A Study On Machine Learning Algorithms For Fall Detection And Movement Classification

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Electrical and Computer Engineering
University of Saskatchewan
Saskatoon

By
Amitoz Singh Ralhan

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5A9

Abstract

Fall among the elderly is an important health issue. Fall detection and movement tracking techniques are therefore instrumental in dealing with this issue. This thesis responds to the challenge of classifying different movement types as a part of a system designed to fulfill the need for a wearable device to collect data for fall and near-fall analysis.

Four different fall activities (forward, backward, left and right), three normal activities (standing, walking and lying down) and near-fall situations are identified and detected. Different machine learning algorithms are compared and the best one is used for the real time classification. The comparison is made using Waikato Environment for Knowledge Analysis or in short WEKA. The system also has the ability to adapt to different gaits of different people. A feature selection algorithm is also introduced to reduce the number of features required for the classification problem.

Acknowledgements

I can never overstate my gratitute to my supervisors Dr. Seok-Bum Ko and Dr. Yang Shi who guided me throughout the period of my MSc pursuit. Without their support, inspiration and constructive criticism, I might have digressed from my path. Working with them has been my most rewarding academic experience so far.

I would also like to thank all the members of the FANFARE team, Dr. Anh Dinh, Dr. Daniel Teng, Dr. Li Chen, Dr. Vanina Bello-Haas, Dr. Jenny Basran and Dr. Carl McCrowsky who have been dedicated to the success of this system. It has been a wonderful experience to be a part of the team and contribute my own bit to the project.

I am indebted to many student colleagues who accompanied and shared knowledge with me. My friends, who made my stay here enjoyable and memorable and made me feel at home away from home.

Most importantly, I would like to thank my parents, Jernail Singh Ralhan and Suman Ralhan, and my brother Maninder Singh Ralhan. Their love and support has made this possible. To them I dedicate this thesis.

Contents

List of Tables

List of Figures

List of Abbreviations

LOF	List of Figures
LOT	List of Tables
MATLAB	Matrix Laboratories
FANFARE	Falls And Near Falls Assessment Research and Evaluation
WEKA	Waikato Environment for Knowledge Analysis
GUI	Graphical User Interface
RBF	Radial Basis Function
OFS	Optimum Feature Selection
NBC	Naive Bayesian Classifier
UCI	University of California, Irvine
FANFARE	Falls And Near Falls Assessment Research and Evaluation

Chapter 1

Introduction

This chapter introduces the problem of fall detection. The concept of a fall is discussed along with a description of the issues related to fall detection. As this work involves the use of machine learning techniques for the detection of fall and movement classification, this chapter also introduces the field of machine learning. A section on supervised learning, explains the concept in detail, as this is the sub branch of machine learning that is used for this research. Finally, a discussion on feature selection is included, which gives an overview of the process and its importance for the current work.

1.1 Fall Detection

Fall amongst the elderly is a major health concern. Falls account for approximately half of all injury-related admissions in hospitals in the over 65 age group [1, 2]. Falls are responsible not only for causing disabling fractures and other physical injuries, but also for causing psychological trauma which can reduce the independence and confidence among the elderly [3]. Detection of fall is a peculiar problem as it is ill defined in definition. Although the concept of a fall is ingrained in the common sense but it is difficult to define it precisely and thus hard to identify a means of detection. One of the definitions can be a change from upright/sitting position to reclining or completely lengthened position in an uncontrolled manner and in a very small length of time.

Complex internal neural and muscular models control the body's postural stability [4]. Older adults who have experienced fall or near falls frequently are typically

assessed through documentation methods where questionnaires are used to determine the exact nature of fall and the circumstances under which the falls occurred [5]. These documentation methods have advantages and disadvantages in terms of accuracy, costs and time commitments. The extent and accuracy of recall of the fall activity is always a matter of concern in these methods. The circumstances that led to the fall and the near fall situations are particularly harder to recognize as the older people are often themselves oblivious as to what caused the fall. Therefore, robust and reliable methods for detecting and analyzing the falls is required. It is important to have historical movement data from the patients in order to correctly identify the causes and find patterns or scenarios that lead to a fall. This situation has prompted a great deal of research in fall detection and movement classification. In Chapter 2 different techniques that have been used for fall detection are discussed.

1.2 Machine Learning

Machine learning is the domain of science that explores the ability of machines of *understanding* data. It involves developing algorithms that would enable computers to learn complex patterns and make intelligent decisions based on that. *Learning* itself covers a broad range of processes and is thus hard to define. As regards to machines, it can be said that the machine learns whenever it changes its structure, program or data in such manner that its future performance improves [6]. Machine learning can broadly be categorized into two fields, Unsupervised Learning and Supervised Learning. In the prior, machine tries to identify groups of similar data from a larger dataset. In other words, it tries to form clusters of data based on some criteria such as cost functions. The machine has no prior knowledge of classes the data belongs to, it only tries to identify *natural* clusters or groups of data. Supervised learning on the other hand learns from the test set which contains classified data and predicts the classes of unseen data. The present work is concerned with the supervised learning. The test data classified into different movement types is input to the machine and from this labeled data it will learn the pattern and accordingly

predict the movement types upon receiving new unseen data. Supervised learning is explained briefly in the next section.

1.2.1 Supervised Learning

Supervised learning is a technique for deducing a function from training data. The training data consists of vectors of input data and desired outputs. After *seeing* this data, the machine is expected to find patterns in the data and relate it to the corresponding outputs. This means generalizing the present data to unseen situations in a reasonable way. Not all machine learning algorithms perform the same way for any given situation and training data. Therefore, one of the first decisions to be made is the choice of machine learning algorithm. This generally involves a comparison of performance of different algorithms on a given data set. Then based on results such as accuracy of classification, time taken to build the model, complexity of the algorithm etc. the most suitable algorithm depending on the requirements can be selected.

One of the most important thing in supervised learning is the quality of the training data. If the machine is expected to perform well for unseen situations, then the training data must be exhaustive and accurate enough to enable the machine to build an appropriate model. If the training data does not include the entire gamut of real world situations, then the resulting model after supervised learning is prone to over-fitting, i.e it is not generalized enough and thus performs very good on the seen data, but poorly on the unseen one. In Chapter 3, the collection of training data for the purpose of this thesis is illustrated. It would be evident how the method of collecting the data ensured a good training data that would generate a generalized model once used by a machine learning algorithm. Another criteria is the number and quality of features selected for the classification. Features can be described as entities descriptive of an object. The number of features should be large enough to accurately describe the object and small enough not to encumber the learning process with too much redundant data. The next section discusses this issue.

1.2.2 Feature Selection

Usually, a data set can have hundreds and thousands of features. This huge dimensionality causes a lot of problems in the process of machine learning. This situation is often referred to as "Curse of Dimensionality" [7]. Feature selection, variable selection or attribute selection is the technique of selecting a subset of relevant features that would result in robust models. In theory, more features should result in a better distinguishing capability by the classifier, however, it is not the case, as redundant features not only slow down the process, but also result in over-fitting [8]. Over-fitting is the situation where the classifier is able to recognize the situations similar to the training data set with accuracy, but performs poorly for general situations. In other words, feature selection is the technique to remove the irrelevant and redundant data in order to improve the process of machine learning. Selecting a subset of relevant features improves the performance in the following ways [9]:

- Alleviating the *Curse of Dimensionality*
- Generalizing the model
- Reducing the data required for classification, which is very useful for real time applications
- Faster and cost effective predictors

As will be seen in the later chapters, the feature selection process results in lesser number of features used for movement classification. In real time application of the device, this is important as, more number of features result in slow operation of the device. Lesser features mean that device can work faster and more number of samples can be obtained for each movement type.

1.3 Problems Addressed

Though there is a lot of research going on which deals with the detection of falls in the elderly, and there are many commercial fall detection products available, there

is still a need for more accurate belt worn device.

This thesis primarily addresses two problems. The first is to select the most relevant features needed to do efficient classification. Second is to accurately classify different types of body movements and detect falls and near falls

For the accurate classification, the most appropriate classifier is selected, appropriateness being the accuracy of classification and the speed of model building. Since the device is expected to adapt to the gait and posture of different people, the classifier has to be easily updateable. Still, the most important factor governing the choice of the classifier is the accuracy.

Since the classification has to be done in real time, the number of feature used for classification needs to be kept to a minimum. Too many features will result in slower operation and thus lesser number of data samples classified. The most efficient feature selection algorithm solves this problem by selecting only the most relevant features needed for classification.

1.4 Outline of Thesis

The rest of the thesis is structured as follows. Chapter 2 discusses the related research in the areas of both fall detection and feature selection. Chapter 3 explains the methodology, experimental setup and all the algorithms. Chapter 4 enumerates the results and includes discussions. It is followed by a conclusion and future work.

Chapter 2

Related Work

Detection of fall is an area in which a lot of research is going on. This section describes some of the earlier efforts in the detection of the fall and also the present research going on in this field. Since, a new feature selection algorithm called Optimum Feature Selection is introduced in this thesis, this section also describes the research work in the field of feature selection.

2.1 Fall Detection

Since fall in the elderly is a major health concern, the situation has prompted a great deal of research in fall detection and movement classification. To detect the falls, several different approaches have been used. Sixsmith [10] used an array of infrared detectors for fall monitoring. Other methods such as using video cameras, door alerts, pressure mats etc. have been in place for some time now and are discussed by Miskelly [1]. Noury [11] used infrared position sensors and magnetic switches to monitor the activity. Use of accelerometers and gyroscopes has gained widespread popularity in detecting ambulatory motion and Machine learning algorithms are the most intuitive way of detecting and classifying different types of falls [12, 13, 14]. One of the earliest works involving use of accelerometers for fall detection is by Lord and Colvin [15] in 1991 and then by William [16] which was a belt worn device and detected the shock of impacting the ground and determined if the patient is lying down by using a mercury tilt switch. In [17], a sensor attached to the armpit detects the change in velocity and when this velocity exceeds a threshold, the sequence from upright position to lying posture and the absence of movement after the fall is

tracked. If after the velocity has increased beyond the threshold, the patient is lying on the ground without any movement, then a fall is said to have occured. Most of these devices have the primary objective of distinguishing normal movement from a fall event and often suffer from a high rate of false alarms, which is one of the reasons for keeping these devices from gaining more use in daily life [13]. In [18], the data from the waist mount accelerometer is passed through a Gaussian filter to remove noise and then a 3D body motion model is used to map the data to motion types. Hwang [19] used a combination of tilt meters, gyroscopes and accelerometers to detect the falls.

