# Change Detection in Human Physical Activities

By

Naveed Khan (M.S.)

A Thesis submitted in fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

School of Computing and Information Engineering
Faculty of Computing and Information Engineering

ULSTER UNIVERSITY

June 2018

*(I confirm that the word count of this Thesis is less than 100,000 words.)*

# Dedication

*To my parents & family for their prayers, love and sacrifices.*

# Contents

# List of Figures

# List of Tables

# Acknowledgements

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many individuals. I would like to extend my sincere thanks to all of them.

First and foremost, all praises and thanks to **GOD Almighty**, the Almighty for his guidance, protection, peace of mind, good health and the innumerable favors which he has bestowed upon me in order to finish this research.

I would like to express my sincere gratitude and special appreciation to my supervisors **Professor Sally McClean**, **Dr. Shuai Zhang** and **Professor Chris Nugent** for their constant supervision, guidance, encouragement and support in completing this endeavour.

I also wish to thank my friends, colleagues and administrative staff within the School of Computing and Information Engineering for the help and support they have provided since I joined the department.

I gratefully acknowledge the funding received towards my PhD from Ulster University for awarding me Vice Chancellor Research Scholarship (VCRS) which provide me an excellent mobility opportunity that had made a positive impact on my life.

I would also like to say a heartfelt thank you to my Mom, Dad, brothers and sister for their endless support and always believing in me and encouraging me to follow my dreams. Finally, I truly appreciate my wife, for her support, patience, understanding, consistent encouragement and motivation throughout my PhD.

# Abstract

Activity Monitoring is a key feature of health and well-being assessment that has received immense consideration from the research community over the last few decades. In recent years, smart phones with inbuilt sensors have become popular for the purpose of activity recognition. The sensors capture a large amount of data, which contain meaningful events, in a short period of time. Hence, the capability to detect, adapt and respond to such changes performs a key role in various domains such as to identify changes in patient vital signs in a medical domain or to assist in the process of generating activity labels for the purposes of annotating real-world datasets. The sudden change in mean, variance or both may represent a change point in time series data. A change point can also be used to identify the transition from one activity to another. Change point detection is a technique to process and analyse the sensor data and identify the transition from one underlying time series generation model to another.
In this thesis, the existing Multivariate Exponentially Weighted Moving Average (MEWMA) algorithm has been used to automatically detect such change points for transitions in user activity. The MEWMA approach has the advantage that it does not require any assumptions to be made in relation to the underlying distributions to evaluate multivariate data streams and can run in an online scenario.

Following this, the genetic algorithm (GA) has been used to identify the optimal set of parameters for a MEWMA approach to change point detection. The GA optimizes different parameters of the MEWMA in an effort to find the maximum F-measure, which subsequently identifies the exact location of the change point

from an existing activity to a new one. Furthermore, we benchmark our approach against a similar multivariate approach, namely Multivariate Cumulative SUM (MCUSUM) to automatically detect change points in different user activities. In addition, GA and Particle Swarm Optimization (PSO) are also used to automatically identify an optimal parameter set using different parameters for MEWMA and MCUSUM, so as to maximize the objective function that is F-measure. The evaluation is performed using different metric measures based on real and synthetic datasets collected from accelerometer sensor.The experimental results shows that the proposed approach MEWMA outperforms than the bench mark approach MCUSUM.

Hence, the accurate change point detection in the data enable a system to identify changes in user activities and recognize and monitor good behaviour such as healthy exercise patterns based on these activities.

# List of Abbreviations

| | |
|---|---|
| **3D** | Three Dimensional |
| **AANN** | Auto-associative Neural Networks |
| **aHSIC** | additive Hilbert-Schmidt Independence Criterion |
| **CPD** | Change Point detection |
| **CUSUM** | Cumulative SUM |
| **DTW** | Dynamic Time Wrapping |
| **DWT** | Wavelet Transform |
| **ECG** | electrocardiogram |
| **EED** | Early Drift Detection |
| **EWMA** | Exponentially Weighted Moving Average |
| **FFT** | Fast Fourier Transform |
| **FN** | Flase Negative |
| **FP** | False Positive |
| **GA** | Genetic Algorithm |

| | |
|---|---|
| **GLR** | Generalized Likelihood Ratio |
| **GMM** | Gaussian mixture Model |
| **HAR** | Human activity recognition |
| **I.I.D** | Independently Identically Distributed |
| **KLIEP** | Kullback-Leibler Impotence Estimation Procedure |
| **LCL** | Lower Control Limit |
| **MCUSUM** | Multivariate Cumulative SUM |
| **MEWMA** | Multivariate Exponentially Weighted Moving Average |
| **OCSVM** | One-Class Support Vector Machine |
| **OOB** | Oversampling based Online Bagging |
| **PSO** | Particle Swarm Optimization |
| **SSM** | Sub Space Method |
| **TN** | True Negative |
| **TP** | True Positive |
| **UCL** | Upper Control Limit |
| **UOB** | Undersampling based Online Bagging |

# Note on Access to Contents

"I hereby declare that for 2 years following the date on which the thesis is deposited in Research Student Administration of Ulster University, the thesis shall remain confidential with access or copying prohibited. Following expiry of this period I permit

1. the Librarian of the University to allow the thesis to be copied in whole or in part without reference to me on the understanding that such authority applies to the provision of single copies made for study purposes or for inclusion within the stock of another library.

2. the thesis to be made available through the Ulster Institutional Repository and/or EThOS under the terms of the Ulster eTheses Deposit Agreement which I have signed.*

IT IS A CONDITION OF USE OF THIS THESIS THAT ANYONE WHO CONSULTS IT MUST RECOGNISE THAT THE COPYRIGHT RESTS WITH THE AUTHOR AND THAT NO QUOTATION FROM THE THESIS AND NO INFORMATION DERIVED FROM IT MAY BE PUBLISHED UNLESS THE SOURCE IS PROPERLY ACKNOWLEDGED".

# Chapter 1

# Introduction

This chapter provides a synopsis of the thesis subject matter which characterizes the need for change point detection in various applications and more specifically in health sensor data which is used for monitoring human activities. The problem statement formulation and study rationale along with the summary of objectives and methodologies are discussed. As a conclusion, a synopsis of the key contribution and the thesis structure is described and illustrated.

## 1.1 Overview

Human activity recognition (HAR) has emerged as an active area of research over the past few years. It is an important and challenging field which can support numerous pervasive applications. These applications range from health care and assisted living to industrial areas and surveillance. For example, the objective of health care and assisted living is to perform activity monitoring and recognition and facilitate independent living and support in place for patients. In industry, activity aware applications facilitate workers to perform their tasks smoothly and support them so as to avoid mistakes and preserve workplace safety. Activity recognition technology in the deployment of surveillance and security helps to capture and identify threats of terrorism (Chen et al., 2012).

Over the last decade, significant developments have been made in sensor technologies such as low power, low cost, high capacity, miniaturized sensors (Pantelopoulos and Bourbakis, 2010)(Alemdar and Ersoy, 2010) and data processing techniques. These areas have received immense consideration from researchers following the development and advances in such supporting technologies, to move from low level data collection and transmission towards high level context processing, information integration and activity recognition. Meanwhile, a number of real world problems and their solutions has become progressively reliant on activity recognition.

Activity recognition is a complicated process to be categorized into four basic tasks.

1. To choose and deploy appropriate sensors to objects and environments in order to monitor and capture a user's behaviour along with the state change of the environment.

2. To collect, store and process perceived information through data analysis techniques and/or knowledge representation at appropriate levels of abstraction.

3. To create computational activity models in a way that allows software systems/agents to conduct reasoning and manipulation.

4. To select or develop reasoning algorithms to infer activities from sensor data (Chen et al., 2012).

Hence, the aim of activity recognition research is to assist computers to have similar skills to humans for recognizing people's activities. The objective of recognizing activities of daily living is to provide an activity recognition system with sensing abilities.

## 1.2 The need of Change Detection

The implementation of activity monitoring is a key part of a context aware system because it is critical for understanding human behaviour as well as human centric applications. In recent years, numerous wearable sensor technologies such as accelerometers, GPS, light sensors and gyroscope have been used to provide support in data collection providing low power communication and fast processing (Cleland et al., 2014).

Activity monitoring empowers novel context aware solicitations in various domains such as industrial, educational and medical areas. For instance, a vigorous depiction of daily activities can be valuable in assessing health and wellbeing, to care for elderly people and to gain an appreciation of their ability to live independently. The objective of activity monitoring is therefore to automatically detect the activities of daily life. Body movement can be captured through wearable sensors (e.g. accelerometers) to identify different transitions of movement patterns in performing various activities e.g. sitting, walking, and running (Stikic et al., 2011). The characteristics of such sensors are that they are lightweight, unobtrusive and power efficient.

Change point detection is used to identify the transition from one underlying time series generation model to another. The sudden change in mean, variance or both may represent change points in time series data (Camci, 2010). Change point detection algorithms can be categorized as being online or offline. Online change detection algorithms are used in real time systems to observe, monitor and process data as it becomes available. In the offline scenario, firstly the data is collected and then the change point algorithm is used to collectively process all the data. Online change point detection is sequential, fast and minimizes false alarms. One of the problems that still needs to be addressed within this domain of research is automatic change point detection in user activity when the transition has occurred. This involves selection of an algorithm to form a fundamental component of a real system and accurately detect a change point, which for example solicits autonomously user interaction based on transition within an input stream. Such in-time solicitation can be helpful in various situations like annotating activities for making real world annotated datasets or for detecting variation whilst monitoring patient vital signs for example heart rate.

The Crowd Labelling Application (CLAP)(Cleland et al., 2014) has been developed to collect labelled activity data at large scale in a free-living environment. The purpose is to collect a large amount of data for training and testing in order to improve the generalization ability of the AR models. Also, ground truth labels are recorded that represent the user activities. The implementation framework of CLAP is shown in Figure 1.1. The AR module consists of two primary components the AR module and a labelling prompt module. The AR module is used to identify the stationary and non-stationary activities for example ‘stand still’ and ‘walking’. The three second window with a total of three consecutive windows (i.e. nine seconds of data) has been used to detect an activity in the AR module. In addition, an activity class has been assigned using the Gaussian Mixture Model (GMM). Moreover, once the transition is detected, the AR module initiates the label prompting module by displaying icons on screen and thus ground truth for the activity is recorded.

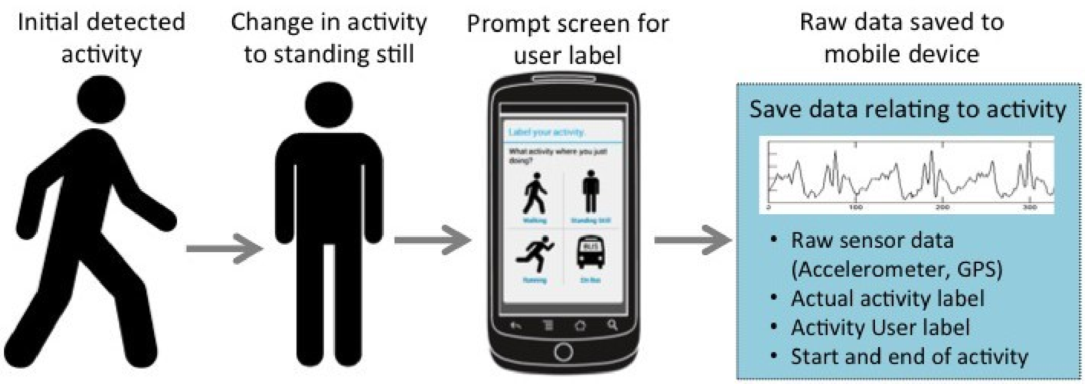



**Figure 1.1:** Framework of Crowd Labelling Application (CLAP) (Cleland et al., 2014)

## 1.3 Change Detection Applications

This section discusses application areas of change detection techniques. In these application areas, change detection refers to the deviation of time points in which the characteristics of a model are subject to abrupt change in characteristics at earlier unknown points. An abrupt change can be considered as any change that occurs either instantly or very fast in the sampling frequency of measurement (Basseville et al., 1993).

### 1.3.1 Quality Control

Change detection has been used in quality control to monitor a continuous manufacturing process to ensure quality. The online quality control procedures evaluate sequential measurement to make decisions. The production process can be classified as in control or out of control. The production process is considered out of control, when disorder happens in the in-control state of production process. Thus, the change detection is used to detect the occurrence of sudden change and the time happens in process. This detection is helpful to maintain the safety of production process, quality and classification of output items (Wetherill and Brown, 1991).

### 1.3.2 Navigation System Monitoring

Various common apparatuses are used in navigation systems for boats, planes, rockets and other moving objects. Example of such systems are radio navigation systems, inertial navigation systems and global satellite navigation for planes (Sturza, 1988). The two sensors gyro and accelerometer are used in inertial navigation systems. These sensors collect the motion and rotation information of moving objects, which is further used to calculate the position and orientation of an object relative to a known starting point. The safety and accuracy of such systems can be achieved by deploying a redundant fault tolerant measurement system. Change detection is used for such problems to identify abnormal measurements in navigation signals, which is highly undesirable. The radio and global satellite navigation systems also required continuous monitoring using redundant measurement to avoid any abnormal measurement. The change detection technique is also applied in these systems to address this issue.

### 1.3.3 Segmentation of Signals

The abrupt change detection based automatic signal segmentation has performed a substantial role in recognition-oriented signal processing (Ukil and Zivanovic, 2006). In the first step, recognition-oriented signal processing required an automatic segmentation of the input signal. The usefulness of a segmentation approach is very pragmatic in numerous biomedical signal process applications such as electrocardiograms (Corge and Puech, 1986), electroencephalogram (Appel and Brandt, 1983) and speech signals (Di Francesco, 1990). The input signals are fragmented using a segmentation approach for homogenous segments and the lengths are adjusted according to the local characteristic of the analyzed signal. Also, the homogeneity for each segment is calculated using the mean or spectral properties. The purpose of the segmentation algorithm is to minimize false alarms, missed detection and low detection delay.

### 1.3.4 Seismic Data Processing

Seismic data analysis is the recording of ground motion. There are a number of factors involved that unsettle the ground such as strong waves in the ocean, sudden change in the atmosphere and human activity (Havskov and Ottemöller, 2010). Such factors cause a higher amplitude in the recording and are identified as seismic events. This kind of events is mainly caused by the abrupt release energy of seismic sources that cause earthquakes. Change detection approaches are used to continuously monitor the seismic data and identify the changes if they happen in the previous and current observations. Thus, an appropriate response can be provided in timely manner.

## 1.4 Problem Statement

Recently, smart phones with inbuilt sensors have received immense consideration from researchers for the purpose of activity recognition. Human daily life activities and behavioural information can be captured through sensors to help in developing telecare applications. A large amount of data can be captured through sensors in a short period of time and used to identify meaningful events. The data can be used to identify a change point which indicates transition to a specific event. Change point detection can be used to classify the transitions occurring in time series data from one model to another. In time series data, the abrupt change in mean, variance or both determines a change point. Different machine learning and statistical techniques can then be used to detect changes automatically from sensor data to identify different activities.

Sample sensor data identifies significant events which can be categorized into different states. However, the huge volume of sensor data is difficult for humans to understand in its raw format. Sensor data which relates to specific events such as sitting, standing, walking and running needs to be identified and classified. Therefore, it is impractical, inaccurate and expensive for humans in term of time to manually perform such tasks.

Automatic change point detection in user activity is still a challenging task when a transition occurred. The output of this study will empower a fundamental component of a real system to accurately detect a change point in user activity. Also, timely solicitation can be useful in different scenarios such as to annotate different activities for generating real world data sets or detecting changes in patient vital signs.

Analysis of the literature reflects that the existing change-point detection methods tend to be sophisticated in nature. Moreover, prior knowledge is often required about the possible change points, and their distribution which could make the implementation of these methods more challenging for an automatic, online change detection application. Furthermore, additional weaknesses could be the observation of numerous estimation parameters, monitoring descriptors and tuning variables. The problem increases when analyzing multivariate data simultaneously. There is also not much previous work on multivariate change-point detection, which takes account of dependencies between different time-series, as well as their individual profiles.

In the literature, a wide variety of sophisticated algorithms has been utilized for change-point detection. The online and offline change point detection may be application specific which requires quick or late response with low or high computational cost. Online change point detection should be lightweight in computational cost unlike offline change point detection. In the literature, the

multivariate approach is not very common and usually used in offline scenarios with high computational cost.
Moreover, we are here interested in running on-line change detection algorithms on a phone or other smart device. We therefore require a lightweight algorithm which is computationally efficient in terms of speed and storage requirements while still achieving high classification accuracy.

## 1.5 Study Rationale and Objectives

Wearable sensors can be used to capture a large amount of data in a short period of time and also to identify meaningful events. Change point detection can be used to identify the transition from one generation model to another. The specific detected change point might indicate a specific event in the data such as sitting, standing, walking and running. This can be identified by a change in the mean, variance or both of a time series data. Therefore, it is impractical, inaccurate and expansive for humans in term of time to manually perform such tasks. The aim of this research is as follows

*"To investigate methods that can automatically detect changes in sensor data streams, with the specific goal to support the annotation of datasets."*

To achieve this aim, a number of objectives have been identified as follows

**Objective 1:** To review the use of existing wearable sensors used for human activity monitoring. In this review, different human activities are identified. Moreover, an overview of state-of -the-art algorithms on change point detection in time series data is presented.

**Objective 2:** To evaluate a multivariate approach for online change and identify the optimal parameter set for accurate change-point detection in activity monitoring with high metric measures such as Accuracy, Precision, G-Means and F-Measure.

**Objective 3:** To implement and evaluate optimization algorithms for the multivariate approach in order to automatically identify optimal parameter set for accurate change-point detection.

**Objective 4:** To develop an evaluation framework to compare different multivariate and optimization approaches for change-point detection. The fusion of such approaches could empower a system to automatically identify optimal parameter set for accurate change detection.

## 1.6 Thesis Structure

This section summarizes the structure of this thesis and highlights the contents of each chapter.

