.123	0.3028	70.0866 %	104.4811 %
22	81	73(90.1235%)	8(9.8765%)	0.121	0.2872	72.1645 %	101.6861 %
23	74	67(90.5405%)	7(9.4595%)	0.1136	0.2948	62.6171 %	100.1948 %
24	83	77(92.7711%)	6(7.2289%)	0.0899	0.2612	44.604 %	83.8404 %
25	72	67(93.0556%)	5(6.9444%)	0.0837	0.2448	40.3877 %	77.5792 %

5.1.1.3 Using Random Forest algorithm

Table 4: Random Forest algorithm on accelerometer

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	81(91.0112%)	8(8.9888%)	0.0944	0.2267	49.7675 %	75.0853 %
2	79	75(94.9367%)	4(5.0633%)	0.1082	0.2475	63.1575 %	86.657 %
3	87	82(94.2529%)	5(5.7471%)	0.0899	0.2306	57.2564 %	84.4028 %
4	90	84(93.3333%)	6(6.6667%)	0.0733	0.1989	54.6707 %	79.2392 %
5	95	90(94.7368%)	5(5.2632%)	0.0911	0.2255	50.8097 %	76.8858 %
6	81	77(95.0617%)	4(4.9383%)	0.112	0.2459	54.383 %	78.0733 %
7	79	74(93.6709%)	5(6.3291%)	0.1044	0.2283	54.6433 %	75.4023 %
8	88	83(94.3182%)	5(5.6818%)	0.0827	0.2117	47.6223 %	73.4213 %
9	85	80(94.1176%)	5(5.8824%)	0.1002	0.2354	55.9227 %	80.3463 %
10	96	91(94.7917%)	5(5.2083%)	0.0996	0.2362	61.9079 %	85.2137 %
11	98	90(91.8367%)	8(8.1633%)	0.1166	0.2517	73.8892 %	91.6863 %
12	88	79(89.7727%)	9(10.2273%)	0.1248	0.2676	80.2372 %	98.4613 %
13	79	72(91.1392%)	7(8.8608%)	0.1272	0.2721	74.2378 %	95.261 %
14	81	74(91.358%)	7(8.642%)	0.1277	0.271	68.2204 %	90.4898 %
15	89	83(93.2584%)	6(6.7416%)	0.1036	0.2463	40.4503 %	69.5559 %
16	78	74(94.8718%)	4(5.1282%)	0.1041	0.2316	38.9149 %	64.042 %

17	85	78(91.7647%)	7(8.2353%)	0.1054	0.2517	45.3282 %	74.8794 %
18	89	85(95.5056%)	4(4.4944%)	0.0936	0.2247	49.3528 %	74.3988 %
19	75	71(94.6667%)	4(5.3333%)	0.084	0.216	42.0209 %	69.728 %
20	75	70(93.3333%)	5(6.6667%)	0.1111	0.2506	61.9631 %	85.7164 %
21	87	80(91.954%)	7(8.046%)	0.1236	0.2666	70.3978 %	91.9716 %
22	81	74(91.358%)	7(8.642%)	0.109	0.2575	65.0117 %	91.1722 %
23	74	69(93.2432%)	5(6.7568%)	0.135	0.2757	74.4017 %	93.6937 %
24	83	77(92.7711%)	6(7.2289%)	0.1051	0.2503	52.1107 %	80.347 %
25	72	64(88.8889%)	8(11.1111%)	0.1378	0.2893	66.5003 %	91.6774

5.1.1.4 Using SVM algorithm

Table 5: SVM algorithm on accelerometer

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	80(89.8876%)	9(10.1124%)	0.1011	0.318	53.3223 %	105.3035 %
2	79	75(94.9367%)	4(5.0633%)	0.0506	0.225	29.5474 %	78.7718 %
3	87	80(91.954%)	7(8.046%)	0.0805	0.2837	51.2525 %	103.8432 %
4	90	83(92.2222%)	7(7.7778%)	0.0778	0.2789	57.984 %	111.1035 %
5	95	87(91.5789%)	8(8.4211%)	0.0842	0.2902	46.9917 %	98.9228 %
6	81	75(92.5926%)	6(7.4074%)	0.0741	0.2722	35.9755 %	86.4207 %
7	79	71(89.8734%)	8(10.1266%)	0.1013	0.3182	52.9874 %	105.1152 %
8	88	82(93.1818%)	6(6.8182%)	0.0682	0.2611	39.2491 %	90.5678 %

9	85	76(89.4118%)	9(10.5882%)	0.1059	0.3254	59.0733 %	111.0813 %
10	96	88(91.6667%)	8(8.3333%)	0.0833	0.2887	51.8058 %	104.1518 %
11	98	90(91.8367%)	8(8.1633%)	0.0816	0.2857	51.716 %	104.074 %
12	88	81(92.0455%)	7(7.9545%)	0.0795	0.282	51.153 %	103.7692 %
13	79	72(91.1392%)	7(8.8608%)	0.0886	0.2977	51.7079 %	104.2053 %
14	81	74(91.358%)	7(8.642%)	0.0864	0.294	46.184 %	98.1715 %
15	89	84(94.382%)	5(5.618%)	0.0562	0.237	21.9362 %	66.9352 %
16	78	67(85.8974%)	11(14.1026%)	0.141	0.3755	52.7172 %	103.8204 %
17	85	77(90.5882%)	8(9.4118%)	0.0941	0.3068	40.4716 %	91.2586 %
18	89	82(92.1348%)	7(7.8652%)	0.0787	0.2804	41.4729 %	92.869 %
19	75	68(90.6667%)	7(9.3333%)	0.0933	0.3055	46.6899 %	98.6131 %
20	75	68(90.6667%)	7(9.3333%)	0.0933	0.3055	52.0699 %	104.4889 %
21	87	81(93.1034%)	6(6.8966%)	0.069	0.2626	39.2918 %	90.6098 %
22	81	74(91.358%)	7(8.642%)	0.0864	0.294	51.5382 %	104.0752 %
23	74	67(90.5405%)	7(9.4595%)	0.0946	0.3076	52.1333 %	104.5392 %
24	83	77(92.7711%)	6(7.2289%)	0.0723	0.2689	35.856 %	86.2999 %
25	72	64(88.8889%)	8(11.1111%)	0.1111	0.3333	53.6293 %	105.6383 %