There is still a need for more accurate fall and near-fall detection in the medical community. In this work, machine learning techniques are employed, so that not only the falls can be detected, but they can be further classified into subcategories depending on the direction of falls. The normal movements are also further classified into subcategories. The availability of information about different movement types, the direction of fall and pre-fall position helps in pre-fall and post-fall analysis. Accurate classification of the normal movement coupled with the direction of fall, gives the complete sequence of events that led to the fall.

2.2 Feature Selection

Feature selection has been an active and fruitful field of research and development for decades in statistical pattern recognition, machine learning, data mining and statistics. A few such research works were discussed in [8]. When there are hundreds and thousands of features present, not all of them add to the information of the target. Both theoretical analysis and empirical evidence show that along with irrelevant features, redundant features also affect the speed and accuracy of learning algorithms and thus should be eliminated as well [20]. Feature selection algorithms can be broadly classified into two types. The wrapper model uses the predictive accuracy of a predetermined learning algorithm to determine the goodness of the selected subsets. These methods are computationally expensive for data with a large

number of features [21]. The filter model separates feature selection from classifier learning and selects feature subsets that are independent of any learning algorithm. These methods use general statistical measurements such as distance measures, information gain, correlation coefficients, consistency etc. to determine the feature subset.

The Optimum Feature Selection algorithm introduced in this work falls in the filter model. It ranks the features based on distance measurements and then removes redundant features using correlation coefficients. The performance of the algorithm is tested with Naive Bayesian Classifier. Naive Bayesian Classifier is extremely sensitive to the type of features used. If the features are highly correlated, they can get high weightage and reduce the accuracy of classification [22]. Many different approaches have been explored to improve the performance of the Naive Bayesian Classifier. The four main approaches are feature selection, structure extension, local learning and data expansion [23].

Most of the algorithms discussed in [8] try to find a feature set which results in an increase in the accuracy of classification. The result is a relaxed approach to the rejection of irrelevant or redundant features. The Optimum Feature Selection algorithm avoids this restraint and uses a more aggressive approach towards rejecting features, which in some cases might result in a slight fall in classification accuracies.

Chapter 3

Methodology

The entire problem of movement classification can be broken down into the following sequential steps.

- Collection of data accurately to generate a proper training file, which is an accurate representation of all kinds of movements.

- Selection of an appropriate classification algorithm which is able to distinguish different types of movements accurately.

- Feature selection process to explore the possibility of reducing the number of features required for the classification process.

- Generating the final model using the selected features and the classification algorithm.

- Implementing the model into the real time application, with the capability to update the model so as to be adaptable for different users.

In this chapter, the process and methodology of each of these steps will be discussed in detail. The chapter discusses the data collection methods, the algorithms to be compared, the feature selection algorithm used and finally a discussion on the real time implementation. Chapter 4 again lists results for each of these steps and the decisions reached based on the results.

3.1 Data Collection

The device used for data collection in the Falls And Near Falls Assessment Research and Evaluation (FANFARE) project is a Jennic board of JN5139 series and is shown in Figure 3.1.

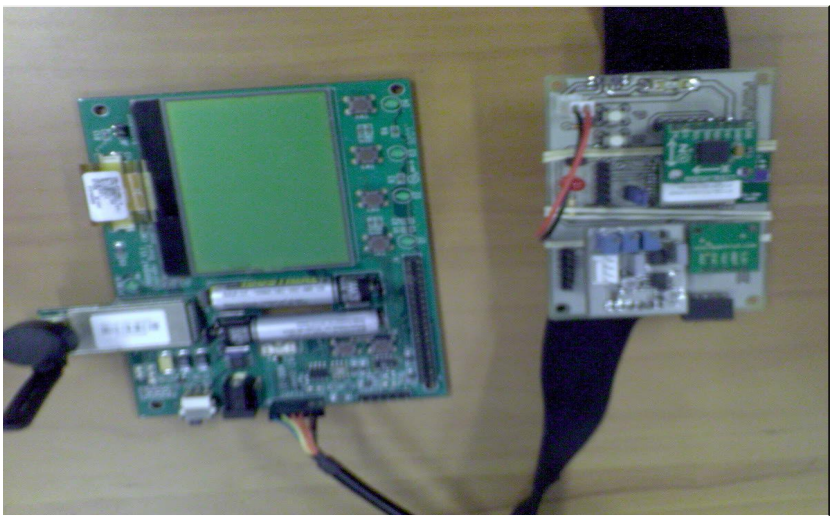

Figure 3.1: Fall Detection Device.

The belt worn device shown in right half of the figure is the end device that communicates with the coordinator which is on the left half of the figure. The coordinator is connected to the computer via cable. The device consists of a high precision three axis accelerometer sensor ST LIS3LV02DQ. It has a range of $\pm 6g$ or $\pm 2g$, which is user selectable and an operating voltage of 2.16V - 3.6V. The gyroscope used is InvenSense IDG-300. This dual-axis gyro is a MEMS device operating at a single supply voltage of 3.0-3.5V. The sensor provides analog outputs of X and Y rates with a full scale of $\pm 500\,^{\circ}$/sec. The data consists of 3582 data items distributed in 597 rows and 6 columns. The data was collected for 7 types of movements. Out of the 7, 3 are normal movements - Walking, Standing and Lying down. The rest are four types of fall activities - Forward, Backward, Left and Right. The next section describes the manner in which the data was collected for each of these movements. All the data

was collected by wearing the belt worn device at the chest level. Figure 3.2 shows the interaction between the end device and the coordinator and also the classification steps.

The training data is collected by imitating the normal movements and fall activities a number of times until we obtain sufficient number of samples for the reliable machine learning process. The output of training data collection is a number of samples having a feature set corresponding to the axis of accelerometer and gyroscope. Depending on the type and number of sensors used, the number of features can range anywhere between 5 to 15. This training data set is labeled for different movement types and then is subjected to a feature selection process, which reduces the feature set to lesser number of features based on the relevance and information content of each feature. This final reduced feature set is then used to generate the classification model. A classification model is a set of parameters which is the output of a machine learning algorithm. These parameters can then be used to classify new unseen data. In probabilistic methods for example, the parameters can be mean and standard deviation. From these parameters, the likelihood of a movement type can be calculated for every new unseen sample of data. Quality of the training data determines the accuracy of the classification model. Once the classification model is generated, it is input into the end device. Now, the device can determine the movements of the subject wearing the device. Every new sample of data is classified as one of the movement types based on the parameters of the classification model. To make the device adaptable to individual posture and gait, the data collected from sensors is fed back to the classification model and used to update the features. In this manner, the device adapts to a particular user over a period of time. In the inevitable cases of false alarms, the classification should be rectified and then fed back to the classification model, ensuring that the parameters are updated correctly. All the classification results are wirelessly transmitted to the coordinator, which is connected to a computer displaying the movements in real time. In case the result of the classification, is a $Fall$, then an alarm is raised.

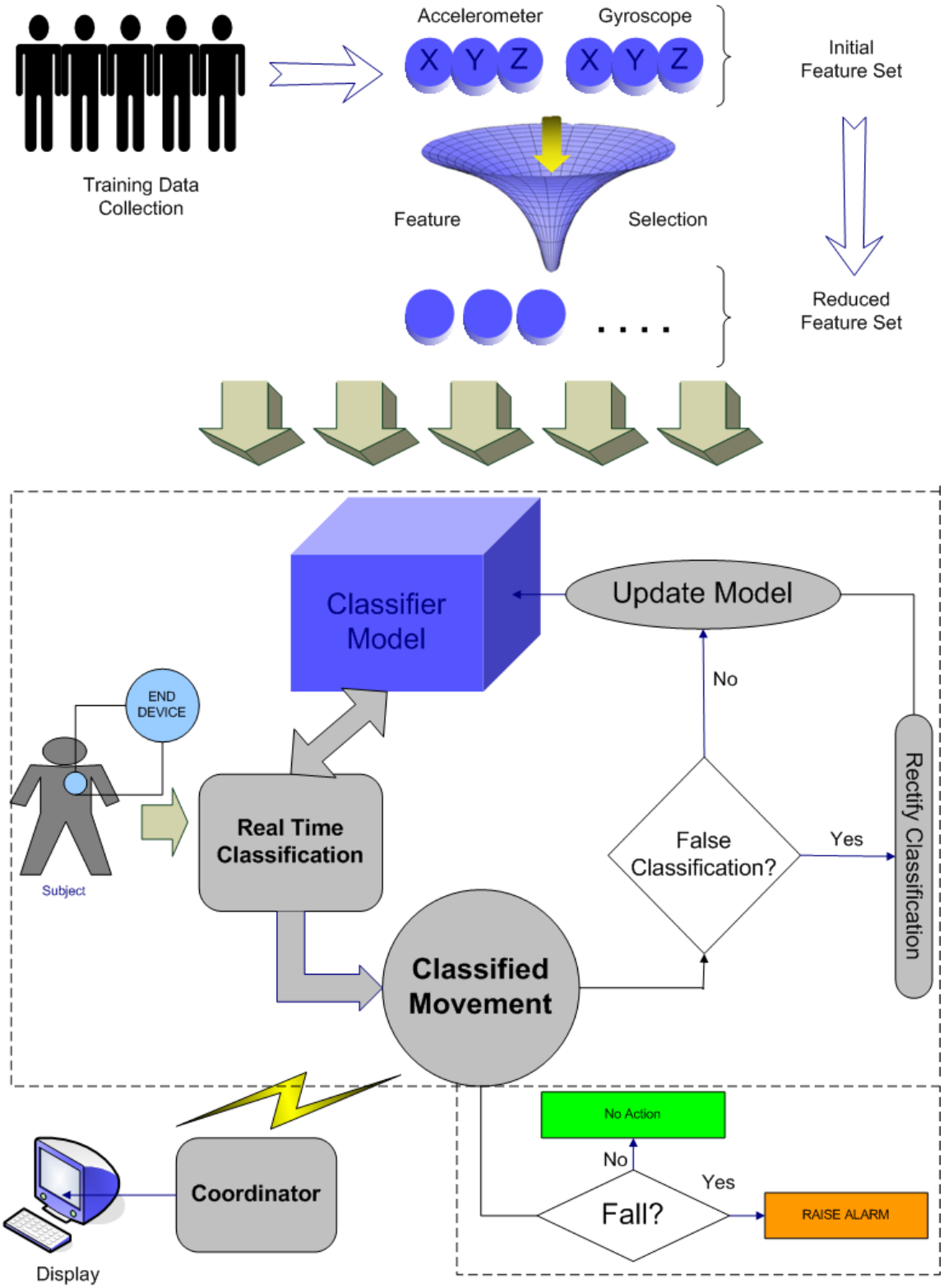

Figure 3.2: Block Diagram of Experimental Device

3.1.1 Movement Types

This section describes the manner in which data is collected for different movement types. The movements are imitated to be as close as possible to the normal routine movements. Here are all the movement types and the methodology of collecting data for each one of them.

