M.Sc. Engg. Thesis

Adaptive Mobile Notifications Generation based on Physical Activities

by

Md. Mobasshir Arshed Naved

Submitted to

Department of Computer Science and Engineering

in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)

Dhaka 1000

September 2016

Dedicated to my loving family and friends

Author's Contact

Md. Mobasshir Arshed Naved
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology (BUET).
Email: mobasshir_arshed@yahoo.com

The thesis titled "Adaptive Mobile Notifications Generation based on Physical Activities", submitted by Md. Mobasshir Arshed Naved, Roll No. **0412052057P**, Session April 2012, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on September 20, 2016.

Board of Examiners

1. ____________________________
Dr. Md. Yusuf Sarwar Uddin — Chairman (Supervisor)
Associate Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

2. ____________________________
Dr. M. Sohel Rahman — Member (Ex-Officio)
Head and Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

3. ____________________________
Dr. S. M. Farhad — Member
Associate Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

4. ____________________________
Dr. Atif Hasan Rahman — Member
Assistant Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

5. ____________________________
Dr. Rajesh Palit — Member (External)
Associate Professor
Department of Electrical and Computer Engineering
North South University, Dhaka.

Candidate's Declaration

This is hereby declared that the work titled "Adaptive Mobile Notifications Generation based on Physical Activities" is the outcome of research carried out by me under the supervision of Dr. Md. Yusuf Sarwar Uddin, in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Md. Mobasshir Arshed Naved

Md. Mobasshir Arshed Naved

Candidate

Acknowledgment

I express my heart-felt gratitude to my supervisor, Dr. Md. Yusuf Sarwar Uddin for his constant supervision of this work. He advised me a lot in every aspect of this work and guided me with proper directions whenever I sought one. His patient hearing of my ideas, critical analysis of my observations and detecting flaws (and amending thereby) in my thinking and writing have made this thesis a success.

I would also want to thank the members of my thesis committee for their valuable suggestions. I thank Dr. S. M. Farhad, Dr. Atif Hasan Rahman and specially the external member Dr. Rajesh Palit.

In this regard, I remain ever grateful to my beloved parents, who always exist as a source of inspiration behind every success of mine I have ever made.

Abstract

Physical activity is an important factor that is considered for the prevention of diseases like diabetes or hypertension and for rehabilitation. Besides the advancement of technology and availability of smart-phones creates the opportunity to utilize the power of smartphone's sensors, for example, accelerometers, to support cost-effective behavioral intervention to promote physical activities. In this thesis, we attempt to identify basic physical activities of a user from smartphone's 3D accelerometer data and then suggest the user through mobile phone notifications the recommended level of physical activities he/she should undergo.

In our work, we analyze an existing dataset containing accelerometer data with labeled physical activities, namely standing, walking, stair-up and stair-down and a few others, and learn the patterns identifying various activities. Once the patterns are learned, we identify series of activities that a certain user performs from its mobile phone accelerometer data determining what portion of time the user spends in what activities. Based on this information, we develop suggestions of performing activities for that user by analyzing his/her current and required amount of physical activities within a time window using some predefined standard (amount of activities he/she must undergoes to be fit and healthy). These suggestions are propagated to the user suggesting him/her to make further engagement in physical activities through mobile notification system.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Regular Physical activity is essential for physical and mental wellness of almost everyone. Being physically active helps one to become pro-active and to stay independent. Regular physical activity over long periods of time can produce long-term health benefits also. Hence health experts say that every single person should be active every day to maintain their health. In addition, regular exercise and physical activity can reduce the risk of developing some diseases and disabilities that develop as people grow older. In some cases, exercise is an effective treatment for many chronic illness such as arthritis, heart disease, diabetes and so on [1]. Exercise also helps people with high blood pressure, balance problems, or difficulty in walking. It can be said that to lead a serene life, physical activities plays an undeniable role.

Due to daily work pressure and sometimes because of human nature, most often people forget to take part in doing certain amount of physical activities. Hence there is a need for a reminder that will motivate every individual to perform regular activities.

1.1 Motivation

With the advancement of technology during the last decade the evolution and availability of inexpensive and wearable sensors such as the accelerometer, GPS receptor, cameras and microphones along with computational expansion in terms of both hardware and software, has unlocked a new field of opportunities in the mobile applications domain. These mentioned sensors can be easily found in any smartphone, widespread in almost all the mobile telecommunications markets. This sensoring world is changing the paradigm of human relation with machines.

As said earlier, physical activity plays an important role by decreasing risk of obesity, diabetes, hypertension and many other chronic diseases [1]. The availability of smartphones pave the way to support cost-effective behavioral intervention to promote physical activities.

Smartphones can potentially help encourage people to exercise and record their physical activity. As these are equipped with powerful sensors as accelerometers, gyroscopes, and orientation sensors, these can be used to recognize, monitor and recommend various types of physical activities. Different sensor readings can indicate different kinds of activity. For examples, unchanging accelerations over time can indicate sedentary activity such as sitting or standing, whereas sharply varying vertical accelerations can indicate running or climbing stairs. Most often obese people and diabetic patients have to do a certain amount of exercise every day to lose their weight and to control their blood sugar level respectively [2, 3]. Not only for them but also for everyone, smartphones become the best tool to record individuals daily activities, because people usually carry their phones every day and everywhere. Hence people could check out their daily activity record on the phone at the end of each day or at some free times of a day, and they will have a clear picture of their physical activity performance during the day. Moreover smartphones can be used to suggest each of these individuals about their lacking in physical activities and can be used to recommend certain amount of activities that will fulfill the lacking.

1.2 Contribution of this Thesis

The overall goal of this thesis is the development of a physical activity recommendation system, with two main objectives. On the one hand, the goal is to monitor how far individuals meet professional recommendations. This means the intensity estimation of performed activities: to distinguish activities of light, moderate and vigorous effort. On the other hand, to provide suggestion to maintain a healthy rate of involvement in physical activities. In this thesis, we attempt to identify basic physical activities of a user from smartphones accelerometer data and then suggest the user through mobile phone notifications the recommended level of physical activities he/she should undergo.

We build this model from a dataset collected for previous works in heterogenity in activity recognition [4] having as inputs the timestamps, the X, Y, Z axes acceleration values and a label of which activity is taking place for each record. The framework for the development of feature extraction relies on the knowledge about each activity considered in order to capture relevant characteristics aiming the best discrimination between activities. Then we find a modeling approach for physical activity recognition that enables the minimization of the number of features needed to have a good classification model. We have tried out various state of the art activity recognition classification and selected the one which suits our objective. We build the activity suggestion system that will provide suggestion by calculating the need of physical activities an user must undergo to maintain a balanced life on regular intervals. This system will communicate with the user by invoking notifications in the smartphones of them.

The contribution of our thesis can be summarized as follows:

- We classify physical activities from smartphones accelerometer dataset.
- We measure the level of the extracted activities (energy expenditure) within a time window.
- We generate suggestions by analyzing users current and required amount of performing activities on the basis of predefined standard.
- We calculate the number of notifications for notifying users to motivate them in performing suggested amount of activities.

The remainder of this thesis is organized as follows. Chapter 2 introduces some related works. Chapter 3 describes Activity Recognition using 3D accelerometer data in details. Chapter 4 discusses the Notification and Suggestion system in brief. Chapter 5 provides some evaluation of the system with different cases. Conclusion and future work are discussed in Chapter 6.

Chapter 2

Background and Related Work

More attention has been paid to activity recognition in the field of mobile communication because of the increasing availability of accelerometers in consumer products, like smart-phones, and because of its many potential applications. In this chapter we start the work by analyzing the state of the art of activity recognition based on accelerometer data. In relation to the background about this topic the overview can be found in section 2.1.

The data mining approach for activity recognition involves many techniques as feature extraction and learning methods. The feature extraction paradigm is approached in section 2.2 along with classification techniques. Finally in section 2.4 we point out various monitoring research works that are carried away by the researchers in recent times.

2.1 Accelerometer Sensor based Activity Recognition

In order to deal with activity recognition based on accelerometer data, there exist numerous numbers of approaches though it can be said as a topic of recent interest. The common approach to the single user activity recognition problem [5, 6, 7], is to apply the techniques proposed via a two-stage process. First they collected accelerometer raw data by extracting from a batches of data (normally called sliding windows) and derived features from them. Then they have applied one or more classifiers to recognize different activities considered in their works.

Bao & Intille [5] have the most referenced work in this topic and can be found almost in all the works done after 2004. They have collected a sample from 20 subjects in an unsupervised way, using a network of sensor placed simultaneously in different parts of the body. In this work [5] they have two conclusions very important for the direction of the research done afterward. The first indicates that there are good possibilities to create a generalized model. Secondly they have concluded that the most discriminative accelerometer in terms of which activity is taking place was placed on the hip, what can be a good indicator for the pocket in pants is a good placement to collect the input data for the purpose of having a good model and this conclusions can be also found on other works [7].

In another study, Krishnan et al. [7] examined seven lower body activities using data collected from ten subjects wearing three accelerometers. This method was tested in supervised and semi-naturalistic settings. Tapia et al. [8] collected data from five accelerometers placed on various body locations for twenty-one users and used this data to implement a real-time system to recognize thirty gymnasium activities. A slight increase in performance was made by incorporating data from a heart monitor in addition to the accelerometer data. Mannini and Sabitini [9] used five triaxial accelerometers attached to the hip, wrist, arm, ankle, and thigh in order to recognize twenty activities from thirteen users. Various learning methods were used to recognize three postures (lying, sitting, and standing) and five movements (walking, stair climbing, running, and cycling). Foerster and Fahrenberg [10] used data from five accelerometers in one set of experiments and from two of those accelerometers in another for activity recognition. Thirty one male subjects participated in the study and a hierarchical classification model was built in order to distinguish between postures such as sitting and lying at specific angles, and motions such as walking and climbing stairs at different speeds. Subramayana et. al. [11] addressed similar activities by building a model using data from a tri-axial accelerometer, two micro-phones, photo-transistors, temperature and barometric pressure sensors, and GPS to distinguish between a stationary state, walking, jogging, driving a vehicle, and climbing up and down stairs.

While these systems using multiple accelerometers or a combination of accelerometers and other sensors were capable of identifying a wide range of activities, they are not very practical because they involve the user wearing multiple sensors distributed across their body. This could work for some short term, small scale, highly specialized applications (e.g., in a hospital setting) but would certainly not work for the applications that we want to foresee.

Some researches have focused on the use of a single accelerometer for activity recognition. Long, Yin, and Aarts [12] collected accelerometer data from twenty four users using a triaxial accelerometer worn without regard for orientation at the users waist. Data was collected naturalistically, and decision trees as well as a Bayes classifier combined with a Parzen window estimator were used to recognize walking, jogging, running, cycling, and sports. Albert et. al. [13] used a single accelerometer attached to the left waists of five users. Standing, sitting, walking, lying, and running were all recognized with high accuracies using fuzzy c-means classification.

Another approach was done by Miluzzo et al. [14] by the development of a mobile application that involves several classifiers, some of them working on the phone and some back-end classifiers, producing several levels of classification. The mobile, from the raw accelerometer data collected by the built in accelerometer, calculates the mean, the standard deviation of the acceleration and the number of peaks in each batch of data, applying the sequence based sliding windows [15]. Fom these features they propose a decision tree was trained using J48 in WEKA workbench [16], to classify which activity is taking place. In this work they deal with the activity recognition problem in a different framework from the previous works, where all the sensoring and classification takes place on the mobile device that people already use, but they dont deal with the mobile orientation problem, and the results in the test dataset are not that good. Yang [17] developed an activity recognition system using the Nokia N95 phone to distinguish between sitting, standing, walking, running, driving, and bicycling. This work also explored the use of an activity recognition model to construct physical activity diaries for the users. Lin et al. [18] focused on physical activity recognition by varying the position and orientation of smart-phone and by using SVM classifier.

2.2 Feature Extraction and Classification Techniques

To extract features and classify a dataset, the naive approach is to look inside a machine learning software package, e.g. Weka Explorer, MATLAB, R and try to find the algorithm that adjusts the best to the dataset. Although there are several proposals in literature and software packages in order to deal with different kinds of problems and characteristics in the datasets, there will be always a problem to deal with a noisy dataset and/or, even more, an inadequate description of the space domain. There are already learning techniques to deal with noisy dataset [19] as decision trees or k-NN but the construction of features that represent the space in the optimal way, creating domains in the space represented without overlapping it will always improve the results of a classifier.

Preece and Goulermas' [20] work can be said as the main reference in terms of feature extraction for this thesis once it is comparing different techniques to extract features. These authors made a comparison of 14 different works where several feature extraction techniques, in order to classify activities from accelerometer data, were classified as time-varying acceleration signal, frequency analysis and wavelet analysis. Features extracted from the three axes as means, standard deviation [21], first quartile and third quartile or correlations between axes are commonly used for the classification process but they need some modifications in order to be reliable in a real application environment. Bayat et al. [22] proposed a low pass filtering mechanism and also provide suggestions about how to pick the features. He also compared between various combination of classifiers that might induce more accurate result.

2.3 Activity Monitoring

There exist few works on activity monitoring and promotion. Vathsangam et al. [23] presented a pilot study on using an application named Strive to monitor physical activity of users and to motivate them to stay active. They provided some users with the app and advised them to do their normal activities. After a week they collected the data from the users and show them a visual (graphical demonstration) to point out the amount of activity they had undergone.

By Showing these data they tried to motivate the users to become more active by looking into their daily activity graph. Lane et al. [24] also developed a mobile application to monitor and promote physical activity, social interaction and sleep time. Besides there are some other works [25, 26] that also deal with monitoring and promotion facilities.

Chapter 3

Activity Recognition from 3D Accelerometer Data

Our goal of this thesis is to promote and suggest physical activities to ensure the wellness of the users. By frequent monitoring of users' activities through the accelerometer reading from their smartphones, we are able to provide suggestions on the basis of some benchmark conditions (such as total calories burned on a given time period an so on). The system that we build to support the goal is comprised of two major components: *Activity Recognition System* and *Activity Suggestion System*. The first component is responsible for training a model for activity recognition from a time series of 3D accelerometer data. It predicts users activity sequence (using the trained model) by analyzing their smartphone's accelerometer data. The former component comprised of activity suggester and notification generator uses the sequence of activities to determine an aggregated level of physical activities for an individual in terms of some measurable quantity (i.e. amount of calorie expenditure). The activity suggester suggests activities if the level of activities fall short a certain preset rate; the notification generator generates the notification that contains the suggestions and propagates it to users' smart-phones. The overall scenario can be depicted using the following Figure 3.1.

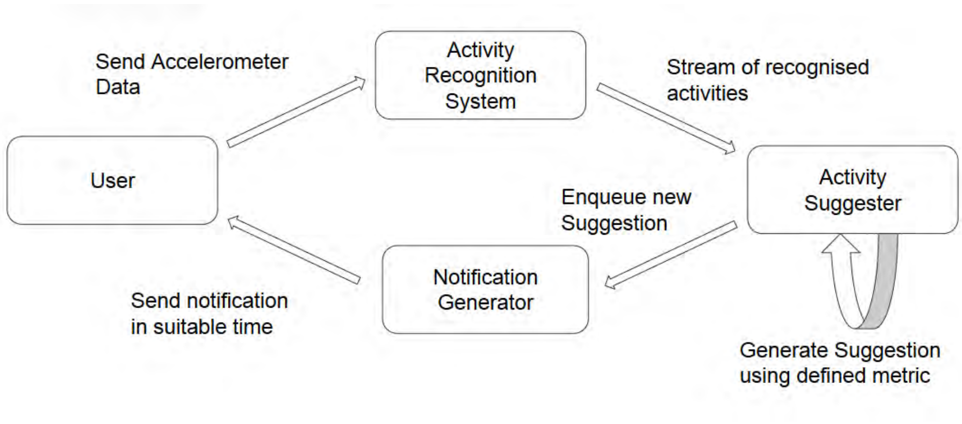


Figure 3.1: Block Diagram of Overall System

In this chapter, we focus on the *Activity Recognition system*; we describe how to build a model that can recognize different activities for a span of time from mobile phones accelerometer data. As we use smartphone's accelerometer to get data that comprised the basis of this work, hence we start this chapter with the overview of accelerometer data.

3.1 Accelerometer Data

An accelerometer is a sensor that can measure the force acting upon it, be it from physical acceleration or from the Earths gravity. Most accelerometers measure acceleration along either 2 (X and Y) or 3 axes (X, Y and Z)(Figure 3.2). The majority of smartphones are fitted with a triaxial accelerometer (three axes), since the two most used smartphone software platforms (Android and iPhone) require this type of sensor.