**Chapter 2: A Taxonomy of Change-Point Detection in Activity Monitoring**

This chapter provides an overview of wearable sensors with a focus on change detection in human activity monitoring. The review begins with basic concepts, definitions on wearable sensors, different activities and multivariate date related to different human physical activities. An overview of state-of -the-art algorithms on change point detection is presented. This chapter concludes with discussion of some challenges associated with multivariate data and change point detection.

**Chapter 3: Parameter Exploration for Online Change Detection in Activity Monitoring**

This chapter describes the online change point detection algorithm MEWMA for accurate change point detection in activity monitoring. The multivariate approach is used due to the nature of accelerometer data with 3 dimensions, which are not independent to each other. The different parameters are explored manually with the aim of achieving better performance and accurate change point detection.

**Chapter 4: Parameter Optimization for Online Change Point Detection in Activity Monitoring Using Genetic Algorithm (GA)**

This chapter presents the genetic algorithm (GA) to automatically identify an optimal parameter set for each activity so as to maximize the fitness function such as F-measure. A genetic algorithm is used to mimic the process of evolution by taking a population of strings, which encodes possible solutions, and combining them based on the fitness function to produce solutions that are high performing. The F-measure is used as a fitness function to find the overall effectiveness of the activity by combining the precision and recall.

**Chapter 5: Evaluation Framework to Analyze Different Multivariate Approaches and Optimization Techniques**

This chapter discusses the evaluation framework of different multivariate approaches for change point detection in activity monitoring. Different optimization approaches are also used to automatically identify an optimal parameter set for accurate change-point detection. The different metric measures are used for evaluation of obtained results for accurate change-point detection. Also, a $t$-test is also performed to find statistical significance for all evaluation metrics and the results justified that MEWMA with PSO is statistical significance with 95% confidence achieved for all metrics measures.

**Chapter 6: Conclusion and Future Work**

This chapter presents the conclusion to this work and highlights the future direction.

## 1.7 Published Work

This section lists the conference and journal papers outputs form the thesis.

**Conference articles**

- Khan, N., McClean, S., Zhang, S. & Nugent, C. 2015. Parameter Optimization for Online Change Detection in Activity Monitoring Using Multivariate Exponentially Weighted Moving Average (MEWMA). International Conference on Ubiquitous Computing and Ambient Intelligence. Springer, pp. 50-59.
- Khan, N., McClean, S., Zhang, S. & Nugent, C. 2016. Using genetic algorithms for optimal change point detection in activity monitoring. 29th International Symposium on Computer-Based Medical Systems (CBMS). IEEE, pp.318-323.
- Khan, N., McClean, S., Zhang, S. & Nugent, C. 2016. Change Point Detection Using Multivariate Exponentially Weighted Moving Average (MEWMA) for Optimal Parameter in Online Activity Monitoring. International Conference on Ubiquitous Computing and Ambient Intelligence. Springer, pp. 1-10.

**Journal articles**

- Khan, N., McClean, S., Zhang, S. & Nugent, C. Optimal Parameter Exploration for Online Change-Point Detection in Activity Monitoring Using Genetic Algorithms. Sensors, **2016**.16(11), 1784-1799.
- Patterson, T., Khan, N., McClean, S., Nugent, C., Zhang, S., Cleland, I. & Ni, Q. Sensor-Based Change Detection for Timely Solicitation of User Engagement. IEEE Transaction on Mobile Computing, **2017**, 16(10), 2889-2900.

**Proposed Submission**

- Khan, N., McClean, S., Zhang, S. & Nugent, C. Evaluation of statistical techniques for online change point detection in activity monitoring. (*to be submitted to IEEE Transactions on Biomedical Engineering.*)

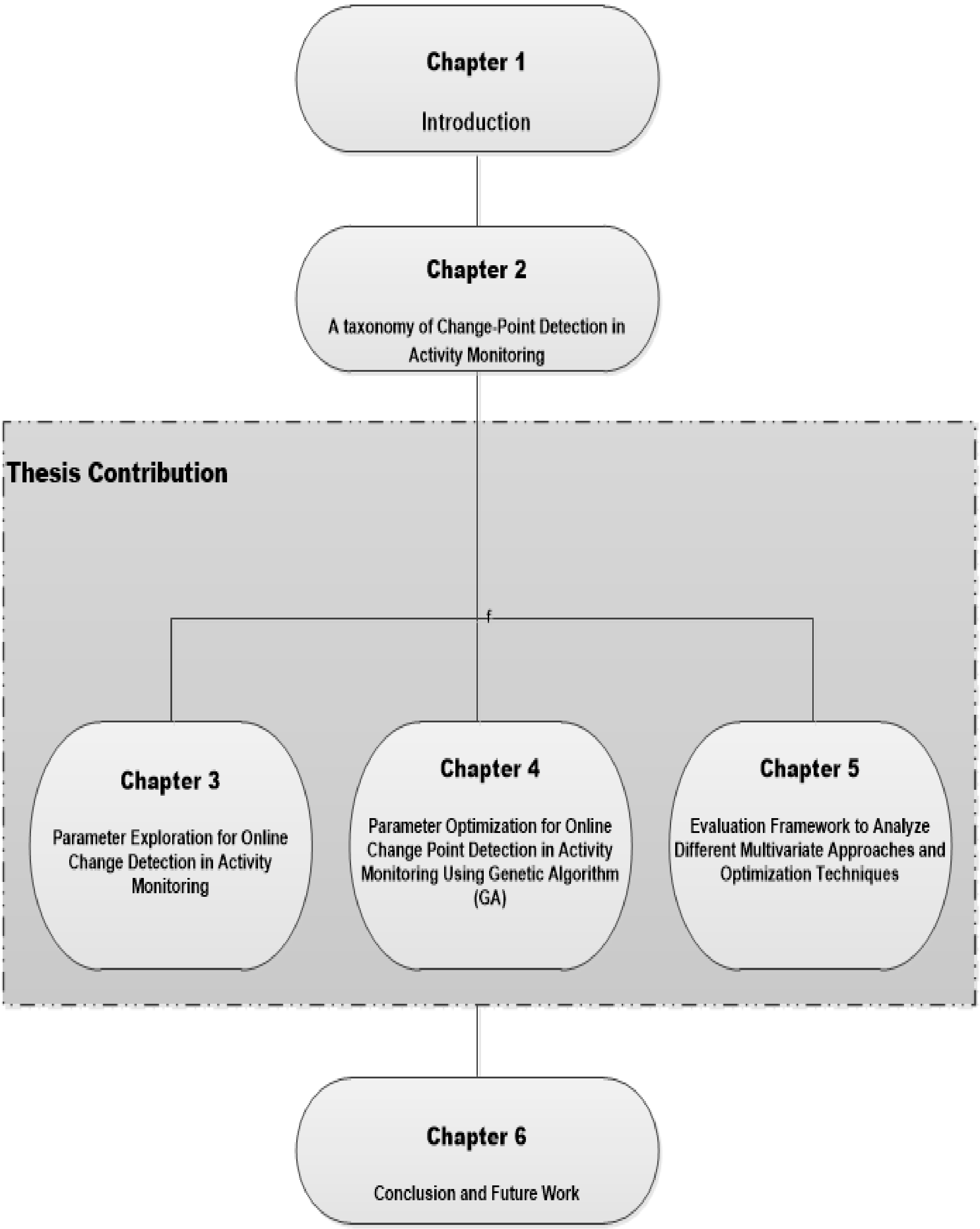



**Figure 1.2:** Overview of Thesis Organization

# Chapter 2

# A Taxonomy of Change-point Detection in Activity Monitoring

## 2.1 Introduction

This chapter provides an overview of wearable sensors with a focus on change detection in human activity monitoring. The review begins with basic concepts, definitions on wearable sensors, different physical activities and multivariate data related to human physical activities. An overview of state-of-the-art algorithms on change point detection is presented. This chapter concludes with discussion of some challenges associated with multivariate data and change point detection.

## 2.2 Sensors

Sensors are sophisticated devices that is used to detect and respond to signals. They "are devices that acquire information about stimuli in the outside world" (Francis et al., 2009). A sensor converts a physical parameter such as blood pressure, temperature or humidity etc. into a signal that can be measured electrically. However, sensors are different in type, purpose, output signal and technical infrastructure. Sensors generate signals in response to stimuli that can be measured. In recent years, these sensors have become low cost, wireless, and deployable in real world, mobile settings (Cook and Krishnan, 2015). There are various types of sensors available for activity learning and monitoring. Here we focus on health sensors that are attached to an individual that is performing an activity. The different kind of wearable sensors is given in Figure 2.1.

### 2.2.1 Wearable sensors

Wearable sensors are widely discussed in the literature for the purpose of activity recognition. Moreover, the sensors attached to an individual body collect data about gesture, movement and actions performed by an individual. These sensors can be sewn into smart garments, worn as watches or placed on the human body. However, in recent years due to technology advancement, these wearable sensors are also embedded in smart phones that are routinely carried by an individual as they perform their daily life activities.

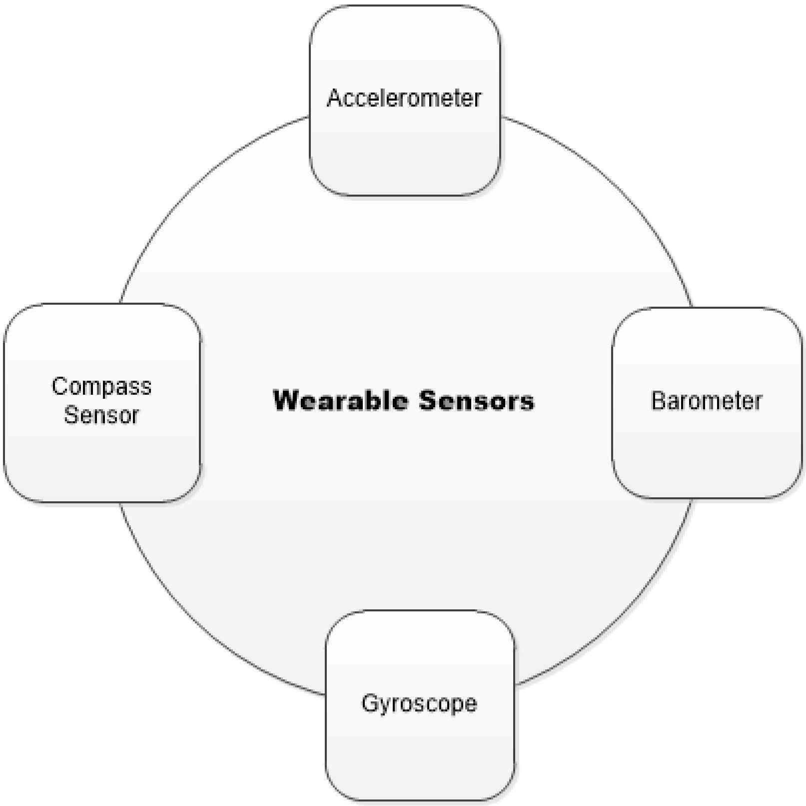



**Figure 2.1:** Different wearable sensors used to capture human movement

A number of wearable sensors are used such as accelerometer, gyro and Magnometer in the literature for the purpose of activity monitoring.

#### 2.2.1.1 Accelerometer

The most common sensor used in activity monitoring is an accelerometer which is either worn or carried by an individual. Such kind of sensor is very effective and efficient in monitoring body actions that involves repetitive motions such as sit, walk, stand, run etc. (Chen et al., 2012). The accelerometer is an electromechanical device which measures acceleration forces. The force can be static like the constant force of gravity or dynamic like moving or vibrating accelerometers. Acceleration can be detected in two or three axes to sense motion and orientation (Chen et al., 2012). The three-dimensional accelerometer sensor measures acceleration along the x, y and z axes as shown in Figure 2.2. The acceleration can be calculated as change in velocity over time as $a = \frac{\Delta v}{\Delta t}$. A time stamp can also be returned with the three axes readings when the person carries the device so the change in direction or velocity can help in detecting the change in acceleration, which makes such sensors optimal for detecting different type of movements. Furthermore, accelerometers are heavily used in smart phones to measure the physiology of the human movements. Also, the acceleration data can identify notional patterns is a specific time-period, which is useful and helpful in detecting and recognition complex human activities. However, an actometer, only measures the body movement and acceleration in different directions. The device 'counts' the amount of movement and the higher the count, the more active the wearer (Eaton, 1983). These acceleration changes are converted into signals which are used for further processing as shown in Figure 2.3. The start and end points of a motion can be identified by using the acceleration changes in the data. These signals provide information to facilitate context awareness such as motion, acceleration and gesture.

#### 2.2.1.2 Gyroscope

The gyroscope sensor is used to measure the earth's gravity, which helps to determine the orientation. The model has a freely-rotating plate known as a rotor, attached onto a spinning axis in the centre of a big and stable wheel

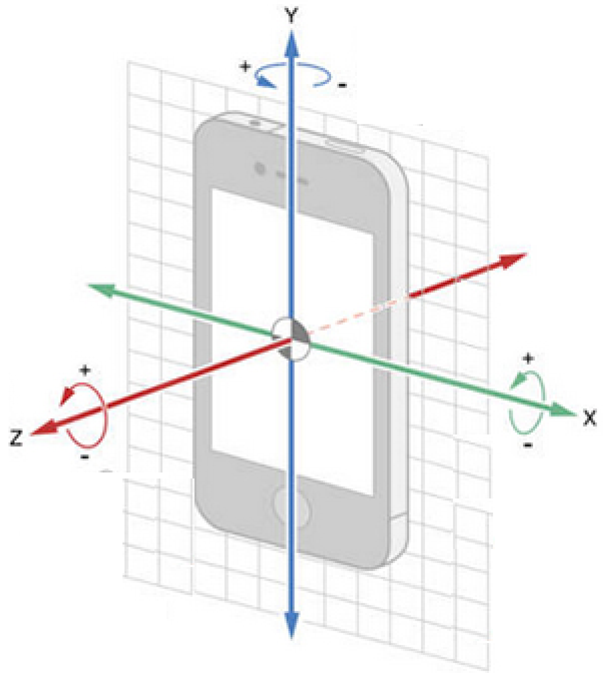



**Figure 2.2:** A smart phone with 3 axis accelerometer signal

(Chen et al., 2012). In contrast to an actometer, the gyro measures the change in angular velocity over time and it is calculated as $v = \frac{\Delta\theta}{\Delta t}$.

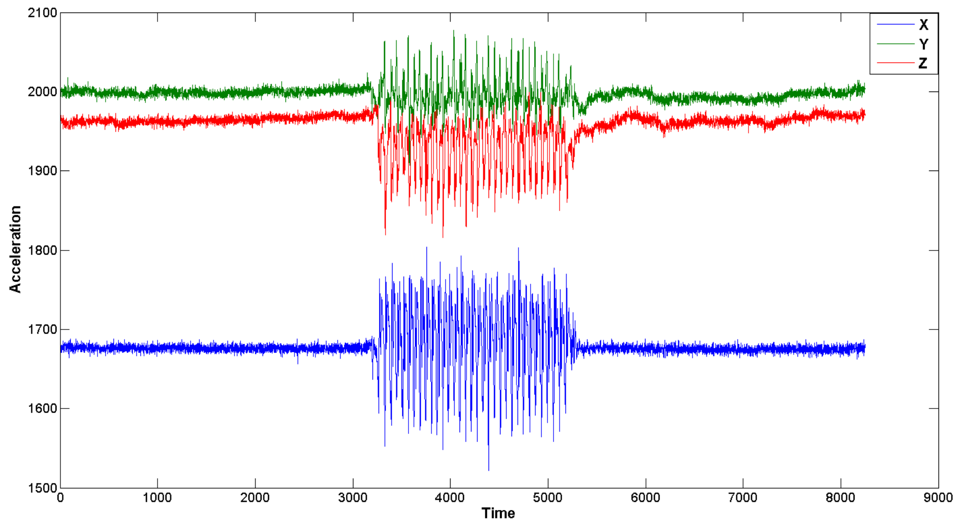



**Figure 2.3:** A sample of 3 axis of an accelerometer signal for the stand to walk activity

However, the gyro sometimes experiences accumulated error due to drift but is mostly used in combination with other sensors to provide more information about the motion.

#### 2.2.1.3 Digital compass

The digital compass is equipped with a sensor called a magnetometer with a simple orientation in relation to earth's magnetic field (Calzada et al., 2014). It can auto rotate the digital map reliant on the physical orientation. It is also valuable for detecting and locating metallic objects within its sensing radius. The human activities can be modelled based on their proximity to these detected objects.

#### 2.2.1.4 Barometer

The Barometer sensor is widely used to measure the atmospheric pressure. The pressure is calculated to forecast short term changes in weather. Hence, the primary aim is to detect atmospheric change and pressure, specifically weight of air. Also, the barometer sensor is used to detect useful events such as a closing door.

#### 2.2.1.5 Pedometer

The pedometer is an electromechanically device that is used to detect the motion of hands or hips and count the number of steps taken by an individual. However, the actometer is used to capture the body movement rather than time (Eaton, 1983). As discussed earlier, a number of body-attached sensors have been used to capture and evaluate the body movement in free living environment as shown in Figure 2.1. However, the accelerometer is widely being accepted and becoming popular for activity monitoring in a free living environment (Mathie et al., 2004). The accelerometers have considerable advantages in monitoring of human movement. As such they measure the accelerations in motion along reference axes. Also, using accelerometers, the frequency and intensity of physical activity are analysed simultaneously, which makes them preferable to the pedometer and actometer, which are attenuated by impact or tilt. Additionally, some accelerometers can also measure the gravity to provide tilt sensing information with respect to a reference plane when they rotate with objects. Such features of accelerometry data provides sufficient information about the monitoring of different physical activities (Yang and Hsu, 2010).