Lying Down

To collect data for the Lying Down position, the subject is lying down flat on the back. Small controlled movements like slowly turning to left or right side or turning over completely and lying down flat on stomach are also included in this movement class. While shifting positions, no sudden movement is made and the transition in smooth.

Standing

For this movement type, the subject is standing upright at a fixed position. Slow controlled movements like leaning forward, backward, left or right is also considered standing stance. Minor shifting of feet to change the direction, the subject is facing is also included in the standing position.

Walking

Walking involves normal paced walking or even taking few steps from the standing position. The data collection process takes into consideration any direction changes or even walking backwards. Slight leaning in all the four directions is also allowed while walking, just as in standing position.

Fall Forward/Backward/Left/Right

Falling in either direction involves sudden changes in the acceleration values. For fall forward, the subsection goes from standing position to a lying down position while going in a forward direction in an uncontrolled manner in a fraction of second.

Near Fall

Near fall can be described as a state in which the subject is in a precarious position and is on the verge of falling. Every fall is preceded by a near fall situation. So, the samples just before the actual fall, are classified as near fall situations. In these experiments, the near fall situation is not further classified as left/right or forward/backward. Too much leaning in any direction results in the near fall situation.

Figure 3.3 shows a snapshot of the collected data. The first column is the time stamp, followed by acceleration values in X,Y and Z axis respectively. The last two columns are the Gyroscope values for the X and Y axis respectively. The device is aligned in a way that the Y axis accelerometer is directed upwards from ground, Z axis accelerometer is directed outwards from chest and X axis accelerometer is directed from right to left.

	A	B	C	D	E	F
1	4:14:11 PM	0.044	-0.056	1.014	-0.366	0
2	4:14:11 PM	0.044	-0.056	1.014	-1.832	0
3	4:14:11 PM	0.044	-0.056	1.014	-0.366	-0.733
4	4:14:11 PM	0.044	-0.056	1.014	-0.733	-0.733
5	4:14:11 PM	0.044	-0.056	1.017	0	0
6	4:14:11 PM	0.044	-0.053	1.017	0	0
7	4:14:12 PM	0.044	-0.053	1.014	-0.733	0
8	4:14:12 PM	0.044	-0.056	1.014	0	-0.366
9	4:14:12 PM	0.044	-0.056	1.014	-0.366	0
10	4:14:12 PM	0.044	-0.056	1.014	-0.733	0
11	4:14:12 PM	0.044	-0.056	1.014	0	-0.733
12	4:14:12 PM	0.044	-0.056	1.014	-1.099	0
13	4:14:12 PM	0.044	-0.056	1.014	-0.733	-1.099
14	4:14:12 PM	0.044	-0.056	1.017	-1.099	0
15	4:14:12 PM	0.044	-0.053	1.017	-0.733	0

Figure 3.3: Sample data file collected from accelerometer and gyroscope

While experimenting, all the falls started from an upright position and ended in

an almost lying position on a couch at a height of about 1 feet from the ground. In the repeated experiments, the time taken for the subject to go from an upright position to a lying down position on the couch is calculated to be about 1/5th to 1/7th of a second. So for each fall activity, there are around 5-6 samples of data rows available considering the sampling rate of 40 samples/sec at which the device operates. The procedure was repeated over and over again around 18-20 times for each type of fall activity. Collecting data for falling backward, left or right involved the same procedure as for forward fall, in respective directions.

3.2 Algorithm Comparison

As discussed earlier in Section 1.2 machine learning involves providing a training file to the learning algorithm in which the data is correctly classified. The machine then learns the patterns involved in the data and can automatically classify future situations.

More accurate classification techniques will ensure lesser number of false alarms (normal movements detected as falls) and fewer undetected falls. Therefore, selection of a suitable classification algorithm is the first and the most important step to solving any classification problem. Since all classification algorithms behave differently towards different type of data, depending upon the number of attributes and the nature of outliers among other factors, a comparison of different algorithms applied to a particular data set is useful in selecting the best algorithm for the task [24]. This section introduces five different classification algorithms applied to accelerometer and gyroscope data collected from the fall detection device. A brief description of each of the algorithm is provided and their respective characteristics discussed. In Chapter 4, we look at the results of the comparison of these algorithms which will lead us to the final choice of our algorithm. The comparison is made using Waikato Environment for Knowledge Analysis or in short WEKA [25]. WEKA is an open source software consisting of a number of machine learning algorithms. A brief introduction of this software is given later in this section.

3.2.1 Algorithms to be compared

In this section, five different machine learning algorithms are presented. All these algorithms differ greatly in the approach they use for learning and are popular algorithms for supervised learning. These are the two reasons for the selection of these algorithms for comparison.

Naive Bayesian Classifier

Naive Bayesian Classifier is a simple probabilistic classifier. It assumes that every feature related to a class is independent of each other [26]. So, the probability of occurrence of a class C, provided the features F_1 through F_N is

$$P(C|F_1, F_2...Fj) = P(C) \prod_{j=1}^{N} P(F_j|C). \tag{3.1}$$

The classifier learns the conditional probability of each attribute from the training data. Classification is done by calculating the probability of C given the values of features F_1 through F_N and then predicting the class with the highest probability value. Though the independence assumption is far reaching and often inaccurate in real world data, this method performs surprisingly well for most of the classification problems [27].

Radial Basis Function (RBF)

A radial basis function network is an artificial neural network which uses radial basis functions as activation functions [28]. RBF networks typically have three layers. An input layer, a hidden layer with non linear RBF function and a linear output layer. The output φ of the network for an input x is given by

$$\varphi(x) = \sum_{i=1}^{N} a_i \rho(\parallel x - c_i \parallel), \tag{3.2}$$

where c_i is the center for neuron i and a_i is the corresponding weight. A successful implementation of these networks requires appropriate values of the center and the weights [26].

Support Vector Machine

Support Vector Machines are a class of linear classifiers that simultaneously minimize the empirical classification error and maximize the geometric margin. The process involves creating a hyperplane in a n-dimensional space, that would separate two data sets with the highest margin [29]. Reaching such a hyperplane is essentially a solution of the following optimization problem.

$$\text{minimize} \quad \frac{1}{2}\| W \|^2 \quad \text{subject to} \quad C_i(W.x_i - b) \geq 1, \tag{3.3}$$

where $1 < i < n$ and W is the vector normal to hyperplane. b is the offset for vector W.

C4.5

C4.5 is a type of decision tree that uses Shannon's entropy as a criterion for selecting the most discriminatory feature.

$$\text{Entropy(S)} = \sum_{i=1}^{c} -p_i \log_2(p_i). \tag{3.4}$$

The entire data set is split using any one of the features and the resultant information gain is measured. The process is repeated for every feature and the one with the highest information gain is selected for splitting the data [26]. This becomes the first decision node of the tree and the process is repeated for every node until the final node or the leaves are reached.

Ripple Down Rule Learner

Ripple Down Rule Learner is a type of machine learning algorithm that comes under the general class of rule learners. Like all rule learners, it induces a set of rules from the data. It generates the default rule first and then the exceptions for the default rule with the least (weighted) error rate. The process is repeated until the final leaf is reached which has only one default class and no exceptions [26].

3.2.2 WEKA

WEKA (Waikato Environment for Knowledge Analysis) [25] is a java based data mining software developed at the University of Waikato, New Zealand and is available as a free software under GNU public license. It supports different data mining tasks such as data preprocessing, clustering, classification and regression. It is frequently used by researchers because of its ease of use and open source environment [30, 24]. Graphical User Interfaces contained in the software make it easy to use and better to visualize the data. The data is processed as an arff file. The tool is capable of reading data from a number of file formats like a csv file or a SQL database and convert the data internally to arff format [31]. Figure 3.4 shows a snapshot of the WEKA GUI. Data file corresponding to the movement data collected is shown open in the figure. The data file only contains three features which are the acceleration values in the three axis. The bar plot is a visual representation of the number of classes and the number of samples in each class. In the figure, 8 classes can be seen. Different classification, clustering and feature selection algorithms can be selected from the tool bar at the top.

Cross Validation

Cross validation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. In any application, where the final goal is the prediction of classes, the cross validation estimates how accurately the predictive model will behave in practice. The results from cross validation not only predict the performance of the classifier in practice, but also aid in the selection of the classifier or model [20].