5.1.2 Gyroscope Dataset

5.1.2.1 Using AdaBoost M1 algorithm

Table 6: AdaBoost M1 algorithm on gyroscope

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	80(89.8876%)	9(10.1124%)	0.1569	0.3113	82.737 %	103.096 %
2	79	71(89.8734%)	8(10.1266%)	0.1112	0.2599	64.9157 %	90.9677 %
3	87	79(90.8046%)	8(9.1954%)	0.1474	0.3083	93.8748 %	112.8574 %
4	90	85(94.4444%)	5(5.5556%)	0.0833	0.2352	62.1153 %	93.682 %
5	95	85(89.4737%)	10(10.5263%)	0.1679	0.3183	93.711 %	108.4998 %
6	81	68(83.9506%)	13(16.0494%)	0.2026	0.3514	98.4169 %	111.5732 %
7	88	80(90.9091%)	8(9.0909%)	0.1199	0.2925	77.1021 %	107.6191 %
8	79	72(91.1392%)	7(8.8608%)	0.1051	0.2723	61.3147 %	95.3065 %
9	81	72(88.8889%)	9(11.1111%)	0.1581	0.3129	84.4987 %	104.4864 %
10	89	76(85.3933%)	13(14.6067%)	0.2213	0.3587	86.411 %	101.2893 %
11	78	62(79.4872%)	16(20.5128%)	0.2591	0.3943	96.8583 %	108.9989 %
12	85	74(87.0588%)	11(12.9412%)	0.1946	0.3275	83.6952 %	97.4324 %
13	89	80(89.8876%)	9(10.1124%)	0.1463	0.3067	77.1584 %	101.5492 %
14	75	63(84%)	12(16%)	0.1672	0.3397	83.6247 %	109.6454 %
15	75	67(89.3333%)	8(10.6667%)	0.1715	0.3372	95.6713 %	115.3338 %
16	87	79(90.8046%)	8(9.1954%)	0.1395	0.3036	79.475 %	104.767 %
17	81	72(88.8889%)	9(11.1111%)	0.1377	0.3128	82.1365 %	110.7404 %

18	74	66(89.1892%)	8(10.8108%)	0.141	0.3079	77.6838 %	104.668 %
19	83	72(86.747%)	11(13.253%)	0.1647	0.3352	81.6902 %	107.5835 %
20	72	61(84.7222%)	11(15.2778%)	0.1584	0.3418	76.4751 %	108.3223 %
21	79	71(89.8734%)	8(10.1266%)	0.1705	0.318	89.1996 %	105.0505 %
22	88	78(88.6364%)	10(11.3636%)	0.1395	0.3001	80.3163 %	104.1012 %
23	85	75(88.2353%)	10(11.7647%)	0.1629	0.3234	90.8946 %	110.4049 %
24	96	84(87.5%)	12(12.5%)	0.1495	0.3278	92.9433 %	118.253 %
25	98	89(90.8163%)	9(9.1837%)	0.1311	0.2894	83.0537 %	105.3984 %

5.1.2.2 Using Naïve Bayes algorithm

Table 7: Naïve Bayes algorithm on gyroscope

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	79(88.764%)	10(11.236%)	0.1295	0.311	68.264 %	102.9998 %
2	79	70(88.6076%)	9(11.3924%)	0.1503	0.2795	87.7001 %	97.8384 %
3	87	75(86.2069%)	12(13.7931%)	0.1908	0.3536	121.535 %	129.4654 %
4	90	84(93.3333%)	6(6.6667%)	0.0796	0.2441	59.3355 %	97.2581 %
5	95	86(90.5263%)	9(9.4737%)	0.19	0.3153	106.0387 %	107.4784 %
6	81	67(82.716%)	14(17.284%)	0.214	0.3634	103.9418 %	115.3999 %

7	88	82(93.1818%)	6(6.8182%)	0.1045	0.2574	67.2109 %	94.6957 %
8	79	71(89.8734%)	8(10.1266%)	0.1397	0.298	81.5063 %	104.3282 %
9	81	74(91.358%)	7(8.642%)	0.1382	0.2876	73.8615 %	96.0562 %
10	89	71(79.7753%)	18(20.2247%)	0.2432	0.4068	94.9496 %	114.8808 %
11	78	66(84.6154%)	12(15.3846%)	0.2193	0.3639	81.9731 %	100.5927 %
12	85	69(81.1765%)	16(18.8235%)	0.2126	0.3909	91.4057 %	116.2741 %
13	89	80(89.8876%)	9(10.1124%)	0.1424	0.2948	75.0837 %	97.62 %
14	75	67(89.3333%)	8(10.6667%)	0.152	0.03035	76.0492 %	97.978 %
15	75	68(90.6667%)	7(9.3333%)	0.1414	0.2975	78.8929 %	101.7629 %
16	87	77(88.5057%)	10(11.4943%)	0.1489	0.3392	84.8406 %	117.027 %
17	81	73(90.1235%)	8(9.8765%)	0.1249	0.2852	74.4714 %	100.9538 %
18	74	67(90.5405%)	7(9.4595%)	0.1307	0.2945	72.0392 %	100.1088 %
19	83	69(83.1325%)	14(16.8675%)	0.2274	0.3706	112.7949 %	118.9674 %
20	72	63(87.5%)	9(12.5%)	0.1739	0.3388	83.9218 %	107.3805 %
21	79	69(87.3418%)	10(12.6582%)	0.1667	0.3357	87.2443 %	110.9024 %
22	88	77(87.5%)	11(12.5%)	0.1713	0.322	98.6027 %	111.6791 %
23	85	76(89.4118%)	9(10.5882%)	0.1505	0.302	83.9718 %	103.098 %
24	96	85(88.5417%)	11(11.4583%)	0.1405	0.3144	87.3585 %	113.4347 %
25	98	90(91.8367%)	8(8.1633%)	0.1327	0.2863	84.0467 %	104.2832 %

5.1.2.3 Using Random Forest algorithm

Table 8: Random Forest algorithm on gyroscope

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	80(89.8876%)	9(10.1124%)	0.1476	0.2871	77.8506 %	95.0769 %
2	79	71(89.8734%)	8(10.1266%)	0.12	0.2534	70.0273 %	88.6909 %
3	87	79(90.8046%)	8(9.1954%)	0.1517	0.3	96.6476 %	109.809 %
4	90	82(91.1111%)	8(8.8889%)	0.122	0.2719	90.9521 %	108.3017 %
5	95	85(89.4737%)	10(10.5263%)	0.1723	0.3161	96.1567 %	107.7504 %
6	81	70(86.4198%)	11(13.5802%)	0.1978	0.3466	96.0546 %	110.0465 %
7	88	81(92.0455%)	7(7.9545%)	0.1322	0.2828	84.9871 %	104.0631 %
8	79	72(91.1392%)	7(8.8608%)	0.1468	0.2855	85.6874 %	99.9471 %
9	81	74(91.358%)	7(8.642%)	0.164	0.3045	87.6177 %	101.6835 %
10	89	67(75.2809%)	22(24.7191%)	0.2396	0.3829	93.5358 %	108.1346 %
11	78	62(79.4872%)	16(20.5128%)	0.2688	0.4048	100.4981 %	111.9235 %
12	85	71(83.5294%)	14(16.4706%)	0.1829	0.3278	78.6666 %	97.5224 %
13	89	80(89.8876%)	9(10.1124%)	0.1489	0.2887	78.5023 %	95.6177 %
14	75	64(85.3333%)	11(14.6667%)	0.172	0.3164	86.0428 %	102.145 %
15	75	68(90.6667%)	7(9.3333%)	0.152	0.3044	84.7995 %	104.1121 %
16	87	78(89.6552%)	9(10.3448%)	0.1564	0.3009	89.1268 %	103.8086 %
17	81	74(91.358%)	7(8.642%)	0.1484	0.295	88.4984 %	104.4359 %