The accelerometer in smartphones is used by the operating system of the phone to perform orientation-sensitive tasks (such as rotating the screen to match the view of the user), as well as by various applications installed on the phone. With this useful capability already built into the phone, our aim is to use this sensor for the task of capturing the forces acting on the phone due to the owner's activities. The data thus produced will be the starting point for training an automated system to recognize the person's activity from acceleration data.

For obtaining a system to recognize activities, we build upon the large body of work in the Data Mining field [4]. The input data considered for this work will be streaming data from

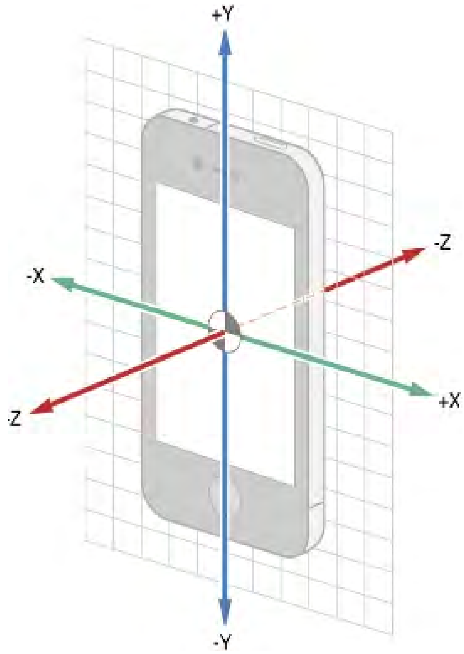


Figure 3.2: Accelerometer readings of a smartphone

accelerometer sensors collected from smartphones with Android systems (although the proposed method could work just as well on alternative smartphones). More formally, the definition of our initial problem is: from a stream of accelerometer data, $Acc = \{acc_1, acc_2, \cdots, acc_n\}$ with $acc_i = (acc_x,\ accy_y,\ acc_z)$,we predict activities from a predefined set of activities $A = \{a_1, a_2, \cdots, a_k\}$. We measure the aggregated level of physical activities an user is performing. On the basis of this, we provide suggestion of doing further activities through mobile phone notification system. After getting the accelerometer data as input we move onto the specific task of activity recognition. The physical activity recognition process consists of several basic functionalities or steps. These are described in the later sections of this chapter.

3.2 Overview of Activity Recognition System

The block diagram enlisted in Figure 3.3 simplifies the activity recognition process that we used in our thesis.

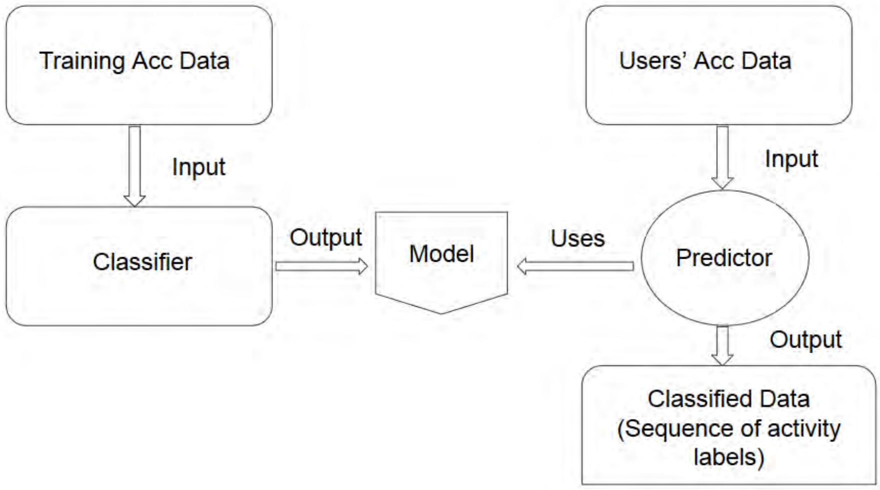


Figure 3.3: Block Diagram Activity Recognition Process

The detailed mechanism of Activity Recognition System is explained starting with Data preparation in the following sections.

3.3 Data Preparation

3.3.1 Noise Reduction

Filtering noise out of sensor data is an important first step while working with any system. It is quite common that accelerometer data possess noises. To mitigate its effect we applied a low-pass filter [27]. A simple low-pass filter for data in time domain is a smoothing function. In other words, the filtered signal is smoother and less dependent on short changes. We used a low-pass filter to reduce the influence of sudden changes on the accelerometer data. It uses a low-value filtering factor (β) to generate a value that uses 20% of the unfiltered acceleration data and 80% of the previously filtered value. This factor was chosen empirically.

$$Low_{acc_x} = \beta \times Prev_{acc_x} + (1 - \beta) \times Cur_{acc_x}$$

$$Low_{acc_y} = \beta \times Prev_{acc_y} + (1 - \beta) \times Cur_{acc_y}$$

$$Low_{acc_z} = \beta \times Prev_{acc_z} + (1 - \beta) \times Cur_{acc_z}$$

where $\beta = 0.8$ and *Prev*, *Cur* and *Low* are previous, current and low pass filtered value of input accelerometer data respectively.

3.3.2 Selection of Sliding Window

Classifying accelerometer data into physical activities falls under the general problem of discrete time-series classification. In general, associating time-varying data with classes present problems for machine learning algorithms inferring rules from a set of examples, where each example is considered separately as an input vector with an output vector. This assumption, that separating the examples produces no informational loss, requires careful construction of the input vector. Importantly, the relevant temporal information included in the time-series has to be accounted for. The most common way to account for this temporal information in activity recognition [28] is to use a *sliding-windows* approach. It helps to make the model actualized. This sampling decision can be a sequence-based window with a fixed dimension of particular size or a timestamps-based window of a fixed duration as explained by Babcock et. al [15].

The sequence-based window of size l consists in deciding a dimension, l, to extract information from the incoming data. When new data arrives to the processor, it dismisses the older data. Lets assume a data stream $x_{1i},\ x_{2i},\ \cdots,\ x_{ni}$ where i is the variable and a window of dimension l , and $l < i$. The extraction of features in this case will start from 1 to l and then when the $l + 1$ instance arrives the first instance is dismissed for this local classification.

To extract features, the main challenge is to decide a dimension of this window, but normally this decision is made from understanding the problem we are trying to classify, or from recommendations if the literature related to the problem has already applied this technique. Although there are different approaches to decide the dimension of this window, it is always convenient to have in mind that longer windows will be richer in terms of information, thus producing normally better features to be used in the classification process. Smaller windows will have the ability to reflect more quickly different classes. There will be always a trade-off between quality of features produced and the ability to recognize changes in terms of the classification. In our work we have selected this time window to be of 10 seconds in length.

Another consideration is the amount of jump between consecutive windows. However, features extracted between two consecutive windows will not contain much new information and will increase the running time of the algorithms. Previous research in activity recognition has successfully employed 50% overlap between adjacent windows [29, 5]. We use this assumption in our work too.

3.3.3 Feature Extraction

In order to get efficient activity classication, the feature extraction phenomena from smartphone accelerometer is very crucial. We performed the features extraction for both time and frequency domain, in order to understand which is the most important to do the classication. In the time domain the features were based on the following studies [30] [31] and the features in the frequency domain were based on [32] [33]. We have selected a total of 14 features for this thesis work.

3.3.3.1 Time-domain Features

These are features derived directly from the window, and usually of a statistical nature. These features include mean vector, standard deviation vector, euclidean norm of mean vector, euclidean norm of the standard deviation, correlation values and 25^{th} and 75^{th} percentile values.

For our system we get time series accelerometer data indexed by time as input. Let $acc(t)$ be the time series data that comprised of data from 3 three axes and can be written as $acct(t) = (acc_x(t), acc_y(t), acc_z(t))$. The number of data present in a *sliding window* is defined by N. We can calculate the time-domain features as follow:

- **Mean vector:** The mean vector among x axis can be expressed as:

$$\overline{acc_x} = \frac{1}{N}\sum_{t=1}^{N} acc_x(t)$$

And similar expression follows for other axes.

- **Standard deviation vector:** The standard deviation module of x axis component can be written as:

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{t=1}^{N}[acc_x(t) - \overline{acc_x}]^2}$$

  Similar kind of expression follows for other axes.

- **Euclidean norm of mean vector:** The module of mean vector can be calculated as:

$$||\overline{acc}|| = \sqrt{\overline{acc_x}^2 + \overline{acc_y}^2 + \overline{acc_z}^2}$$

- **Euclidean norm of the standard deviation:** The calculation of standard deviation module of each component is as follows:

$$||\sigma^i|| = \sqrt{{\sigma_x^i}^2 + {\sigma_y^i}^2 + {\sigma_z^i}^2}$$

- **Correlation values:** The correlation helps to establish the relationship between the axes and understand in which direction the signal presented a higher variation. The correlation between two axis (x and y) is obtained as follows:

$$corr_{xy} = \frac{1}{N-1}\sum_{t=1}^{N}\frac{acc_x(t) - \overline{acc_x}}{\sigma_x} \times \frac{acc_y(t) - \overline{acc_y}}{\sigma_y}$$

  Similarly we can have expression for correlation between x, z and y, z axes.

3.3.3.2 Frequency-domain Features

These are features derived from the Fourier transform of the windowed data. These features include the specific frequency components, the amplitude of spectrum, the peak frequency in the spectrum and number of peak values under a certain value in the spectrum.

- **FFT:** A Fast Fourier Transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse. A Fourier transform converts time to frequency and vice versa; an FFT rapidly computes such transformations. The calculation of the Fast Fourier Transform, that refers to a way the discrete Fourier Transform (DFT) can be calculated efficiently, by using symmetries in the calculated terms. The symmetry is highest when n is a power of 2, and the transform is therefore most efficient for these sizes. We have done the DFT for y axis data only.

$$ACCY_m = \sum_{n=0}^{N-1} acc_{y_n} e^{-j\frac{2\pi}{N}mn}$$

where $m = 0, 1, 2, \cdots, N-1$.

- **Peak Frequency:** Some activities produce quite clear peaks in the Fourier-spectra and extracting features at these points lead to valuable features. To extract features at these points a clear definition of what constitutes a peak needs to be formulated. In this thesis a simple peak measure is used.

After extraction, all features were organized and normalized in a suitable manner for the subsequent classication process.

3.4 Classification

After the step of data pre-processing and computation of all additional attributes for feature extraction as described in previous section, the final attribute vector gets created. This attribute vector is passed to the classification model which tries to recognize (classify) the appropriate activity of the user.

3.4.1 Classification Techniques

There exists numerous numbers of classification techniques or we can say learning algorithms, starting with simple Naive Bayes, k-Nearest Neighbors, Linear Regressions or non-linear models as Decision Trees, Neural Networks and then moving to ensemble learning methods; it is easy to achieve the conclusion that this area is already huge and growing. At the same time there is a concern in relation to the decision of which learning methodology should be implemented to a specific problem. Here in our case we have gone through 4 classification models to predict the incoming activities. They are: k-Nearest Neighbors, Generalized Linear Regression, SVM and Random Forest model. They are discussed below:

3.4.1.1 Support Vector Machine (SVM)

First developed in the late seventies [34] support vector machines, or SVM, have received a great deal of attention from the machine learning community. It has by now developed a strong mathematical foundation and rigorous statistical analysis, which could be contrasted by the previous methods which rely on heuristics or analogies of human learning. The idea behind SVM is to find the plane which maximizes the margin between the input data of two classes. As we intended to differentiate between more than two classes (here activities) we have to deal with multi-class SVM classification formulation.

3.4.1.1.1 Multi-class SVM: As by nature SVM is a binary classifier, some technique has to be applied to achieve multi-class classification. A simple approach is to train several binary SVMs and combine their outputs into a single classification. To achieve this result two most common methods are:

- **One-against-one approach:** This approach trains $n(n-1)/2$ classifiers, where n is the total number of classes that needs to be classified. Here each classifier is trained to distinguish between two classes. The class that has been predicted by the majority of the classifiers is considered to be the output of the complete classifier.

- **One-against-all:** This approach trains n classifiers, one for each label. Each single SVM is trained with one class marked as positive and all other classes treated as negative. The class of the SVM with the highest decision value is chosen as the output of the complete classifier

3.4.1.2 k-Nearest Neighbors

k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification. Here the input consists of the k closest training examples in the feature space. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor. It is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

3.4.1.3 Random Forests

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.

3.4.1.4 Generalized Linear Regression

Generalized Linear Models provide a unified way to fit responses that do not fit the usual requirements of least-squares fits. We have used this method to classify various activities by generating model of separate activities. Later we predicted the outcome using this generated model.

Among all the described model we have found that multi-class SVM classifier provides classification with greater accuracy. So we stick with this classifier for the later use.

3.4.2 Classifier Performance Measurement

Algorithms researchers usually calculate the accuracy of the algorithm (i.e., the percentage of correctly classified examples) and the error predictions to evaluate the performance of different classifications. To estimate these measures there exist various methods and different kinds of measures that can be used to select the best model from available options available. The most common technique to do so is called cross validation.

3.4.2.1 Cross Validation

The idea behind this validation techniques is to train the algorithm on a subset of the data and then evaluate it on the part of the data that is unseen during training. This is how a cross validation procedure starts, but when the training and evaluation is done, it switches a chunk of the training data with the evaluation data and retrains the algorithm. This procedure is then iterated until all the data has been trained and evaluated on. The average of the performance measure for all iterations is the cross validated estimate of the algorithm. The most known way of splitting the data is k-fold cross-validation. In k-fold cross-validation data is split randomly into k sets of approximately equal size. In each iteration the algorithm is trained on $k-1$ sets and evaluated on the left out set. Most commonly values set for k are: all samples, also called leave-one-out, 10 and 5. In our case we have used Leave-one-subject-out cross validation that is leave one user data out as a test set and use other users data as a train set. Besides this we have employed tradition 10 fold cross validation technique to ensure the proper validation of classified data.

3.4.2.2 Confusion Matrix, Accuracy and F-score

We use three matrices to evaluate the performance of each algorithm throughout the thesis. They are: The confusion matrix, accuracy and F-score. Among these the confusion matrix is the most informative one. It comprises of a table that compares all classification values to the true class value. The other two matrices are calculated from this table. However, the amount of information included in the table makes the measure difficult to comprehend, and cluttered, thus it is used only when other measures, are not sufficient enough. Accuracy is renowned as the simplest measure and is defined as all correctly identified examples divided by the total number of examples present in the classification data. This measure is easy to understand and ubiquitous in activity recognition. Most of referenced articles about activity recognition employ this measure.

For multi-class problems, class specific performances is taken into account for further depth analysis. Let consider a multi-class problem with errors defined with respect to a particular class A, represented by the following table:

	classifier outputs (class A)	classifier outputs (others)
true class (A)	true positive(t_p)	false negative (f_n)
true class (others)	false positive(f_p)	true negative (t_n)

Table 3.1: Classifier Outcome

Accuracy is defined for a single class as the true positives divided by the sum of true positives and false negative. It somehow neglects the false positives of the outcome. As a result this measure can not differentiate between a classifier being overly sensitive to a class and one having high discrimination for that class.

The third measure that we used in the thesis, the F-score is a performance measure that takes this effect into account by putting equal emphasis on finding the true classes (i.e. single class accuracy) that is called recall in information retrieval and being discriminative is called precision (i.e. not returning many false negatives).

$$precision = \frac{t_p}{t_p + f_p}$$

$$recall = \frac{t_p}{t_p + f_n}$$

The harmonic mean of these two measures form the F-score which lies between 0 and 1.

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

3.5 Training and Test Dataset

To recognize activities from newly arrived accelerometer data, a generated classification model is needed. This model is created using the training datasets. Training dataset contains accelerometer data but here the data is associated with activity level. So that we are able to differentiate between accelerometer data of different activities. In our thesis we provided a label to each accelerometer data on the basis of activity level. Such as for set of activities $A = \{a_1, a_2, \cdots, a_k\}$, we provide labels $1, 2, 3, \cdots, k$. This accelerometer data along with the labels of activities then fetched into appropriate classifier using a particular time window for detection that hereafter generates the model that is used for classifying new accelerometer data (can be termed as test data). This model is generated by following above mentioned steps and the performance of the classifier that is used is measured. If it provides the best results, then this model becomes the core component that is used later for detection of activities from newly coming test datsets. The output or the detection of the activities results in a sequence of label that we provided before. This label points to the activities that are performed by the user.