There are different types of cross validations, but the one used most often is the K fold cross validation. In this technique, the data is divided into K subsets. Out of the K subsets, $K-1$ subsets are used to train the data and the 1 set used to validate the model. This process is repeated K times, using each subset exactly once for validation. The most commonly used values for K are 10 and 5. In the present

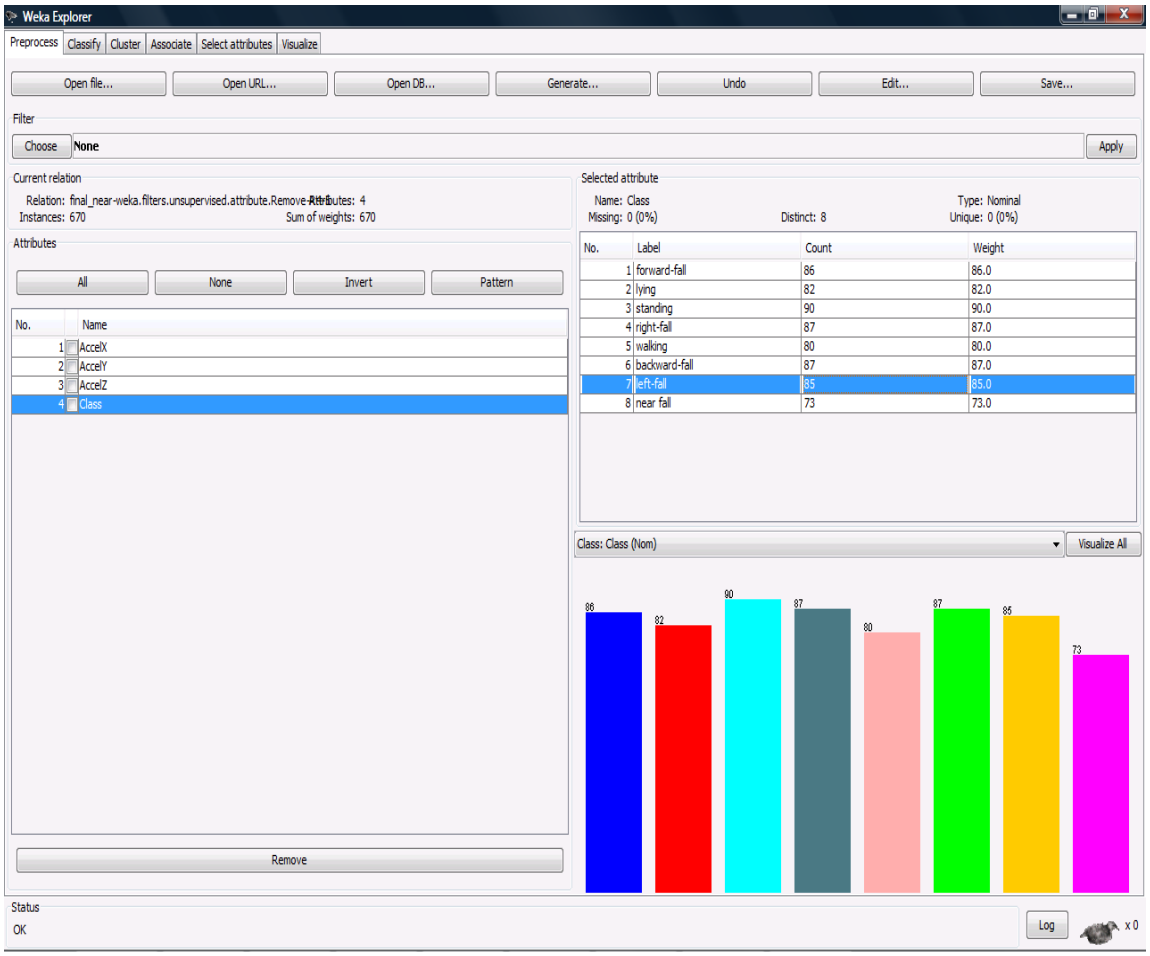

Figure 3.4: WEKA

experiments, both 10 and 5 fold cross validation are used. The data sets used to assess the performance of the feature selection algorithm have been validated using 5 fold cross validation, as some of these data sets are very large in size, and 10 fold cross validation takes a lot of memory and time.

3.3 Feature Selection

This section describes the Optimum Feature Selection algorithm. Feature selection, as introduced in Chapter 1, is an important prior step to any classification problem which reduces the dimensionality and thus the amount of data required for training. In the fall/near fall and movement classification, the end device transmits data to the coordinator wirelessly. Too many features would consume a lot of bandwidth.

Each node sends 5 columns of data (more if heart rate is included), and in case more than one node is used, then there is a lot of data to deal with and transmit. Once the training data is ready, it can be used with a feature selection algorithm to filter out the most valuable features which would not make any considerable difference to the accuracy or in many cases can even increase the accuracy.

As discussed in Section 2.2 there are many feature selection algorithms varying in complexity and approach. The algorithm presented here has a very aggressive approach towards reducing the number of features which reduces the number of features required to a great extent. The resultant features result in the same classification accuracy or higher. Even if in some cases the accuracy drops a little, it is acceptable, as the main aim is to reduce the size of data as much as possible. For this research, as described later in Section 3.5 and seen in Section 4.4 slight change in instance classification accuracies does not effect the performance of the device, as each movement type has at least 4 corresponding instances and thus is detected without fail.

3.3.1 Description

The algorithm is based on the concept of class discrimination ability of features. This means how well a feature can distinguish between different classes [7]. The features can be ranked using some distance measure and an optimum number of features selected from the feature set (these features would be able to distinguish between the classes the best and thus better classify the data). However, this does not mean that the features cannot be reduced further. There can still be features in the set that are redundant and not adding any more information to the decision making. In the next step these features can be removed. Next section describes the algorithm in detail.

3.3.2 Algorithm

This section discusses the Optimum Feature Selection algorithm in detail. The nominal values of the data sets were replaced by numbers. In the first step, the entire data set is normalized to confine the values in the range 0 to 1. This is important as the ranking step involves Euclidean distances and standard deviations and for it to be consistent, all the data values should lie in the same range. The rank of a particular feature is calculated using the following equation

$$\text{Rank} = \frac{\text{sum of distances between means of each class}}{\text{sum of standard deviation within each class}} \tag{3.5}$$

This approach is the Fisher distance ranking method [7]. For a particular feature if the distance between the means values for different classes is large and the deviations within the same class are low, then that feature can better distinguish between the classes. The higher the value of Rank, the better discrimination ability of the feature, or more valuable the feature is. The major steps of the algorithm are listed below

1. Normalize the data sets.
2. Rank the Features and select top features based on parameter θ.
3. Identify and separate features with correlation coefficient more than γ.
4. Off the separated feature keep the feature with maximum correlation with the target and remove remaining.
5. Use the final feature set for classification.
6. Compare the accuracies of the original and the reduced data set.

Once the features have been ranked, the remaining steps of the algorithm are governed by two parameters θ and γ. The prior decides what fraction of the ranked features will be selected and the latter is the threshold of correlation coefficient between any two features, beyond which the two features can be considered to possess redundant information.

The top $\frac{1}{\theta}$ features are used to form the reduced feature set. In case there are missing values in the dataset, they are not used in any of the calculations and are ignored. Then the correlation matrix of the reduced data set is analysed. The features with correlation coefficient greater than γ are separated. Out of these features, only the feature having maximum correlation with the target in the original unreduced data set is kept, the rest are discarded. If the number of features in a dataset is N, then the following relation is used for the parameters θ and γ.

$$\begin{aligned} \text{For } N < 15 \quad (\theta, \gamma) &= (2, 0.7) \\ 15 \leq N < 30 \quad (\theta, \gamma) &= (3, 0.6) \\ N > 30 \quad (\theta, \gamma) &= (7, 0.7) \end{aligned}$$

These relations were deduced after testing different combinations of θ and γ for the first 9 datasets. From these initial results, the optimum relation between the number of features and the parameters θ and γ were found to be the one mentioned in the equation above (optimality being similar or better accuracy with maximum reduction in the number of features). The relation was used for the rest of the datasets and as we will see in Section 4.3, it holds good for all the datasets. This leaves us with the final feature set. Finally, Naive Bayesian Classifier is used to classify this reduced data set and accuracies compared.

3.3.3 Algorithms compared

The above mentioned Feature selection algorithm performance is compared with two other feature selection algorithms. The results of this comparison are discussed in Section 4.3.1. This section gives a brief description of the two.

CfsSubset Evaluation

Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

Subsets of features that are highly correlated with the class while having low intercorrelation are preferred [32].

Consistency Subset Evaluation

Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes.

Consistency of any subset can never be lower than that of the full set of attributes, hence the usual practice is to use this subset evaluator in conjunction with a Random or Exhaustive search which looks for the smallest subset with consistency equal to that of the full set of attributes [33].

3.4 Near-Fall Assessment

Near-fall is a special case in the classification problem as it is even more ambiguously defined than the other fall scenarios. Near-fall can intuitively be defined as a situation in which the subject is at risk of falling down. This precarious situation is itself a subjective issue and can vary for different people. For the present experiments, near-fall situation is defined as an extremely inclined position in any direction or a huge deviation from the normal sway.

Detection of near-fall is important for giving warning to the patients, intimating them of their precarious situation so that they may hold on to something and avoid any potential falls. In Section 4.2 the results for near-fall classification are listed. These results are obtained data from a single node. Section 4.5 lists the same results for data obtained using two nodes.

3.5 Real Time Classification

Once the final feature set and the classification algorithm are identified, the model can be implemented into the device. Following are the steps involved and the capabilities required of the real time implementation.

- Upload the model parameters into the device. (Depending upon the type of algorithm selected, it could be tree structure, probability distributions, surface boundaries etc.)
- Classify each incoming sample of data into one of the movement types based on the classification model.
- Send the classification result wirelessly to the co-ordinator.
- Update the model with newly classified data. This is to achieve adaptability.
- In case of a misclassification, update the model with the corrected data class.

  Chapter 4 addresses the model chosen and Section 4.4 discusses the model, the updation process and the results.

3.6 Fall assessment using two nodes

The motivation behind using multiple nodes is the greater accuracy achieved in classifying different movements and also more number of movement types that can be detected. As can be seen in Section 4.5, the use of two nodes enables us to classify an additional movement type i.e. *Sitting*, which was not possible using only one node. Also, there is better distinction between standing and walking. This section discusses the experimentation involving two nodes. In the tests so far, only one belt worn device, worn at chest height was used for the data collection. This new experiment involves data collection from two nodes worn on different parts of the body. Here is the detail of the setup and the data collection process.

- The second device consists of three axis accelerometer and single axis gyroscope.
- The first device is still worn around chest and the second device on the thigh just above the knee.
- Both devices are on the left side of the body.

- The methodology for data collection is the same as used with the single node setup.
- An additional movement type can be classified by using the two nodes. This is the *Sitting* position.
- The data for the *Sitting* position is collected by making the subject sit on a chair and allowing slight shifting of feet and sway of upper body to account for normal sitting positions like upright, leaned back or leaned over.
- The resultant data file has 9 features and 9 classes as shown in Figure 3.5.

Section 4.5 lists the results of the classification using the data from two nodes. It also shows the results of the feature selection algorithm on this data and the corresponding changes in accuracy and number of features.