18	74	68(91.8919%)	6(8.1081%)	0.1553	0.301	85.5731 %	102.315 %
19	83	71(85.5422%)	12(14.4578%)	0.1813	0.3192	89.9388 %	102.4494 %
20	72	62(86.1111%)	10(13.8889%)	0.1861	0.33	89.829 %	104.5953 %
21	79	69(87.3418%)	10(12.6582)	0.1868	0.3326	97.7618 %	109.8579 %
22	88	79(89.7727%)	9(10.2273%)	0.1451	0.3006	83.5352 %	104.2507 %
23	85	79(92.9412%)	6(7.0588%)	0.1499	0.2906	83.6215 %	99.2078 %
24	96	84(87.5%)	12(12.5%)	0.1431	0.302	88.9765 %	108.9743 %
25	98	88(89.7959%)	10(10.2041%)	0.1501	0.2985	95.0927 %	108.7442 %

5.1.2.4 Using SVM algorithm

Table 9: SVM algorithm on gyroscope

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	80(89.8876%)	9(10.1124%)	0.1011	0.318	53.3223 %	105.3035 %
2	79	72(91.1392%)	7(8.8608%)	0.0886	0.2977	51.7079 %	104.2053 %
3	87	80(91.954%)	7(8.046%)	0.0805	0.2837	51.2525 %	103.8432 %
4	90	84(93.3333%)	6(6.6667%)	0.0667	0.2582	49.7006 %	102.8618 %
5	95	86(90.5263%)	9(9.4737%)	0.0947	0.3078	52.8656 %	104.9234 %
6	81	72(88.8889%)	9(11.1111%)	0.1111	0.3333	53.9632 %	105.8433 %
7	88	81(92.0455%)	7(7.9545%)	0.0795	0.282	51.153 %	103.7692 %

8	79	72(91.1392%)	7(8.8608%)	0.0886	0.2977	51.7079 %	104.2053 %
9	81	73(90.1235%)	8(9.8765%)	0.0988	0.3143	52.7817	104.9497 %
10	89	76(85.3933%)	13(14.6067%)	0.1461	0.3822	57.034 %	107.9297 %
11	78	66(84.6154%)	12(15.3846%)	0.1538	0.3922	57.5097%	108.4368 %
12	85	74(87.0588%)	11(12.9412%)	0.1294	0.3597	55.6484 %	107.0102 %
13	89	80(89.8876%)	9(10.1124%)	0.1011	0.318	53.3223 %	105.3035 %
14	75	67(89.3333%)	8(10.6667%)	0.1067	0.3266	53.3598 %	105.4218 %
15	75	68(90.6667%)	7(9.3333%)	0.0933	0.3055	52.0699 %	104.4889 %
16	87	79(90.8046%)	8(9.1954%)	0.092	0.3032	52.389 %	104.6272 %
17	81	74(91.358%)	7(8.642%)	0.0864	0.294	51.5382 %	104.0752 %
18	74	67(90.5405%)	7(9.4595%)	0.0946	0.3076	52.1333 %	104.5392 %
19	83	74(89.1566%)	9(10.8434%)	0.1084	0.3293	53.784 %	105.6954 %
20	72	64(88.8889%)	8(11.1111%)	0.1111	0.3333	53.6293 %	105.6383 %
21	79	71(89.8734%)	8(10.1266%)	0.1013	0.3182	52.9874 %	105.1152 %
22	88	80(90.9091%)	8(9.0909%)	0.0909	0.3015	52.3322 %	104.5786 %
23	85	77(90.5882%)	8(9.4118%)	0.0941	0.3068	52.5096 %	104.7285 %
24	96	88(91.6667%)	8(8.3333%)	0.0833	0.2887	51.8058 %	104.1518 %
25	98	90(91.8367%)	8(8.1633%)	0.0816	0.2857	51.716 %	104.074 %

5.1.3 Accelerometer & Gyroscope combined dataset

5.1.3.1 Using AdaBoost M1 algorithm

Table 10: AdaBoost M1 algorithm on accelerometer and gyroscope combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	82(92.1348%)	7(7.8652%)	0.0979	0.281	51.6348 %	93.054 %
2	79	74(93.6709%)	5(6.3291%)	0.0714	0.2143	41.6796 %	75.0316 %
3	87	81(93.1034%)	6(6.8966%)	0.0826	0.2567	52.5969 %	93.9615 %
4	90	86(95.5556%)	4(4.4444%)	0.0462	0.1921	34.4764 %	76.5157 %
5	95	89(93.6842%)	6(6.3158%)	0.0705	0.2399	39.3375 %	81.7848 %
6	81	77(95.0617%)	4(4.9383%)	0.097	0.245	47.0957 %	77.8052 %
7	88	81(90.9091%)	8(9.0909%)	0.1049	0.2897	67.4456 %	106.6057 %
8	79	71(89.8734%)	8(10.1266%)	0.1279	0.304	74.6404 %	106.4217 %
9	81	75(92.5926%)	6(7.4074%)	0.1098	0.2875	58.6607 %	96.0042 %
10	89	83(93.2584%)	6(6.7416%)	0.0879	0.24	34.3384 %	67.7797 %
11	78	71(91.0256%)	7(8.9744%)	0.1132	0.2733	42.3015 %	75.5511 %
12	85	81(95.2941%)	4(4.7059%)	0.081	0.2376	34.8442 %	70.6672 %
13	89	83(93.2584%)	6(6.7416%)	0.0687	0.2415	36.2083 %	79.9598 %
14	75	70(93.3333%)	5(6.6667%)	0.0712	0.2362	35.5965 %	76.2319 %
15	75	67(89.3333%)	8(10.6667%)	0.1009	0.2677	56.2691 %	91.5724 %
16	87	78(89.6552%)	9(10.3448%)	0.106	0.2917	60.3931 %	100.6303 %
17	81	75(92.5926%)	6(7.4074%)	0.0811	0.2569	48.3795 %	90.9354 %