## 2.3 Activities

Wearable sensors have been used widely in the literature to monitor different activities performed by an individual. Physical activity is defined as "any bodily movement produced by skeletal muscles resulting in energy expenditure above resting level" (Caspersen et al., 1985). The objective of human activity analysis is to identify actions and intent of a user from an observation. Thus, activity learning is a significant concept because it is critical for understanding human behaviour. An individual can perform the number of activities in his daily life such as walk, sit, run, upstairs, downstairs etc. The different types of activities are categorized such as Actions, Transitioning activities and Activities of Daily Living are shown in Figure 2.4. The Actions are considered as a movement of specific parts such as throwing or holding something, bending arm or shaking feet (Murao and Terada, 2014). The physical activities such as sitting, standing, walking and running are considered low level activities because it involved body-wide movement and postures. Moreover, high level activities are a collection of low- level activities and consist of a number of activities in a sequence such as watching TV, driving etc. (Huynh, 2008).
The complexity of these activities varies and depends on the type of activities performed. Activities which are static in nature such as standing or sitting are easier to evaluate than the activities like walking or running because they are periodic in nature. Moreover, the activities such walking, running, upstairs and downstairs are also very hard to separate because of high motion similarities in movement patterns (Khan, 2011). Therefore, sensors which are able to detect motion and acceleration have gained interest in motion aware system develop-

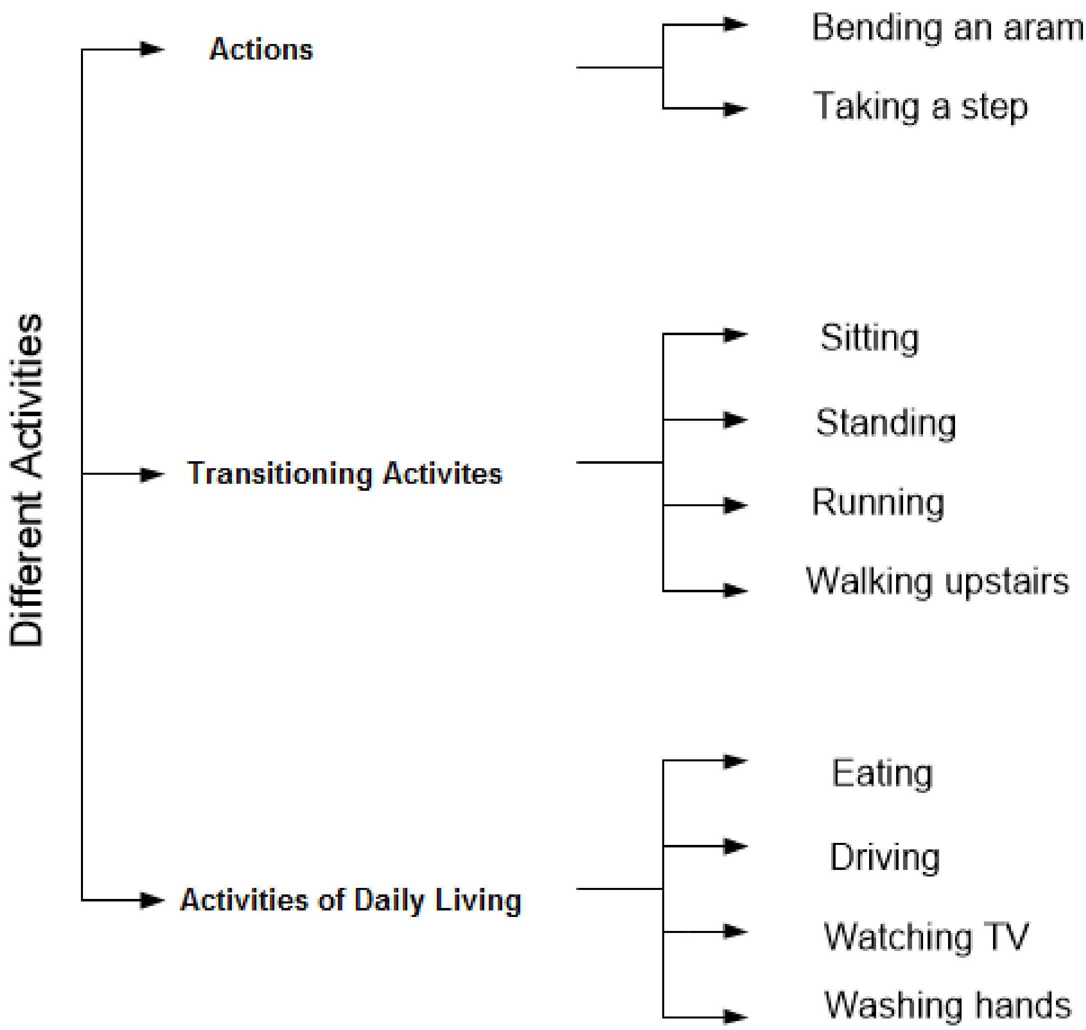



**Figure 2.4:** Different kind of activities

ment. The most widely used inertial sensor is the accelerometer which can be used as an expedient tool for assessment of human motion in a free living environment (Mathie et al., 2004).

In the literature, various methods have been used to retrieve valuable information from raw sensor data. The main steps can be classified as raw sensor data acquisition, segmentation and windowing, feature extraction and detection. The block diagram of the whole procedure is shown in Figure 2.5.

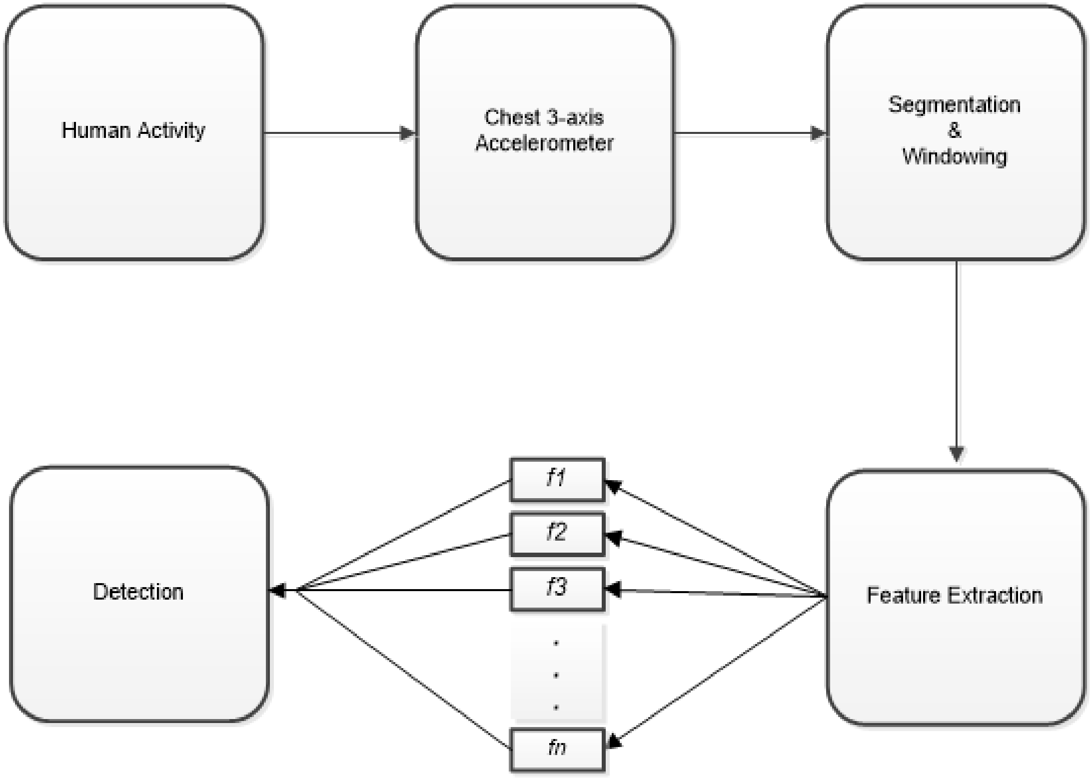



**Figure 2.5:** Block Diagram of the whole procedure

## 2.4 Segmentation

As discussed in section in 2.3, once the activities are captured then the segmentation algorithms are used to analyse the input data. Segmentation is the process of dividing the input data into segments in order to get useful information from a continuous stream of data. The classifier cannot produce meaningful events without proper selection of segmentation or subsequence (Lovrić et al., 2014). An input subsequence with high discriminative power enables the classifier to produce better results. The segmentation is divided into classes supervised and unsupervised. In supervised segmentation, the sample data is provided along with ground truth labels from which appropriate segment boundaries can be learned while, in unsupervised segmentation, features of the data are utilized alone, without supervised guidance to identify the boundaries in the sequence data. In online scenario, the window is defined simultaneously as the data arrives from the continuous real-time activity. However, in the offline scenario, first data is collected and then segmentation is applied to all data as a whole. Moreover, continuous sensor data is difficult to analyse and challenging for detecting change in continuous activity. Therefore, in this study we are focusing on unsupervised segmentation technique to properly segments continuous input data streams. The different methods that have been used for segmentation of time series data are as follows.

### 2.4.1 Sliding window algorithm

In this approach, a small consecutive sequence is initialized for a specific time of 1s, 2s, or 3s until the end of the time series data. Each segment from the left is traversed at the first data point of the time series, then the algorithm tries to approximate the data at the right with increasingly longer segments (Keogh et al., 2001). At a specific point $i$, the potential segment for an error is greater than the user threshold, thus the sub-sequence until $i-1$ is transformed into a segment. The approximation error of the linear segment is calculated using the sum of squares, or the residual error. This is calculated by taking all the vertical differences between the best-fit line and the actual data points, squaring them and then summing them together.Then $i$ is incremented and the process continues until the whole time series is converted into a piecewise approximation. The error is computed in the same way for algorithms in Figures 2.7 and 2.8. The pseudo code of the sliding window algorithm is shown in Figure 2.6. The sliding window algorithm is interesting due to its simplicity, intuitiveness, online property and has computational cost of $O(nL)$ where $L$ denotes the segment average length (Avci et al., 2010). The analysis of sequential patterns in stream data are very important because it can be exploited to improve the prediction accuracy of our classifiers. Therefore, the sliding window algorithm has been used in diverse areas such as to analyze weather forecast data (Yahmed et al., 2015), ECG data analysis (Jeon et al., 2014) and statistical data analysis (Yu et al., 2014). The aim is to find and explore the number of intervals and width of intervals and to process them.

```
Input: A time series T= t1,t2,...,tn (sequence), max_error (e)
Output: number of segmented time series T= t̄1, t̄1..., t̄N (sub
sequence)

begin
Seg_TS= Sliding_Window(T, max_error)
anchor=1; //starting point
while(i< n) // not finished segmenting time series
    i=2;
    while calculate_error(T[anchor:anchor+1])< max_error
        set i equal to i+1;
    end while
    Seg_TS = concat(Seg_TS,create_segment(T[anchor:anchor+(i-1)]);
    anchor=anchor+i;
    end while
end
```

**Figure 2.6:** Sliding Window Algorithm

### 2.4.2 Top-Down Algorithm

A top down algorithm has also been used for segmentation by dividing the time series data into a number of segments and splitting the data at their best locations. The two segments are evaluated and checked that the approximation error is below a user threshold. If not, the algorithm continues recursively and divides into subsequence's until the threshold condition is fulfilled.

The pseudo code of Top down algorithm is shown in Figure 2.7. This algorithm works recursively with a computational cost of $O(n^2M)$ where $M$ identifies the segment's number(Keogh et al., 2001). The Top Down algorithm is used in data mining and image processing fields. In the data mining field, the approach has been used at multiple abstraction level in order to identify a framework for mining sequence databases (Li et al., 1998). Also, (Shatkay and Zdonik, 1996) has used it for time series databases that could be able to support and approximate distinct and complex queries. Moreover, in image processing, the approach has been used on images to detect discontinuities that are potentially indicative of object boundaries (Borenstein and Ullman, 2008).

```
begin
        Seg_TS = Top_Down(T , max_error)
        best_so_far = inf;
    for i = 2 to length(T) - 2 //Find best place to split the time series.
            improvement_in_approximation = improvement_splitting_here(T,i);
        if improvement_in_approximation < best_so_far
                breakpoint = i;
                best_so_far = improvement_in_approximation;
        end
    end for
    // Recursively split the left segment if necessary.
    if calculate_error(T[1:breakpoint]) > max_error
            Seg_TS = Top_Down(T[1: breakpoint]);
    end
    // Recursively split the right segment if necessary.
    if calculate_error( T[breakpoint + 1:length(T)] ) > max_error
            Seg_TS = Top_Down(T[breakpoint + 1: length(T)]);
    end
```

**Figure 2.7:** Top Down Algorithm

### 2.4.3 Bottom up Algorithm

The Bottom up approach is the complement of the top down algorithm and initializes segmentation of time series data by dividing the data of length $n$ into $n/2$ segments at best approximation.

The adjacent sub sequences are fused to get larger segments of time series data and computational cost is also calculated for this process simultaneously. The algorithm is used to iteratively fuse the lowest cost pair till stopping criteria are met. The adjacent sections $i$ and $i+1$ are fused and the algorithm stores values. Initially, the new segment with its right neighbour merging cost is calculated and then the $i-1$ segment with its new larger neighbour merging cost is recalculated. The computational cost is the same as sliding window algorithm. The pseudo code for the algorithm is shown in Figure 2.8.

The approach has been used extensively in piecewise linear approximation of time series data (Van Laerhoven and Schiele, 2009). Furthermore, the algorithm has been used in system dynamics and data mining in order to monitor and analyze various events occurs in the data (Wirsch, 2014). In medical data analysis, the algorithm has also been used to provide a high level of representation for a medical pattern matching system (Hunter and McIntosh, 1999).

```
begin
  Seg_TS = Bottom_Up(T , max_error)
  for i=1:2:length(T)  //Create initial fine approximation.
      Seg_TS = concat(Seg_TS, create_segment(T[i:i+1]));
  end;
  for i=1:length(Seg_TS) - 1 //  cost of merging segments.
   merge_cost(i)=calculate_error([merge(Seg_TS(i),Seg_TS(i+1))]);
  end;
 while min(merge_cost)< max_error // While not finished.
  index = min(merge_cost);   // Find "cheapest" pair to merge.
  Seg_TS(index) = merge(Seg_TS(index),Seg_TS(index+1)); // Merge
them.
  delete(Seg_TS(index+1));  // Update records.
  merge_cost(index) = calculate_error(merge(Seg_TS(index),
Seg_TS(index+1)));
  merge_cost(index-1) = calculate_error(merge(Seg_TS(index-
1),Seg_TS(index)));
 end;
end;
```

**Figure 2.8:** Bottom Up Algorithm

As discussed earlier, all segmentation algorithms have been used to segment and extract relevant information from a large time series dataset. However, the selection of appropriate segmentation algorithm must be based on a consideration of features such as whether the algorithm is online or offline and the minimum time complexity utilized in the processing of data. In our study, we used a sliding window algorithm because of minimum time complexity utilization which can run in an online scenario. The comparison summary of the aforementioned segmentation algorithms is shown in Table 2.1.

**Table 2.1:** Summary of three segmentation algorithms

| Algorithm | Online | Time Complexity |
|---|---|---|
| Top-Down | No | $O(n^2M)$ |
| Bottom Up | No | $O(nL)$ |
| Sliding Window | Yes | $O(nL)$ |

## 2.5 Feature Extraction

The objective of feature extraction is to find and extract more prominent features of the data to accurately represent the original data. Specifically, the large volume of input data can be represented as a minimum set of features denoted as a feature vector and the process is known as feature extraction. Accurate change point detection can be achieved by using these feature vectors in a data segment to identify various activities. The extracted feature can also be used to give as an input to classification algorithms (Preece et al., 2009). The features extracted from sensor data can be categorized into three domains, namely Time domain, Frequency domain and Time-frequency domain.

### 2.5.1 Time domain features

The time domain features are provided for analysis of mathematical functions, physical signals or time series data. The time domain indicates the variation in amplitude of signal with time. In the time domain, the extracted features determine the key observation of waveform features and signal statistics. The following features are used in the time domain to extract useful features (Dargie, 2009) mean, variance, covariance, standard deviation and correlation.

#### 2.5.1.1 Mean

The mean is calculated as the average of the input data. The mean is calculated using Equation 2.1.

$$\mu = \frac{\sum x_i}{N} \tag{2.1}$$

Where $\sum x_i$ is the sum of all values and $N$ is the number of the data values.

#### 2.5.1.2 Variance

The variance determines how distant each number is from the mean in the dataset. The variance measures the differences between each value in the dataset from the mean, squaring the differences and divides the sum of the square by the total number of values in the dataset. The variance is calculated using Equation 2.2

$$\sigma^2 = \frac{\sum (x_i - \mu)^2}{N - 1} \tag{2.2}$$

where $x_i$ denotes each value in the dataset, $\mu$ is the mean of dataset and $N$ is the total number of values in the dataset.

#### 2.5.1.3 Standard deviation

The standard deviation is used to measure and evaluate the extent of variation or dispersion for a set of data values. A calculated standard deviation close to zero is considered as the data point close to mean otherwise, a high value represents the situation where the data points are scattered over a large range of values. The standard deviation is the square root of the variance. The standard deviation can be calculated using Equation 2.3.

$$S_x = \sqrt{\frac{\sum (x_i - \mu)^2}{N - 1}} \tag{2.3}$$

where $x_i$ s the values in the dataset, $\mu$ is the mean and $N$ is the total number of values in the dataset.

#### 2.5.1.4 Covariance

The covariance is used to measure the association between two variables such as $x$ is associated with changes in a second variable $y$. In other words, covariance measures the extent to which two variables are linearly associated. The covariance can be calculated using Equation 2.4.

$$COV(x, y) = \frac{1}{n-1}\left(\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})\right) \tag{2.4}$$

where $x_i$ and $y_i$ are the values of dataset, $\overline{X}$ and $\overline{Y}$ is the mean and $n$ is the size of dataset.

#### 2.5.1.5 Correlation

Correlation is a statistical measurement of the relationship between two or more variables. Correlation measuring how well a linear relationship fits the observed data.The correlation of two variables could be positive, negative or zero. The variables are said to be positively correlated where higher values of one variable are associated with high values of the other variable while for negative correlation, the high values of one variable are associated with the low values of the other. A zero correlation means that the variables are uncorrelated. The correlation lies between $+1$ and $-1$. The correlation can be calculated using Equation 2.5.

$$\rho_{x,y} = \frac{COV(x, y)}{\sigma_x \sigma_y} \tag{2.5}$$

where $COV(x, y)$ is the covariance and $\sigma_x$ , $\sigma_y$ are the standard deviations of $x$ and $y$.