	A	B	C	D	E	F	G	H	I	J	K
1	AccelX1	AccelY1	AccelZ1	GyroX1	GyroY1	AccelX2	AccelY2	AccelZ2	GyroX2	Class	
2	0.009	0.967	-0.091	-0.733	-7.326	-0.04	-1	-0.04	0.04	Standing	
3	0.003	0.958	-0.062	4.029	-14.652	-0.04	-1	-0.03	0.04	Standing	
4	0	0.967	-0.053	6.96	-21.978	-0.04	-0.99	-0.04	0.04	Standing	
5	-0.009	0.964	-0.053	12.821	-24.542	-0.04	-1.01	-0.03	0.06	Standing	
6	-0.018	0.967	-0.047	16.85	-23.443	-0.03	-1	-0.04	0.04	Standing	
7	0.358	0.879	-0.446	74.725	-15.751	-0.08	-0.98	-0.05	-0.28	Walking	
8	0.446	0.932	-0.443	87.912	4.029	-0.02	-1.01	0	-0.33	Walking	
9	0.372	0.915	-0.375	76.19	17.949	0.03	-0.99	-0.12	-0.24	Walking	
10	0.103	0.469	0.871	2.564	-9.89	0.37	-0.21	1.02	-0.47	Lying	
11	0.041	0.507	0.812	5.861	-5.495	0.4	-0.23	1.01	-0.5	Lying	
12	0.041	0.528	0.75	2.564	18.315	0.57	-0.2	0.91	-0.18	Lying	
13	-0.062	0.618	0.759	-1.832	-9.89	0.45	-0.18	0.97	0.08	Lying	
14	-0.029	0.721	-0.695	2.564	-1.465	-0.04	-0.05	1.13	-0.17	Sitting	
15	-0.026	0.724	-0.689	3.297	-1.832	-0.04	-0.05	1.13	-0.17	Sitting	
16	-0.026	0.727	-0.683	2.564	-0.733	-0.04	-0.05	1.13	-0.17	Sitting	
17	-0.023	0.733	-0.677	0.733	-2.198	-0.03	-0.05	1.13	-0.18	Sitting	
18	-0.029	0.736	-0.683	5.861	-1.465	-0.03	-0.05	1.13	-0.18	Sitting	
19	0.449	0.667	1.585	-28.938	-166.3	-0.12	-1.01	0.03	0	Near Fall	
20	-0.403	0.531	-1.555	-130.403	398.535	-0.12	-0.99	0.04	0	Near Fall	
21	-4.822	-0.865	1.644	-7.692	109.158	0.57	-0.27	-0.34	0.03	left	
22	-1.489	-0.607	0.205	-25.275	-158.974	1.68	-0.54	0.01	0.16	left	
23	2.263	-0.361	0.95	169.597	115.385	-2.77	0.32	0.62	0.04	right	
24	1.155	1.621	1.378	26.007	77.289	-1.12	-0.47	0.57	-0.02	right	
25	-0.885	0.411	3.473	-20.147	541.392	0.27	-0.64	0.66	-0.26	backward	
26	-0.258	0.211	1.26	-92.308	531.392	0.84	-1.04	0.77	0.08	backward	
27	-0.352	0.513	-4.277	-34.799	-942.857	-0.94	0.46	-1.21	-0.1	forward	

Figure 3.5: Data From Two Nodes

Chapter 4

Results

4.1 Performance comparison of different classifiers

The data collected from the fall detection device as explained in Section 3.1 is labeled for all the different movement types and then used as training data for the five different machine learning algorithms explained in Section 3.2.1.

To measure the accuracy of the machine learning algorithms namely Naive Bayesian Classifier, Radial Basis Function Network, Support Vector Machine, C4.5 and Ripple Down Rule Learner, same method is applied to all the five algorithms, i.e. the entire data set is used to train each algorithm and subsequently 10 fold cross validation method is used to test the generated classification model. In WEKA, every row of data is considered as an instance and the features in the data are known as attributes. Results of the simulation show different parameters such as correctly and incorrectly classified instances, mean absolute and root mean squared error, confusion matrix etc. Table 4.1 shows the classification accuracies of different machine learning algorithms on the test data that contains 7 classes and 597 instances. It also shows the number of correctly and incorrectly classified instances.

As is evident from the Table 4.1 Naive Bayesian Classifier gives the highest accuracy at 97.32%, and takes the least time for model building. Support Vector Machine performs poorly in accuracy and particularly in the time taken to build the model. C4.5 tree is a close competitor both in terms of accuracy and speed. Table 4.2 lists different error measures for each of the classifiers. Again, it can be seen that, Naive Bayesian Classifier has the least values for all the error measures. With a root mean square error value of 0.07, Naive Bayesian Classifier performs many times better than

others, especially support vector machines which shows a root mean square error of 0.30. Other error measures mentioned in the table also point to the same conclusion that Naive Bayesian Classifier outperforms all other algorithms compared.

Table 4.1: Comparison of each algorithm

Algorithm	**Correctly Classified Instances % (value)**	**Incorrectly Classified Instances % (value)**	**Time Taken [1]%(seconds)**
Naive Bayesian Classifier	97.32 (581)	2.68 (16)	0.01
Support Vector Machine	92.29 (551)	7.70 (46)	14.16
Radial Basis Function	95.81 (572)	4.19 (25)	8.01
C4.5	94.64 (565)	5.36 (32)	.04
Ripple Down Rule Learner	92.78 (554)	7.20 (43)	0.16

The Naive Bayesian Classifier outperforms all other classifiers. These results concluded that Naive Bayesian Classifier should be the choice for classifier. All the further experiments used Naive Bayesian Classifier and the feature selection algorithm's performance also tested on Naive Bayesian Classifier performance on the feature subsets.

[1]using Intel Core2 Duo 1.67Ghz processor and 2GB RAM

Table 4.2: Classification errors for each algorithm

Algorithm	**Mean Absolute Error**	**Root Mean Squared Error**	**Relative Absolute Error %**	**Root Relative Squared Error %**
Naive Bayesian Classifier	0.01	0.07	4.53	21.47
Support Vector Machine	0.20	0.30	83.96	86.77
Radial Basis Function	0.01	0.11	4.99	30.28
C4.5	0.02	0.12	7.52	34.04
Ripple Down Rule Learner	0.02	0.14	8.40	40.99

Table 4.3 shows the confusion matrix for the classification results obtained from Naive Bayesian Classifier. Confusion matrix is a visualization tool typically used in supervised learning, which makes it easy to analyze the classification results. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. From the table it can be seen that the first row has 86 instances, which is the actual number of instances corresponding to $class\ A : Forward\ Fall$. However first column has 87 instances, suggesting that 1 instance from some other class has been misclassified as $Class\ A$.

From the table it can be seen that all the instances belonging to $Forward\ Fall$,$Backward\ Fall\ Lying$ and $Standing$ have been correctly classified. There is misclassification in other movement types, with $Left\ Fall$ having 8 instances misclassified, 2 of which as $Right\ Fall$ and 6 as $Walking$.

Table 4.3: Confusion matrix for Naive Bayesian Classifier

A	B	C	D	E	F	G	
86	0	0	0	0	0	0	A:Forward Fall
0	82	0	0	0	0	0	B:Lying
0	0	90	0	0	0	0	C:Standing
1	2	0	82	0	0	2	D:Right Fall
0	0	0	0	77	0	3	E:Walking
0	0	0	0	0	87	0	F:Backward Fall
0	0	0	2	6	0	77	G:Left Fall

4.2 Near Fall Analysis

The previous section showed that the Naive Bayesian Classifier is most suitable for the movement classification. In this section, the classifier is used to classify new data which also includes the $NearFall$ situation. Now the total number of classes in the dataset is 8 and the number of instances are 670. The classification accuracy is 99.40%, with 4 samples belonging to Near Fall class getting misclassified. Table 4.4 shows the confusion matrix for the classification result. As can be seen, there are only 4 misclassified instances. One $Forward\ Fall$ is misclassified as $Near\ Fall$. There are two instances of false alarm and one instance of undetected fall.

Table 4.4: Confusion matrix for Naive Bayesian Classifier (including near fall)

A	B	C	D	E	F	G	H	
85	0	0	0	0	0	0	1	A:Forward Fall
0	82	0	0	0	0	0	0	B:Lying
0	0	90	0	0	0	0	0	C:Standing
0	0	0	87	0	0	0	0	D:Right Fall
0	0	0	0	80	0	0	0	E:Walking
0	0	0	0	0	87	0	0	F:Backward Fall
0	0	0	0	0	0	85	0	G:Left Fall
1	0	0	1	0	0	1	70	H:Near Fall

4.3 Optimum Feature Selection

Optimum Feature Selection algorithm is used in this fall detection and movement classification problem to reduce the number of features required for correctly classifying the different movements. This algorithm is generalized in nature and can be used as a feature selection algorithm for different data sets. A feature selection algorithm must be able to perform well on different kinds of data sets. To prove that this algorithm can perform well for a wide range of data sets, the UCI data sets discussed in the next section are used for testing its performance. The UCI data sets are invariably used in the machine learning community to compare the performance of different algorithms.

4.3.1 UCI Datasets

To evaluate the performance of the Optimum Feature Selection Algorithm, 18 different datasets from the UCI (University of California, Irvine) repository [34] are used as shown in Table 4.5.

Table 4.5: Description of UCI data sets

Dataset	**Number of Features**	**Number of Classes**	**Number of Instances**
Breast Cancer	9	2	286
German Credit	21	2	1000
KrVsKp	36	2	3196
Mushroom	22	2	8124
Vote	17	2	435
Diabetes	9	2	768
Lung Cancer	57	3	32
Car	7	4	1728
Hepatitis	20	2	155
Ionosphere	35	2	351
Liver Disorder	7	2	345
Sick	30	2	3772
Vehicle	17	4	846
Tic-Tac-Toe	10	2	958
Spambase	58	2	4601
Sonar	61	2	208
Page Blocks	11	5	5473
Iris	5	3	150

The datasets used are from many different fields and contain a range of number of features, classes and instances. Such a selection was made to better analyze the effectiveness of the algorithm in selecting relevant features. The number of attributes in the datasets range from below 10 to above 50. Data set *Iris* has the least number of features at 2 and *Sonar* has the most number of features at 61. The number of instances is mostly of the order of few hundreds, but there data are sets with as few instances as 32 and as many as 8124. Also, they come from different fields like medicine, finance etc. The experimental results show that the algorithm works well on all of these datasets and is thus fairly generalized.