18	74	67(90.5405%)	7(9.4595%)	0.0909	0.2721	50.0859 %	92.4835 %
19	83	77(92.7711%)	6(7.2289%)	0.0725	0.2549	35.9846 %	81.81 %
20	72	67(93.0556%)	5(6.9444%)	0.0819	0.2543	39.5197 %	80.5815 %
21	79	71(89.8734%)	8(10.1266%)	0.1186	0.2922	62.0539 %	96.535 %
22	88	80(90.9091%)	8(9.0909%)	0.1017	0.2899	58.5163 %	100.5405 %
23	85	77(90.5882%)	8(9.4118%)	0.1005	0.2697	56.0824 %	92.0621 %
24	96	87(90.625%)	9(9.375%)	0.0887	0.2691	55.1464 %	97.0885 %
25	98	87(88.7755%)	11(11.2245%)	0.1238	0.3098	78.4309 %	112.8613 %

5.1.3.2 Using Naïve Bayes algorithm

Table 11: Naïve Bayes algorithm on accelerometer and gyroscope combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	78(87.6404%)	11(12.3596%)	0.1285	0.3132	67.7733 %	103.6982 %
2	79	75(94.9367%)	4(5.0633%)	0.0684	0.2245	39.9167 %	78.6014 %
3	87	80(91.954%)	7(8.046%)	0.11	0.2798	70.0681 %	102.4238 %
4	90	85(94.4444%)	5(5.5556%)	0.058	0.2278	43.2565 %	90.7387 %
5	95	87(91.5789%)	8(8.4211%)	0.1122	0.2814	62.5871 %	95.9384 %
6	81	69(85.1852%)	12(14.8148%)	0.1532	0.3397	74.4215 %	107.8528 %
7	88	81(92.0455%)	7(7.9545%)	0.074	0.2389	47.5759 %	87.8967 %
8	79	73(92.4051%)	6(7.5949%)	0.1059	0.2814	61.7758 %	98.4963 %
9	81	75(92.5926%)	6(7.4074%)	0.0927	0.2689	49.5663 %	89.7917 %

10	89	83(93.2584%)	6(6.7416%)	0.0796	0.2559	31.0923 %	72.2631 %
11	78	72(92.3077%)	6(7.6923%)	0.0999	0.2759	37.356 %	76.2757 %
12	85	78(91.7647%)	7(8.2353%)	0.0989	0.2853	42.5384 %	84.8693 %
13	89	85(95.5056%)	4(4.4944%)	0.0577	0.1979	30.4289 %	65.5392 %
14	75	69(92%)	6(8%)	0.0965	0.2541	48.2896 %	82.0123 %
15	75	69(92%)	6(8%)	0.1014	0.253	56.5484 %	86.5321 %
16	87	78(89.6552%)	9(10.3448%)	0.1146	0.3038	65.3018 %	104.8372 %
17	81	76(93.8272%)	5(6.1728%)	0.0782	0.2339	46.6136 %	82.8016 %
18	74	70(94.5946%)	4(5.4054%)	0.0749	0.2325	41.2525 %	79.0306 %
19	83	75(90.3614%)	8(9.6386%)	0.1005	0.2778	49.8267 %	89.159 %
20	72	67(93.0556%)	5(6.9444%)	0.0926	0.2631	44.7152 %	83.3919 %
21	79	70(88.6076%)	9(11.3924%)	0.1282	0.3251	67.096 %	107.3847 %
22	88	83(94.3182%)	5(5.6818%)	0.0579	0.2175	33.3582 %	75.4285 %
23	85	78(91.7647%)	7(8.2353%)	0.0987	0.2574	55.0775 %	87.8747 %
24	96	88(91.6667%)	8(8.3333%)	0.0985	0.2834	61.2377 %	102.2518 %
25	98	89(90.8163%)	9(9.1837%)	0.0986	0.2682	62.4566 %	97.6968 %

5.1.3.3 Using Random Forest algorithm

Table 12: Random Forest algorithm on accelerometer and gyroscope combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	82(92.1348%)	7(7.8652%)	0.1066	0.2293	56.2254 %	75.9363 %
2	79	75(94.9367%)	4(5.0633%)	0.1059	0.2197	61.8279 %	76.9014 %
3	87	82(94.2592%)	5(5.7471%)	0.1005	0.2332	63.9924 %	85.361 %
4	90	86(95.5556%)	4(4.4444%)	0.0688	0.1716	51.2745 %	68.3638 %
5	95	89(93.6842%)	6(6.3158%)	0.1093	0.2339	60.9717 %	79.7219 %
6	81	77(95.0617%)	4(4.9383%)	0.1252	0.245	60.7986 %	77.793 %
7	88	79(89.7727%)	9(10.2273%)	0.1182	0.2655	75.9988 %	97.6754 %
8	79	73(92.4051%)	6(7.5949%)	0.1544	0.2931	90.1195 %	102.6031 %
9	81	75(92.5926%)	6(7.4074%)	0.1252	0.259	66.9009 %	86.4912 %
10	89	85(95.5056%)	4(4.4944%)	0.1162	0.2451	43.364 %	69.2137 %
11	78	73(93.5897%)	5(6.4103%)	0.1224	0.2335	45.7681 %	64.5594 %
12	85	79(92.9412%)	6(7.0588%)	0.1054	0.2297	45.3282 %	68.3176 %
13	89	84(94.382%)	5(5.618%)	0.094	0.218	49.5897 %	72.1831 %
14	75	71(94.6667%)	4(5.3333%)	0.0907	0.2067	45.3559 %	66.7184 %
15	75	70(93.3333%)	5(6.6667%)	0.1203	0.2494	67.0957 %	85.3106 %
16	87	79(90.8046%)	8(9.1954%)	0.1387	0.2767	79.042 %	95.4841 %
17	81	76(93.8272%)	5(6.1728%)	0.1	0.2295	59.637 %	81.2606 %
18	74	68(91.8919%)	6(8.1081%)	0.12	0.2494	66.1349 %	84.7551 %

19	83	78(93.9759%)	5(6.0241%)	0.1022	0.2355	50.6765 %	75.5868 %
20	72	66(91.6667%)	6(8.3333%)	0.1394	0.2742	67.3047 %	86.8919 %
21	79	72(91.1392%)	7(8.8608%)	0.127	0.261	66.433 %	86.2289 %
22	88	83(94.3182%)	5(6818%)	0.096	0.2291	55.2759 %	79.472 %
23	85	78(91.7647%)	7(8.2353%)	0.1139	0.2431	63.5366 %	82.9969 %
24	96	87(90.625%)	9(9.375%)	0.105	0.2411	65.2753 %	86.9792 %
25	98	90(91.8367%)	8(8.1633%)	0.1204	0.253	76.2811 %	92.1628 %