3.6 Summary

In short for physical activity recognition, accelerometer data works as input. Then this data is processed thoroughly using filterization techniques. Later sliding window is selected along with the extraction of features that are used for classifying various activities. Then using the features set and using suitable classification model the inputted accelerometer data results in specific set of activities. This recognized activities creates the possibility of recommending and suggesting further activities on the basis of some pre-defined factors which is discussed in the following chapter.

Chapter 4

Activity Suggestions and Adaptive Notifications

Recognized sequence of activities works as input for the activity suggestion system that suggests few of the activities from a varied activity set in order to fulfill a certain quota of activities in a particular time period. This suggested activities are treated as mobile notification that help an user to maintain a recommended amount of physical work in his/her day to day life. This chapter explains the activity suggestion procedure as well as the providing adaptive notifications also.

4.1 Activity Suggestion System

Activity Suggestion System is the core component of this thesis work. After the activity recognition phase this component comes into play. Activity recognition provides us with a sequence of activity labels, which indicates what activity the user is performing for a certain span of time. These labels determines the intensity as well as the continuation of a certain activity. Let $A = \{a_1, a_2, \cdots, a_k\}$ define a set of activities and they are associated with labels $1, 2, 3, \cdots, k$ respectively. So we are able to determine an activity by the activity label it holds. Such as activity label of 1 represents the a_1 activity from the activity set and so on. As we discussed in the previous section, training data is labeled with activity level or name and we later generated the classifier model by using these training accelerometer data along with our provided labels. When a new data encounters, then the model becomes responsible for the classification of these data and results in a sequence of recognized activity labels that can be used for appropriate activity recognition for a particular time frame. The block diagram of Figure 4.1 shows the output (sequence of activity labels) that is used by the suggestion system to generate suggestion notification:

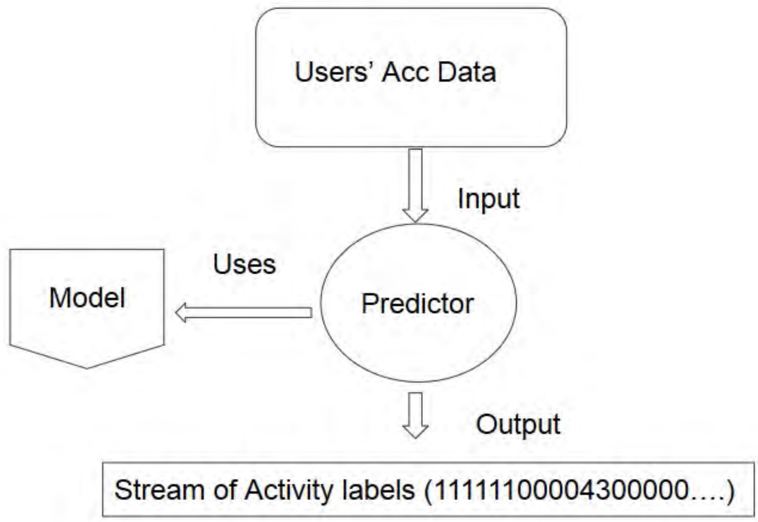


Figure 4.1: Sequence of activity labels results from Activity Recognition system

The recognized activity label sequence sometimes may contain some noisy data. To get rid of these anomalies we have certain mechanisms. These are described in the next section.

4.1.1 Anomaly Reduction

The time frame of our sliding window is of 10 seconds in length. So the recognition system or the classifier provided us with a label for each 10 second chunk of accelerometer data. When we get a sequence of data like "1111kkkkk4422" (here $1, 2, 3, \cdots, k$ are labels as before), we can interpret it as a total of 2 minutes data (as there are 12 labels and each of which is for 10 second window length of accelerometer data) of whose first 40 seconds a_1 activity is performed followed by 40 seconds of a_k and 20 seconds of a_4 and 20 seconds of a_2 activity.

Sometimes we find singleton activity that is performed only for a very short period of time (minimum duration of 10 sec). After and before this activity no similar activities are performed, rather they are all different around. This indicates that this very activity is misclassified amid a series of other activities. Such types of classification may be resulted in due to some noisy value from the classifier (Figure 4.2).

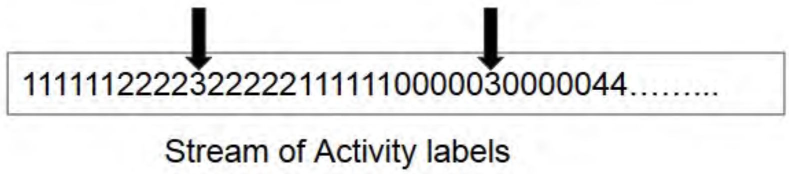


Figure 4.2: Presence of anomaly in activity detection

In order to eliminate or reduce such types of anomalies, we have considered the following two approaches.

- Treating each of the activities that is performed as an *independent* entity. In this approach if the activity labels that results from classification are small in width then we will treat this activity as a non dominant one and will decide that this activity label is the result of an anomaly. So we will consider that the activity that precedes or succeeds this certain activity would be the correct activity and will replace the anomalous activity with the correct one. Here the most critical problem is how to define the smallest width to detect anomaly. Here by observing practical scenario and analyzing several data sources we have empirically fixed the length of time that a certain activity has to be performed to be considered as proper activity rather than the anomalous one is around 20 seconds. So any activity having performing time less than this value will be treated as anomalous activity and the correction scheme mentioned above will be applied.

- Treating each activity as *dependent* to certain extent on one another. In this approach we will consider the fact that every activity of the activity set has dependency on each other. For this reason we have to determine a dependency matrix for all the activities of the activity set. By using the dependency value between two activities and the posterior probabilities that results from model classification along with activity label we can derive a relation of occurring an activity just after a certain kind of other activity. Using this approach we can also mitigate the effect of anomalous activity.

We have followed the first approach in this thesis. As we were not able to find the dependency matrix in existing works and because of this insufficient information of clear dependency in collected data, we currently skipped the second approach to avoid misleading correcting scheme and thus stick to the first approach for reducing anomalous activities.

4.1.2 Assigning Weights to Activities

It is easy to observe that each activity in activity set A differs in term of intensity and energy expenditure associated with that activity. Different activity has different level of physical engagement from a user perspective. For example, climbing a stair up is certainly more engaging then standing still, and biking is surely higher than stair down. In order to provide a more realistic suggestion about physical activities, each of the activity from the set has to be treated according to their energy expenditure level. Let we have a set of activities consisting of several activities such as standing, sitting, walking, stairs climbing, biking and so on. The intensity for each of them differs quite a bit. For example, in our case standing and sitting might fall under activities of low intensity where walking might fall under activity of moderate and climbing stairs or biking might fall under activity of vigorous intensity.

To quantify the level of physical engagement in doing a certain activity, we introduce a *weight factor* that is associated for each member of the activity set. So, let $W = \{w_1, w_2, \cdots, w_k\}$ be the weight set for activities in activity set A. One way to determine weight factors is to determine how much calorie is burned per activity per unit time. The work at [35] lists a couple of activities with their associated calorie expenditure per day. We use these values. We normalize these values and was able to prepare a reliable weight set associated with the activity set. In the process, we were able to interpret our work of suggesting physical activities on the basis of energy level expenditure (amount of work to be done) to the problem of suggesting activity to maintain calorie expenditure of an user.

4.1.3 Problem Definition of Activity Suggestion

We have a series of accelerometer data as input at a certain sampling rate. As mentioned before, we made chunks of accelerometer data. The width of each chunk equals to 10 seconds. Each of the chunk is predicted to be an activity from the activity set $A = \{a_1, a_2, \cdots, a_k\}$. We consider a series of such activities (i.e., 60 activities in a total of 10 minutes) which represents an activity bundle. This length of such bundle in time (i.e., 10 minutes) constitutes an *epoch*, whose length is denoted by τ.

Each user is expected to burn a certain amount of calorie per epoch (which in turn corresponds to a certain level of physical activities rendered in that epoch). Let ρ be the prescribed calorie expenditure per epoch for an user. So at epoch t (starting from epoch 0), the user is supposed to burn a total of $\rho \times t$ calorie. But the user may not make the full expenditure depending on the physical engagement he/she made upto that epoch. let $C(t)$ be the current cumulative expenditure upto epoch t by that user. So, the difference between the expected amount and the actual performed one gives the amount of *deficit* upto that time. Hence, we have

$$deficit = \rho \times t - C(t)$$

.

A positive value of deficit suggests that the user in indeed falling behind the expectation in terms of calorie expenditure which needs to rectified by sending active notifications and activity suggestions. This suggestion will only be triggered if the level of deficit falls short a certain threshold. In our notification system, we want to give each user a choice to determine this threshold, which gives the user a control knob to decide when he wants to get notified. We allow this by a control factor, $\alpha < 1$, by which the overall expectation of the calorie expenditure for that user is lowered. That is, the amount of expenditure barely enough to avoid notification is to remain slight above ($\alpha \times \rho \times t$). In this, we recalculate the modified deficit (denoted as Δ) as follows:

$$\Delta = (\alpha \times \rho \times t - C(t))$$

.

If $\Delta > 0$, then the system will provide activity suggestion for the user for the next epoch $t + 1$. The value of α depends on individual user's choice. And, ρ is dependent on user's age, weights, BMI (Body Mass Index), health condition and other physical features. This value should be fixed after consulting with domain experts or physicians.

The problem of activity suggestions now becomes the following. Given Δ for a given epoch, the user needs to receive as suggestions consisting of physical activities along with their possible duration of engagement that can lead him to fulfill the deficit. Let t_i be the suggested duration of activity a_i for a given epoch. Our notification system tries to determine these t_i's.

In this regard, we assume that a user would like to have more time to relax than performing activities continually within a time frame. So we propose a model that ensures longest relax time within the time frame by minimizing total engagement time in performing suggested activities. While suggesting activities, we also want to ensure that the user does not engage more than a certain maximum rate. In order to avail highest relax time, a user can trivially be suggested with the most expensive physical activities that burns large amount of calorie within a short time. But that high workout may be detrimental to health of that person and may harm the user instead of benefiting. Arguably, there is no point that a user is performing expensive physical activities when he can achieve the same by doing another activity with lower physical engagement albeit a bit longer time. For this, we introduce a maximum allowable rate at which a user can burn calorie during the period of his engagement. This rate is denoted as γ. Figure 4.3 depicts the parameters we have encountered and provides graphical interpretation of them.

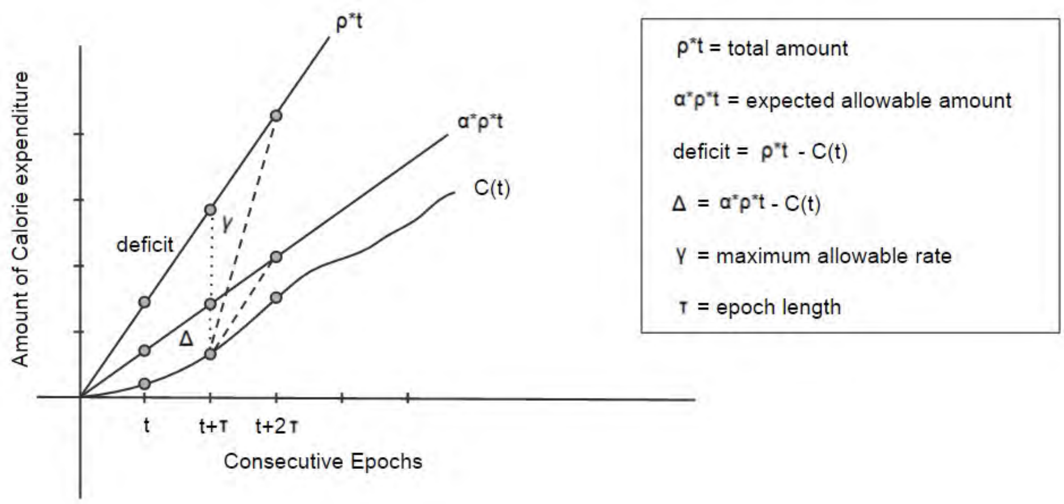


Figure 4.3: Parameters of Activity Suggestion System

Therefore, the problem of activity suggestion can be formulated with the following optimization problem, which asks to find duration t_i's for activity a_i so as to—

$$\begin{aligned}
\text{minimize} \sum_{i=1}^{k} t_i \\
\text{subject to} \sum_{i=1}^{k} t_i \times w_i \geq \Delta + \alpha \times \rho \\
\sum_{i=1}^{k} t_i \leq \tau \\
\sum_{i=1}^{k} t_i \times w_i \leq \gamma \times \sum_{i=1}^{k} t_i \\
t_i \geq 0, \text{real}
\end{aligned} \tag{4.1}$$

The objective function finds the total time of physical engagement among suggested activities. The first constraint ensures that the total calorie expenditure by the suggested activities exceeds deficit Δ, whereas the second on limits total time to be within the epoch duration (τ). The third one limits the rate of calorie expenditure within the limit the set by γ. Note that γ can vary from epoch to epoch depending in the amount of deficit the user actually has.

In our work, we tried to determine γ in two ways.

- It can be extracted from activity label data of previous epoch by considering the most likely activity an user might want to undergo.
- It can be calculated from mean amount of the activities an user has to undergo to fulfill the prescribed amount of expenditure (ρ).

The formulation we just mentioned does not, however limit the number of activities a user needs to perform during an epoch. If the suggestion eventually goes for a long set of activities, it will be cumbersome for the user to perform (even may hurt him); instead we can limit how many number of activities the system can suggests. Let us introduce a binary variable x_i to denote whether activity a_i is chosen or not. Considering in each epoch, the system should suggest at most two activities, we can have the following formulation:

$$\begin{aligned}
\text{minimize} & \sum_{i=1}^{k} x_i \times t_i \\
\text{subject to} & \sum_{i=1}^{k} x_i \times t_i \times w_i \geq \Delta + \alpha \times \rho \\
& \sum_{i=1}^{k} x_i \times t_i \leq \tau \\
& \sum_{i=1}^{k} x_i \leq 2 \\
& \sum_{i=1}^{k} x_i \times t_i \times w_i \leq \gamma \times \sum_{i=1}^{k} t_i \\
& t_i \geq 0, x_i \in \{0,1\}
\end{aligned} \tag{4.2}$$

This problem is an instance of a non-linear IP (Integer Programming) problem, which is hard. Due to hardness, we do not proceed with this formulation rather stick to our first one.

4.1.4 Algorithm for Activity Suggestion

The problem posed in Equation 4.1 is quite similar to a variant of Coin Change Problem [36]. This variant of coin change problem asks a certain amount of money to be constructed from a given set of coin denominations using the minimum number of coins. Here, in our context, the target money is the amount of the deficit (Δ) in the current epoch plus the expected amount of expenditure in the next epoch, coins are our performing time for activities and the denominations correspond to the weights (calorie ratings) of activities. But there is an important difference. This is the boundary value condition (the third constraint), which is dependent on the value of max rate (γ).

The algorithm *activitySuggestion* takes care of the boundary value condition. This algorithm selects a subset from the available weight-set (corresponding to activity set) on the basis of the maximum allowable rate(γ). Then it feeds the subset to the *activitySelection* algorithm to generate suggestion. The algorithm tries to suggest the activities with rate smaller than γ. But if we calculate γ by considering the most likely activity then, in order to cope up with the need of physical activities to be performed, sometimes γ is adjusted.

We devise a dynamic algorithm named *activitySelection*, inspired from [36]. The technique is very close to way we construct a certain amount of money in practice (we choose minimum number of denominations needed to construct the amount). The algorithm outputs a set of recommended activities along with a set of time of doing each of the suggested activities. Let $S[amnt]$ be the minimum period of performing activities having respective weights from weight-set (calorie expenditure rating). Here, $amnt = \alpha \times \rho + \Delta$. In the optimal solution to suggest activities for burning $amnt$ amount of calories, there exist some activity a_i with associated weight w_i, where $w_i \leq amnt$. Furthermore, the remaining amount of calorie expenditure from $amnt$ in the optimal solution must themselves be the optimal solution to make suggestion for $amnt - w_i$. Thus, if w_i, the corresponding calorie burning weight of activity a_i is the first weight in the optimal solution to make suggestion for $amnt$ amount, then $S[amnt] = 1 + S[amnt - w_i]$; i.e., one period of performing w_i activity plus $S[amnt - w_i]$ periods of performing activities to optimally make suggestion for $amnt - w_i$ amount of calorie expenditure. We dont know which activity a_i having weight w_i is the first weight used in the optimal solution to make suggestion for *amnt* amount; however, we may check all k such possibilities (subject to the constraint that $w_i \leq amnt$), and the value of the optimal solution must correspond to the minimum value of $1 + S[amnt - w_i]$, by definition. Furthermore, when making suggestion for amount $\Delta \leq 0$, the value of the optimal solution is clearly 0, as we are not providing any suggestion in such case. We thus have the following recurrence:

$$S[amnt] = \begin{cases} min_{i:w_i \leq amnt}\{1 + S[amnt - w_i]\}, & \text{if } \Delta > 0. \\ 0, & \text{otherwise.} \end{cases}$$

Both the algorithms are listed below.