Moreover, the cross correlation has also been used in the literature to evaluate the similarity between acceleration signals from various axes for same or various segments (Bao and Intille, 2004).

### 2.5.2 Frequency domain features

The frequency domain is used to analyze data using a mathematical function or a signal with respect to frequency. Analysis of the frequency domain is most widely used in signals or functions that are periodic over time. Transformation is the most important concept of this domain and is used to convert a function from the time domain to frequency domain. The frequency spectrum provides an alternate view of the signal which identify how much of the signal lies within each frequency band over a range of frequencies. The frequency domain features can be obtained using Fourier transforms for example, the window of sensor data can be transformed to the frequency domain using the Fast Fourier Transform (FFT). The output of this transformation gives a set of coefficients which identify the amplitude of the signal frequency and energy distribution of the signal. There are different metrics such as maxima, energy, entropy and median entropy used to represent the spectral distribution of the signal energy.

#### 2.5.2.1 Maxima

In the frequency spectrum, the $n$-maxima is used for different capacities to compare the dominant frequencies. The significant size of frequency sample can be obtained by summing the $i^{th}$ maximum and dividing by the total number of maxima. After completing this, evaluation is made by using the $n^{th}$ maximum and inspecting its deviation from the average maximum. For example, for human movement $n = 10$ and for a car $n = 1000$ are normally used because mostly low frequencies arise in human movement.

#### 2.5.2.2 Energy

The spectrum structure is revealed by the set of sensor reading of spectrum energy. In this scenario, spectrum energy indicates the overall energy analysis of the sensor readings. To evaluate the spectrum, first it is divided into $n$ sub-bands and then the energy in each band is normalized using the total energy of the spectrum. The correlation test is applied to each sub band to evaluate numerous measurements (Dargie, 2009). A higher correlation between measurements signifies a small variation between sub bands energy. This paper concluded that correlation can be measured by using the average difference of sub band energies. It is calculated as the sum of the squares of the magnitude of the frequency content as given in Equation 2.6.

$$S_E = f(n)^2 \tag{2.6}$$

where $f$ is the number of frequency components.

#### 2.5.2.3 Entropy

The normalized information entropy of discrete FFT coefficient can be calculated using entropy. The entropy also helps in identifying change in signals with identical energy values but corresponding to numerous activity patterns. Entropy is used to identify activities with simple acceleration patterns and also with more complex patterns (Bao and Intille, 2004).
For example, uniform movement of legs is observed in cycling but in running more complex patterns are observed and therefore many FFT components are displayed. This difference indicates the entropy is higher frequency domain for running than cycling. To find spectral entropy, first the FFT of the signal is calculated then the Power Spectral Density (PSD) of the signal is calculated by simply squaring the amplitude spectrum and scaling it by number of frequency bins using Equation 2.7.

$$\hat{P}(\omega_i) = \frac{1}{N}|X(\omega_i)|^2 \tag{2.7}$$

Now, normalize the calculated PSD by dividing it by a total sum using Equation 2.8. The normalized signal is further used to find the entropy and can be calculated using equation 2.8.

$$p_i = \frac{\hat{P}(\omega_i)}{\sum_{j=1}^{n} \hat{P}(\omega_j)} \tag{2.8}$$

The normalised signal is further used to find the Power Spectral Entropy can be calculated using equation 2.9.

$$H = -\sum_{j=1}^{n} p_i \ln p_i \tag{2.9}$$

### 2.5.3 Time frequency domain features

This approach is used to evaluate features of both time and frequency domains. The wavelet approach is used to analyse complex signals and the primitive intent is to detect transitions between different activities (Preece et al., 2009). The original signal is divided into a series of coefficients which incorporate the temporal and spectral information about the original signal. These coefficients help to identify a localized temporal instance, which is a change in frequency characteristics of the original signal. This approach is used in the sensor signal to identify points in the signal which helps in detecting change from one activity to another. The wavelet analysis is used to determine time-frequency features that represent the original signal.

#### 2.5.3.1 Wavelet Coefficients

A number of studies has shown that time-frequency features can be analysed by the discrete wavelet transform (DWT) (Sekine et al., 2000) (Nyan et al., 2006). In this approach, the original time domain signal with maximum frequency $f$ is divided into a linear approximation through a low pass filter and a high pass filter. This decomposition enables the half band filter to accurately regenerate the original signal without losing any information. However, in the consequent decomposition, the approximation signal divides again from the previous level to the second approximation with more observable coefficients (Mallat, 1989). This operation is repeated until the appropriate decomposition level. In the literature, various set of features have been extracted using the discrete wavelet transform (DWT) from sensors such as accelerometers. The wavelet analysis can be applied to body worn sensors to decompose signals into several coefficients. These coefficients correspond to the specific data of each frequency band. The various types of DWT such as Haar and Daubachies are used to decompose the signal. In (Sekine et al., 2000), the first set of wavelet features is proposed using accelerometer data. The accelerometer signal is decomposed using the wavelet transform and the feature characterize as the signal power measurement. The sum of squares is used to calculate the coefficients at level 4 and 5. The acceleration data is sampled at $250Hz$. This approach can also be used to identify wavelet coefficients for lower sampling frequency. In body worn sensors, wavelet analysis has been used to identify the points of change in frequency content of sensor signals. In (Nyan et al., 2006), wavelets were used to identify such points in order to evaluate the transition times among various type of gait data analysis. The above extraction techniques have been widely used in the literature for feature extraction from the observed data (Preece et al., 2009). The characteristics, such as cost, robustness and expressive power, of features extracted using different extraction techniques were analyzed. The time domain features have advantages over other techniques such as they do not require the laborious task of framing and fourier transformation, avoid complexity of pre-processing and consume less energy (Dargie, 2009). Therefore, they can be deployed in resource constrained environment. Hence, in our experiments, we have used accelerometer data so time domain features are used for feature extraction form the observed data.

## 2.6 Different approaches for change point detection in time series data

Change point detection can be used to classify the transitions occurred in time series data from one model to another. In time series data, the abrupt change in mean, variance or both determines a change point (Camci, 2010).Change point algorithms can be categorized as online or offline. The online change detection algorithms are used in real time systems and used to observe, monitor and process data concurrently as available. However, in the offline scenario, first the data is collected and then the change point algorithm used to collectively process all data. A number of algorithms have been used in the literature to detect change in time series data. An activity-recognition algorithm was previously used to detect changes in daily life activities with the help of a Gaussian mixture classifier (Cleland et al., 2014) based on mobile data. Some activities, such as stationary and nonstationary, were classified as standing-still and running, respectively. The authors have used three consecutive windows of nine seconds each in the entire activity-detection process in their proposed solution. Moreover, some activities such as stand-still and walking could be detected and labelled simultaneously at changeover points. Some of the limitations of the approach were the short delay that caused incorrect detection of user activity and unsuitability in the real-time scenarios in such situations when the user transitions from nonstationary "walking" to stationary "standing-still". The cumulative sum control chart (CUSUM) is a technique that is effective in detecting small shifts, using the mean of the process in cardiovascular events (Zhang et al., 2011). The authors have used some core methods in order to evaluate physiological monitoring modules. The core methods are the hierarchal online activity recognition method and the biometric extraction method. In the hierarchal online activity-recognition method, first the pre-processing is performed using a finite impulse response filter. In the second step, the fast Fourier transform (FFT) has been used to convert the signal from the time domain to the frequency domain and extract the mean and energy feature from the pre-processed data. Finally, those features having direct impact on the performance of the activity recognition algorithm were selected. In the biometric extraction method, first the heart rate values are extracted from the echocardiogram (ECG) signal. The FFT was applied to attenuate low frequency noise and eliminate waveform irregularities from the signal. Finally, the 2-pass filter was used to find the local maxima of the ECG signal and detect the significant R-peaks. However, CUSUM cannot detect sudden shifts in accelerometer data and is therefore ineffective for such changes. An independent random sequence has been used by (Jain and Wang, 2015) to detect changes using a univariate change detection algorithm. In the first step, the index of the changeover for the most likely point inside the processing window is calculated. Secondly, the hypothesis is verified for the expected change point whether correct or incorrect for the expected change. The algorithm utilized small memory, computationally inexpensive and did not required knowledge of underlying distribution. Furthermore, the authors in (D'Angelo et al., 2011), have proposed a fuzzy Bayesian change-point detection technique using the posterior probability of the current run length in time-series data. The proposed technique works in two folds. First, the fuzzy set technique is applied to cluster and transform the initial time series data into a new time series with a beta distribution. Secondly, the new time-series data is further used by a Bayesian change-point model to detect the change points. Then, the change points' positions were estimated using the Metropolis-Hastings algorithm. The advantage of using this approach is that it does not require a priori knowledge of the distribution, but it is computationally expensive. Similarly, the One-Class Support Vector Machine algorithm (OCSVM) (Vlasveld, 2014) has been used for change detection in human activities. The high dimensional hyper sphere has been used to model the sensor input data. Moreover, the radi of hyper sphere is used to analyze and evaluate the distribution of change point detection. The increase or decrease in changes corresponds to various activities. The data is modelled by a high dimensional hyper sphere. Change point detection is the distribution based on the analysis of radii of the hyper sphere, which changes i.e. increases or decreases corresponding to various events. The early drift detection (EED) approach has been used for detecting small and abrupt change in time series data (Baena-García et al., 2006). In this approach, the distance between two classification errors was used and calculated from the average distance and its standard deviation. If the calculated distance is small, the change is detected otherwise the new point belongs with the previous points. The event detection in human-activity monitoring can significantly reduce transmissions (Brusey et al., 2009).The transition between postures is difficult to classify and therefore remains unlabelled. The data is captured through accelerometer sensors placed on different parts of the body. Moreover, a posture-activity monitoring system has been developed that can classify posture from the observed data. Time-based filtering, a naïve voting scheme, and an exponentially weighted voting scheme have been used to improve the posture classification accuracy. The exponentially weighted voting scheme outperforms than other schemes in event detection. Also, the transmission is reduced from an original 10 Hz to about 600 event transmissions in 30 min. The kernel density estimator approach has been used in (Chen et al., 2012). In this approach, the density estimation ratios have been calculated for population of data. Furthermore, these estimation ratios were used to identify the change points in the data. This approach has the advantage of automatic model selection and the convergence property. However, the disadvantages include difficulty in calculating density estimation for high-dimensional data, which can be slow and less robust. The template matching algorithm known as dynamic time wrapping (DTW) (Murao and Terada, 2014) has been developed for gesture detection. For each activity one template is used and the number of templates identifies the processing time for recognition. In DTW, one template from each activity has been used with a number of templates that directly affects the computational performance. Furthermore, the subspace identification algorithm has been proposed in (Kawahara and Sugiyama, 2009) for change point detection in time series data. The objective is that subspace is used to span by the columns of an extended observability matrix which is approximately equal to the one spanned by the sub sequences of time series data. In this technique, the estimation of change point detection is based on subspace identification, the extended observability matrix column space of subspace method (SSM) and assessing newly arrived data based on this subspace. The advantage of this approach is to handle rich amount of data more precisely than conventional approaches due to its implicit utilization. The Kullback-Leibler Impotence Estimation Procedure (KLIEP) has been developed for change point detection in time series data (Sugiyama et al., 2008). The density estimation ratios of population data are used in KLIEP algorithm. This approach has the advantages of convergence properties and automatic model selection. However, the limitations are that the density estimation for high dimensional data is difficult to calculate which makes it slow, less robust and also have a convex optimization problem. A change detection method with feature selection for high dimensional time series data has been proposed in (Yamada et al., 2013) known as additive Hilbert-Schmidt Independence Criterion (aHSIC). It used weighted sum of HISC values between features and their associated pseudo binary labels. The HSIC is also known as a kernel based independence measure because it used feature selection during its detection measure estimation. The advantage of this approach is that it uses those features which have more impact on abrupt changes occurs in the data. The Auto-associative Neural Networks (AANN) (Hu et al., 2007) has been used to detect anomaly detection in multivariate time series data. The AANN consist of three layers called input, hidden and output layer. The number of neurons in the input and output layers are the same while there are less neurons in the hidden layer. The AANN is trained using the input layer to encode the data using the input layer and forms principle components at the bottleneck or hidden layer. Moreover, the principle components are decoded to original data using the output layer. The network is trained using the input data and testing data is then applied to detect changes for anomaly detection in time series data. The early detection of an anomaly might help in fault diagnostic to take timely action for maintenance. The proposed approach is very effective for anomaly detection but an immediate convergence of AANN required a high percentage of normal data for training. Also, the time complexity is quite high and not suitable to be use in a real time scenario.

The Information-Theoretic approach (Dasu et al., 2006) has been used to detect change in multi-dimensional data streams. This approach is a nonparametric approach and requires no assumption of underlying distribution. The relative entropy also called Kullback-Leibler distance has been used to measure the difference between two distributions. Moreover, the theory of bootstrapping using statistical methods has been used to identify the statistical significance of calculated measurements. However, more complex methods are required for K-L distance to increase the change detection performance and power significance of measurement.

### 2.6.1 Episode detection algorithm

The Minimum Description Length (MDL) principle is used in (Tatti and Vreeken, 2012) to detect and identify the sequential patterns that encapsulate the best data in the sequence data. The sequential data is encoded using sets of serial episodes and the encoded length is further used as a quality score. Here two heuristic approaches such as SQS-Candidates and SQS-Search have been used to efficiently identify the best patterns in the sequence data. The SQS-Candidates approach was used to filter a candidate collection over a large data set and SQS-Search was used to efficiently mine models directly from data. The accurate detection of an abnormal episode in a sequence data within a fixed size window has been investigated in (Atallah et al., 2003). The episode is considered as an ordered collection of subsequences of a large data stream. The problem of finding frequent episodes in an event sequences with respect to window size constraint is difficult to analyse and evaluate. Probabilistic analysis, similarly to a parallel episode case and an arbitrary set of serial episodes, has been used to find an abnormal episode in sequence data streams. A specific threshold has been used for rejecting the null hypothesis for each window size and identifying an abnormal episode. In (Laxman et al., 2007a) a windows-based counting algorithm known as WINEPI has been proposed for discovering frequent episodes in sequential data. The framework consists of defining episodes as partially ordered sets of events, and looking at windows on the sequence. The WINEPI algorithm has been used for finding all episodes from a given class of episodes that are frequent enough. The algorithm was based on the discovery of episodes by only considering an episode when all its sub episodes are frequent, and on an incremental checking of whether an episode occurs in a window. The implementation shows that the method was efficient in detecting episodes but has high computational cost. A new notion for episode frequency has been proposed which is based on the non-overlapped occurrences of an episode in the given data sequence. An efficient counting algorithm (based on finite state automata) has been presented in (Laxman et al., 2007b) to obtain the frequencies for a set of candidate episodes. This algorithm has the same order of worst case time and space complexities as the windows-based counting algorithm of (Laxman et al., 2007a). However, through some empirical investigations, the non-overlapped occurrences based algorithm has been found to be much more efficient in terms of the actual space and time needed, and, on some typical data sets, it runs several times faster than the windows-based algorithms. Another important advantage of the non overlapped occurrences count is that it facilitates a formal connection between discovery of frequent episodes and learning of generative models for the data sequence in terms of a specialized family of Hidden Markov Models (Laxman et al., 2005).

### 2.6.2 Android activity recognition API

Activity recognition API in Android systems is an API (ActivityRecognition, 2018) that automatically detects user-activities like still, running, walking, cycling, tilting, and driving etc. by periodically reading short bursts of sensor data. Such APIs are currently implemented and used in many fitness apps such as GoogleFit in order to provide the information about the user activities e.g. the distance the user travelled, and the steps taken. Mobile Apps can integrate activity recognition without dealing with complexity of pattern analysis on raw sensor data through the activity recognition API (Zhong et al., 2015). Therefore, rather than to go through the whole process of data collection, feature extraction and classifier training, software developers can utilise this AR service through an API. Initially four types of activities were supported: Stationary, On Foot, Cycling and in Vehicle. In an update, three more activities were added: Walking, Running and Tilting. According to its documentations1, the Android AR service makes use of low-power, on-board sensors to recognise the user's current physical activity with efficient energy consumption (ActivityRecognition, 2018).

Analysis of the literature reflects that the current change-point detection methods tend to be more sophisticated in nature. Modelling the data distribution in a multidimensional data stream is a challenging task, where most of the approaches discussed in (Jain and Wang, 2015), (Zhang et al., 2011), (Dasu et al., 2006), (D'Angelo et al., 2011), (Baena-García et al., 2006),(Brusey et al., 2009) and (Murao and Terada, 2014) have been applied only for univariate data and most of the approaches discussed in (Kawahara and Sugiyama, 2009), (Sugiyama et al., 2008), (Yamada et al., 2013) and (Hu et al., 2007) have been applied offline. Moreover, prior knowledge is often required about the possible change points and their distribution which could make the implementation of these methods more challenging for an automatic, online change detection application. Furthermore, the other weaknesses could be the observation of numerous estimation parameters, monitoring descriptors and tuning variables. The difficulty increases when multivariate data are analysed simultaneously.

# Chapter 3

# Parameter Exploration for Online Change Detection in Activity Monitoring

## 3.1 Introduction

This chapter discusses online change detection in univariate and multivariate data in activity monitoring. In recent years, smart phones with inbuilt sensors have become popular for the purpose of activity recognition. The sensors capture a large amount of data in a short period of time which contains meaningful events. The change points in this data are used to specify transition to a distinct event which can be used in various scenarios such as to identify patient vital signs in the medical domain or requesting activity labels for generating real-world labelled dataset. Within this chapter a detailed overview of the Multivariate Exponentially Weighted Moving Average (MEWMA) algorithm has been provided; MEWMA is an existing technique and used as an innovative approach for change point detection. Moreover, the online univariate approach from the literature has been discussed and implemented for change point detection. This provide a benchmark for evaluation of MEWMA. This chapter evaluates the online univariate approach and standard MEWMA results and also improve the standard MEWMA for change point detection by exploring and tuning the different parameters such as $\lambda$ (the "forgetting factor"), window size and significance values.