5.1.3.4 Using SVM algorithm

Table 13: SVM algorithm on accelerometer and gyroscope combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	89	87(97.7528%)	2(2.2472%)	0.0225	0.1499	11.9032 %	49.7004 %
2	79	76(96.2025%)	3(3.7975%)	0.038	0.1949	22.3551 %	68.5301 %
3	87	84(96.5517%)	3(3.4483%)	0.0345	0.1857	22.121 %	68.2285 %
4	90	89(98.8889%)	1(1.1111%)	0.0111	0.1054	8.3789 %	42.2276 %
5	95	92(96.8421%)	3(3.1579%)	0.0316	0.1777	17.7115 %	60.6562 %
6	81	77(95.0617%)	4(4.9383%)	0.0494	0.2222	24.1104 %	70.6793 %
7	88	85(96.5909%)	3(3.4091%)	0.0341	0.1846	22.0949 %	68.1949 %
8	79	76(96.2025%)	3(3.7975%)	0.038	0.1949	22.3551 %	68.5301 %

9	81	78(96.2963%)	3(3.7037%)	0.037	0.1925	19.9359 %	64.4717 %
10	89	85(95.5056%)	4(4.4944%)	0.0449	0.212	17.6271 %	60.0124 %
11	78	76(97.4359%)	2(2.5641%)	0.0256	0.1601	9.627 %	44.3685 %
12	85	83(97.6471%)	2(2.3529%)	0.0235	0.1534	10.1576 %	45.6849 %
13	89	87(97.7528%)	2(2.2472%)	0.0225	0.1499	11.9032 %	49.7004 %
14	75	74(98.6667%)	1(1.3333%)	0.0133	0.1155	6.7132 %	37.3861 %
15	75	72(96%)	3(4%)	0.04	0.2	22.4927 %	68.7071 %
16	87	83(95.4023%)	4(4.5977%)	0.046	0.2144	26.3509 %	74.1673 %
17	81	78(96.2963%)	3(3.7037%)	0.037	0.1925	22.2919 %	68.4487 %
18	74	73(98.6486%)	1(1.3514%)	0.0135	0.1162	7.5099 %	39.6955 %
19	83	80(96.3855%)	3(3.6145%)	0.0361	0.1901	18.0212 %	61.1184 %
20	72	70(97.2222%)	2(2.7778%)	0.0278	0.1667	13.5036 %	53.0034 %
21	79	78(98.7342%)	1(1.2658%)	0.0127	0.1125	6.6667 %	37.2742 %
22	88	87(98.8636%)	1(1.1364%)	0.0114	0.1066	6.5789 %	37.0625 %
23	85	84(98.8235%)	1(1.1765%)	0.0118	0.1085	6.6059 %	37.1277 %
24	96	93(96.875%)	3(3.125%)	0.0313	0.1768	19.5479 %	63.93 %
25	98	95(96.9388%)	3(3.0612%)	0.0306	0.175	19.5059 %	63.8711 %

5.1.4 Accelerometer, Gyroscope and Magnetometer combined dataset

5.1.4.1 Using AdaBoost M1 algorithm

Table 14: AdaBoost M1 algorithm on accelerometer, gyroscope and magnetometer combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	68	65(95.5882%)	3(4.4118%)	0.0442	0.1878	22.625 %	61.471 %
2	67	64(95.5224%)	3(4.4776%)	0.0475	0.1997	21.6133 %	61.3621 %
3	59	54(93.1034%)	4(6.8966%)	0.0667	0.2469	33.6409 %	80.376 %
4	62	59(95.1613%)	3(4.8387%)	0.0443	0.2021	17.215 %	57.2228 %
5	66	62(93.9394%)	4(6.0606%)	0.056	0.2133	27.981 %	68.8857 %
6	71	69(97.1831%)	2(2.8169%)	0.0304	0.167	16.1416 %	55.7012 %
7	62	56(90.3226%)	6(9.6774%)	0.0948	0.2987	44.7985 %	93.7694 %
8	68	65(95.5882%)	3(4.4118%)	0.0427	0.1989	21.8893 %	65.0971 %
9	70	65(92.8571%)	5(7.1429%)	0.0682	0.2496	35.7706 %	82.6806 %
10	68	65(95.5882%)	3(4.4118%)	0.0412	0.1928	23.8842 %	67.4539 %
11	72	70(97.2222%)	2(2.7778%)	0.0358	0.1686	21.8293 %	60.5346 %
12	65	61(93.8462%)	4(6.1538%)	0.0749	0.2574	48.5014 %	95.7348 %
13	64	57(89.0625%)	7(10.9375%)	0.1039	0.2947	50.4894 %	93.856 %
14	60	57(95%)	3(5%)	0.0374	0.1694	17.1877 %	52.3729 %
15	69	66(95.6522%)	3(4.3478%)	0.046	0.1957	26.9749 %	68.9185 %
16	63	58(92.0635%)	5(7.9365%)	0.0893	0.2856	42.8024 %	90.3103 %
17	62	52(83.871%)	10(16.129%)	0.1563	0.3493	60.7317 %	98.9066 %

18	66	60(90.9091%)	6(9.0909%)	0.0978	0.2913	40.0153 %	84.7039 %
19	70	69(98.5714%)	1(1.4286%)	0.0197	0.1244	10.3033 %	41.2033 %
20	64	61(95.3125%)	3(4.6875%)	0.0601	0.2248	26.2512 %	67.6889 %
21	63	60(95.2385%)	3(4.7619%)	0.049	0.2119	23.4633 %	66.988 %
22	61	57(93.4426%)	4(6.5574%)	0.0675	0.2426	25.8681 %	68.2167 %
23	73	70(95.8904%)	3(4.1096%)	0.0448	0.1982	24.3912 %	66.9601 %
24	59	55(93.2203%)	4(6.7797%)	0.0741	0.2535	30.2837 %	73.7122 %
25	70	66(94.2857%)	4(5.7143%)	0.0616	0.2323	32.2896 %	76.9654 %

5.1.4.2 Using Naïve Bayes algorithm

Table 15: Naïve Bayes algorithm on accelerometer, gyroscope and magnetometer combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	68	66(97.0588%)	2(2.9412%)	0.0403	0.1736	20.6275 %	56.797 %
2	67	64(95.5224%)	3(4.4776%)	0.0734	0.2299	33.3768 %	70.6525 %
3	58	55(94.8276%)	3(5.1724%)	0.0466	0.174	23.5311 %	56.62 %
4	62	59(95.1613%)	3(4.8387%)	0.0482	0.219	18.7423 %	62.011 %
5	66	63(95.4545%)	3(4.5455%)	0.0609	0.2186	30.4359 %	70.6029 %