Algorithm 1 Suggest Activities on the basis of γ

procedure ACTIVITYSUGGETION($weightSet$, γ, $amountToBePerformed$, τ)
    sort $weightSet$ in non-decreasing order
    $lower \leftarrow$ immediate lower value than γ from $weightSet$
    $higher \leftarrow$ immediate higher value than γ from $weightSet$
    $maximumTime \leftarrow$ time to complete the $amountToBePerformed$ using γ
    $maxAmount \leftarrow$ amount of activity achieved by using τ and max weight from $weightSet$
    if $maximumTime$ exceeds τ **then**
        if $amountToBePerformed$ exceeds $maxAmount$ **then**
            $suggestion \leftarrow$ maximum activity
            calculate $amountToBeCarried$ for next period
        else
            choose activities from the $weightSet$ that is greater than $higher$
            $suggestion \leftarrow$ suggested list of activities using $activitySelection$ algorithm
        end if
    else
        if time to finish $amountToBePerformed$ using $lower$ is below τ **then**
            choose activities from the $weightSet$ that is smaller than $lower$
        else
            choose activities from the $weightSet$ that is smaller than $higher$
        end if
        $suggestion \leftarrow$ suggested list of activities using $activitySelection$ algorithm
    end if
    return $suggestion$ and $amountToBeCarried$
end procedure

Algorithm 2 Selection of Activities

```
procedure ACTIVITYSELECTION(weightSet, amnt)
    S[0] ← 0
    for i ← 1 to amnt do
        minimum ← ∞
        for j ← 1 to size of weightSet do
            if weightSet[j] ≤ i then
                if 1 + S[i − weightSet[j]] < minimum then
                    minimum ← 1 + S[i − weightSet[j]]
                    index ← j
                end if
            end if
        end for
        S[i] ← minimum
        Index[i] ← index
    end for
    generate path using the Index
    generate activityList using the path, weightSet and S
    return activityList
end procedure
```

4.2 Adaptive Notification Generator

The suggestion system provides a set of suggested activities along their performing time for each epoch. Adaptive notification generator propagates these results to the user on the basis of certain rules. As building adaptive notification generator for mobile devices is itself a huge work and required extensive amount of working to set the rules that guides the system, we hereby opted for a very simple type of notification system. Our predefined rules for propagating suggestions through mobile notification is listed below:

- If the calorie expenditure for t epoch meets the expected calorie expenditure ($\alpha \times \rho \times t$), means when $\Delta \leq 0$ then no notification is sent to the user for epoch $t+1$.
- If $\Delta > 0$, then a list of suggestion is provided to the user that has to be performed in the $t+1$ epoch.
- If the user does not perform the activities and Δ rises further greater than 0, then a reminder notification will be sent for the up coming epoch $t+2$ along with the suggestion

notification.

- If this continues then the rate of sending reminder message will be increased after a certain number of epoch elapsed.

- If Δ rises higher than a preset threshold value then the rate of sending notification will be increased too.

By using these several assumptions we build our Adaptive notification generator.

4.2.1 Notification Generation Paradigm

We have decided the maximum number of notifications per epoch is 3. Among this notifications, the first one contains the suggestion scheme. The later ones include motivational reminders. The suggestion scheme is comprised of activity names along with the time of performing that activity. At the beginning of an epoch the notification with the suggestion scheme is fired. The remaining notifications are sent in preferable time within the epoch length (τ).

4.3 Possible System Design

There exist two alternatives to design the whole system. These are described in the following sections.

4.3.1 Cloud Assisted Suggestion System

In this system, a mobile client is used to connect with a cloud server. The client initially sends accelerometer data which is treated as training data. The cloud server processed the data and use classifiers to learn the model. Previously collected data set can also be used for the learning purpose as we did using some benchmark data [4]. After learning the model retains in the server. After a particular epoch length (τ) the client sends data chunk to the server and the server classifies these data using the classifier model. It predicts the activities and also prepare suggestion for performing certain amount of activities to maintain the expected rate ($\rho \times \alpha$). Then it propagates the suggestion to the mobile client along with the number of notification required to push the user to perform suggested activities. The mobile client generates notifications based on these information and notifies the user about his lacking of performing physical activities.

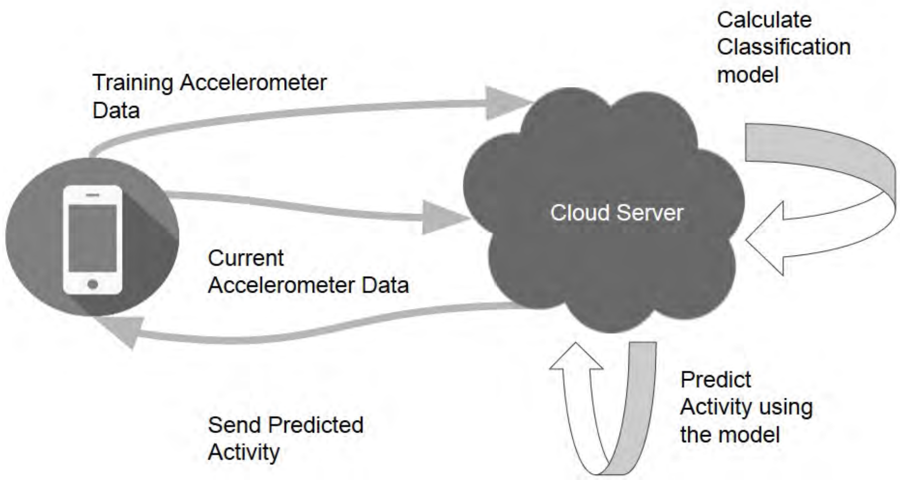


Figure 4.4: Cloud assisted suggestion System

4.3.2 In-device Suggestion System

The training part of this system is similar to the cloud assisted suggestion system. The main difference is that, here the learned model is sent back to the mobile client. So for the recognition and suggestion, the client do not have to send any data to the server. Besides using the model it can predict and suggest the required amount of activities and notifications by itself. Later the client generates suggestions according these information.

Among these two system the first one seems costly as a lot of data have to be sent to the server on a frequent basis. For this thesis, we did not design any of the mentioned system explicitly. We assume that we have the classification model and and stream of user's accelerometer data. We have simulated all the possibilities that can arise and mainly focused on

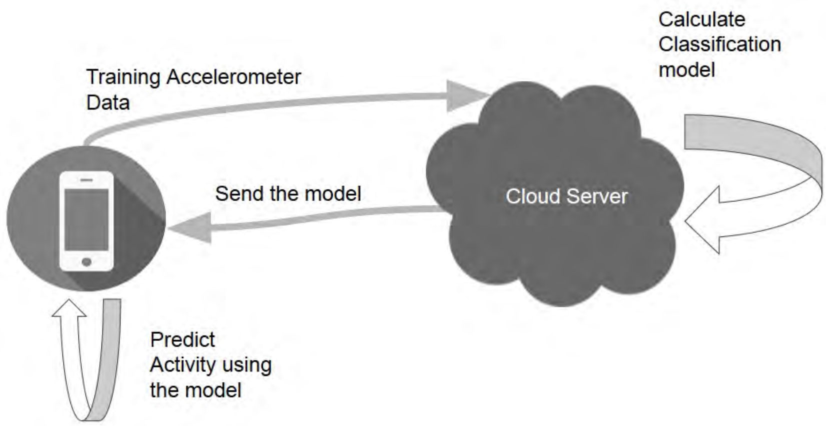


Figure 4.5: In-device suggestion System

determining the suggested amount of activities (with time preferences) along with the number of notifications.

Chapter 5

Experimental Results

This chapter includes some experiments that we performed to select a classifier for activity recognition from time series accelerometer data. Besides this chapter also includes several performance testing experiments on activity suggestion system along with notification handling. First of all we present information about our existing dataset and then move into activity recognition and activity suggestion part. We have used MATLAB for the simulation and classification purpose.

5.1 Training Data and Test Data

We build model from a dataset collected for previous works in heterogeneity in activity recognition [4]. This data set contains time series accelerometer data along with the label of associated activity names. The dataset contains a set of activities and they are {null activity, walking, standing, stairs up, stairs down, biking}. Here, null activity refers to do nothing at all. The dataset contains reading of 9 different users performing these mentioned activities for a significant amount of time using 8 different models of smartphones. It contains more than 15 millions of data. We build training and test dataset from this vast dataset for the experimentation purpose.

We prepared our training dataset in two ways:

1. We treated all data of an user consisting of all the activities that is collected using smartphones of similar sampling frequency as our training data.

2. We treated data of 3 users that is collected using smartphones of similar sampling frequency as our training data.

We carried out our experiment by using both the datasets and found almost similar results. Hence we present the results of our experiment on the basis of the first set of training data throughout the chapter. For the classification purpose, we assigned numeric label to each of the activities of the activity list. Here we selected the activity labels for the set of activities {null activity, walking, standing, stairs up, stairs down, biking} to be 0, 1, 2, 3, 4, 5. So 0 resembles null activity, 1 resembles walking and so on.

We select test data from the remaining dataset. We used different users data from various devices for testing purpose.

5.2 Feature Extraction

We have selected a total of 16 features to detect activities. These include both time and frequency domain features. Time domain features include mean vector, standard deviation vector, euclidean norm of mean vector, euclidean norm of the standard deviation, correlation values and sum of magnitudes up to 25^{th} and 75^{th} percentile values. On the other-hand, frequency domain features comprise of the specific frequency components, the amplitude of spectrum, the peak frequency in the spectrum and number of peak values under a certain value in the spectrum.

5.3 Performance of Classifiers for Activity Recognition

We selected several features for our recognition system (discussed in 3.3.3 section) to be relied on. We first applied low pass filtering to reduce the noise present in the training data. We used a sliding window of 10 seconds for our recognition purpose.

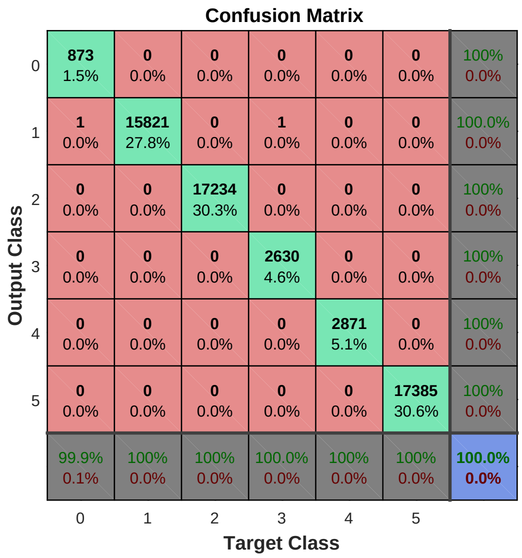
In order to predict the activities from user's accelerometer data, we have to build a model that would classify those data using our training dataset. To serve these purpose, we have to go through several classifiers and few readings regarding classification process. Finally we have opted for a few classifiers: k-NN, *Random Forest, Multi-class SVM* as our potential set of classifiers. After using each of the classifier with various parameter in MATLAB platform, we selected the best one from them.

We used k-NN algorithm by using built-in MATLAB function and specifying various related parameter associated with it. The neighbor size is changed thoroughly to get the proper result. We also opted for the *Random Forest ensemble* by using the *fitensemble* method of the MATLAB to get a proper model using the training data. We also varied the size of the tree that is need for the ensemble method.

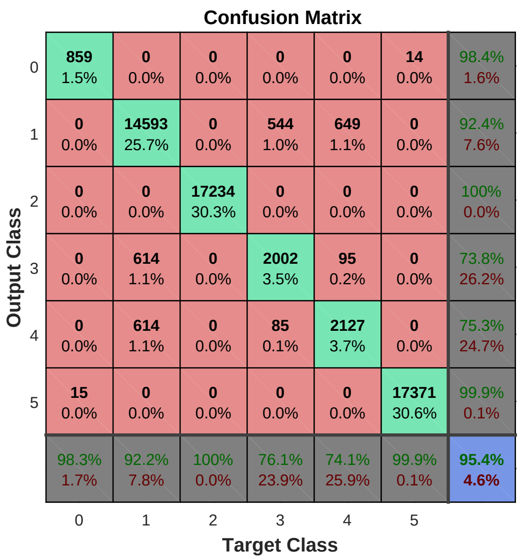
Finally we used MATLAB's multi-class model which is guaranteed by the use of *fitcecoc* method. We provided SVM learner as the learner of this mentioned method. As a result, this multi-class model classifies the data using the SVM classifier and served the need of using multi-class SVM classifier.

5.3.1 Cross Validation Results

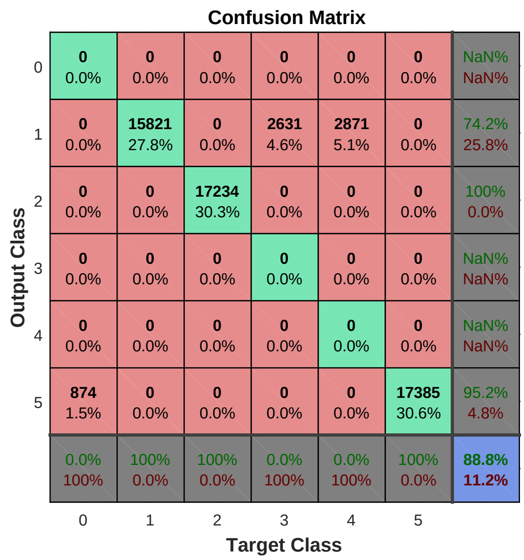
Here we provide the confusion matrix of 10 fold cross validation test using each of the classifiers on our training data in Figure 5.1. The labels : 0 ,1 ,2 ,3, 4, 5 represents null activity, walking, standing, stairs up, stairs down and biking respectively. We used various size of trees for random forest such as 50, 100, 200 and almost got the similar kind of confusion matrix. We also used various neighbor size for k-NN and almost got similar results. For Multi-class SVM, we used "one vs one" and "one vs all" coding for SVM learners. The accuracy, precision and other performance matrices for the above mentioned classifiers are listed below in Table 5.4.



(a) Multi-class SVM (one vs one)



(b) k-NN with neighbor size = 50



(c) Random Forest Ensemble with tree size = 500

Figure 5.1: Outcome Cross-validation of Various Classifiers Confusion Matrix

classifiers	accuracy	precision	recall	F -score
k-NN (neighbor size = 20)	0.9534	0.91144	0.9224	0.9169
k-NN (neighbor size = 50)	0.95327	0.91015	0.9233	0.9167
Random Forest (tree size = 100)	0.88777	0.71256	1	0.83221
Random Forest (tree size = 500)	0.88778	0.71275	1	0.83228
Multi-class SVM (one vs one)	0.99995	0.9998	1	0.9998
Multi-class SVM (one vs all)	0.99993	0.99974	1	0.99987

Table 5.1: Comparison of Performance metrics after Cross-validation

5.3.2 Performance Comparison Using Test Data

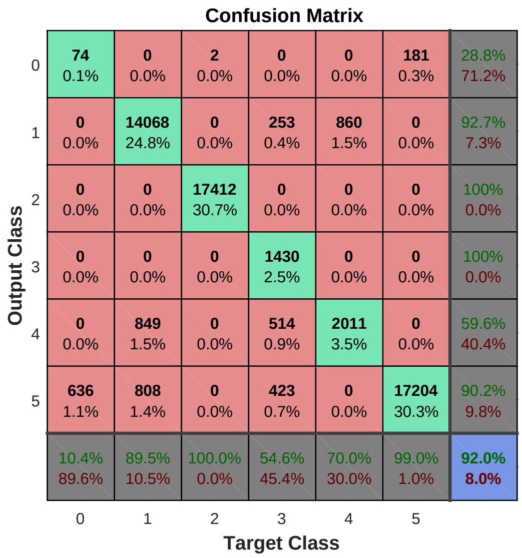
As we have a huge dataset, we made two scenarios to measure the performance of the classifiers. They are:

- Train and test a model using a single user. Here we make the training dataset using several smartphone model's accelerometer data while keeping a few models data for the extraction of test data for that user.