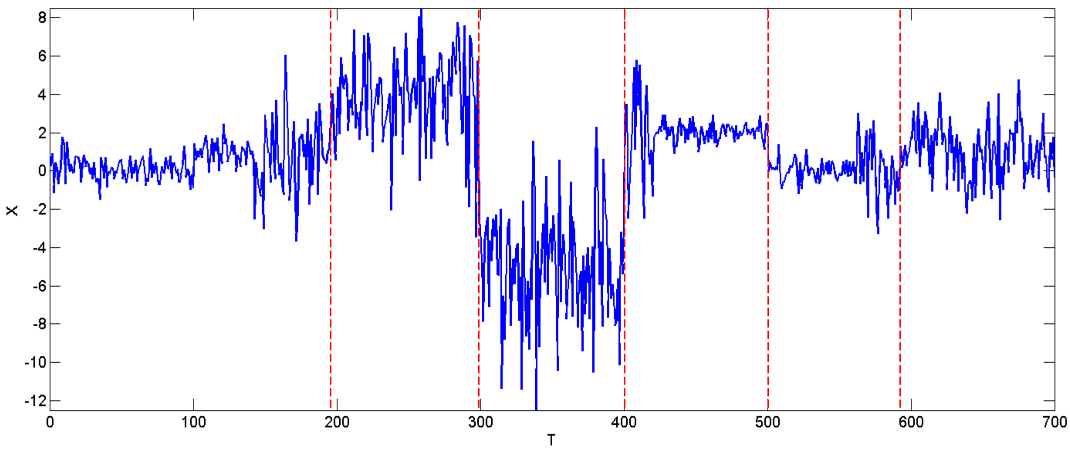
## 3.2 Change Detection

Change detection can be defined as the process of identifying differences in the state of an object by observing it at different times. Hence, a statistical framework is required to develop methods for change detection. In time series data, the point at which the statistical properties of an underlying process change are known as a change point. The change in mean, variance or both can represent the change in time series data (Ross and Adams, 2012). The change detection in mean and variance can be shown in Figure 3.1. Also, the change in mean, variance and covariance can be shown in Table 3.1.

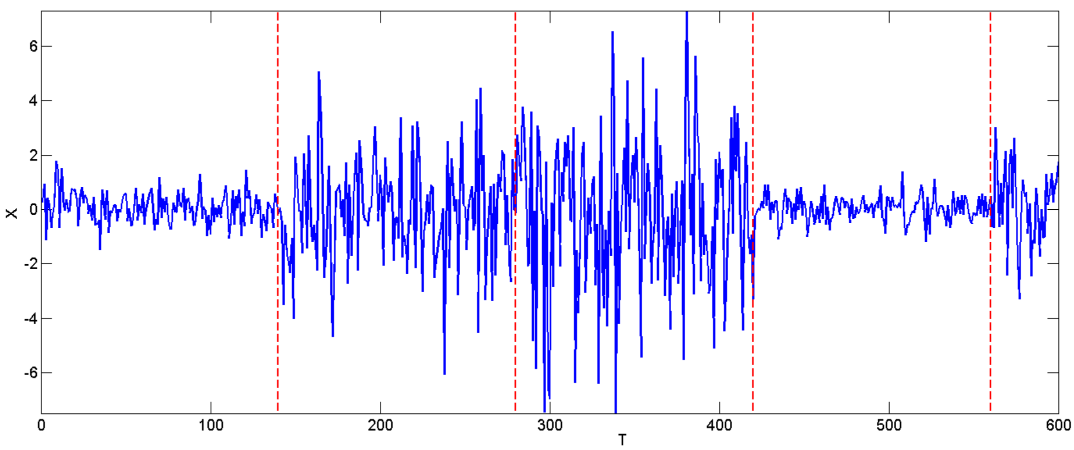
In the literature, the primary focus has been on offline change detection, when the data has been received and analyzed as a whole for change point detection. However, this thesis mainly focus on online change detection where data is observed, monitored and analyzed sequentially as soon as the new data becomes available. The change detection methods are used to identify and locate the possible change points in the previous observations. The change detection methods can be classified into parametric and non-parametric. In parametric method, the detection method requires distribution knowledge about the data while in non-parametric, the detection methods does not require the assumption of the knowledge distribution for the data (Eckley et al., 2011).
As presented in Table 3.1, the mean and variance is changing abruptly from region to region and detected as a change point which is shown in Figure 3.1.

**Table 3.1:** Mean and Variance

| **Mean ($\mu$)** | **variance ($\sigma$)** |
|---|---|
| 0.4292 | 0.5036 |
| 3.7432 | 5.8373 |
| $-5.0938$ | 21.0956 |
| 1.9392 | 1.2143 |
| 0.000091 | 2.1943 |




**(a)** Change in Mean




**(b)** Change in Variance

**Figure 3.1:** (a) and (b) Change points in time series data (Ross and Adams, 2012)

Furthermore, the dataset was collected using 3-axis accelerometer by (Zhang et al., 2011) has been used to collect and captured data for different activities performed by each participant. These activities consist of Sit to Stand, Stand to Sit, Stand-Walk Corridor-Stand, Stand-Walk Downstairs-Stand and Stand-Walk Upstairs-Stand. In all these activities, the $x$, $y$ and $z$ axis show the acceleration magnitude of the input signal and the ground truth changes identifies the actual change happens in these activities. The sit to stand and stand to sit activity are shown in Figure 3.2 and Figure 3.3 respectively. In these activities, the mean has changed abruptly than the variance and covariance when the participant transitioned from one activity to another such as sit to stand and stand to sit as shown in Figure 3.2 and Figure 3.3 respectively.

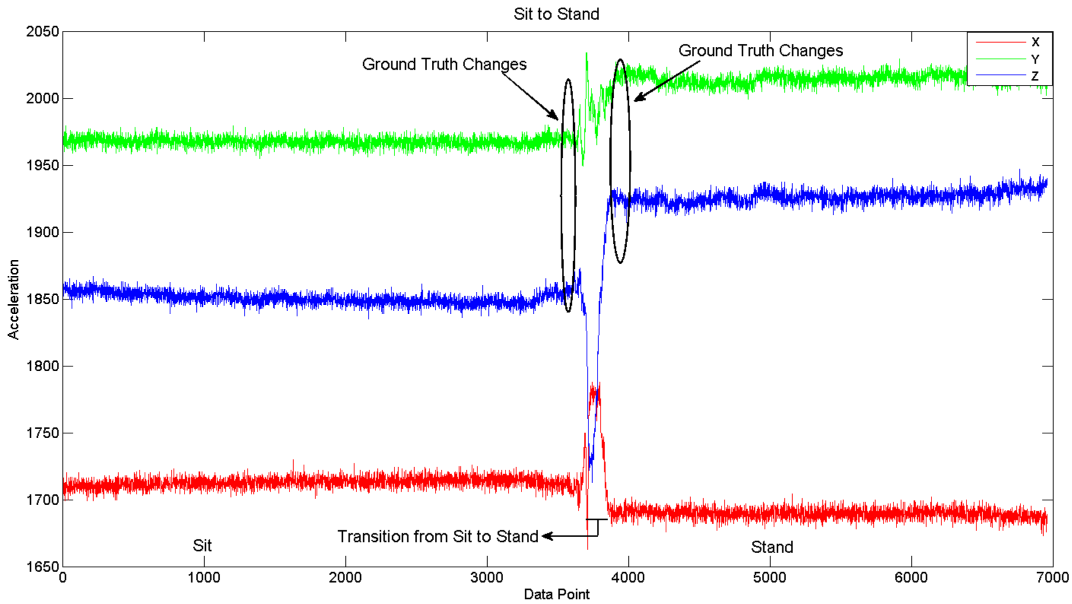



**Figure 3.2:** Sit to Stand (Zhang et al., 2011)

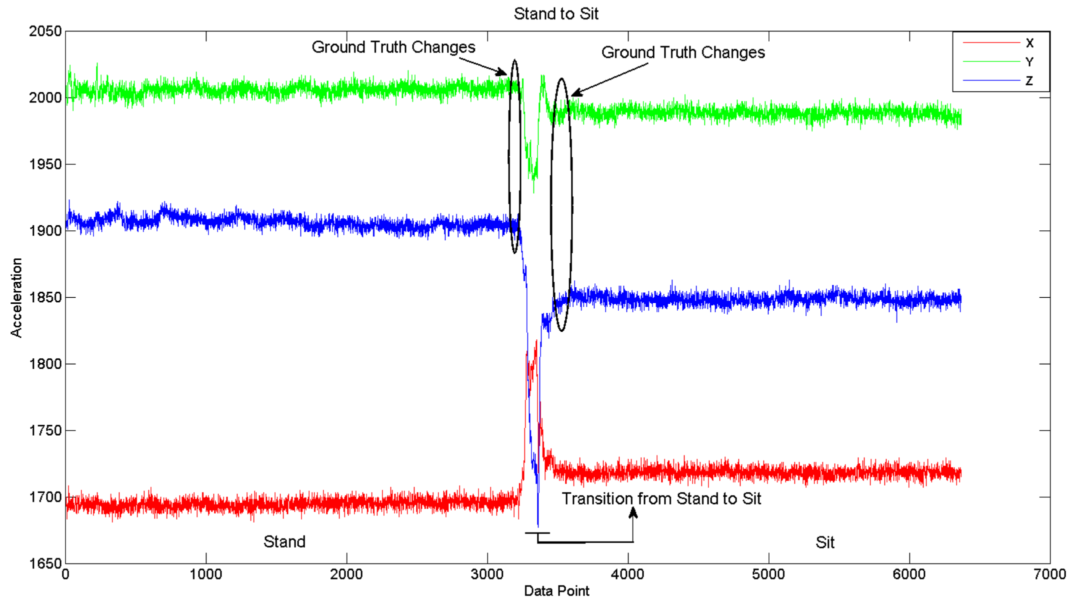



**Figure 3.3:** Stand to Sit (Zhang et al., 2011)

However, for the stand - walk corridor- stand activity, the mean, variance and covariance have changed suddenly when the participant transitioned from stand to walk and then stand as shown in Figure 3.4. Likewise, for the Stand-Walk Up Stairs-Stand and Stand-Walk Downstairs-stand activities, the mean, variance and covariance have also changed abruptly when the participant transitioned from stand to walk Down Stairs or Upstairs and then stand as shown in Figure 3.5 and Figure 3.6 respectively.

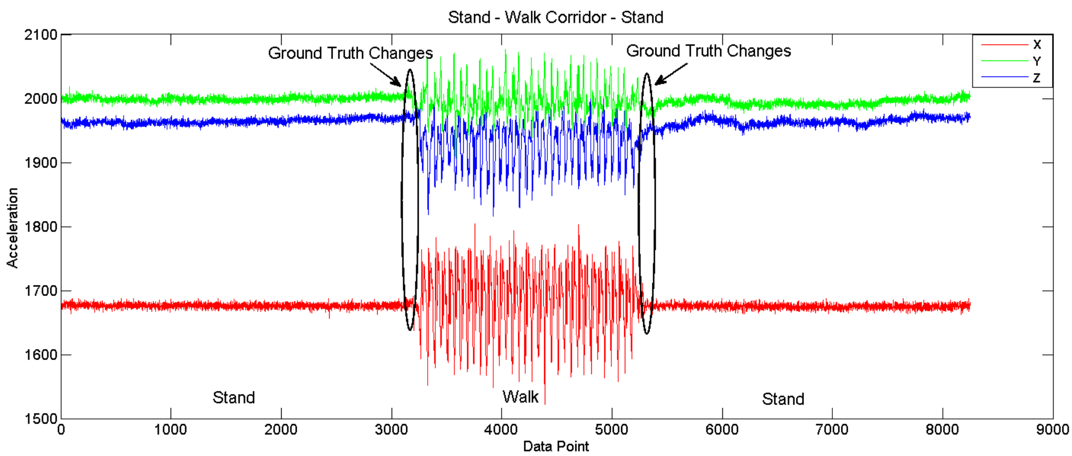



**Figure 3.4:** Stand - Walk Corridor - Stand (Zhang et al., 2011)

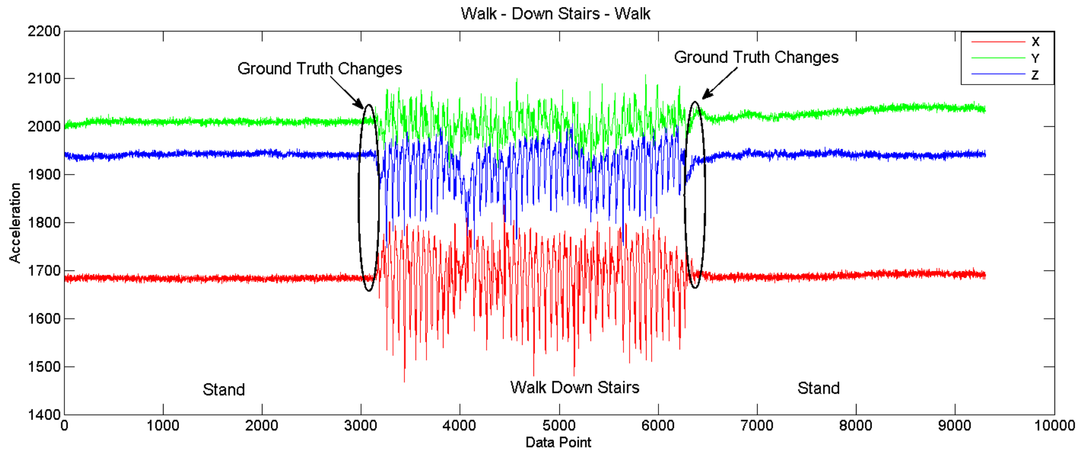



**Figure 3.5:** Stand - Walk Downstairs - Stand (Zhang et al., 2011)

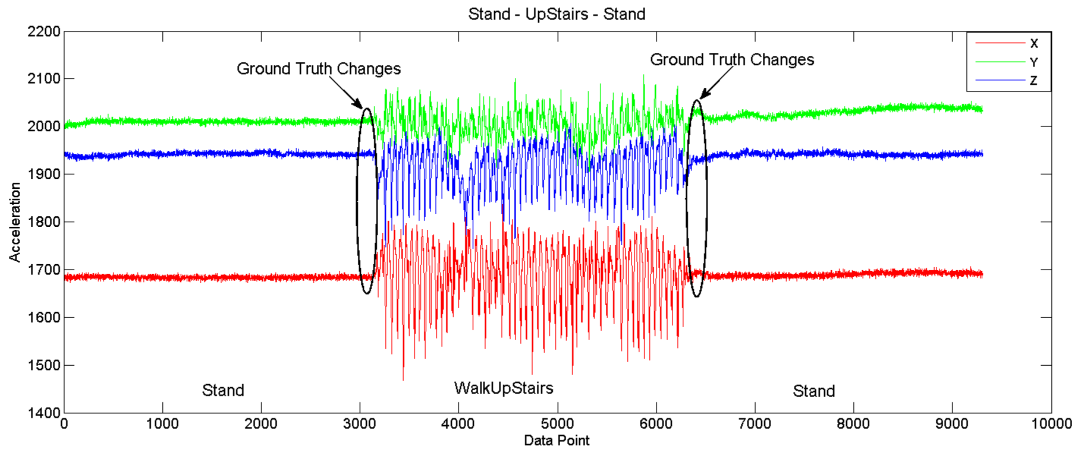



**Figure 3.6:** Stand - Walk Upstairs - Stand (Zhang et al., 2011)

Moreover, change detection is used to find and specify change points in the data streams, where each change point in a current distribution identifies the occurrence of significant change. Both univariate and multivariate data can be used for change detection analysis. However, in literature most of the existing approaches have used univariate data (Bifet and Gavalda, 2007),(Jain and Wang, 2015),(Kifer et al., 2004),(Takeuchi and Yamanishi, 2006) for change detection and very few approaches have been used for change detection in multivariate data (Kuncheva and Faithfull, 2014), (Song et al., 2007). Also, these proposed approaches did not consider multivariate data streams for change point detection.

## 3.3 General Change Point Models

The analysis of time series data is based on number of variables which could be the number of objects that can be analyzed as sample in an experiment. The time series data can be univariate or multivariate. The univariate data is simple to analyze because it involves one variable for analysis. However, multivariate data is more complex to analyze as it involves more than one variable for analysis. In multivariate data analysis, the difficulty increases when multivariate data are analysed simultaneously. The multivariate data gives richer picture of the data as compared to univariate data. In this thesis, we mainly focus on online change point detection in multivariate data streams.

Suppose a time series data consists of data $x_{1:n} = (x_1, x_2, ..., x_n)$.Each observation can be identified at time $t$. The $x_i$ can be considered as univariate or can be extended to multivariate data. A number of change points $m$ can be assumed in the data and the position for the change points is represented by $\tau_1, \tau_2, ..., \tau_n$.The order of the change points is considered as sequential so that $\tau_i < \tau_j$ if and only if $i < j$. The data is divided into $(m + 1)$ segments for identifying $m$ change points, hence, the $i^{th}$ segment consists of data $y_{\tau_{i-1}} + 1 : \tau_i$ In each segment the data is to be assumed as independently identically distributed $(i.i.d)$ (Sharkey and Killick, 2014). The sequence distribution can be represented as:

$$X_i = \begin{cases} F_0 & if \ i \leq \tau_1 \\ F_1 & if \ \tau_1 < i \leq \tau_2 \\ F_2 & if \ \tau_2 < i \leq \tau_3 \\ ... & \\ F_n & if \ i > \tau_n \end{cases}$$

Hence, in traditional parametric approaches, the information about assumption of underlying distributional form $F_i$ is required before and after the changepoint which is a limitation of such approaches. However, an alternative technique is required which do not require such assumptions about underlying distribution about data because most of the real-world problems and processes do not have explicit and well-defined behaviour (Hawkins and Deng, 2010). The imprecise assumptions about the distributional form of data can greatly affect the rate of false positive specifically in sequential change detection (Ross and Adams, 2012). Therefore, change detection approaches with the distribution free characteristics need to be develop that can provide high performance. In addition, change detection methods can be used online to detect and identify change point in online settings as soon as the data becomes available.