6	71	68(95.7746%)	3(4.2254%)	0.0557	0.2147	29.5497 %	71.6034 %
7	62	58(93.5484%)	4(6.4516%)	0.0811	0.2572	38.3337 %	80.7289 %
8	68	62(91.1765%)	6(8.8235%)	0.0813	0.2509	41.6507 %	82.1026 %
9	70	65(92.8571%)	5(7.1429%)	0.0771	0.2371	40.4021 %	78.5399 %
10	68	65(95.5882%)	3(4.4118%)	0.0489	0.2072	28.3654 %	72.4864 %
11	72	68(94.4444%)	4(5.5556%)	0.0511	0.2086	31.1344 %	74.9029 %
12	65	58(89.2308%)	7(10.7692%)	0.1146	0.3147	74.262 %	117.039 %
13	64	59(92.1875%)	5(7.8125%)	0.0994	0.2876	48.3055 %	91.5808 %
14	60	55(91.6667%)	5(8.3333%)	0.0747	0.2608	34.2665 %	80.6526 %
15	69	68(98.5507%)	1(1.4493%)	0.0161	0.1092	9.4373 %	38.4509 %
16	63	54(85.7143%)	9(14.2857%)	0.1462	0.3297	70.0265 %	104.2448 %
17	62	53(85.4839%)	9(14.5161%)	0.1807	0.3499	70.1974 %	99.0709 %
18	66	59(89.3939%)	7(10.6061%)	0.1233	0.3027	50.4471 %	87.9963 %
19	70	67(95.7143%)	3(4.2857%)	0.0418	0.1843	21.9324 %	61.0673 %
20	64	59(92.1875%)	5(7.8125%)	0.0914	0.2772	39.954 %	83.4617 %
21	63	60(95.2381%)	3(4.7619%)	0.0519	0.2074	24.8747 %	65.5654 %
22	61	54(88.5246%)	7(11.4754%)	0.1052	0.2934	40.3321 %	82.5052 %
23	73	68(93.1507%)	5(6.8943%)	0.0645	0.2188	35.1077 %	73.9022 %
24	59	55(93.2203%)	4(6.7797%)	0.0836	0.2276	34.1455 %	66.1929 %
25	70	65(92.8571%)	5(7.1429%)	0.0683	0.232	35.8129 %	76.8604 %

5.1.4.3 Using Random Forest algorithm

Table 16: Random Forest algorithm on accelerometer, gyroscope and magnetometer combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	68	64(94.1176%)	4(5.8824%)	0.0887	0.1846	45.4142 %	60.4183 %
2	67	63(94.0299%)	4(5.9701%)	0.1058	0.2008	48.1058 %	61.6832 %
3	58	54(93.1034%)	4(6.8966%)	0.0941	0.1999	47.4991 %	65.0803 %
4	62	59(95.1613%)	3(4.8387%)	0.0837	0.1918	32.5223 %	54.2944 %
5	66	61(92.4242%)	5(7.5758%)	0.0911	0.2012	45.4724 %	65 %
6	71	67(94.3662%)	4(5.6338%)	0.0828	0.1794	43.9678 %	59.8087 %
7	62	57(91.9355%)	5(8.0645%)	0.109	0.2258	51.5062 %	70.8765 %
8	68	65(95.5882%)	3(4.4118%)	0.0918	0.1952	46.9958 %	63.8712 %
9	70	66(94.2857%)	4(5.7143%)	0.0897	0.1885	47.0276 %	62.4338 %
10	68	65(95.5882%)	3(4.4118%)	0.0757	0.1803	43.9225 %	63.0903 %
11	72	69(95.8333%)	3(4.1667%)	0.0857	0.1964	52.2557 %	70.5245 %
12	65	59(90.7692%)	6(9.2308%)	0.1118	0.23	72.4526 %	85.5262 %
13	64	59(92.1875%)	5(7.8125%)	0.1347	0.2488	65.4375 %	79.2417 %
14	60	58(96.6667%)	2(3.3333%)	0.0995	0.1981	45.6721 %	61.2724 %
15	69	66(95.6522%)	3(4.3478%)	0.0751	0.1828	44.0644 %	64.3723 %
16	63	58(92.0635%)	5(7.9365%)	0.1243	0.2478	59.5468 %	78.3385 %
17	62	55(88.7097%)	7(11.2903%)	0.1645	0.2773	63.9168 %	78.5007 %
18	66	59(89.3939%)	7(10.6061%)	0.1389	0.2539	56.8298 %	73.8314 %

19	70	67(95.7143%)	3(4.2857%)	0.0776	0.1692	40.6624 %	56.0523 %
20	64	60(93.75%)	4(6.25%)	0.0939	0.2172	41.0373 %	65.3935 %
21	63	61(96.8254%)	2(3.1746%)	0.0683	0.1549	32.7013 %	48.978 %
22	61	57(93.4426%)	4(6.5574%)	0.1005	0.2058	38.532 %	57.8654 %
23	73	72(98.6301%)	1(1.3699%)	0.0685	0.156	37.2856 %	52.6974 %
24	59	54(91.5254%)	5(8.4746%)	0.1069%	0.2171	43.6983 %	63.1352 %
25	70	66(94.2857%)	4(5.7143%)	0.09	0.1913	47.1774 %	63.3783 %

5.1.4.4 Using SVM Algorithm

Table 17: SVM algorithm on accelerometer, gyroscope and magnetometer combined

No. of Reading	Total number of instances	Correctly Classified Instances	Incorrectly Classified Instances	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
1	68	61(89.7059%)	7(10.2941%)	0.1029	0.3208	52.7197 %	104.9954 %
2	67	59(88.0597%)	8(11.9403%)	0.1194	0.3455	54.2802 %	106.1711 %
3	58	52(89.6552%)	6(10.3448%)	0.1034	0.3216	52.1968 %	104.6873 %
4	62	53(85.4839%)	9(14.5161%)	0.1452	0.381	56.3971 %	107.8698 %
5	66	59(89.3939%)	7(10.6061%)	0.1061	0.3257	52.9628 %	105.1892 %
6	71	64(90.1408%)	7(9.8592%)	0.0986	0.314	52.3426 %	104.6999 %