This is the outline of the experimental setup.

- Naive Bayesian Classifier algorithm is run on each of the datasets using 5 fold cross validation in WEKA.
- Optimum Feature Selection algorithm is run for each of the datasets in MATLAB and the final feature set extracted.
- Naive Bayesian Classifier is run again in WEKA for each of the datasets using the features selected by the Optimum Feature Selection algorithm.
- Accuracies compared.

The results clearly show that the algorithm effectively selects the minimum number of features while maintaining or increasing the classification accuracy by the Naive Bayesian Classifier. Table 4.6 compares the classification accuracies of Naive Bayesian Classifier on the original dataset and on the reduced data set obtained from the Optimum Feature Selection algorithm. Out of 18 datasets 11 datsets saw an increase in accuracy while 2 datasets saw no change in accuracy. 5 sets had a drop in accuracy, but as Table 4.7 shows, even in these cases the reduction in number of features used is significant. The average increase in the accuracies is 0.7%. At 5.5173%, the dataset *Vote* shows the maximum increase in accuracy, while at -4.86%, the dataset *Car* shows the maximum drop. Since, the aim of the Optimum

Feature Selection algorithm is to achieve maximum reduction in the number of features, keeping the accuracies similar or better, the change in accuracies is not huge for any data set, but, the reduction in number of features is significant.

From Table 4.7 we can see that the algorithm has been able to reduce the number of features to a great extent. In the dataset *Sonar* only 1 out of 61 features was used and the resultant accuracy was also 2.17% higher, as can be seen from Table 4.6. On an average there is a reduction of 84.50% in the number of features used. Out of the total 18 data sets, 17 have shown a decrease of more than half of the original number of features. This means that the test data required to build the model can be significantly reduced. This suggests that using only 15.5% of the features in the original dataset, similar classification accuracies have been achieved.

The data sets that saw a decrease in the classification accuracy after applying Optimum Feature Selection are *German Credit* (-1.2%), $Mushroom$ (-2.1%), *Diabetes* (-1.17%), Car (-4.86%) and $Tic{-}Tac{-}Toe$ (-1.45%) as can be seen in Table 4.6. The percent reduction in the number of features for each of these data sets are 80.95%, 81.82%, 55.56%, 57.14% and 50% respectively. This suggests that though there is a slight decrease in the classification accuracy, the reduction in number of features is considerable.

Table 4.6: Accuracy comparison between Naive Bayesian Classifier alone and Naive Bayesian Classifier with Optimum Feature Selection algorithm

Dataset	**NBC(%)**	**NBC with OFS(%)**	**NBC with OFS vs NBC alone (%)**
Breast Cancer	72.72	74.12	+1.40
German Credit	75.4	74.2	-1.20
KrVsKp	87.89	89.90	+2.00
Mushroom	95.69	93.57	-2.12
Vote	90.11	95.63	+5.52
Diabetes	76.43	75.26	-1.17
Lung Cancer	78.12	81.25	+3.13
Car	85.12	85.12	-4.86
Hepatitis	82.58	82.58	0
Ionosphere	82.62	82.90	+0.28
Liver Disorder	55.36	55.50	0.14
Sick	92.60	95.42	+2.82
Vehicle	44.90	45.50	+0.60
Tic-Tac-Toe	68.26	69.72	-1.45
Spambase	79.61	84.00	+4.39
Sonar	66.83	69.00	+2.17
Page Blocks	91.37	92.14	+0.77
Iris	96.00	96.00	0
Average	**79.06**	**79.76**	**+0.7**

Table 4.7: Comparison of number of features selected by Optimum Feature Selection algorithm and the number of features in original data set

Dataset	**Original number of Features**	**Number of Selected features by OFS**	**Reduction(%)**
Breast Cancer	9	5	44.44
German Credit	21	4	80.95
KrVsKp	36	5	86.11
Mushroom	22	4	81.82
Vote	17	1	94.12
Diabetes	9	4	55.56
Lung Cancer	57	3	94.74
Car	7	3	57.14
Hepatitis	20	6	70
Ionosphere	35	4	88.57
Liver Disorder	7	3	57.14
Sick	30	3	90
Vehicle	17	2	88.23
Tic-Tac-Toe	10	5	50
Spambase	58	8	86.21
Sonar	61	1	98.36
Page Blocks	11	4	63.64
Iris	5	2	60
Average	**24**	**3.72**	**84.5**

Feature selection comparison

This section compares the performance of the Optimum Feature Selection algorithm with two other feature selection algorithms explained in Section 3.3.3 on the 18 UCI datasets. The next two sections compare the results for single and double node data respectively.

Table 4.8 shows the performance of cfs subset evaluation algorithm on the accuracy of classification for all the 18 UCI datasets. As can be seen from the table, the average increase in classification accuracy is 1.79%. The maximum increase is seen for the data set *Ionosphere* at 8.83%. The greatest decrease for the accuracy is for data set *Car* at -15.12%. Out of the 18 data sets, 3 showed a fall in accuracy while 1 showed no change at all. The remaining 14 data sets showed some level of improvement in accuracy.

Table 4.9 shows the reduction in number of features by this algorithm. The average reduction in the number of features is 70.83%. The maximum reduction is shown for the data set *Lung Cancer* at 85.96%. The least reduction is achieved for the data set *Vehicle* at 35.3%

Table 4.8: Accuracy comparison between Naive Bayesian Classifier alone and Naive Bayesian Classifier with Cfs Subset Eval algorithm

Dataset	**NBC(%)**	**NBC with Cfs Subset Eval(%)**	**NBC vs NBC with Cfs Subset Eval (%)**
Breast Cancer	72.72	74.12	+1.40
German Credit	75.4	74.5	-0.9
KrVsKp	87.89	92.20	+4.30
Mushroom	95.69	98	+2.31
Vote	90.12	96	+5.88
Diabetes	77.43	77.08	+0.65
Lung Cancer	78.12	84.37	+6.25
Car	85.12	70	-15.12
Hepatitis	82.58	88.38	+5.80
Ionosphere	82.62	91.45	+8.83
Liver Disorder	55.36	55.65	+0.29
Sick	92.60	96.42	+3.82
Vehicle	44.9	46.81	+1.91
Tic-Tac-Toe	69.72	72.55	+2.83
Spambase	79.61	79.07	-0.54
Sonar	66.83	68.27	+1.44
Page Blocks	91.37	94.5	+3.12
Iris	96.00	96.00	0
Average	**79.05**	**80.85**	**+1.80**

Table 4.9: Comparison of number of features selected by Cfs Subset Eval and the number of features in original data set

Dataset	**Original number of Features**	**Number of selected features by Cfs Subset Eval**	**Reduction(%)**
Breast Cancer	9	5	44.44
German Credit	21	3	85.71
KrVsKp	36	7	80.55
Mushroom	22	4	81.82
Vote	17	5	70.58
Diabetes	9	4	55.55
Lung Cancer	57	8	85.96
Car	7	1	85.71
Hepatitis	20	10	50
Ionosphere	35	14	60
Liver Disorder	7	1	85.71
Sick	30	6	80
Vehicle	17	11	35.3
Tic-Tac-Toe	10	5	50
Spambase	58	15	74.14
Sonar	61	19	68.85
Page Blocks	11	6	45.45
Iris	5	2	60
Average	**24**	**7**	**70.83**

Table 4.10 shows the performance of consistency subset evaluation algorithm on the accuracy of classification for all the 18 UCI datasets. The average increase in classification accuracy for all the data sets is 2.15%. The maximum increase in accuracy is seen for the data set *Lung Cancer* at 9.02%. The greatest decrease in accuracy is for data set *Sonar* at -0.96%. In total, 5 data sets show no change in accuracy, 2 show a fall in accuracy and the remaining 11 data sets had some level of increase in classification accuracy.

Table 4.11 shows the reduction in number of features by this algorithm. The average reduction in the number of features is 63.42%. The maximum reduction in feature number is achieved for the data set *Lung Cancer* at 92.98%. For this particular data set, the algorithm accomplished a great decrease in the number of features and also a considerable increase in the classification accuracy. In 3 data sets namely *Diabetes*, *Car* and *vehicle*, this algorithm could not achieve any reduction in the number of features.

Table 4.10: Accuracy comparison between Naive Bayesian Classifier alone and Naive Bayesian Classifier with Consistency Subset Eval algorithm

Dataset	**NBC(%)**	**NBC with Consistency Subset Eval(%)**	**NBC with Consistency subset Eval vs only NBC (%)**
Breast Cancer	72.73	75.16	+2.43
German Credit	75.4	75.2	-0.2
KrVsKp	87.89	94.33	+6.44
Mushroom	95.69	98.52	+2.83
Vote	90.11	91.03	+0.92
Diabetes	77.43	76.43	0
Lung Cancer	78.12	87.15	+9.03
Car	85.12	85.12	0
Hepatitis	82.58	84.51	+1.93
Ionosphere	82.62	84.62	+2.00
Liver Disorder	55.36	55.65	+0.29
Sick	92.60	94.70	+2.1
Vehicle	44.9	44.9	0
Tic-Tac-Toe	69.72	72.33	+2.61
Spambase	79.61	86.96	+7.35
Sonar	66.83	65.86	-0.97
Page Blocks	91.37	91.37	0
Iris	96	96	0
Average	**79.06**	**81.21**	**+2.15**

Table 4.11: Comparison of number of features selected by Consistency Subset Eval and the number of features in original data set

Dataset	**Original number of Features**	**Number of selected features by Consistency Subset Eval**	**Reduction(%)**
Breast Cancer	9	2	77.77
German Credit	21	7	66.67
KrVsKp	36	6	83.33
Mushroom	22	5	77.27
Vote	17	10	41.18
Diabetes	9	9	0
Lung Cancer	57	4	92.98
Car	7	7	0
Hepatitis	20	12	40
Ionosphere	35	7	80
Liver Disorder	7	1	85.71
Sick	30	12	60
Vehicle	17	17	0
Tic-Tac-Toe	10	8	20
Spambase	58	25	56.89
Sonar	61	14	77.05
Page Blocks	11	10	9.09
Iris	5	2	60
Average	**24**	**8.78**	**63.42**

As is evident from the results listed in these tables, the cfs subset evaluation algorithm and the consistency subset evaluation algorithms perform good in increasing the accuracy of the classification, but poorly on the number of features reduced. A comparison of accuracy performance is depicted in Figure 4.1. Optimum Feature Selection algorithm increases the average accuracy of the 18 data sets by 0.5%, while the Cfs Subset Eval and Consistency Subset Eval algorithms increased the average accuracy by 1.79% and 2.15%.