### 3.3.1 The Online Univariate Change Detection Algorithm

The online univariate algorithm for change point detection is presented by (Jain and Wang, 2015). In univariate analysis, the data has one variable which do not describe any relationship and its purpose is to represent, summarize and find patterns in the data. Unlike other algorithms such as cumulative sum control chart (CUSUM) and generalized likelihood ratio (GLR), this algorithm does not required knowledge about the underlying distribution before and after the change point. The change detection algorithms are classified into two categories namely, that of detecting changes in dependent and independent random sequences. The independent random sequences with distributions parameterized by scalar parameters while dependent sequences characterized by a multi-dimensional parameter vector (Basseville et al., 1993).

The change detection problem in second category (independent random sequence) is more complicated as compared to first category (dependent random sequence). In order to make dependent random sequence to independent one, a systematic approach is used to detect change in a dependent random sequence and perform a “whitening” transform and inverse "whitening" transform). The original whitening transform , together with the inverse transform $T^{-1}$ produce an identity mapping as shown in Figure 3.7. The purpose of transformation is “peeling away” the dependency induced by the dependency models to reveal the “driving force” behind the system change (Jain and Wang, 2015). Further, the driving force can be treated as an independent random process, which implies that the change-detection algorithm in the first category are again applicable.

Specifically, the proposed algorithm is implemented for change detection in univariate independent random sequences. The algorithm works on two steps: hypothesis and verification. In the first step, the index of the most likely changeover point inside the processing window is calculated. In the second step, the hypothesis is verified as to whether the change is correct (change occurs) or incorrect (no change).

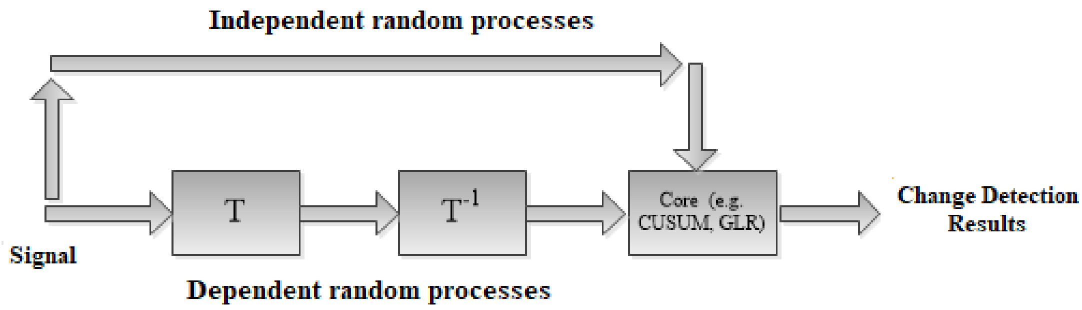



**Figure 3.7:** Unified framework for change detection problem in both independent and dependent random processes (Jain and Wang, 2015)

The mathematical formulation for change detection in an independent random sequence is described by (Jain and Wang, 2015).

Consider a data stream $X_1, X_2, ..., X_n$ of length $n$. The data stream consists of random variables and can be formed in two distributions such as $X_1, X_2, ..., X_{k-1}$ assumed to be I.I.D having distribution $D_1$ and $X_k, X_{k+1}, ..., X_n$ assumed to be I.I.D with distribution $D_2$.The changeover point such as $k$ is unknown and could occur anywhere between 1 and $n$.

The algorithm is implemented to detect change in online applications in which the data is continuously streamed from source to receiver. Therefore, a sliding window is used to analyse the data stream for change detection. The change point can occurs within the window to apply over a data stream. A non-overlapping window over the data stream is applied over a successive group of $n$ data items.

Hence, the algorithm does not have enough information about an underlying distribution, therefore, certain statistics of the input data (data stream) are calculated to find the change points in the data. The algorithm operates by sliding an analysis window of length $n$ over the data stream, $X_i$, $i = 1, ...n$ and examines if the changeover point occurs within the window. To simplify the notion, the $n$ data items within the analysis window $X_1, ..., X_n$ regardless of their true positions in the stream. The algorithm operates on a slightly longer data stream $X_{1-m}, ..., X_0, X_1, ..., X_n, X_{n+1}, ..., X_{n+m}$ , or the processing window is padded with $m$ extra data items at each end $(X_{1-m}, ..., X_0, \quad and \quad X_{n+1}, ..., X_{n+m})$. This padding is necessary to ascertain with accuracy the occurrence of the changeover point from index 1 up to index $n$. The statistics is denoted by $f$. Then for an index $l$, $1 < l \leq n$ within a processing window, the following four functions of $l$ (the means and variances statistics before and after component distributions, separated at $l$) are calculated using Equation 3.1 and 3.2.

$$\begin{aligned} \overline{f}_1(l) &= \frac{\sum_{i=1-m}^{l-1} f(X_i)}{n_1} \\ S_{f_1}^2(l) &= \frac{\sum_{i=1-m}^{l-1} (f(X_i) - \overline{f}_1)^2}{n_1 - 1} \end{aligned} \tag{3.1}$$

and

$$\begin{aligned} \overline{f}_2(l) &= \frac{\sum_{i=l}^{n+m} f(X_i)}{n_2} \\ S_{f_2}^2(l) &= \frac{\sum_{i=l}^{n+m} (f(X_i) - \overline{f}_2)^2}{n_2 - 1} \end{aligned} \tag{3.2}$$

where $n_1 = m+l-1$ , $n_2 = n+m-l$ and $n_1+n_2 = n+2m$. Moreover, If the window stream contains two distributions, intuitively, each individual distribution should be homogenous in $f$ while the two distributions should be distinct in $f$. Furthermore, if the above procedure identifies $e$ , $1 < e \leq n$, as the changeover point, by calculating the mean and variance in Equation 3.1 and 3.2 respectively, of the sampling distribution for $f(X_i)$ in the range of $1 - m \leq i < e$ (denoted as $\overline{f}_1(e)$ and $S_{f_1}^2(e)$) and $e \leq i \leq n + m$ (denoted as $\overline{f}_2(e)$ and $S_{f_2}^2(e)$). The null hypothesis is that no changeover has occurred within the window $(\overline{f}_1(e) = \overline{f}_2(e))$.

**Initialization**

$k_{best} = k_{curr} = 2;$

$n1_{best} = n1_{curr} = m + 1; \qquad n2_{best} = n2_{curr} = n + m - 1;$

$\bar{f}1_{best} = \bar{f}1_{curr} = \frac{\sum_{i=1-m}^{1} f(X_i)}{n1_{best}}; \qquad \bar{f}2_{best} = \bar{f}2_{curr} = \frac{\sum_{i=2}^{n+m} f(X_i)}{n2_{best}};$

$s_{b_{best}} = s_{b_{curr}} = \frac{n1_{best} n2_{best}}{(n1_{best} + n2_{best})} (\bar{f}1_{best} - \bar{f}2_{best})^2;$

**Hypothesis generation**

for $l = 3, \cdots, n$ do {

$k_{curr}{+}{+}; \qquad n1_{curr}{+}{+}; \qquad n2_{curr}{-}{-};$

$\bar{f}1_{curr} \leftarrow \frac{m+l-1}{m+l} \bar{f}1_{curr} + \frac{f(X_l)}{m+l}; \qquad \bar{f}2_{curr} \leftarrow \frac{n+m-l+1}{n+m-l} \bar{f}2_{curr} - \frac{f(X_l)}{n+m-l};$

$s_{b_{curr}} \leftarrow \frac{n1_{curr} n2_{curr}}{(n1_{curr} + n2_{curr})} (\bar{f}1_{curr} - \bar{f}2_{curr})^2;$

if $(s_{b_{curr}} > s_{b_{best}})$ {

$k_{best} = k_{curr};$

$n1_{best} = n1_{curr}; \qquad n2_{best} = n2_{curr};$

$\bar{f}1_{best} = \bar{f}1_{curr}; \qquad \bar{f}2_{best} = \bar{f}2_{curr};$

$s_{b_{best}} = s_{b_{curr}};$

}

}

**Hypothesis validation**

$s^2_{\bar{f}1_{best}} = \frac{\sum_{i=1-m}^{k_{best}-1} (f(X_i) - \bar{f}1_{best})^2}{n1_{best} - 1}; \qquad s^2_{\bar{f}2_{best}} = \frac{\sum_{i=k_{best}}^{n+m} (f(X_i) - \bar{f}2_{best})^2}{n2_{best} - 1};$

$t = \frac{\bar{f}1_{best} - \bar{f}2_{best}}{s};$

$s = \sqrt{\frac{2MSE}{n_h}}; \qquad MSE = \frac{n1_{best} s^2_{\bar{f}1_{best}} + n2_{best} s^2_{\bar{f}2_{best}}}{n + 2m - 2}; \qquad n_h = \frac{2}{\frac{1}{n1_{best}} + \frac{1}{n2_{best}}};$

if $t$ is larger than the corresponding value from the $t$-table

Accept $k_{best}$ as a real changeover index

else

Repeat the procedure for the next window location

**Figure 3.8:** The pseudo-code for online univariate algorithm (Jain and Wang, 2015)

The hypothesis validation step is used to identify whether change occurs or not at specific index $e$. Hence, a $t$ statistic is calculated using Equation 3.3 to reject the null hypothesis or accept that a change point occurs in the data.

$$t = \frac{\overline{f}_1(e) - \overline{f}_2(e)}{s} \tag{3.3}$$

where $s$ is the estimated standard error of the difference between the means, which is 2.

$$s = \sqrt{\frac{2MSE}{n_h}} \quad MSE = \frac{n_1 S^2_{f_1}(e) + n_2 S^2_{f_2}(e)}{n + 2m - 2} \quad , n_h = \frac{2}{\frac{1}{n_1} + \frac{1}{n_2}} \tag{3.4}$$

MSE is the mean-squared-error, $n_h$ is the harmonic mean of the two sample sizes, and $n + 2m2$ is the degrees of freedom (DOF) (Welkowitz et al., 2011) (Triola, 2006). The hypothesis generation and verification steps are explained for an online univariate algorithm is shown in Figure 3.8.

### 3.3.2 Exponentially Weighted Moving Average (EWMA) Control Chart

The Exponentially Weighted Moving Average (EWMA) was introduced by (Lowry et al., 1992) for detecting shifts in the process. The EWMA is a statistic for monitoring the process that averages the input data in such a way that give less weight to the historical data and more weight to current data.

As the EWMA considered present and past information of the input data and therefor more efficient and fast in detecting small shift in the data (Montgomery, 2009)

In EWMA, the decision depends on the EWMA statistic, which is an exponentially weighted average of all prior data, including the most recent measurement (Lucas and Saccucci, 1990). The Univariate EWMA statistics can be defined using Equation 3.4.

$$\mathbf{Z}_i = \lambda\mathbf{X}_i + (1-\lambda)\mathbf{Z}_{i-1} \qquad i = 1, 2, 3, ..., n \tag{3.5}$$

where $\mathbf{Z}_i$ is the $i^{th}$ EWMA vector, $\lambda$ is the value between $0 < \lambda \leq 1$ that weights the current and historical data and $\mathbf{X}_i$ is the $i^{th}$ input observation vector, $i = 1, 2, 3, ..., n$.

However, the extension of EWMA is Multivariate EWMA (MEWMA) which is used to simultaneously monitor two or more related process of the input observation. The following section give detail explanation of MEWMA.

#### 3.3.2.1 Multivariate Exponentially Weighted Moving Average (MEWMA) change point detection algorithm

The Multivariate Exponentially Weighted Moving Average (MEWMA) is a statistical control method to monitor simultaneously two or more correlated variables and also provide sensitive detection of small and moderate shifts in time series data. The MEWMA statistic incorporates information of all prior data including historical and current observation with a user-defined weighted factor (Khoo, 2004)(Pan and Jarrett, 2014). Moreover, MEWMA can be used to detect shift of any size in the process. MEWMA has achieved better performance to detect small and moderate changes than other multivariate control charts like the T-Square and Shewhart control chart (Bersimis et al., 2007). MEWMA can be defined using the following Equation 3.5.

$$\mathbf{Z}_i = \wedge\mathbf{X}_i + (1-\wedge)\mathbf{Z}_{i-1} \qquad i = 1, 2, 3, ..., n \tag{3.6}$$

where $\mathbf{Z}_i$ is the $i^{th}$ MEWMA vector, $\wedge$ is the diagonal matrix with elements $\lambda_i$ which weight the current and historical data for $i = 1, ..., p$ where $p$ is the number of dimensions and $0 < \lambda_i \leq 1$, and $\mathbf{X}_i$ is the $i^{th}$ input observation vector,

$i = 1, 2, 3, ..., n$. The out-of-control statistics is defined in Equation 3.6.

$$\mathbf{T}_i^2 = \mathbf{Z}_i' \mathbf{\Sigma}_i^{-1} \mathbf{Z}_i < h \tag{3.7}$$

where $\mathbf{\Sigma}$ is the variance covariance matrix of $\mathbf{Z}_i$ and $h(> 0)$, chosen to achieve a specified in-control value. Multivariate analysis is used to measure more than one characteristic of a system and also evaluate the relationship among these characteristics. In general the $\lambda$ value lies between 0 and 1,but, the standard value used in literature for MEWMA algorithm is $\lambda = 0.3$ (Lucas and Saccucci, 1990).

In multivariate analysis, the data points $\mathbf{X}_1, ..., \mathbf{X}_n$ is a subsequence of a data stream where $n$ is the length of a data stream. Each data point $\mathbf{X}_i$ is a vector of $n$ sensor observations. The data points in the data stream may be from various distributions, for example, $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, ..., \mathbf{X}_{i-1}$ and $\mathbf{X}_i, \mathbf{X}_{i+1}, ..., \mathbf{X}_n$ can be from distributions $D_1$ and $D_2$ respectively. The aim of the algorithm is to determine and classify the position of change points $i$ in the data stream. In the first step, the MEWMA algorithm calculates the exponentially weighted moving average of the multivariate input observations to find the change points. Hence, the sliding window version of the algorithm is used to analyze the data points sequentially and incremented by 1 each time to process all data within a window. In each window, the index variable, $i$ slides subsequently to determine the global statistics for each index $i$, $1 < i \leq n$. Each input vector of multivariate data is used to find the MEWMA vector represented by $\mathbf{Z}_i$. In addition, the variance and covariance matrix of $\mathbf{Z}_i$ is calculated recursively and represented by $\mathbf{\Sigma}_i$, Once the T-squared statistic is calculated as shown in Equation (3.7), the significance value $h$ is used to identify the confidence of the entire window. If the T-squared value is greater than $h$, then $x_i$ will be labelled as a change point within a data stream. The accelerometer data analysis identifies the actual values of specific change points which may represent an increase or decrease in the data. Therefore, the sliding window algorithm detects a number of adjacent change points which highlights the significant change point in the data. Adjacent change points can be classified by defining a new parameter $k$, which indicates sequential adjacent change points and is considered as an indicative point of the same event and therefore would be merged.

### 3.3.3 Sliding Window for change detection

The $x$, $y$ and $z$ axis of an accelerometer signal is divided into multiple windows of different sizes $(1s, 1.5s, 2s, 2.5s, 3s.3.5s, 4s)$. The sliding window algorithm as discussed in section 2.4.1 is applied on each window. In the next step, each window is transformed into a feature vector. The feature vector consists of time domain features which were calculated from each window data. The extracted features were obtained by processing the $x$, $y$ and $z$ axis of an accelerometer data. The time domain statistical features such as mean, variance and covariance were calculated to form a feature vector. The multiple window division of an accelerometer signal and feature extraction were shown in Figure 3.9.

Once the T-statistics for each algorithm is calculated discussed earlier in section 3.3.1 and 3.3.2 then the number of possible values for the signicance values (0.005, 0.01,0.025,0.05) are used to identify the confidence of each window. Hence, if the T-statistic value is greater than the significance value then the $x_i$ will be labelled as change point within the window. The process is continued till the end of time series data. The total number of windows of different sizes (1s, 1.5s, 2s, 2.5s, 3s. 3.5s, 4s) are shown in Figure 3.10. The number of window is inversely proportional to window sizes because as the window size increase the total number of window decreases as shown in Figure 3.10.