7	62	55(88.7079%)	7(11.2903%)	0.1129	0.336	53.3349 %	105.4725 %
8	68	61(89.7059%)	7(10.2941%)	0.1029	0.3208	52.7197 %	104.9954 %
9	70	63(90%)	7(10%)	0.1	0.3162	52.4194 %	104.757 %
10	68	62(91.1765%)	6(8.8235%)	0.0882	0.297	51.1718 %	103.933 %
11	72	68(94.4444%)	4(5.5556%)	0.0556	0.2357	33.8773 %	84.6488 %
12	65	60(92.3077%)	5(7.6923%)	0.0769	0.2774	49.8298 %	103.1425 %
13	64	57(89.0625%)	7(10.9375%)	0.1094	0.3307	53.1395 %	105.3258 %
14	60	53(88.3333%)	7(11.6667%)	0.1167	0.3416	53.5519 %	105.6308 %
15	69	63(91.3043%)	6(8.6957%)	0.087	0.2949	51.0398 %	103.8353 %
16	63	56(88.8889%)	7(11.1111%)	0.1111	0.3333	53.2347 %	105.3978 %
17	62	53(85.4839%)	9(14.5161%)	0.1452	0.381	56.3971 %	107.8698 %
18	66	57(86.3636%)	9(13.6364%)	0.1364	0.3693	55.7762 %	107.3609 %
19	70	63(90%)	7(10%)	0.1	0.3162	52.4194 %	104.757 %
20	64	57(89.0625%)	7(10.9375%)	0.1094	0.3307	47.7972 %	99.5738 %
21	63	56(88.8889%)	7(11.1111%)	0.1111	0.3333	53.2347 %	105.3978 %
22	61	55(90.1639%)	6(9.8361%)	0.0984	0.3136	37.7149 %	88.191 %
23	73	66(90.411%)	7(9.589%)	0.0959	0.3097	52.1998 %	104.5911 %
24	59	51(86.4407%)	8(13.5593%)	0.1356	0.3682	55.4019 %	107.0723 %
25	70	65(92.8571%)	5(7.1429%)	0.0714	0.2673	37.4424 %	88.5358

5.2 Accuracy Comparison

5.2.1 Accelerometer

Table 18: Accuracy Comparison of Accelerometer

Algorithm	Correctly classified instances (avg.)
AdaBoost M1	92.75396 %
Naïve Bayes	91.85616 %
Random Forest	93.038132%
SVM	91.387312 %

5.2.2 Gyroscope

Table 19: Accuracy Comparison of Gyroscope

Algorithm	Correctly classified instances (avg.)
AdaBoost M1	88.3978 %
Naïve Bayes	88.1782544 %
Random Forest	88.32832 %
SVM	90.064612 %

5.2.3 Accelerometer and Gyroscope

Table 20: Accuracy Comparison of accelerometer and Gyroscope combined

Algorithm	Correctly classified instances (avg.)
AdaBoost M1	92.059 %
Naïve Bayes	91.931188 %
Random Forest	93.066596 %
SVM	97.1035 %

5.2.4 Accelerometer, Gyroscope and Magnetometer

Table 21: Accuracy Comparison of Accelerometer, Gyroscope and Magnetometer Combined

Algorithm	Correctly classified instances (avg.)
AdaBoost M1	93.937664 %
Naïve Bayes	92.981368 %
Random Forest	93.841988 %
SVM	89.441768 %

5.3 Final Results on Accuracy Comparison

When the Accelerometer and Gyroscope sensors were used individually, it appears that the Accelerometer provides better accuracy threshold for all 4 algorithms while the Gyroscope sensor provides less accuracy threshold (88% - 90%). In the case of using Accelerometer, Random Forest algorithm provides highest accuracy (93%) while other 3 algorithms provide 92% accuracy threshold. Using Gyroscope individually and in case of using Accelerometer and Gyroscope both sensors together we can see that among the 4 algorithms, Support Vector Machine (SVM) provides us with the most accurate data more than 97% and the other 3 algorithms are not very far in accuracy as well. But when Accelerometer, Gyroscope and Magnetometer all 3 sensors are used all at once, we find AdaBoost and Random Forest algorithms to be most accurate with the threshold of 93% while the accuracy threshold of SVM goes below 90%.

Chapter 6

Conclusion

6.1 Summary

A Fall can be lethal for people of all age, especially the young and the old. Due to their nimble body structure, elderly people suffer the most due to fall and as the number of working-class people is increasing rapidly the intensive care for the elderly are decreasing as rapidly. To provide proper health care for the elder class, fall detection and immediate response from nearby medical unit has become a must. In our thesis work we tried to portray the significance of machine learning in fall detection and which algorithm accompanied with certain sensor can be the best tool for accurately detecting fall. Though many researchers in their research work has used particular body sensors and posture detectors, we tried to use android smartphone to detect fall from other physical postures and WEKA software to test the accuracy of algorithms. The SVM we believed to be most accurate turns out not so much in different situations when used with multiple sensors. AdaBoost M1, Random Forest and Naïve Bayes have also proven to be closer in accuracy threshold to SVM than we initially presumed.

6.2 Future Work

There is much room for improvement in the field of detecting falls and we can continue our research and improve our results in the following ways.

1. For better accuracy in detecting various body movements, we can use particular body sensors that can be attached to the chest, torso, ankles and wrists and can use more algorithms and datasets to better train for devices.

2. Due to prejudice, ethical views mobility and comfortability many elderly persons may not support the idea of wearing sensors to their body parts and to reduce their anxiety

regarding technological devices, we can use less sensors that provide even better accurate data rather than using a great deal of them.

3. As smartphones are available now throughout the world, we can develop a certain app for fall detection with various sensors like Accelerometer and Gyroscope that will provide better accuracy in detecting falls and also not be a nuisance to the elderly folks.
4. Only detecting fall won't be enough if the medical response team doesn't get the patient the proper care they need in time. A mobile device can be used in such case that has access to all the nearest healthcare centers of the patient so that when the fall happens, the quick response team can provide for the person in need of proper care.

6.3 Scope for improvement

As we did not have many people available to perform the data collection process the height sample had a comparatively small range and less variation. We recognize the fact that the physical attributes are discrete from one person to another. Therefore, for a larger group of people the accuracy may differ for the algorithms used.

References

[1] https://www.macrotrends.net/countries/WLD/world/life-expectancy#:~:text=The%20current%20life%20expectancy%20for,a%200.24%25%20increase%20from%202018

[2] https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/files/documents/2020/Sep/un_pop_2020_pf_ageing_10_key_messages.pdf

[3] https://www.macrotrends.net/countries/BGD/bangladesh/life-expectancy#:~:text=The%20current%20life%20expectancy%20for,a%200.39%25%20increase%20from%202018

[4] Abbate, S., Avvenuti, M., Corsini, P., Vecchio, A., and Light, J., “Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network : A Survey,” in Yen Kheng Tan (Ed.), Wireless Sensor Networks: Application-Centric Design, Chap 1, pp. 1-20,InTech, Rijeka, Croatia, 2010.

[5] https://www.cdc.gov/mmwr/volumes/67/wr/mm6718a1.htm#F1_down

[6] Gibson, M., Andres, R., Isaacs, B., Radebaugh, T. & Worm-Petersen, J. (1987). The prevention
of falls in later life. a report of the kellogg international work group on the prevention of falls by the elderly., Danish Medical Bulletin **34**(4): 1–24.

[7] Yu, X. Approaches and principles of fall detection for elderly and patient. in e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on. 2008. IEEE.