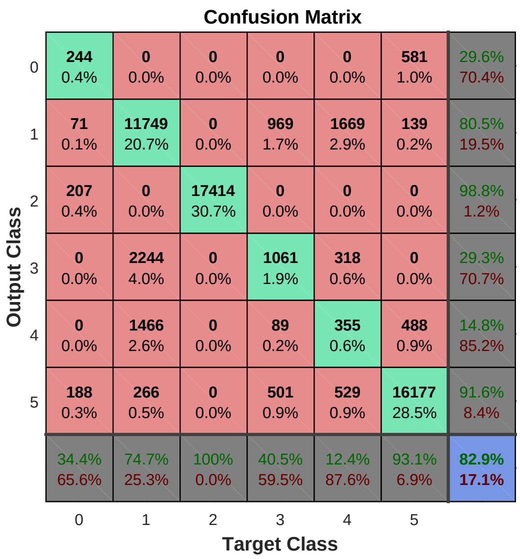
- Train using one and test using another user. In this scenario we train the model using all available accelerometer data of an user and then test another users data with respect to the previous user.

Here the confusion matrix and table of performance matrices are provided for each of the case stated above.

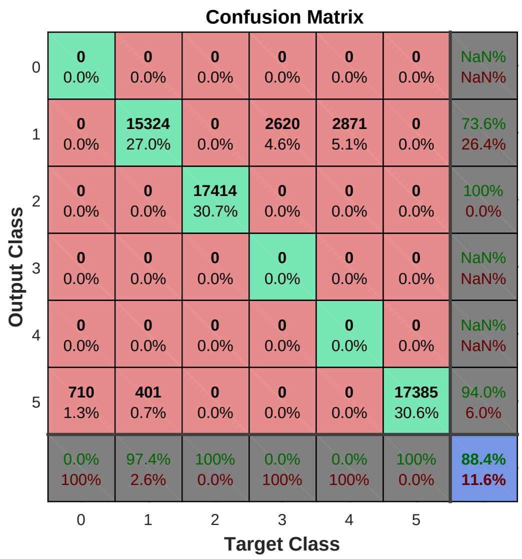
5.3.2.1 Train and Test Using Same User



(a) Multi-class SVM



(b) k-NN with neighbor size = 50



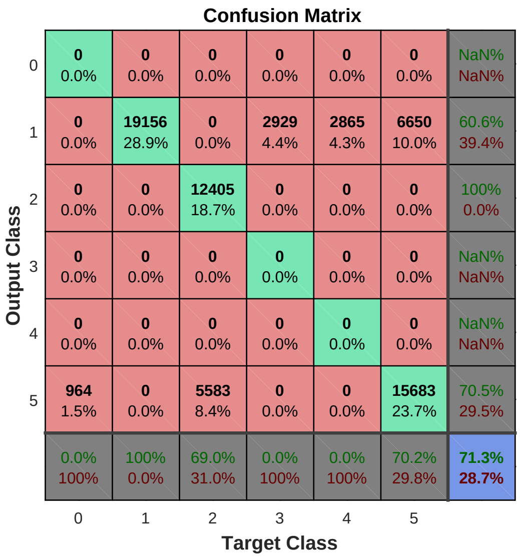
(c) Random Forest Ensemble with tree size = 500

Figure 5.2: Confusion Matrix for different classifiers using a single user data

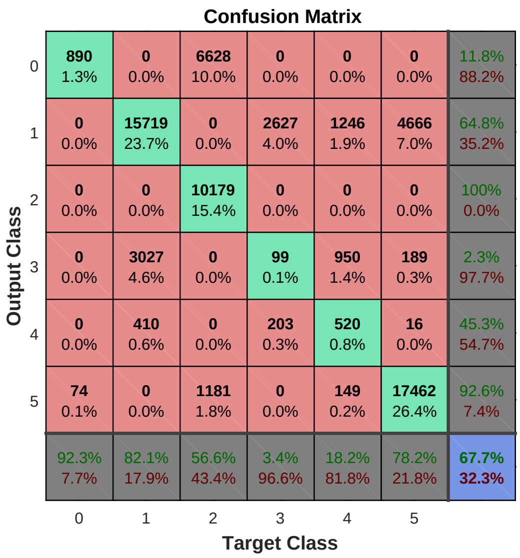
classifiers	accuracy	precision	recall	F -score
Multi-class SVM (one vs one)	0.9202	0.8306	0.8946	0.8614
k-NN (neighbor size = 50)	0.8286	0.6714	0.7472	0.7073
Random Forest (tree size = 500)	0.8836	0.7119	0.9745	0.8228

Table 5.2: Comparison of Performance metrics using a single user data

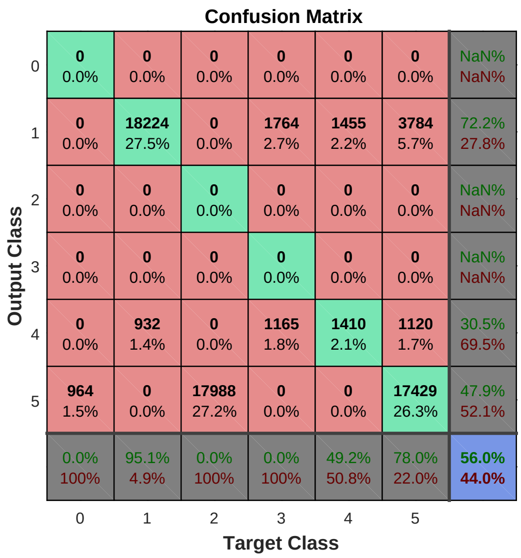
5.3.2.2 Train and Test Using Different User



(a) Multi-class SVM



(b) k-NN with neighbor size = 50



(c) Random Forest Ensemble with tree size = 500

Figure 5.3: Confusion Matrix for different classifiers using multiple users

classifiers	accuracy	precision	recall	F -score
Multi-class SVM (one vs one)	0.7132	0.5021	1	0.6685
k-NN (neighbor size = 50)	0.6774	0.4672	0.5954	0.7128
Random Forest (tree size = 500)	0.5596	0.3922	0.9513	0.5554

Table 5.3: Comparison of Performance metrics using multiple users

We used the variation of these classifiers by:

- Changing the neighbor size of k-NN.
- Changing the tree size of *Random forest*.
- Changing the coding of *SVM*.

All the variations produced somewhat similar results. As it is seen from Figure 5.2 and 5.3 Multi-class SVM classifiers produce the best output among these three classifiers, so we selected this classifier to build a classifier-model that will be used to predict user's accelerometer data in future. Besides it is also clear from the matrix and performance matrices that when an user's data is predicted using his own training data then the classification becomes more accurate. So it will be better to use the same user's data as training data to have more sounding prediction about the outcome (i.e activities).

5.4 Experimental Evaluation of Activity Suggestion System

In this section, we analyze extensive experiments to evaluate our proposed activity suggestion system. We experimented about the performance of the system discussed in section 4.1.3 in MATLAB by varying different parameters. We have assigned weights to each of the activity using [35] and online calorie expenditure calculator. Rather we normalized the values of the activities to a smaller one. These weights are dependent on user's age and weights, BMI and some other factors. For our experimental case, we set the values 0, 5, 3, 8, 4, 7 (normalized) for null activity, walking, standing, stairs up, stairs down and biking respectively (Table 5.4). It interprets as 1 minute of walking burns 5 calories and so on. The suggestion system can suggest any of the activities from the set except null activities as its weight equals to 0. After setting these weights, we varied some of our tunable parameters: prescribed calorie expenditure (ρ), user preference (α) and epoch length (τ) to watch the behavior of our system.

Activity name	Calorie expenditure	Assigned weight
Walking	298 cal/hr	5 cal/min
Standing	172 cal/hr	3 cal/min
Stairsup	470 cal/hr	8 cal/min
Stairsdown	234 cal/hr	4 cal/min
Biking	425 cal/hr	7 cal/min

Table 5.4: Calorie expenditure chart for 165 lbs, 170 cm male user [35].

5.4.1 Performance Metrics

In our experiment, we focus on three different performance metrics. These metrics are as follows:

- The number of notifications sent over a period.
- The amount of suggested activities to overcome calorie deficit per epoch.
- Total relaxation time offered per epoch.

5.4.2 Impact of Variation of Epoch Length (τ)

When we varied the epoch length, the number of notifications sent decreases with the increase of the epoch length. Again the relaxation time increases for the larger epoch length. Here we provided the various outcomes as a measure of graph when the whole experiment was carried out using the prescribed expenditure rate of 5 cal/min. So, $\rho = 5 \times \tau$ and $\alpha = 0.85$. When $\tau = 12$ minutes, then we have following outcomes.

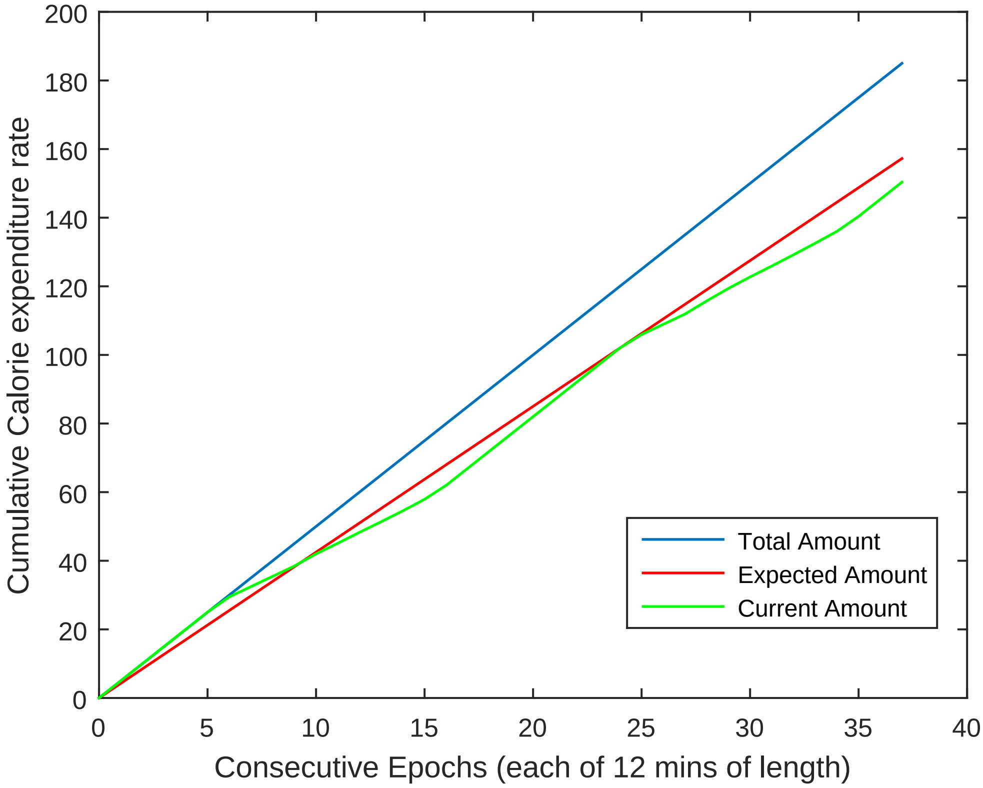
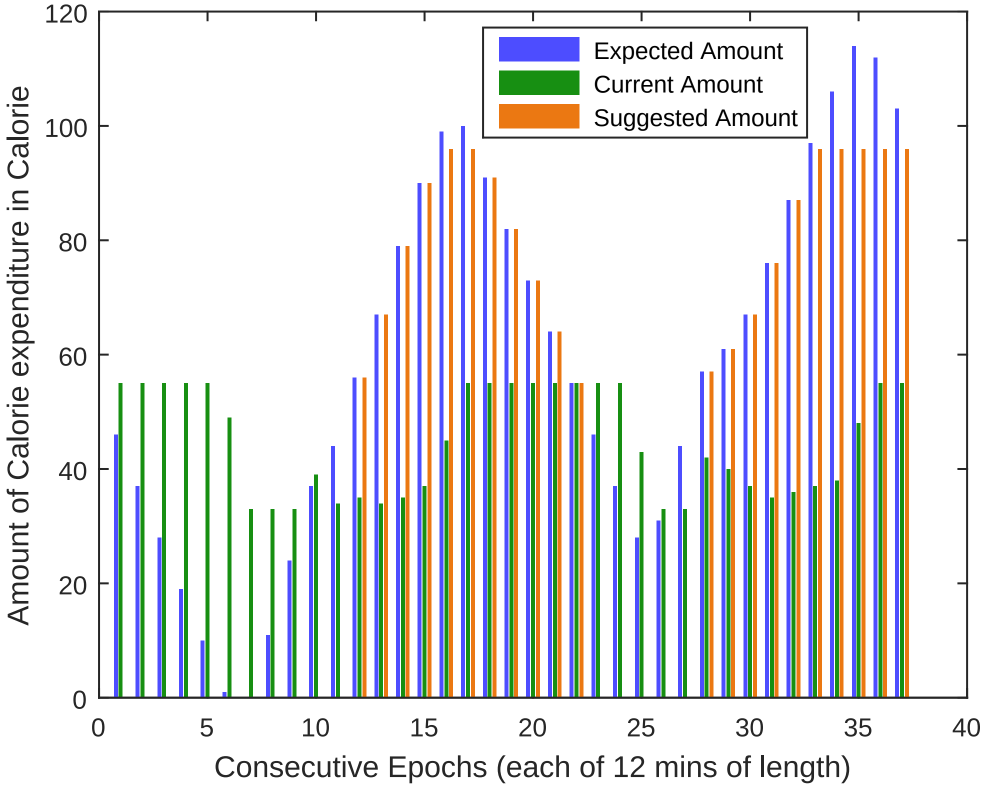
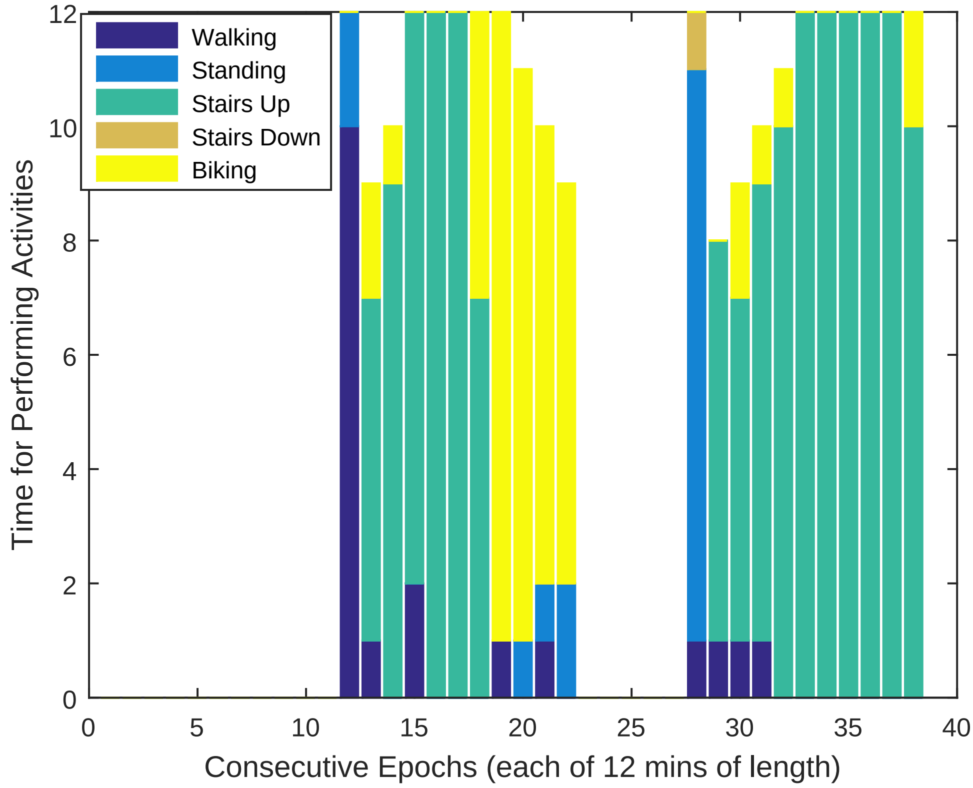


Figure 5.4: Comparison among various expenditure rate (τ = 12 minutes)