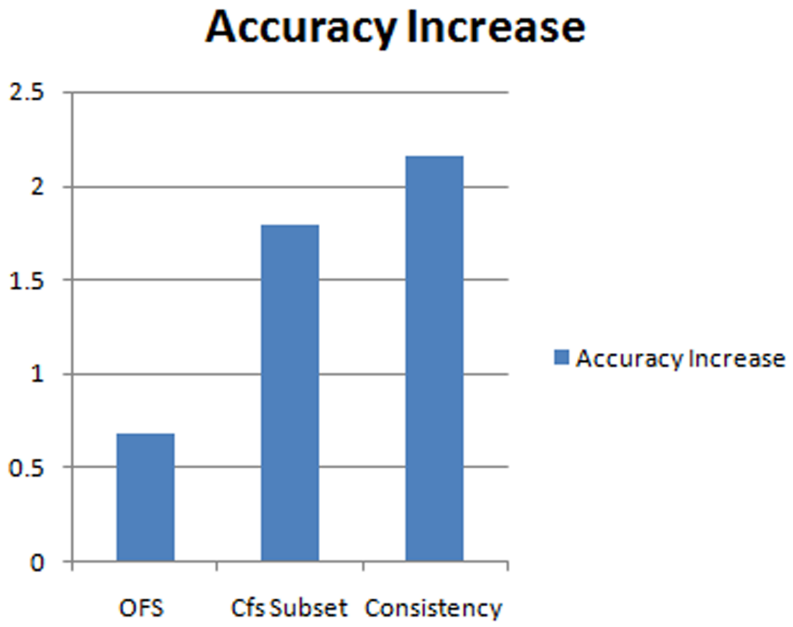

Figure 4.1: Accuracy Comparison of Feature Selection Algorithms

As is evident from Tables 4.6 and 4.7 and depicted in Figure 4.2 the Optimum Feature Selection algorithm performed really good on the number of features reduced. On an average 84.49% features were reduced while maintaining an average increase of 0.5% accuracy. The other two algorithms reduced the features by 70.83% and 63.42%.

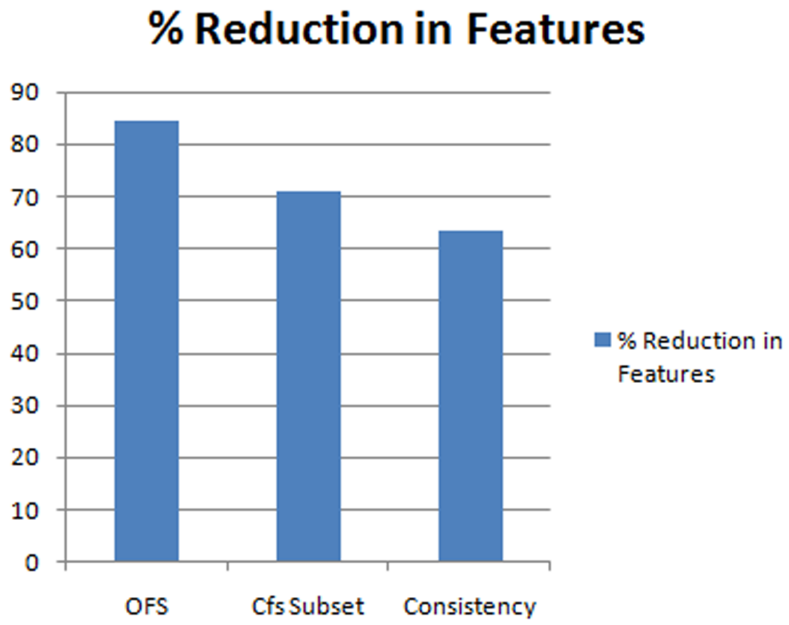

Figure 4.2: Comparison of % Reduction in Features

4.3.2 Single Node Fall Detection

This section investigates the performance of the Optimum Feature Selection algorithm on the classification accuracy for the data collected from single fall detection node. Also, the performance of the other two algorithms is discussed.

The data from the single node consists of 5 features. These are

- Acceleration X axis
- Acceleration Y axis
- Acceleration Z axis
- Gyroscope X axis
- Gyroscope Y axis

The classification results using these 5 features were discussed in Section 4.2. Following are the features selected by the Optimum Feature Selection algorithm.

- Acceleration X axis

- Acceleration Y axis
- Acceleration Z axis

Table 4.12: Confusion matrix using features selected by Optimum Feature Selection algorithm

A	B	C	D	E	F	G	H	
85	0	0	0	0	0	0	1	A:Forward Fall
0	82	0	0	0	0	0	0	B:Lying
0	0	90	0	0	0	0	0	C:Standing
0	0	0	87	0	0	0	0	D:Right Fall
0	0	0	0	80	0	0	0	E:Walking
0	0	0	0	0	87	0	0	F:Backward Fall
0	0	0	0	0	0	84	1	G:Left Fall
1	0	0	5	0	0	0	67	H:Near Fall

Table 4.13: Performance comparison of feature selection algorithms on data from single node

	NBC with OFS	NBC with Cfs Subset Eval	NBC with Consistency Subset Eval
Features used	3	5	4
Accuracy (%)	98.80	99.40	99.40

Using only these three features, the accuracy of classification is 98.80%, which is 0.60% lower than the accuracy achieved using all five features. Table 4.12 shows the confusion matrix for the results.

As can be seen from the confusion matrix, the increase in misclassification is only due to 3 additional features belonging to near fall being classified as some fall activity. Table 4.13 compares the performance of the three algorithms.

As is evident from the discussions in this section, both Cfs Subset Evaluation and Consistency Subset Evaluation algorithms are able to maintain the accuracy, but could not reduce the number of features considerably. Optimum Feature Selection algorithm on the other hand reduces the number of features considerably.

4.4 Real Time Classification

Section 3.5 discussed the methodology and steps involved to achieve real time classification. Based on the classifier selected and the chosen features, the final real time implementation is listed below.

- The selected classifier is the Naive Bayesian Classifier, so the parameters of the model are the means and standard deviations of each feature within each class and the prior probabilities of each class.

- For each incoming sample of data, the posterior probability of each class is calculated, assuming a Gaussian distribution and using the equation

$$P = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x-\mu)^2}{2\sigma^2} \tag{4.1}$$

- This probability is calculated for each feature within each class and all are multiplied together and with the prior probability to give the posterior probability of any given class.

- The class with the maximum posterior probability is assigned to the sample and sent wirelessly to the coordinator.

- Every sample of data, after classification is used to update the model of the respective class by updating the means and deviations of each feature.

Following are the results of the real time implementation.

- Every fall event is detected correctly.
- 3-4 samples of data for each fall activity.
- Leaning in any direction is detected as a near fall situation.
- Fluctuation between standing and walking positions.

4.5 Double Node Analysis

Section 3.6 described the two nodes and data collection process. Using two nodes, the movements can be divided into 9 different classes. This section lists the classification results for the data collected for two nodes. The results of the feature selection process and the consequent change in accuracies is also discussed.

The data consists of 458 instances and 9 features, divided into 9 different classes. Table 4.14 shows the confusion matrix for the classification results.

Table 4.14: Confusion matrix for Naive Bayesian Classifier (Double Node)

A	B	C	D	E	F	G	H	I	
23	0	0	0	0	0	0	0	0	A:Forward Fall
0	51	0	0	0	0	0	0	0	B:Lying
0	0	101	0	0	0	0	0	0	C:Standing
0	0	0	22	0	0	0	0	0	D:Right Fall
0	0	0	0	100	0	0	0	0	E:Walking
0	0	0	0	0	23	0	0	0	F:Backward Fall
0	0	0	0	0	0	20	0	0	G:Left Fall
0	0	0	0	0	0	0	17	0	H:Near Fall
0	0	0	1	0	0	0	0	100	I:Sitting

The accuracy of classification is 99.78%, with 1 sample of sitting position misclassified as right fall. This can be the result of an outlier in the samples of the sitting position. Selection of the most relevant features from this set of 9 features is done using the feature selection algorithm. Table 4.15 shows the comparison of the performance of three different classification algorithms.

Table 4.15: Performance comparison of feature selection algorithms on data from double node

	NBC with OFS	NBC with Cfs Subset Eval	NBC with Consistency Subset Eval
Features used	5	3	9
Accuracy (%)	97.82	95.41	99.78

The Optimum Feature Selection selected the following 5 features

- Acceleration X axis (Chest device)
- Acceleration Y axis (Chest device)
- Acceleration Z axis (Chest device)
- Acceleration Y axis (Thigh device)
- Acceleration Z axis (Thigh device)

From Table 4.15, the accuracy of classification using the reduced feature set is 97.82%. The Cfs Subset Eval algorithm reduced the features from 9 to 3, but with a considerable drop in accuracy. Consistency Subset Eval algorithm is not able to reduce the number of features at all and thus the accuracy values remains the same. Table 4.16 shows the confusion matrix for the result.

As seen in Table 4.16, the bulk of the resulting misclassification is between the walking and the standing positions. There are still no undetected falls. Since, accu-

Table 4.16: Confusion matrix for Naive Bayes Classifier on reduced feature set (Optimum Feature Selection)

A	B	C	D	E	F	G	H	I	
23	0	0	0	0	0	0	0	0	A:Forward Fall
0	51	0	0	0	0	0	0	0	B:Lying
0	0	100	0	0	0	0	1	0	C:Standing
0	0	0	22	0	0	0	0	0	D:Right Fall
0	0	6	0	92	0	0	2	0	E:Walking
0	0	0	0	0	22	1	0	0	F:Backward Fall
0	0	0	0	0	0	20	0	0	G:Left Fall
0	0	0	0	0	0	0	17	0	H:Near Fall
0	0	0	1	0	0	0	0	100	I:Sitting

rate distinction between standing and walking is not a priority issue, the reduction in feature set from 9 features to 5 features is a big advantage.