[8] Baranzini, Federico, et al. "Fall-related injuries in a nursing home setting: is polypharmacy a risk factor?." *BMC health services research* 9.1 (2009): 228.

[9] Lach, Helen W., et al. "Falls in the elderly: reliability of a classification system." *Journal of the American Geriatrics Society* 39.2 (1991): 197-202.

[10] Hartholt, Klaas A., et al. "Societal consequences of falls in the older population: injuries, healthcare costs, and long-term reduced quality of life." *Journal of Trauma and Acute Care Surgery* 71.3 (2011): 748-753.

[11]https://www.cdc.gov/injury/features/older-adult-falls/index.html#:~:text=About%2036%20million%20older%20adults,in%20more%20than%2032%2C000%20deaths.

[12] Vellas, Bruno J., et al. "Fear of falling and restriction of mobility in elderly fallers." *Age and ageing* 26.3 (1997): 189-193.

[13] Ge, Yujia, and Bin Xu. "Detecting Falls Using Accelerometers by Adaptive Thresholds in Mobile Devices." *JCP* 9.7 (2014): 1553-1559.

[14] Abbate, Stefano, et al. "A smartphone-based fall detection system." *Pervasive and Mobile Computing* 8.6 (2012): 883-899.

[15] Zhang, T., Wang, J., Liu, P., and Hou, J.. *Fall Detection by Wearable Sensor and One-Class SVM Algorithm*. Lecture Notes in Control and Information Science, issue 345, pp. 858–863, 2006.

[16] Otanasap, Nuth. "Pre-impact fall detection based on wearable device using dynamic threshold model." *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. IEEE, 2016.

[17] Tapia, Emmanuel Munguia, et al. "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor." *2007 11th IEEE international symposium on wearable computers*. IEEE, 2007.

[18] Bourke, Alan K., and Gerald M. Lyons. "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor." *Medical engineering & physics* 30.1 (2008): 84-90.

[19] Islam, Md Milon, et al. "A Review on Fall Detection Systems Using Data from Smartphone Sensors." *Ingénierie des Systèmes d Inf.* 24.6 (2019): 569-576.

[20] http://www.bookrags.com/research/accelerometer-woi/#gsc.tab=0

[21] Evenson, Kelly R., and James W. Terry Jr. "Assessment of differing definitions of accelerometer nonwear time." *Research quarterly for exercise and sport* 80.2 (2009): 355-362.

[[22] https://docs.idew.org/code-internet-of-things/references/physical-inputs/accelerometer

[23] Scarborough, James Blaine. The Gyroscope. Interscience Publ., 1958.

[24] http://hyperphysics.phy-astr.gsu.edu/hbase/gyr.html

[25] https://link.springer.com/chapter/10.1007/0-387-25786-1_6

[26] https://aeronsystems.com/gyroscopes-and-their-types/

[27] https://www.elprocus.com/gyroscope-sensor/

[28] Diao, Zhanlin, et al. "Analysis and compensation of MEMS gyroscope drift." 2013 Seventh International Conference on Sensing Technology (ICST). IEEE, 2013.

[29] Guri, Mordechai & Daidakulov, Andrey & Elovici, Yuval. (2018). MAGNETO: Covert Channel between Air-Gapped Systems and Nearby Smartphones via CPU-Generated Magnetic Fields. Future Generation Computer Systems. 115. 10.1016/j.future.2020.08.045.

[30] https://www.allaboutcircuits.com/technical-articles/understanding-and-applying-the-hall-effect/#:~:text=Regarding%20accuracy%2C%20currently%20available%20Hall,effect%20devices%20are%20particularly%20suitable.

[31] https://electricalfundablog.com/hall-effect-principle-history-theory-explanation-mathematical-expressions-applications/

[32] Özdemir, Ahmet Turan. "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice." Sensors 16.8 (2016): 1161.

[33] X. Zhang and L. Gao, "A novel auto-calibration method of the vector magnetometer," 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, 2009, pp. 1-145-1-150, doi: 10.1109/ICEMI.2009.5274904.

[34] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018. S2CID 206787478.

[35] Joachims, Thorsten (1998). "Text categorization with Support Vector Machines: Learning with many relevant features". *Machine Learning: ECML-98*. Lecture Notes in Computer Science. Springer. **1398**: 137–142. doi:10.1007/BFb0026683. ISBN 978-3-540-64417-0.

[36] Pradhan, Sameer S., et al. "Shallow semantic parsing using support vector machines." Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004. 2004.

[37] Vapnik, Vladimir N.: Invited Speaker. IPMU Information Processing and Management 2014).

[38] Barghout, Lauren. "Spatial-Taxon Information Granules as Used in Iterative Fuzzy-Decision-Making for Image Segmentation". Granular Computing and Decision-Making. Springer International Publishing, 2015. 285–318.

[39] *A. Maity (2016). "Supervised Classification of RADARSAT-2 Polarimetric Data for Different Land Features". arXiv:1608.00501[cs.CV].*

[40] *DeCoste, Dennis (2002). "Training Invariant Support Vector Machines" (PDF). Machine Learning.* ***46****: 161–190. doi:10.1023/A:1012454411458. S2CID 85843.*

[41] *Maitra, D. S.; Bhattacharya, U.; Parui, S. K. (August 2015). "CNN based common approach to handwritten character recognition of multiple scripts". 2015 13th International Conference on Document Analysis and Recognition (ICDAR): 1021–1025. doi:10.1109/ICDAR.2015.7333916.*

[42] Gaonkar, Bilwaj; Davatzikos, Christos; "Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification".

[43] Cuingnet, Rémi; Rosso, Charlotte; Chupin, Marie; Lehéricy, Stéphane; Dormont, Didier; Benali, Habib; Samson, Yves; and Colliot, Olivier; "Spatial regularization of SVM for the detection of diffusion alterations associated with stroke outcome", *Medical Image Analysis*, 2011, 15 (5): 729–737.

[44] Statnikov, Alexander; Hardin, Douglas; & Aliferis, Constantin; (2006); "Using SVM weight-based methods to identify causally relevant and non-causally relevant variables", *Sign*, 1, 4.

[45] McCallum, Andrew. "Graphical Models, Lecture2: Bayesian Network Represention" (PDF). Retrieved 22 October 2019.

[46] Boosting Algorithms: AdaBoost, Gradient Boosting and XGBoost". *hackernoon.com*. May 5, 2018. Retrieved 2020-01-04

[47] Rojas, R. (2009). AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting. Freie University, Berlin, Tech. Rep.

[48] Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). An Introduction to Statistical Learning. Springer. pp. 316–321.

[49] Mitja Luštrek and Boštjan Kaluža, Fall Detection and Activity Recognition with Machine Learning, Jožef Stefan Institute, Department of Intelligent Systems, July 16, 2008