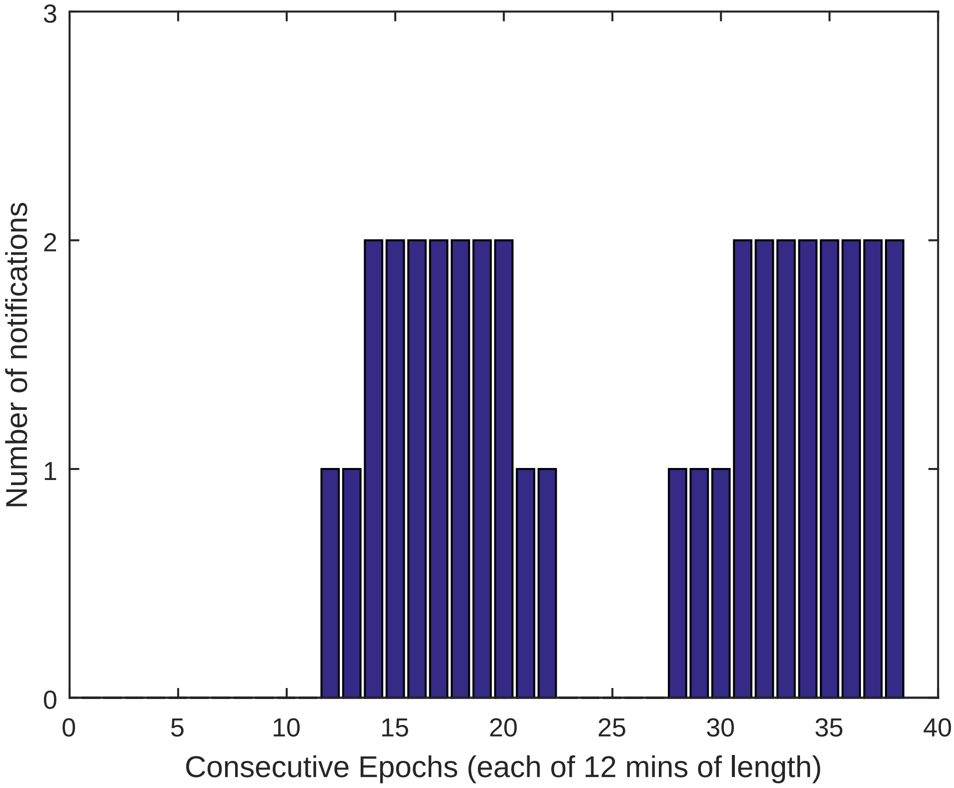


(a) Expected, Current and Suggested amount of calorie expenditure

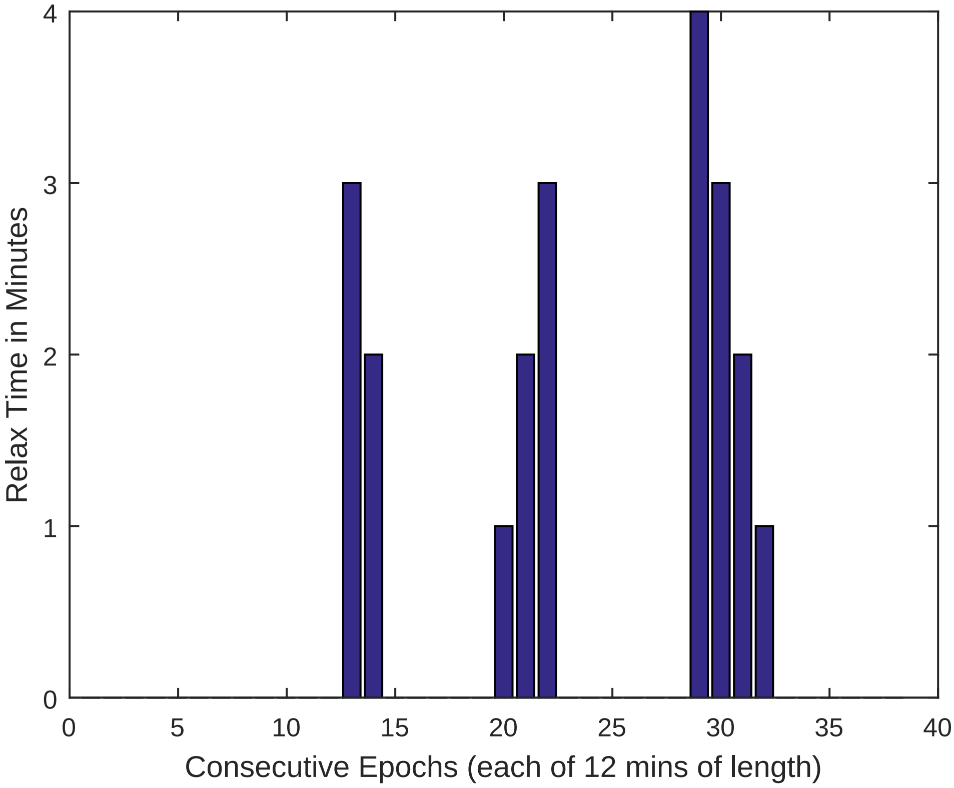


(b) Suggestion of Activities to maintain Calorie expenditure

Figure 5.5: Outcome of Various performance metrics (Suggestion) $\tau = 12$ minutes



(a) Number of Notifications sent



(b) Relaxation Time

Figure 5.6: Outcome of Various performance metrics (Notification count and relaxation time) $\tau = 12$ minutes

Here, Figure 5.4 illustrates the cumulative expenditure rate's comparison. The expenditure rate is calculated by dividing total amount of calorie expenditure of an epoch by the epoch length(τ). When the current cumulative expenditure rate falls between the actual and expected cumulative expenditure rate then no suggestion is provided. Suggestion is provided only when the current cumulative rate falls down the expected one. Figure5.5a illustrates the amount of expected, current and suggested calorie expenditure of each epoch of 12 minutes in length. After observing a particular epoch, suggestion is provided for the next epoch on the basis of deficit as discussed before. Figure 5.5b shows the amount of activities need to be performed to achieve the goal of calorie expenditure. Figure 5.6a and Figure 5.6b shows the number of notifications fired and the amount of relaxation time respectively. When we suggest an activity then relaxation time comes into play. Otherwise it is treated as NaN (which is interpreted as 0 in the graph).

Similarly when τ = 30 minutes, then we have following graphs (Figure 5.7, Figure 5.8, Figure 5.9). It is clear that in this case relaxation time increases but the number of notification decreases. Normally such trends occurs but it can not be guaranteed as user's preference of performing activities can not be predicted.

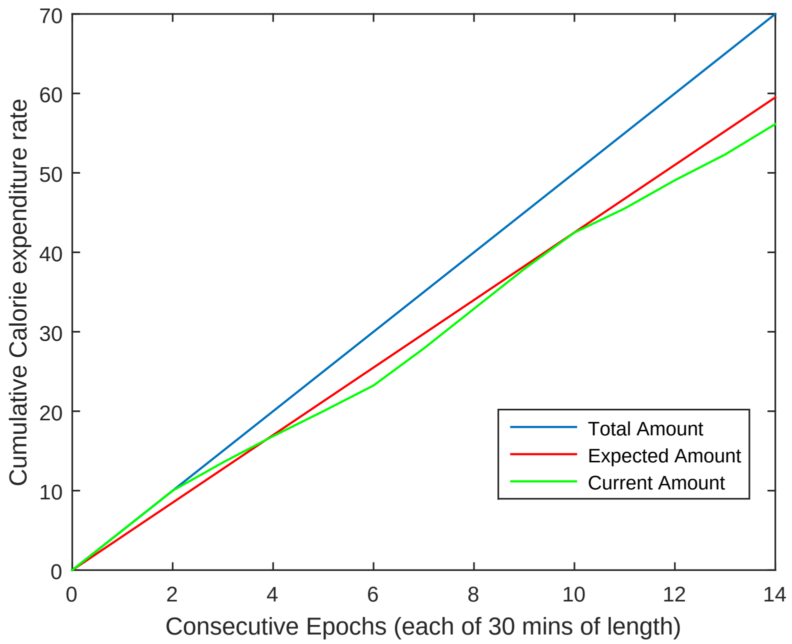
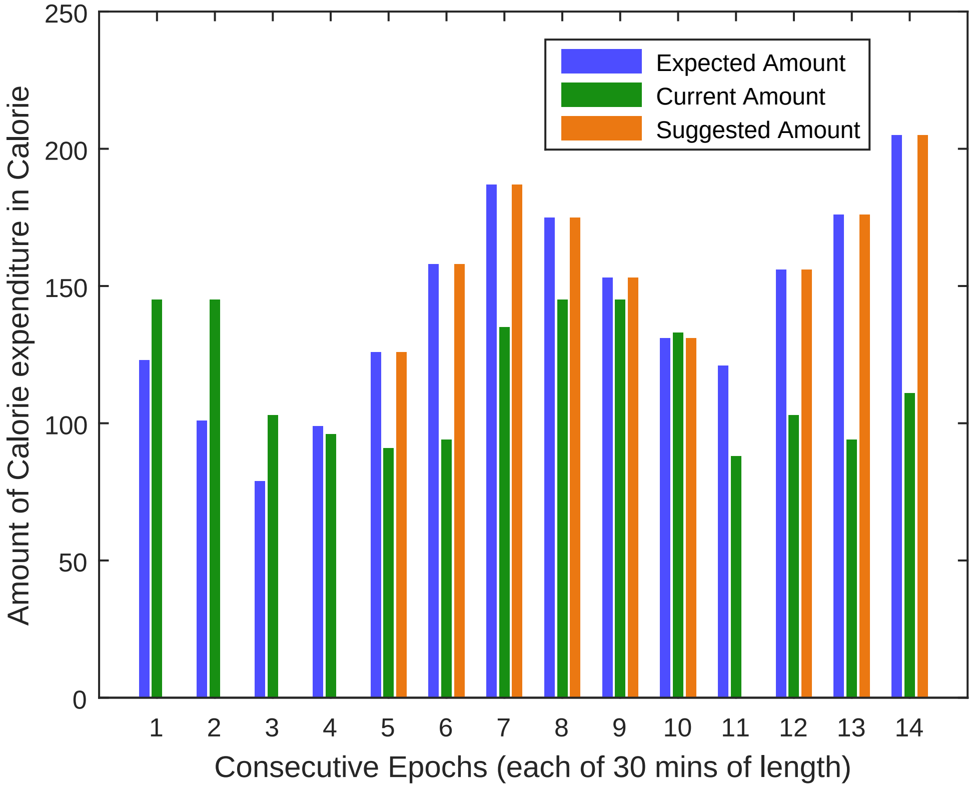
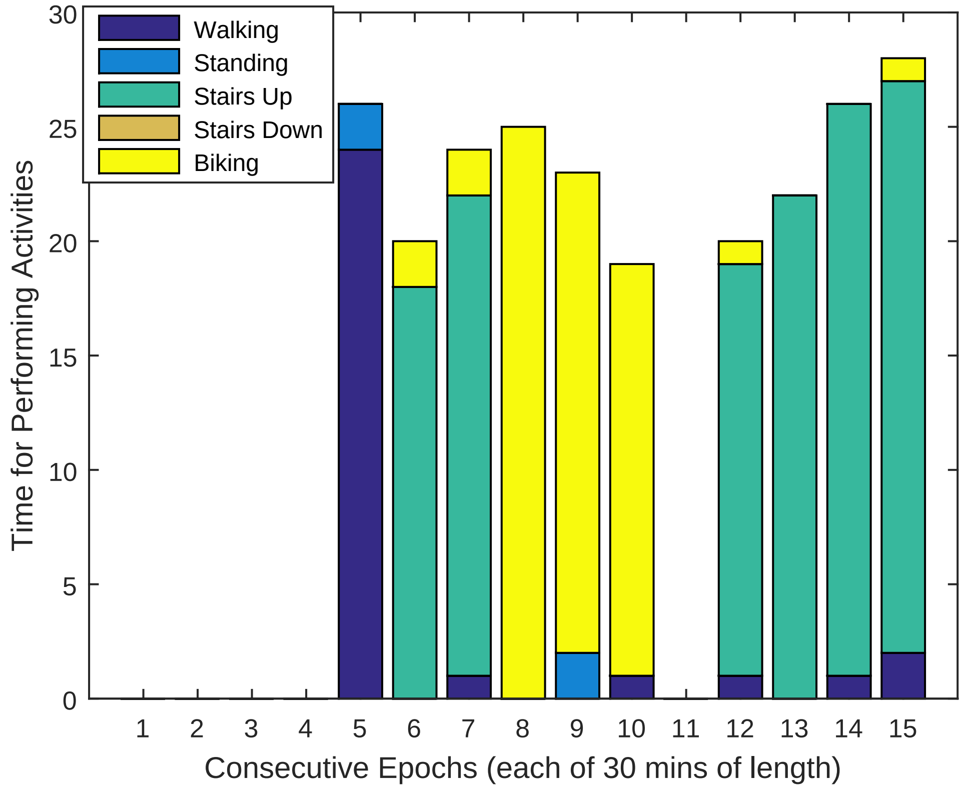


Figure 5.7: Comparison among various expenditure rate (τ = 30 minutes)

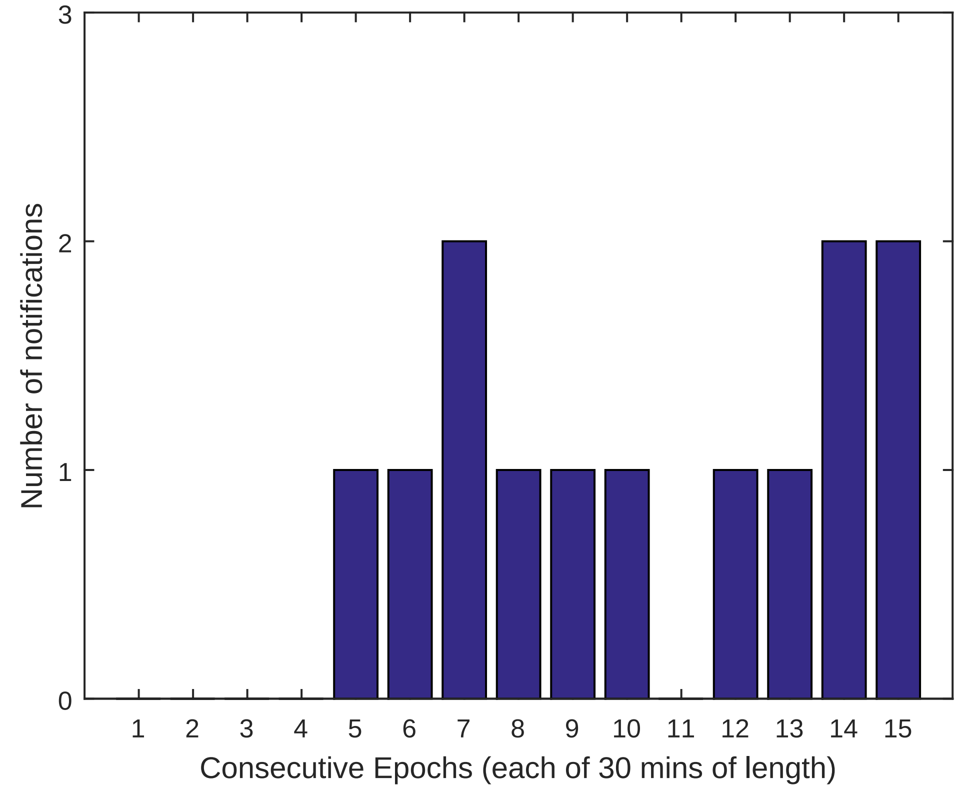


(a) Expected, Current and Suggested amount of calorie expenditure

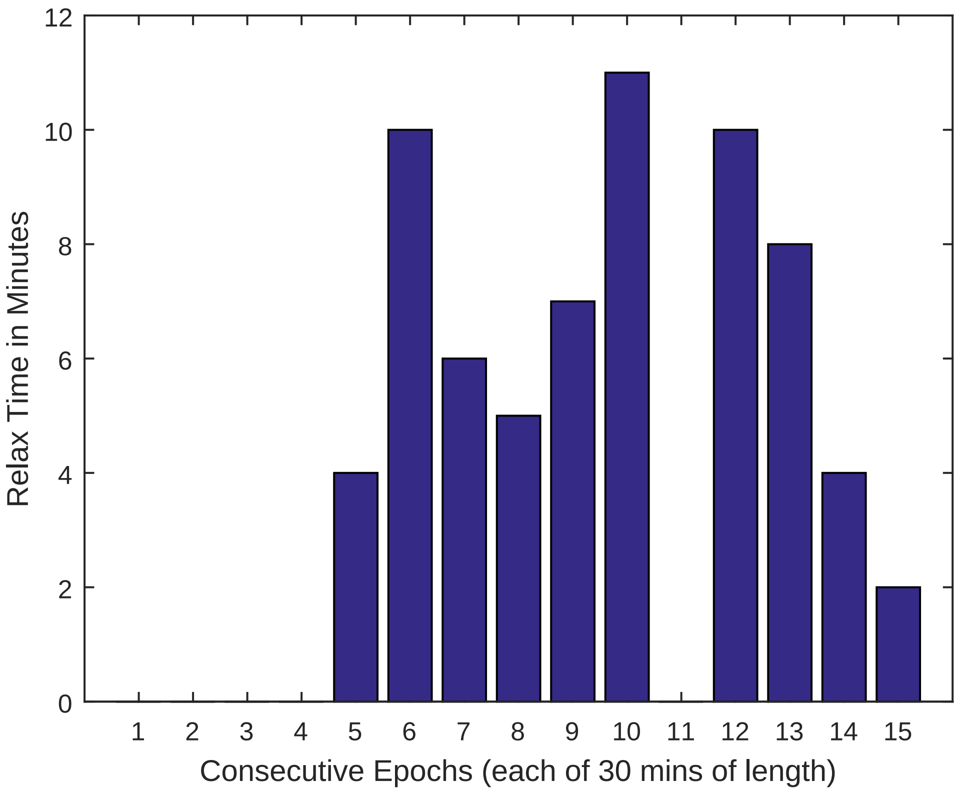


(b) Suggestion of Activities to maintain Calorie expenditure

Figure 5.8: Outcome of Various performance metrics (Suggestion) when $\tau = 30$ minutes



(a) Number of Notifications sent



(b) Relaxation Time

Figure 5.9: Outcome of Various performance metrics (Notification count and relaxation time) when $\tau = 30$ minutes

5.4.3 Comparison between High and Low Performing Users

From the user data set we have selected two users of whom one is performing activities in a high rate and the other one is performing in a quite low rate. We show their cumulative expenditure rate in Figure 5.10 and Figure 5.12 respectively. The number of notifications or the amount of suggestions for the user who is performing at a high rate is very low. Sometimes it almost reduces to zero. Almost negligible amount of suggestions are made for him/her. For the high performing user (whose current rate of performing activities is greater than the expected rate as seen in 5.10) no suggestion is made. As a result no notification is sent. Whereas for the low performing user, the number of notifications increases drastically (Figure 5.14); he/she ought to receive more amount of suggestions to improve his/her calorie expenditure level (Figure 5.13).