The Cfs subset evaluation algortihm reduced the number of features from 9 to 3. However, this is achieved at the cost of reduction in accuracy to 95.41%. Table 4.17 shows the confusion matrix for these results.

As is evident from the confusion matrix for reduced feature subset by Cfs subset evaluation algorithm, the misclassification rate is high. There are in total 21 misclassified samples, of which 9 are undetected fall samples. Though, the number of features selected is less than the ones selected by Optimum Feature Selection algorithm, but misclassification rate is too high to be accepted.

Table 4.17: Confusion matrix for Naive Bayesian Classifier on reduced feature set (Cfs Subset Eval)

A	B	C	D	E	F	G	H	I	
23	0	0	0	0	0	0	0	0	A:Forward Fall
0	50	0	0	0	0	1	0	0	B:Lying
0	0	100	0	1	0	0	0	0	C:Standing
1	3	0	14	0	0	4	0	0	D:Right Fall
0	0	0	0	98	0	1	1	0	E:Walking
0	0	0	0	0	21	1	1	0	F:Backward Fall
0	0	0	1	4	0	14	1	0	G:Left Fall
0	0	0	0	0	0	0	17	0	H:Near Fall
0	0	0	0	0	0	1	0	100	I:Sitting

Chapter 5

Conclusion and Future Work

The introduced device prototype developed by the FANFARE team can detect and classify different types of falls and different types of normal movements. Near-fall situations are also identified correctly and an early warning capability can be integrated to the device.

Naive Bayesian Classifier, which is a well established and popular algorithm for many machine learning problems is used for the fall analysis and proved to be efficient in classifying different movement types accurately and detect fall and near-fall situations. Using this classifier for the real time movement classification has resulted in a machine learning model which is easy to update, as it involves the recalculation of means and deviations and is computationally fast.

There is a fluctuation in classification between walking and standing, which intuitively speaking, can be rectified using more number of nodes. The adaptability of the device ensures that the model keeps adapting to an individual's peculiar gait, posture and lifestyle. The default model caters to the general conception of normal movements and falls. The model can be $fine\ tuned$ by changing the prior probabilities of each class. Even after this tuning, the device can continuously keep improving the classification process by updating the model continuously. The event of a false alarm poses a problem as now, the user must tell the device that it is a normal movement and then the corresponding data should not update the means and deviations of the $fall$ class. The algorithms to update the model are in place, however, the hardware capability to track and send the correct data for update, in the event of a misclassification is not yet in place.

The Optimum Feature Selection algorithm complements the Naive Bayesian Clas-

sifier by selecting only the most relevant features necessary for accurate classification of movement. There is a need to use this algorithm as the existing popular feature selection algorithms are unable to reduce the number of features considerably. This algorithm is shown to perform well not only for the movement classification problem, but also for other types of data sets. The test with other UCI data sets showed that the algorithm is fairly generalized and can achieve significant reduction in the number of features while keeping the classification accuracies at acceptable levels. The comparison with other feature selection algorithms both for movement classification data and UCI datasets, highlighted the difference between them. Where the other algorithms try to improve the accuracy, while having a relaxed approach to reducing number of features, the Optimum Feature Selection algorithm is suited for higher reduction in the number of features, while maintaining acceptable levels of accuracy.

With the addition of more number of nodes, more movement types can be accurately classified. Optimum Feature Selection algorithm can be used to select only the most relevant features from many features obtained from multiple nodes. As discussed earlier, in the event of a misclassification, the device should be capable of tracking the data and sending the corrected version for updating the model. This will ensure a more flexible, adaptable and robust classification model.

References

[1] F.G. Miskelly. Assistive technology in elederly care. *Age and Ageing, British Geriatrics Society*, 30:455–458, 2001.

[2] P. Hawranik. A clinical possibility: Preventing health problems after the age of 65. *Journal of Gerontological Nursing*, 17:20–25, 1991.

[3] O.P. Ryannen, S.L. Kivela, R. Honkanen, and P. Laipalla. Falls and lying helpless in the elderly. *Z Gerontol*, 25:278–282, 1992.

[4] J. Massion. Postural control system. *Curr. Opin. Neurobiol.*, 4:877–887, 1994.

[5] A. Dinh, Y. Shi, D. Teng, A. Ralhan, L. Chen, V.D. Bello-Haas, J. Basran, S-B. Ko, and C. McCrowsky. A fall and near-fall assessment and evaluation system. *The Open Biomedical Engineering Journal*, 3:1–7, 2009.

[6] J. Nilsson. *Introduction to Machine Learning*. First edition, 2005.

[7] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley, second edition, 2001.

[8] L. Yu and L. Huan. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 4:1205–1224, 2004.

[9] I. Guyon and A. Elosseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[10] J.N. Sixsmith. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Comput.*, 3:42–47, 2004.

[11] N. Noury, H. Thierry, V. Rialle, E. Virone, G. Mercier, G. Morey, A. Moro, and T. Pocheron. Monitoring behaviour in home using a smart fall sensor and position sensors. In *Special Topic Conference on Microtechnologies in Medicine and Biology, Annual International IEEE-EMBS*, pages 607–610, 2000.

[12] F.R. Allen, E. Ambikairajah, N.H. Lovell, and B.G. Celler. An adapted gaussian mixture model approach to accelerometry-based movement classification using time-domain features. In *Proceedings of the 28th IEEE EMBS Annual International Conference*, pages 3600–3603, 2006.

[13] N. Noury, A. Fleury, P. Rumeau, A.K. Bourke, G.O. Laighin, V. Rialle, and J.E. Lundy. Fall detection - principles and methods. In *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, pages 1663–1666, 2007.

[14] Y. Lee, J. Kim, M. Son, and M. Lee. Implementation of accelerometer sensor module and fall detection monitoring system based on wireless sensor network. In *Proceedings of Annual International Conference of the IEEE EMBS*, pages 2315–2318, 2007.

[15] C.J. Lord and D.P. Colvin. Falls in the elderly: Detection and assessment. In *Proceedings of the Annual International Conference of the IEEE EMBS*, 1991.

[16] G. et all William. A smart fall and activity monitor for telecare applications. In *Proceedings of the 20th International Conference of the IEEE EMBS*, 1998.

[17] N. Noury, A. Tarmizi, D. Savall, P. Boissy, G. Barralon, P. Virone, and P. Rumeau. A smart sensor for fall detection in daily routine. In *SICICA*, pages 9–11, 2003.

[18] S. Luo and Q. Hu. A dynamic motion pattern analysis approach to fall detection. In *Proceedings of IEEE International Workshop on Biomedical Circuits and Systems*, pages 1–5–8a, 2004.

[19] J.Y. Hwang, J.M. Kang, Y.W. Jang, and H.C. Kim. Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. In *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, pages 2204–2207, 2004.

[20] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, 1995.

[21] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 1:273–324, 1997.

[22] Chotirat Ann Ratanamahatana and Dimitrios Gunopulos. Feature selection for naive bayesian classifier using decision trees. *Applied Artificial Intelligence*, 17:475–487, 2003.

[23] L. Jiang, D. Wang, Z. Cai, and X. Yan. Survey of improving naive bayes for classification. *Advanced Data Mining and Applications*, 4632:134–145, 2007.

[24] M.F. Othman and T.M.S. Yau. Comparison of different classification techniques using weka for breast cancer. *International Conference on Biomedical Engineering*, 15:520–523, 2007.

[25] Weka at http://www.cs.waikato/ac.nz/ml/weka.

[26] K.J. Cios, W. Pedrycz, R.W. Swiniarski, and L.A. Kurgan. *Data Mining: A Knowledge Discovery Approach*. Springer, first edition, 2007.

[27] H. Zhang and J. Su. Naive bayes for optimal ranking. *Journal of Experimental and Theoretical Artificial Intelligence*, 20:79–93, 2008.

[28] S.V. Chakravarthy and J. Ghosh. Scale-based clustering using the radial basis function network. In *Proceedings of IEEE International Conference on Neural Networks*, pages 897–902, 1994.

[29] C. Doukas, I. Maglogiannis, P. Tragas, D. Liapis, and G. Yovanof. Patient fall detection using support vector machines. *International Federation for Information Processing*, 247:147–156, 2007.

[30] J. Rodriguez, A. Goni, and A. Illarramendi. Real-time classification of ecgs on a pda. *IEEE Transactions on Information Technology in Biomedicine*, 9:23–34, 2005.

[31] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and tecniques*. Morgan Kaufman, first edition, 2005.

[32] M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. page 359, 2000.

[33] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In *13th International Conference on Machine Learning*, page 319, 1996.

[34] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

Appendix A

Classification Code

Here is the MATLAB code for the real time classification process. It is given in two sections. The first one generates the model, and the other classifies a sample of data.

A.1 Classifier Model

function [M,SD,Prior,S] = NBclassify(data)
Function to build the Naive Bayesian Classifier model
M: Mean
SD:Standard Deviation
Prior:Prior probabilities
S: Data separated according to each class

separates all the classes
S=seperating(data);*Function to separate data* M={}; SD={};
Prior=[];

calculation of means and standard deviations for every feature in every class.
rows=length(data(:,1)); for i=1:length(S)
Mi=mean(S1,i);
SDi=std(S1,i);
end

calculation of prior probabilities of each class. for
i=1:length(S)
Prior(i)= length(S1,i(:,1))/rows;
end

A.2 Classification

function [class,L,prob,test] = realtime(M,SD,Prior,x)

this function returns the class of the unknown data x, based on the means(M), Deviations(SD) and Prior probabilities calculated from the test data
class=[]; prob=[]; test=[];

assigning class to new data x, each row at a time.
for
r=1:length(x(:,1))
L=[];

calculation of posterior probability of each class and storing the value in vector L
for j=1:length(Prior)
p=1;

calculation of conditional probability for each feature and multiplying all of them (independence assumption of Naive Bayesian Classifier)
for i=1:length(M1,1(1,:))-1
p= normpdf(x(r,i),M1,j(i),SD1,j(i));
end
L=[L,p];
end

assigning the class which has maximum posterior probability

[maximum;n]= max(L);

class=[class,n];
end
class
1 : forward fall
2: lying
3: standing
4: right fall
5: Walking
6: Backward fall
7: left fall
8: Near fall