## 3.4 Real Dataset and Experimental Setup

The MEWMA approach was evaluated for change point detection on a real dataset of accelerometer data. The dataset used here were collected by (Zhang et al., 2011) which consists of two participants wearing the shimmer wireless

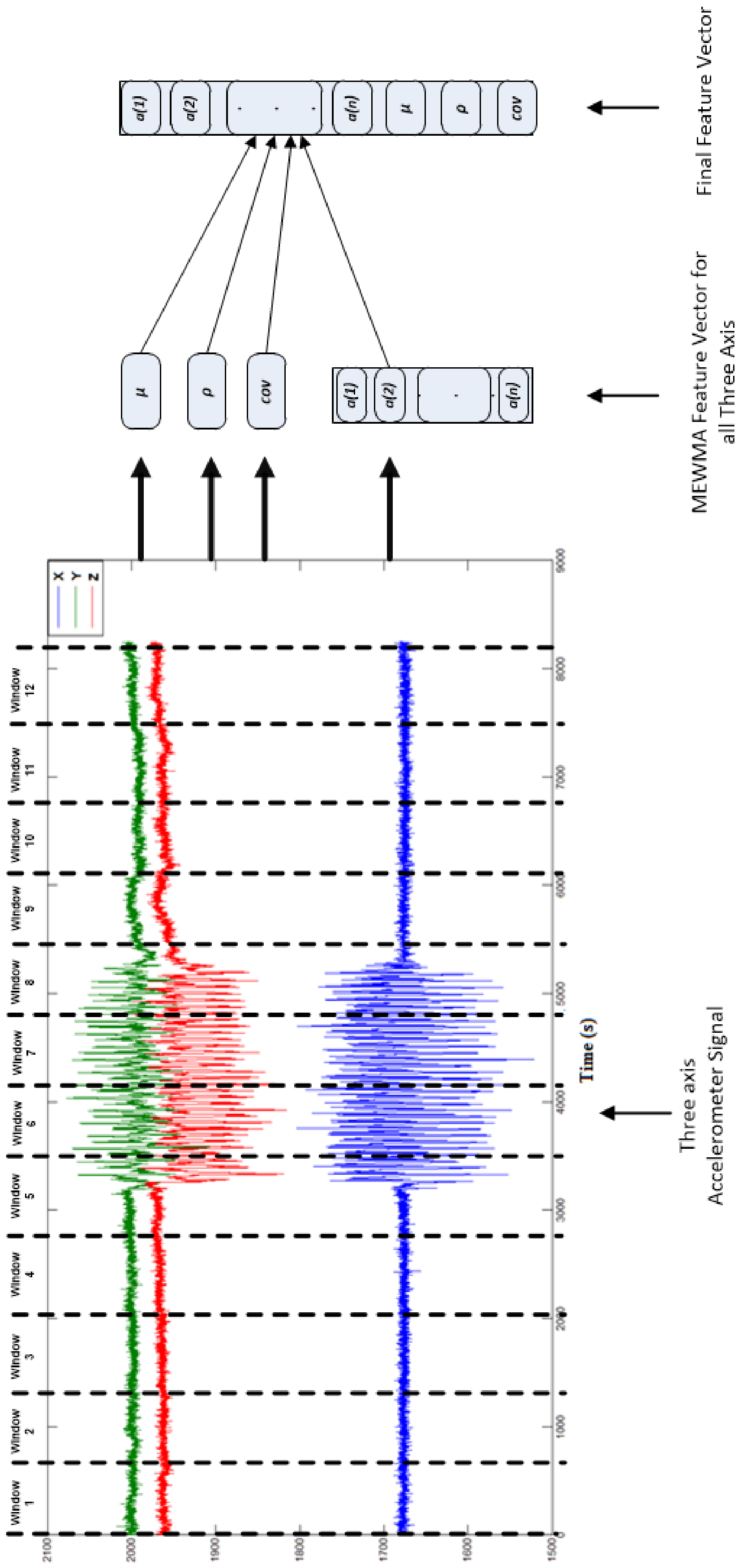



**Figure 3.9:** Multiple window division and Feature vector generation

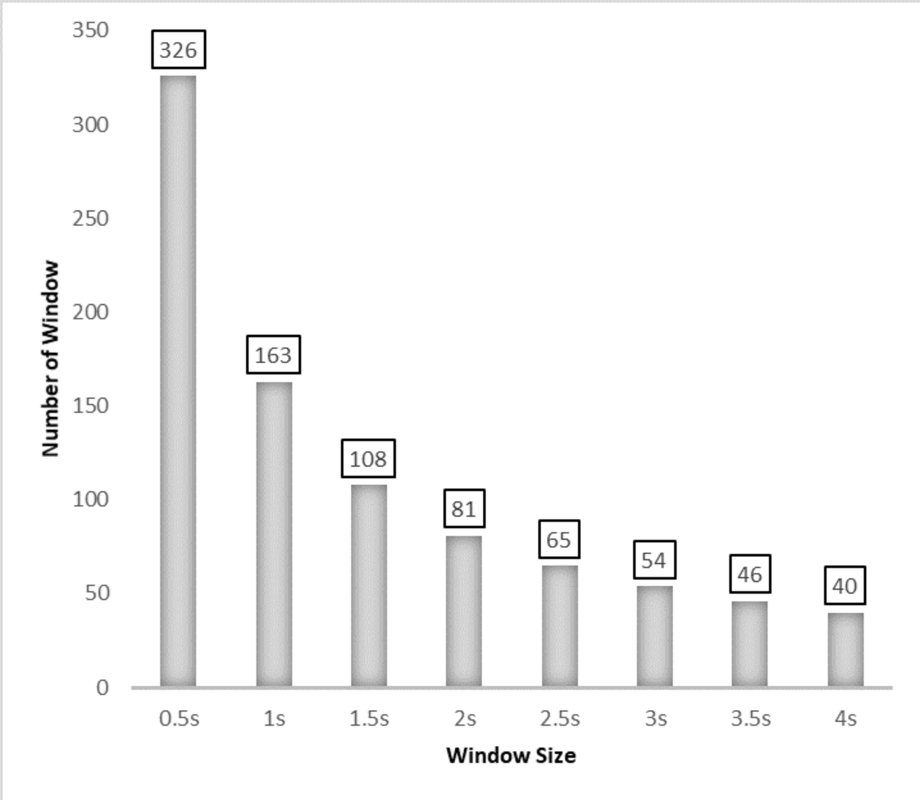



**Figure 3.10:** Window Sizes Vs Total Number of Window

sensing platform (Kuris and Dishongh, 2006).The shimmer sensor was placed in the middle of the participant's left pectoral and at the mid-point between the thigh and knee on the anterior of the participant's right leg. The sensor placement positions on the subject's body enabled anterior-posterior and lateral movements to be captured effectively (Zhang et al., 2011). The five different activities were performed by two participants involving a set of activities consisting of Sit to Stand, Stand to Sit, Stand-Walk Corridor-Stand, Stand-Walk Upstairs-Stand and Stand-Walk Downstairs-Stand. For the static activities such as sit and stand, the participants remained in each state for approximately 30 seconds and then transitioned to other static activities such as sit to stand and stand to sit (Zhang et al., 2011). The dynamic activities were measured by accelerometer data where the subject stood for approximately 30 seconds, followed by an activity performed for 30 seconds and then transition to standing. The data was captured from each participant with sample frequency of 52 *Hz*. The example of Stand to Sit activity and Stand-Walk Downstairs-Stand activity from the dataset can be shown in Figure 3.11 and 3.12. In the data collection, the activity execution of accelerometer data was wirelessly streamed to a receiving computer via the IEEE 802.15.1 Bluetooth communications protocol.

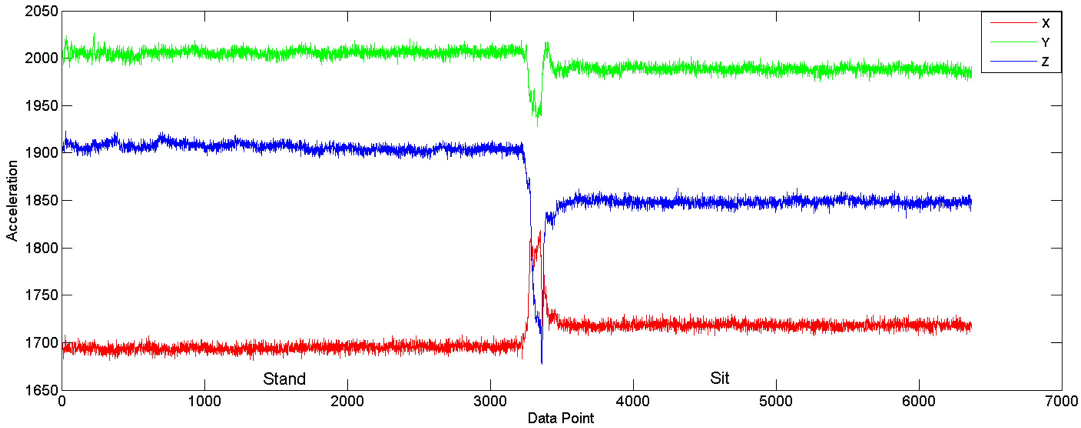



**Figure 3.11:** Stand to Sit

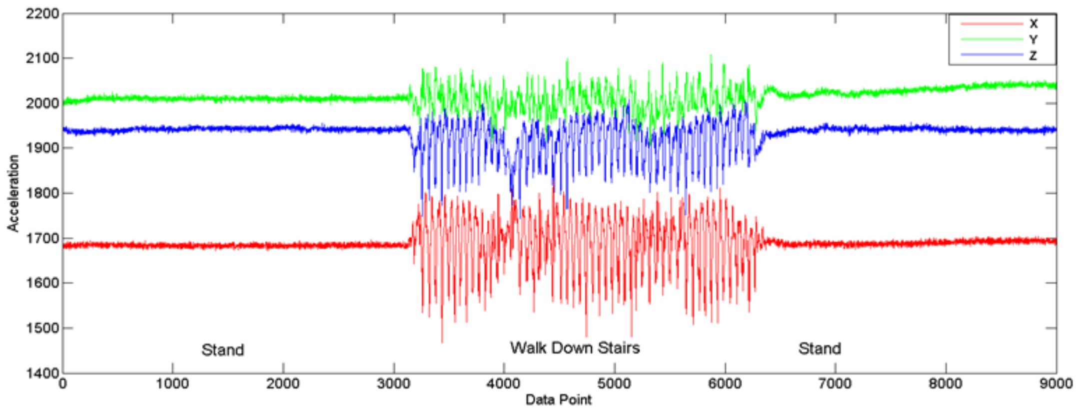
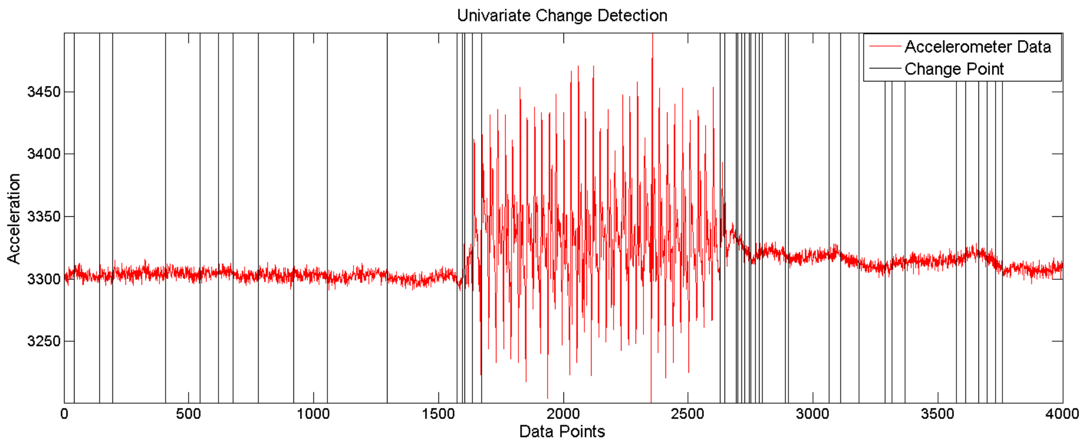



**Figure 3.12:** Stand - Walk Downstairs - Stand

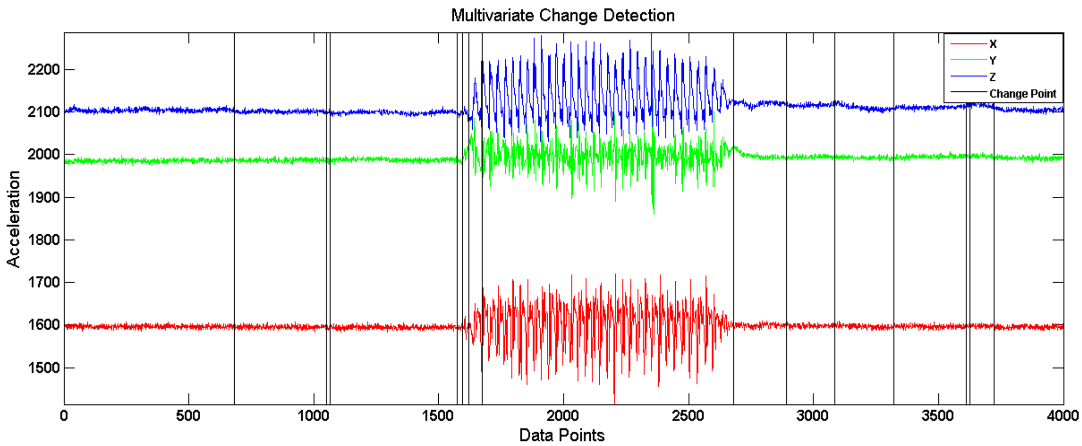
## 3.5 Results and Discussion

In this section,we have performed experiments on a real dataset for univariate and multivariate data and also provide detailed explanation of results comparing the classification performance of the standard MEWMA approach and univariate approach proposed in (Jain and Wang, 2015) for change point detection. For evaluation of univariate data, the magnitude of acceleration is calculated from the $x$, $y$ and $z$ axes of the accelerometer data and used as an input to the univariate algorithm. Likewise, for multivariate data analysis, the $x$, $y$ and $z$ axes of the captured data is used as input to the standard MEWMA approach for change point detection in the data stream. The standard MEWMA ($\lambda = 0.3$) uses the $\lambda$ parameter which weights the current and historical data. However, in our experiments for both algorithms the different window sizes $1s, 1.5s, 2s, 2.5s, 3s.3.5s, 4s)$ and signicance values $(0.005, 0.01, 0.025, 0.05)$ are varied in an effort to nd better performance and accurate change point.The reported window sizes in Table 3.2 and Table 3.3 achieved the high metric measures such as accuracy, precision, G-means and F-measure. All the transitions were occurred in the dataset which we have used for our experiments.The start and end points of the user data is also manually labelled for the purpose of quantitative evaluation of both algorithms. Moreover, for results evaluation, we used different metrics such as accuracy, precision, G- means and F-measure as discussed and explained in section 3.6. In our experiments, these evaluation metrics are used to evaluate change point detection in activity monitoring for the univariate and standard MEWMA approach. The analysis of our experiments for two activities i.e. stand still- walk corridor-stand still and stand still- walk downstairs- stand still for change detection using univariate and standard MEWMA approach are shown in Figure 3.13(a & b) and 3.14(a & b) respectively.

In our experiments, when determining ture positives a quarter second buffer was included at either side of the manually labelled change point to accommodate subjectivity errors inherent in manual labelling. Thus, a detected change point was considered true if its index in the data stream $i$, $i \epsilon z - f/4, ..., z + f/4$ where $z$ is the index in the data stream of the manually labelled change point and $f$ is the sampling frequency in $Hz$. The target of our algorithm is to detect the primary transitions in high level activities such as stand still – downstairs –stand still and stand still– walk corridor- stand still, as presented in Figure 3.13 and 3.14 respectively. The positive and negative detection cases were defined as, the true positive (TP) which is the correctly identified change point and true negative (TN) which are the non-transitional points which are not labelled as




**(a)** Univariate Change Detection for Activity 'Stand still-Walk Corridor-Stand Still' Proposed by Jian and Wang (Jain and Wang, 2015)



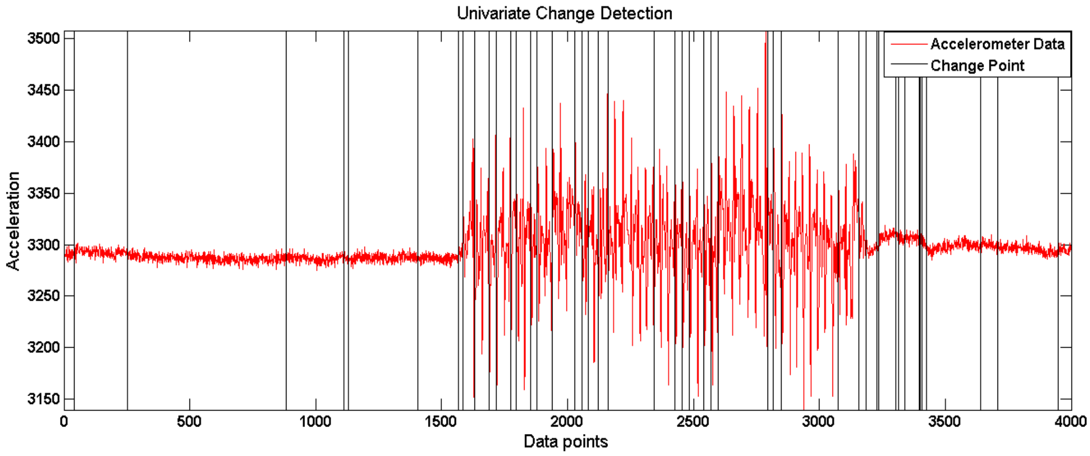

**(b)** Presented Multivariate EWMA change detection with standard value $\lambda = 0.3$ for Activity 'Stand still- Walk Corridor-Stand Still'

**Figure 3.13:** (a and b): Example of Univariate and Multivariate EWMA change detection using sliding window for activity 'Stand still- Walk Corridor-Stand Still'. The window size was 2s and significance value 0.005.

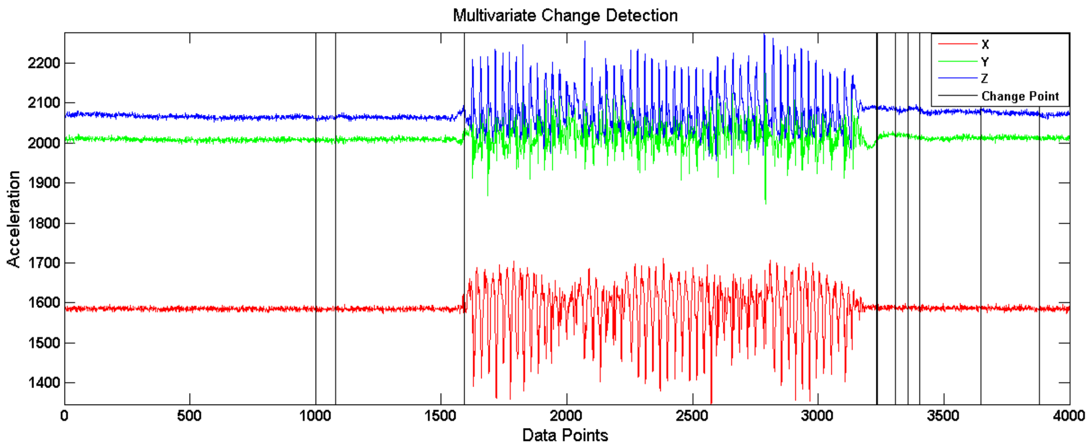
change. The false positive (FP) is the non-transitional point which the algorithm highlights as a change and false negative (FN) occurs when the algorithm is unable to detect changes in the user's activity. The univariate approach identified a maximum number of false positives in the real dataset because of the complex and dynamic nature of the accelerometry data. In Figure 3.13 and 3.14, the univariate and multivariate approach is applied to detect the primary change points in the real dataset. The primary change points are the transitions between high level activities, for example 'stand still'- walk up stairs'- stand still. However, the non-primary change points are detected hugely using univariate approach as presented in Figure 3.13 (a) and 3.14 (a) than the standard MEWMA approach presented in Figure 3.13 (b) and 3.14 (b). The non-primary change points are considered as false positives which can greatly effect to minimize the accuracy, precision, G-Means and F measure. The results of both approaches when applied to a real dataset (discussed earlier in section 3.4) for five different activities {Sit to Stand, Stand to Sit, Stand-Walk Corridor- Stand, Stand-Walk Downstairs-Stand and Stand-Walk Upstairs-Stand } are presented in Table 3.2 and 3.3. In our experiments, we used online Univariate algorithm and standard MEWMA algorithm for change detection in the data. Table 3.2 and Table 3.3, presents results for each activity using the univariate approach presented in (Jain and Wang, 2015) and standard MEWMA approach ($\lambda = 0.3$) for change point detection. For both approaches, the experimental results for each activity with corresponding significance values 0.005, 0.01,0.025,0.05 and window sizes 1s, 1.5s, 2s, 2.5s, 3s. 3.5s, 4s are presented.