For this scenario we calculated all the necessities and the outcome is listed below:

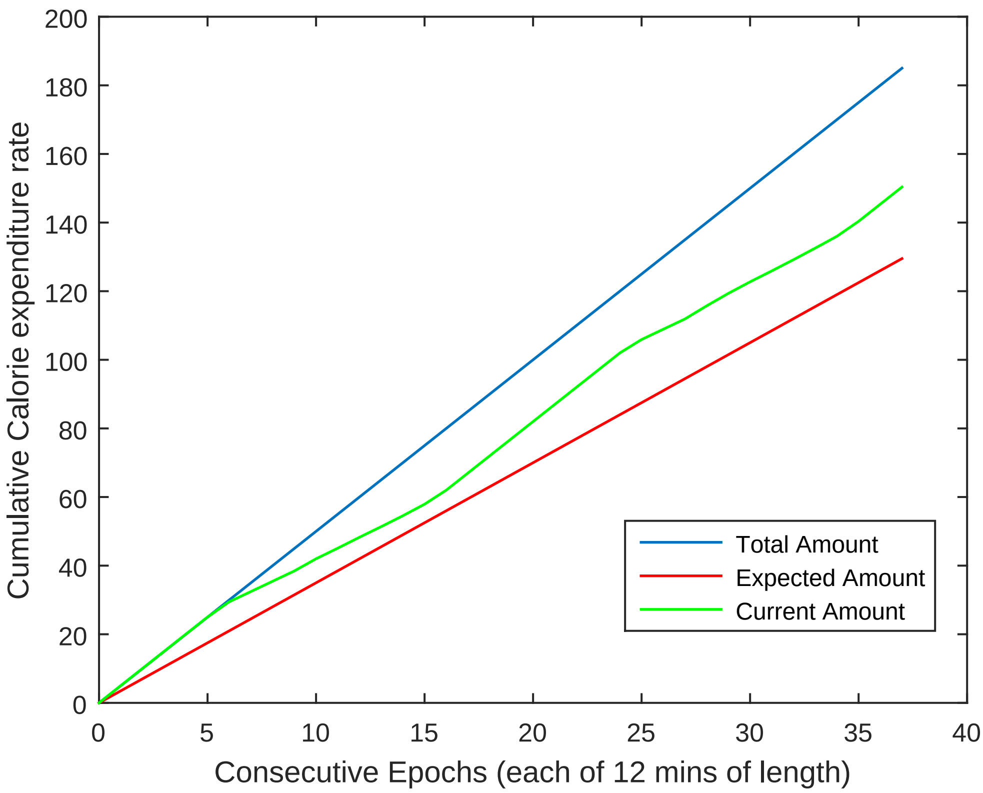


Figure 5.10: Comparison among various expenditure rate for a high performing user

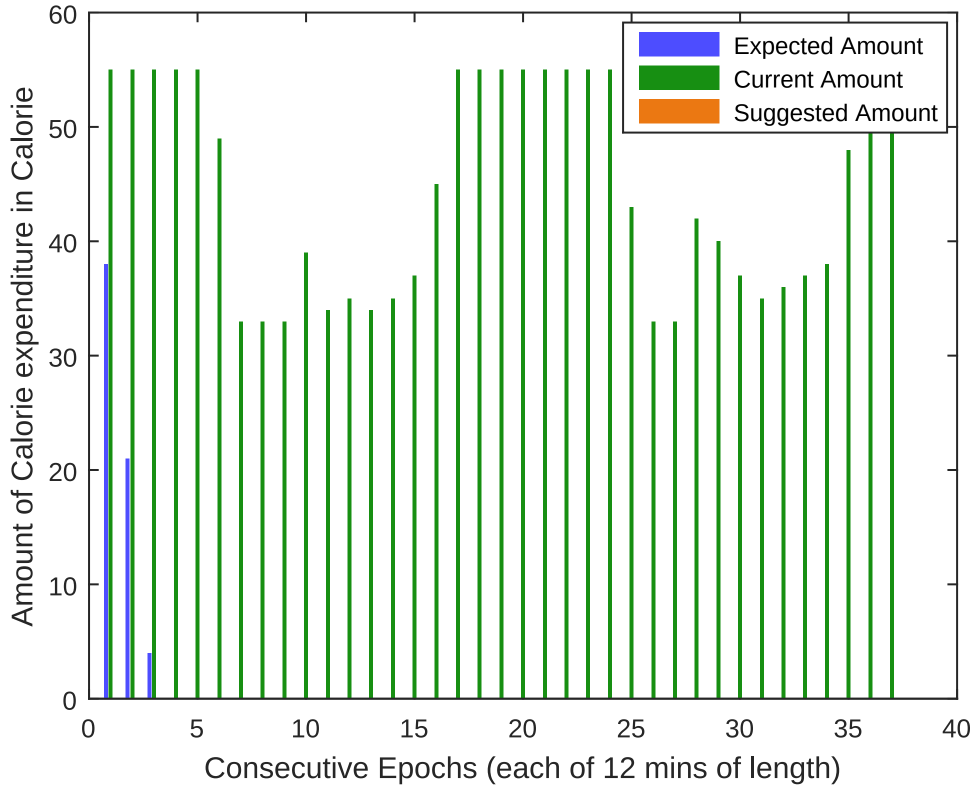


Figure 5.11: Expected, Current and Suggested amount of calorie expenditure for a high performing user

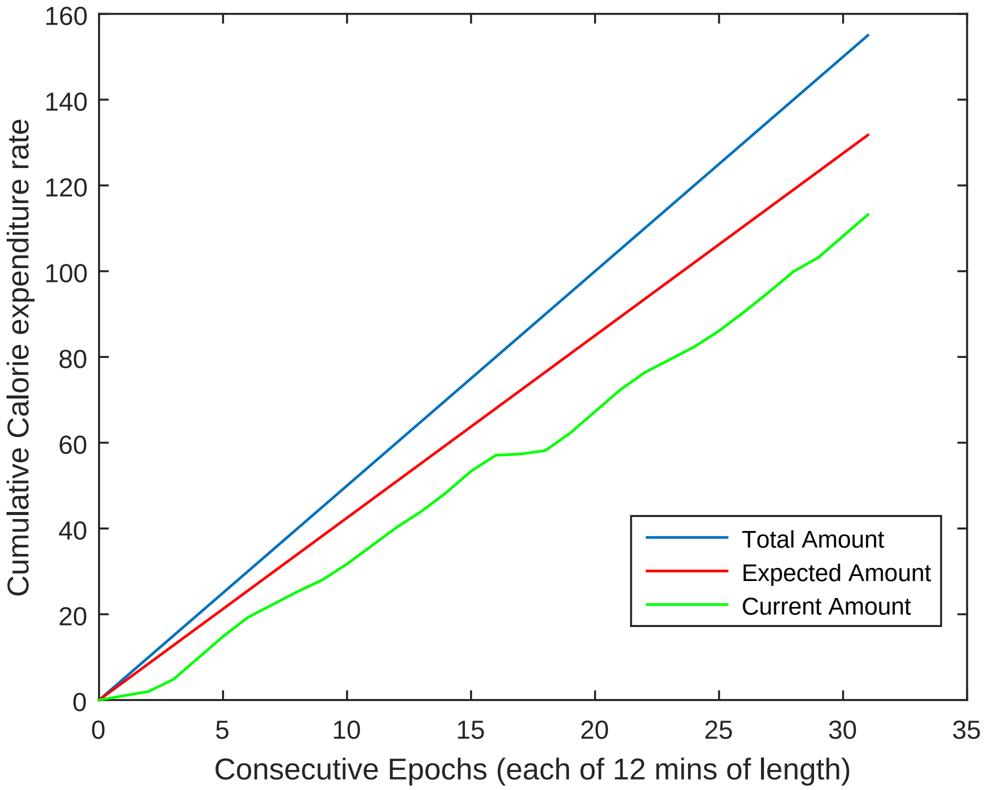
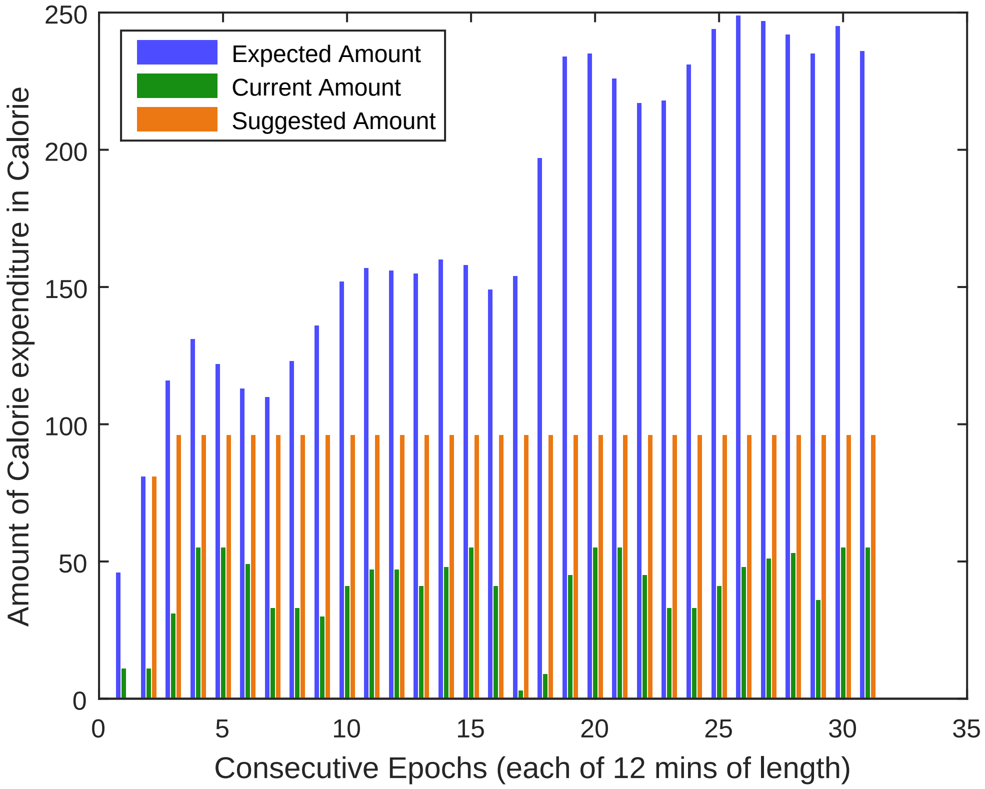
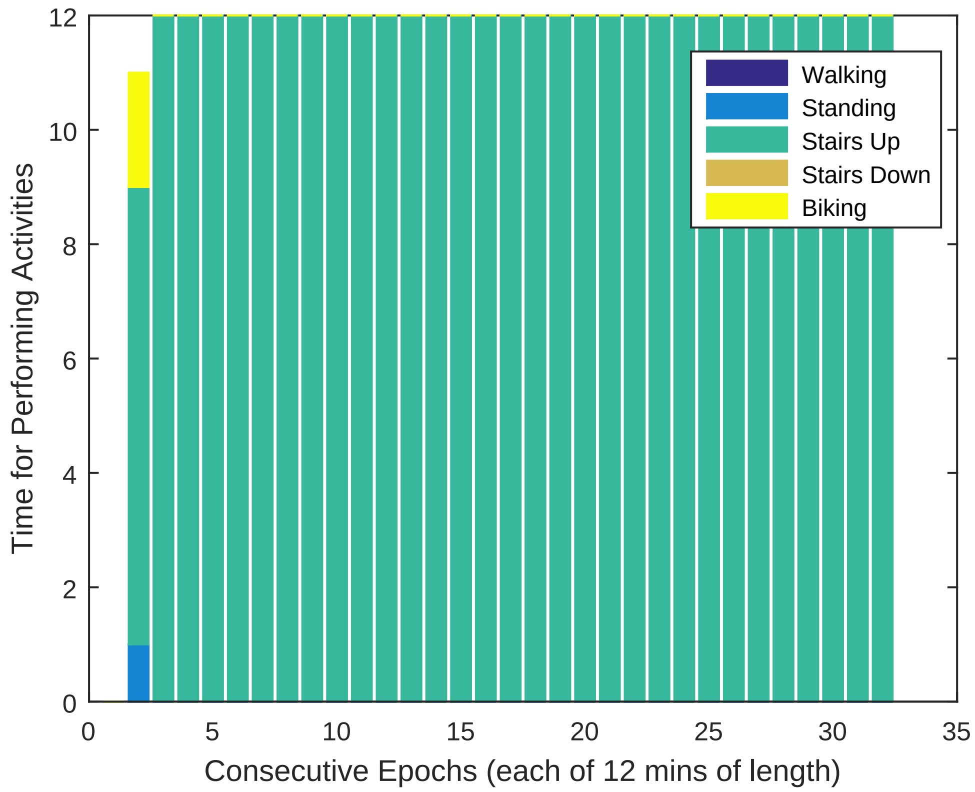


Figure 5.12: Comparison among various expenditure rate for a low performing user

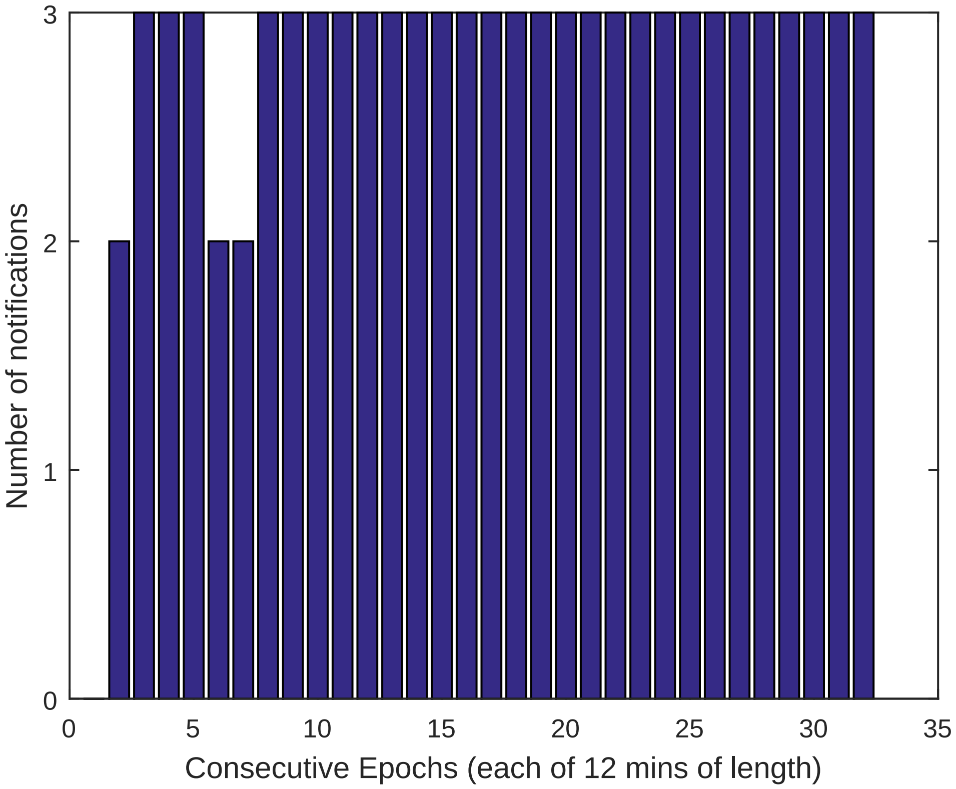


(a) Expected, Current and Suggested amount of calorie expenditure

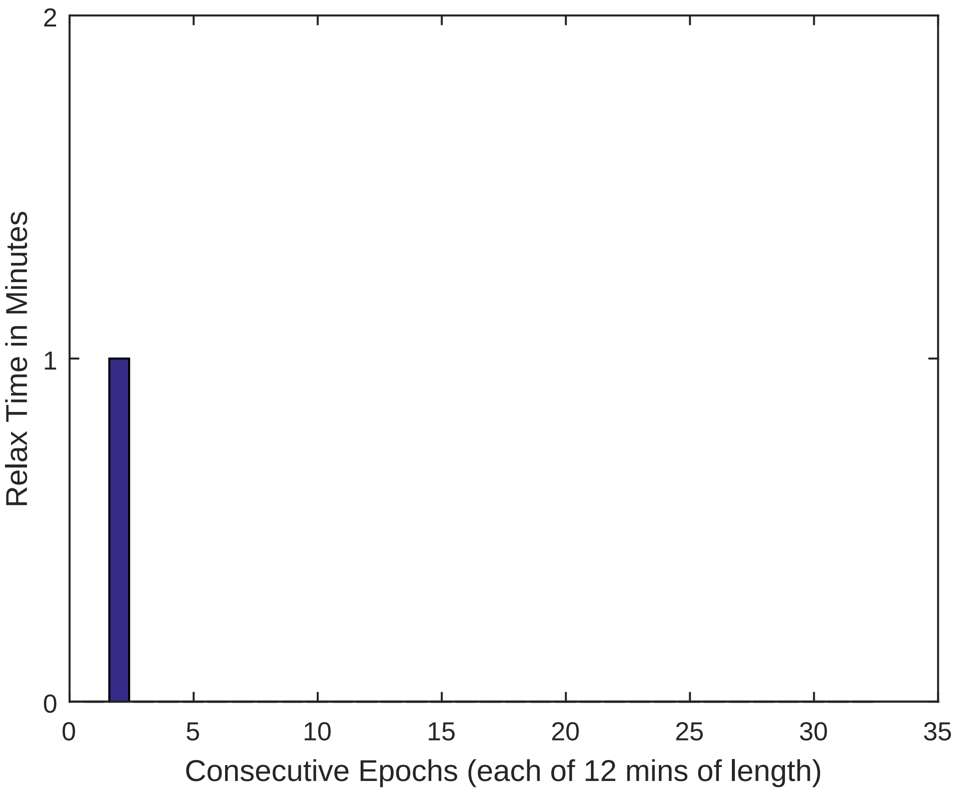


(b) Suggestion of Activities to maintain Calorie expenditure

Figure 5.13: Outcome of Various performance metrics (Suggestion) for a low performing user



(a) Number of Notifications sent

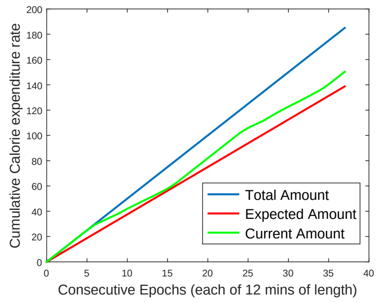


(b) Relaxation Time

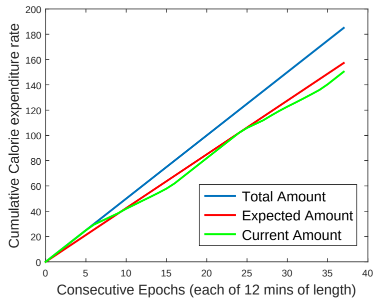
Figure 5.14: Outcome of Various performance metrics (Notification count and relaxation time) for a low performing user

5.4.4 Impact of Variation of Prescribed Calorie Expenditure (ρ) and User Preference (α)

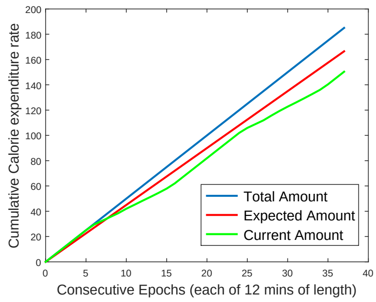
When ρ and α increases then the chance of getting notification increases and the time for relaxation decreases and vice versa. These values can effect the suggestion scheme greatly, as these creates the expected limit of calorie expenditure that an user would have to achieve.



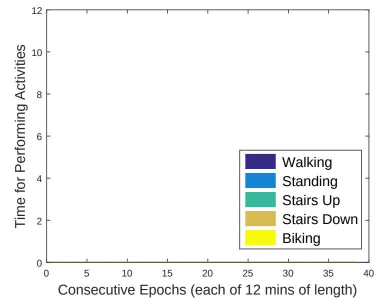
(a) Cumulative expenditure rate when $\alpha = .75$



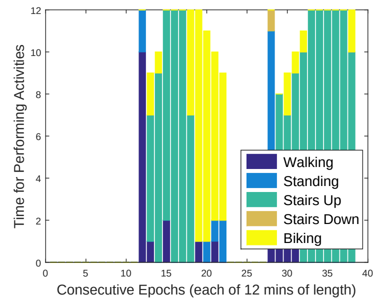
(b) Cumulative expenditure rate when $\alpha = .85$



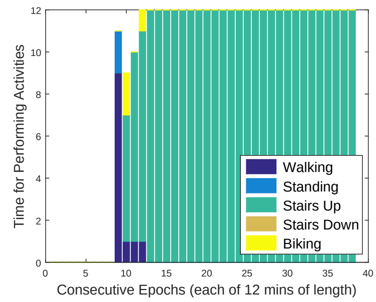
(c) Cumulative expenditure rate when $\alpha = .90$



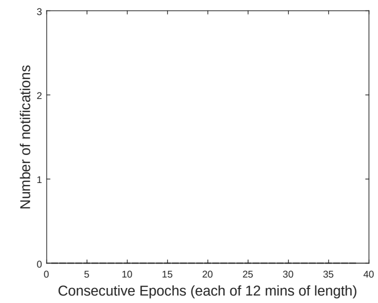
(d) Suggestion of Activities when $\alpha = .75$



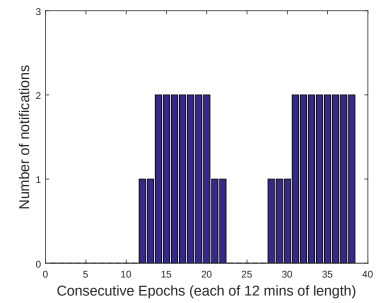
(e) Suggestion of Activities when $\alpha = .85$



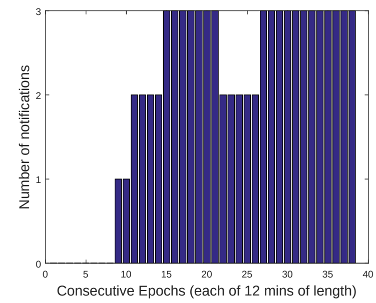
(f) Suggestion of Activities when $\alpha = .90$



(g) Number of Notifications sent when $\alpha = .75$



(h) Number of Notifications sent when $\alpha = .85$



(i) Number of Notifications sent when $\alpha = .90$

Figure 5.15: Effect of α

Figure 5.15 shows the effect of α on suggestion; from which we can see that the user get notifications with suggestions during when the current expenditure rate falls down the expected expenditure rate. When α is adjusted to 0.75, then the number of notification that has to be sent becomes zero as the current cumulative expenditure rate lies with the two rate (expected and actual). But when α gets higher like 0.9 (Figure 5.15c) then the current cumulative rate falls down a lot. In such cases a lot of suggestions are made and notifications are propagated to the user. Moreover when α is set to 0.85 then a moderate amount of notification is fired when the cumulative expenditure rate falls some what below than the expected one.

Chapter 6

Conclusion

In this thesis we derived an activity suggestion system that provides the user with suggestion of a set of activities that ensures the calorie expenditure of an user to stay under an expected level. It also guarantees the maximum relaxation time for an user. Throughout the thesis we dealt with a huge amount of accelerometer data obtained from [4]. We used various classifier to check which one provides the better classification and finally used SVM as our classifier as it produces more accurate result among all the classifiers we used. Then after training the classifier with the training accelerometer data with explicit activity labels we prepared a model that is used to predict activities using user's accelerometer data. This predicted activities are then used as the input of our proposed activity suggestion system.

As there is no much work on such type of activity suggestion system that provide adaptive suggestions/notifications for performing certain type of activity based on user need by evaluating accelerometer data, we have to build the system by using a lot of assumption and boundary conditions. There exist various scopes to work on to make this system more reliable. In future, this work can be extended in following areas:

- Currently we have treated the activities as they are independent of each other. But in reality it is not quite true. There exist some kind of relation between almost each and every activities. So if we can find this dependency factor among various activities, we can use them in limiting anomalous activities more accurately.

- In our work, we are not able to suggest activities in a practical way when the amount of performing activity needed to maintain expected calorie expenditure level rises much further. On that time it becomes quite impossible to suggest an activity that might reduce the deficit (Δ). In such case we suggest the activity that is high in calorie expenditure to be performed throughout the particular epoch. But still we can not bring down the high Δ. This case can be improved further in our future work.

- The Adaptive notification system that we applied in our thesis is a very simple one. More sophisticated techniques can be applied to schedule notification that will help the user to get more engaged to physical activities using the suggestion provided by the suggestion system rather than being irritated.

- In future, we will focus on improving the suggestion system by adding a few more of the practical constraints like preferable time line of performing activities. For example, someone wants to perform light activities at night but vigorous activities in the morning; in such case, he/she should be suggested to perform light activities in night without suggesting to do extensive activities even if it is required and should be suggested to do the extra amount of activities in the following morning that he/she has escaped during the night.

References

[1] S. N. Blair, H. W. Kohl, R. S. Paffenbarger, D. G. Clark, K. H. Cooper, and L. W. Gibbons, "Physical fitness and all-cause mortality: a prospective study of healthy men and women," *Jama*, vol. 262, no. 17, pp. 2395–2401, 1989.

[2] G. N. Healy, K. Wijndaele, D. W. Dunstan, J. E. Shaw, J. Salmon, P. Z. Zimmet, and N. Owen, "Objectively measured sedentary time, physical activity, and metabolic risk the australian diabetes, obesity and lifestyle study (ausdiab)," *Diabetes care*, vol. 31, no. 2, pp. 369–371, 2008.

[3] A. D. Association *et al.*, "Physical activity/exercise and diabetes," *Diabetes care*, vol. 27, no. suppl 1, pp. s58–s62, 2004.

[4] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 127–140, ACM, 2015.

[5] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International Conference on Pervasive Computing*, pp. 1–17, Springer, 2004.

[6] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[7] N. C. Krishnan and S. Panchanathan, "Analysis of low resolution accelerometer data for continuous human activity recognition," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3337–3340, IEEE, 2008.

[8] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *2007 11th IEEE international symposium on wearable computers*, pp. 37–40, IEEE, 2007.

[9] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[10] F. Foerster and J. Fahrenberg, "Motion pattern and posture: correctly assessed by calibrated accelerometers," *Behavior research methods, instruments, & computers*, vol. 32, no. 3, pp. 450–457, 2000.

[11] A. Subramanya, A. Raj, J. A. Bilmes, and D. Fox, "Recognizing activities and spatial context using wearable sensors," *arXiv preprint arXiv:1206.6869*, 2012.

[12] X. Long, B. Yin, and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6107–6110, IEEE, 2009.

[13] M. V. Albert, S. Toledo, M. Shapiro, and K. Koerding, "Using mobile phones for activity recognition in parkinsons patients," *Frontiers in neurology*, vol. 3, p. 158, 2012.

[14] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 337–350, ACM, 2008.

[15] B. Babcock, M. Datar, and R. Motwani, "Sampling from a moving window over streaming data," in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 633–634, Society for Industrial and Applied Mathematics, 2002.

[16] R. E. G. da Costa Cachucho, J. M. Moreira, and J. Gama, "Activity recognition using, smartphone based, accelerometer sensors," 2011.

[17] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pp. 1–10, ACM, 2009.

[18] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *International Conference on Ubiquitous Intelligence and Computing*, pp. 548–562, Springer, 2010.

[19] Z. John Lu, "The elements of statistical learning: data mining, inference, and prediction," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 173, no. 3, pp. 693–694, 2010.

[20] S. J. Preece, J. Y. Goulermas, L. P. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 871–879, 2009.

[21] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *International Symposium on Ubiquitious Computing Systems*, pp. 516–527, Springer, 2006.

[22] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science*, vol. 34, pp. 450–457, 2014.

[23] H. Vathsangam and G. S. Sukhatme, "Using phone-based activity monitors to promote physical activity in older adults: a pilot study," in *Healthcare Innovation Conference (HIC), 2014 IEEE*, pp. 42–47, IEEE, 2014.

[24] N. D. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell, "Bewell: A smartphone application to monitor, model and promote wellbeing," in *5th international ICST conference on pervasive computing technologies for healthcare*, pp. 23–26, 2011.

[25] L. G. Glynn, P. S. Hayes, M. Casey, F. Glynn, A. Alvarez-Iglesias, J. Newell, G. ÓLaighin, D. Heaney, and A. W. Murphy, "Smart move-a smartphone-based intervention to promote physical activity in primary care: study protocol for a randomized controlled trial," *Trials*, vol. 14, no. 1, p. 1, 2013.

[26] M. Aittasalo, S. Miilunpalo, K. Kukkonen-Harjula, and M. Pasanen, "A randomized intervention of physical activity promotion and patient self-monitoring in primary health care," *Preventive medicine*, vol. 42, no. 1, pp. 40–46, 2006.

[27] H. Gjoreski and M. Gams, "Accelerometer data preparation for activity recognition," in *Proceedings of the International Multiconference Information Society, Ljubljana, Slovenia*, vol. 1014, 2011.

[28] S. J. Preece, J. Y. Goulermas, L. P. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensorsa review of classification techniques," *Physiological measurement*, vol. 30, no. 4, p. R1, 2009.

[29] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *AAAI*, vol. 5, pp. 1541–1546, 2005.

[30] N. Lathia, V. Pejovic, K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Smartphones for large-scale behavior change interventions.," *IEEE Pervasive Computing*, vol. 12, no. 3, pp. 66–73, 2013.

[31] M.-W. Lee, A. M. Khan, and T.-S. Kim, “A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation,” *Personal and Ubiquitous Computing*, vol. 15, no. 8, pp. 887–898, 2011.

[32] A. Sharma, A. Purwar, Y.-D. Lee, Y.-S. Lee, and W.-Y. Chung, “Frequency based classification of activities using accelerometer data,” in *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pp. 150–153, IEEE, 2008.

[33] N. Bidargaddi, A. Sarela, L. Klingbeil, and M. Karunanithi, “Detecting walking activity in cardiac rehabilitation by using accelerometer,” in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pp. 555–560, IEEE, 2007.

[34] V. Vapnik, *The nature of statistical learning theory.* Springer Science & Business Media, 2013.

[35] W. L. Haskell, I.-M. Lee, R. R. Pate, K. E. Powell, S. N. Blair, B. A. Franklin, C. A. Macera, G. W. Heath, P. D. Thompson, and A. Bauman, “Physical activity and public health: updated recommendation for adults from the american college of sports medicine and the american heart association,” *Circulation*, vol. 116, no. 9, p. 1081, 2007.

[36] D. Pearson, “A polynomial-time algorithm for the change-making problem,” *Operations Research Letters*, vol. 33, no. 3, pp. 231–234, 2005.