**Table 3.2:** Univariate change detection for 5 different activities

| Activity | Significance value | Win Size | Accuracy % | Precision % | Sensitivity % | G-means % | F-Measure % |
|---|---|---|---|---|---|---|---|
| Sit to Stand | 0.005 | 2s | 98.00 | 28 | 32.30 | 50 | 30 |
| Stand to Sit | | 3s | 98.20 | 30 | 42 | 50 | 35 |
| Stand to Walk Corridor | | 2.5s | 97.50 | 25 | 25 | 40 | 25 |
| Stand to Walk Down Stairs | | 2s | 98.80 | 35 | 26.25 | 50 | 30 |
| Stand to Walk up Stairs | | 2s | 97.00 | 25 | 25 | 40 | 25 |




**(a)** Univariate Change Detection for Activity 'Stand still-Walk Down Stairs-Stand Still' Proposed by Jian and Wang (Jain and Wang, 2015)




**(b)** Presented Multivariate EWMA change detection with standard value $\lambda = 0.3$ for Activity 'Stand still- Walk Down Stairs-Stand Still'

**Figure 3.14:** (a and b): Example of Univariate and Multivariate EWMA change detection using sliding window for activity 'Stand still- Walk Down Stairs-Stand Still'. The window size was 2s and significance value 0.005.

**Table 3.3:** Multivariate EWMA Standard Values for 5 different activities

| Activity | Significance value | Win Size | $\lambda$ | Accuracy % | Precision % | Sensitivity % | G-means % | F-Measure % |
|---|---|---|---|---|---|---|---|---|
| Sit to Stand | 0.005 | 2s | 0.3 | 99.50 | 40 | 40 | 70 | 40 |
| Stand to Sit | | 3s | | 99.60 | 40 | 66.66 | 70 | 50 |
| Stand to Walk Corridor | | 2.5s | | 99.20 | 40 | 40 | 70 | 40 |
| Stand to Walk Down Stairs | | 2s | | 98.50 | 50 | 39.92 | 70 | 44.4 |
| Stand to Walk up Stairs | | 2s | | 97.50 | 30 | 60 | 70 | 40 |

# 3.6 Evaluation

Different performance measures such as accuracy, precision, G- means and F-measure were used for evaluation of the experimental results as follows.

## 3.6.1 Accuracy

The accuracy metric is most commonly used to evaluate the performance of an algorithm in terms of how close a measured value is to the actual (true) value. The accuracy can be measure as the ratio of correctly classified data points to the total number of data points. The accuracy can be calculated using Equation 3.7.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{3.8}$$

High accuracy is achieved for all activities using the univariate approach ranging from 97% to 98.80%. Similarly, for Standard MEWMA, high accuracy was achieved ranging from 97.50% to 99.60%. The reason for getting high accuracy for both approaches is because of disproportionality of high numbers of true negatives in the data. However, standard MEWMA approach obtained higher accuracy for all activities in the dataset than the Univariate approach for the same significance value and window sizes as shown in Table 3.2 and 3.3 respectively.

Thus, for evaluation of both algorithms, we used other metric measures such as precision, G-means and F-measure.

### 3.6.2 Precision

The precision metric is used to find how close the measured values are to each other. Precision can be measured as the ratio of true positive data points to the total classified positive data points. The precision is calculated to avoid unintuitive requests (not a change but algorithm detects it as change) because unintuitive requests may degrade user experience. The precision for univariate and standard MEWMA is calculated using Equation 3.8.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.9}$$

In Table 3.2 and Table 3.3, the sliding window version of the algorithm with the increment of one data point is presented for which the precision is calculated for the detected change points. The proposed standard MEWMA algorithm consistently attained higher precision for all activities in the dataset than the univariate approach as shown in Table 3.3. The maximum precision is achieved for the Univariate approach in the range of 30% to 50% with 0.005 significance value and window sizes 2s to 3s for different activities. However, for standard MEWMA, higher precision is achieved than univariate ranging from 30% to 50% for the same significance value and window sizes. The standard MEWMA improved precision 12% on average compare with the univariate approach for different activities with same significance value and window sizes as shown in Table 3.2 and Table 3.3 respectively.

The class imbalance problem in the dataset is caused by the high sampling frequency in relation to the number of TPs. This highlights the skewed distribution of classes within the dataset and identifies that the minority class is the class of interest (Galar et al., 2012). In our dataset, we have only one true positive (TP) (which represents a correctly identified change point) and a high number of true negatives (TN) (the non-transitional points which are not labelled as change). We used the precision which is the ratio of true positive data points to the total classified positive data points as presented in Equation. 3.8. Therefore, the one or two FP detections reduced the precision to 50% and 30% due to the imbalanced class problem in our real dataset.

### 3.6.3 Sensitivity and Specificity

The sensitivity (also known as Recall) is referred to as the true positive rate (TP) and identifies the proportion of true positive class of interest (the positive class) that is recognized correctly. The sensitivity is calculated as the true positive divided by the total sum of true positive and false negative. Likewise, the specificity is the inverse of sensitivity and referred to as the true negative rate (TN). The specificity is used to identify the class of interest (the negative class) and calculated as the true negative divided by the total sum of true negative and false positive. The sensitivity and specificity is calculated using Equation 3.9 and 3.10.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{3.10}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{3.11}$$

### 3.6.4 G-means

The G-means performance metric is the combination of sensitivity and specificity and defined as the ratio of positive accuracy (sensitivity) and the ratio of negative accuracy (specificity). Thus, the G-means provides activity class-sensitive measure of the detection performance and can be calculated using Equation 3.11.

$$\text{G-means} = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} = \sqrt{sensitivity \times specificity} \tag{3.12}$$

The maximum G-means achieved for the univariate approach ranges from 40% to 50% for different activities with significance value 0.005 and window sizes 2s to 3s. But, the standard MEWMA achieved highest G-means about 70% for all activities using same significance and window sizes as shown in Table 3.2 and Table 3.3. Hence, the standard MEWMA improved on average 24% more than the univariate approach for the same significance value and window sizes.

### 3.6.5 F-measure

The F-measure is used to combine precision and recall and used as measure to find the overall effectiveness of each activity (Cook and Krishnan, 2015). The F-measure can be calculated using the Equation 3.12.

$$\text{F-measure} = \frac{2 \times \ \text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})} \tag{3.13}$$

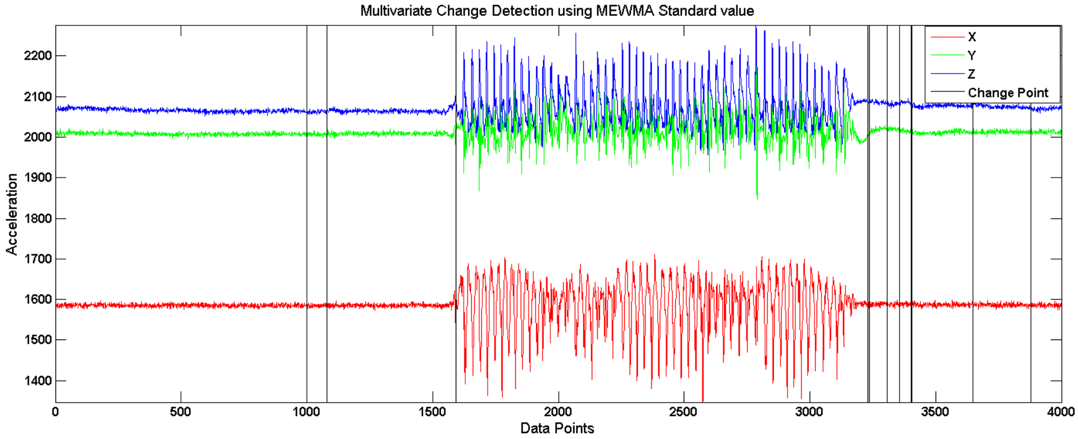
As the F-measure balances the recall and precision.For Standard MEMWA approach, the best F-measure is obtained between 40% to 50% for different activities in dataset while 25% to 35% is achieved for univariate approach. Hence, the standard MEWMA improved F-measure 13% on average more than the univariate approach. The analysis of overall results shows that the proposed standard MEWMA provides better accuracy and also improved on the performance as measured by the other metric measures such as precision, G-means and F-measure by more than 12%, 24% and 13% respectively compare with Univariate approach.

## 3.7 Parameter Exploration for online change detection in activity monitoring

In this section, we analyze and explore our experiments, using the standard MEWMA approach for change point detection discussed in section 3.3.2.1, evaluating the different parameters that can be tuned and explored such as $\lambda$ the parameter which weights current and historical data, window size and significance values with the aim to achieve better performance and accurate change point detection. This section initially summarizes the automatic detection of change points in user activities and then explores the optimal parameter set by evaluating different metric measures discussed earlier in section 3.6 to classify the change point detection in activity monitoring for the MEWMA approach when considering the tuning of its parameters. The tuned parameters help to identify optimal values for each activity by achieving better results for different metrics.

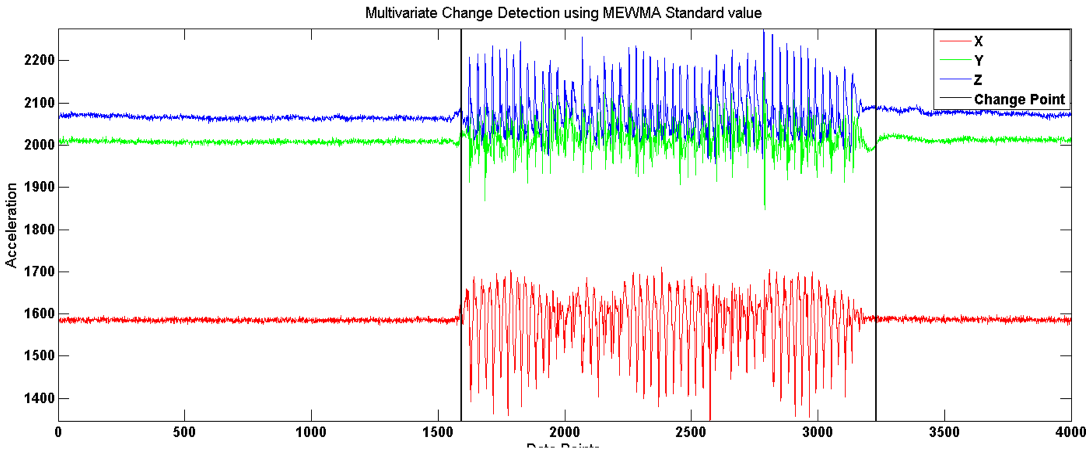
The same real dataset and experimental setup as discussed in section 3.4 is used for our experiments. This section presents the details of the evaluation of the feasibility and performance of the proposed MEWMA on a real dataset for the standard and tuned parameters of MEWMA. For evaluation, the standard MEWMA for which results are presented in Table 3.3 is used as a benchmark. However, we search for optimal parameters by using different values of $\lambda$ ranging from 0.1 to 1. $\lambda$ is the parameter which weights current and historical data and window size (1s, 1.5s, 2s, 2.5s, 3s, 3.5s, 4s).In addition significance values (0.005, 0.01, 0.025, 0.05) are varied in an effort to find better performance and accurate change point detection. The metrics of precision, G-means and F-measure discussed in section 3.6 were used to evaluate change point detection in activity monitoring for the tuned parameters of the MEWMA approach.
The tuned parameters help to identify optimal values for each activity by achieving better results for different metrics. In our experiments, we used the standard MEWMA and tuned (optimal) values for change point detection as presented in Figure 3.15 and Figure 3.16, respectively. The target of our algorithm is to detect the primary transitions in high level activities such as stand still – downstairs –stand still as presented in Figure 3. 16.

The standard MEWMA identified a large number of false positives in the real data because of the dynamic and complex nature of the accelerometery data. Figure 3.15 and Figure 3.16 used the standard and optimal MEWMA respectively to detect the primary change points in real data. Nevertheless, the non-primary




Standard MEWMA

**Figure 3.15:** Example of sliding window change detection results for the activity 'stand still – walk downstairs - stand still'. The window size was 2s second with significance value 0.005 and $\lambda = 0.3$.




Optimal MEWMA

**Figure 3.16:** Example of sliding window change detection results for the activity 'stand still – walk downstairs - stand still' on real data. The window size was 1.5s second with significance value 0.005 and $\lambda = 0.7$.

change points were detected by the standard MEWMA approach as presented in Table 3.3 whereas in Figure 3.16 only the primary change points were detected using MEWMA tuned with optimal values. A number of false positives were detected in the real data using the standard MEWMA approach as presented in Figure. 3.15 which can minimize the accuracy, precision and F measure presented in Table 3.3.

## 3.8 Experimental setup and Results for change detection of each user activity

The experiments were performed on real data set for five different activities such as Sit to Stand, Stand -Walk Corridor-Stand, Stand-Walk Downstairs-Stand and Stand-Walk Up Stairs-Stand. The different parameters of MEWMA such as $\lambda$ the parameter which weights current and historical data, window size and signicance values were tuned with the aim to achieve better performance and accurate change point detection. The following experimental results represents the different values of $\lambda$ ranging from 0.1 to 1 for each activity with corresponding significance values 0.005, 0.01, 0.025, 0.05 and window sizes 1s, 1.5s, 2s, 2.5s, 3s, 3.5s and 4s. Also, the different metrics such as accuracy, precision, G-means and F-measure discussed in section 3.6 were used to evaluate the change point detection for tuned parameters which help to identify optimal values for each activity by achieving better results for different metrics. The following section will discuss the experimental results of MEWMA for optimal parameter selection when considering the tuning of its parameter for each activity.

### 3.8.1 Sit to Stand

The Sit to Stand activity results for accuracy were relatively high about 99.9% for optimal parameters and 99.5% for Standard MEWMA as presented in Table 3.3 and Table 3.4 respectively. The accuracy is high because of considering each class equally important in the dataset even if exist the class imbalance problem in the dataset. The class imbalance problem happens when the total number of a class data (positive) is less than then total number of other classes of data (negative). This highlights the skewed distribution of classes within the dataset and identifies that the minority class is the class of interest (Galar et al., 2012). The accurate change point detection for sit to stand activity using optimal parameter selection can be shown in Figure 3.17.

The maximum precision, G-Means and F-measure for standard MEWMA were achieved is about 40%, 70% and 40% respectively with 0.005 significance value and window sizes 2s as presented in Table 3.3. However, for optimal parameters, the approach attained maximum values are about 66.7% ,100% and 80% with $\lambda$ value 0.5, significance value 0.01 and window size 0.5s as presented in Table 3.4. Also, the higher precision, G-means and F-measure were attained due to the minimum number of false positive detections using optimal parameters than the Standard MEWMA.

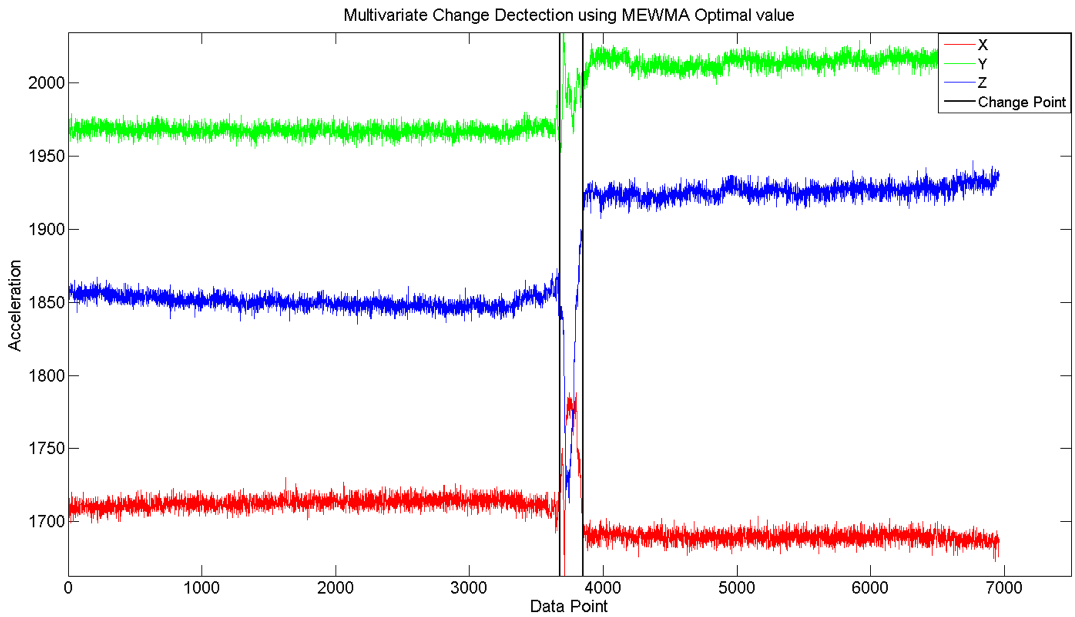



**Figure 3.17:** Sit to Stand

**Table 3.4:** MEWMA optimal parameter for Sit to Stand activity

| Activity | Significance value | Win Size | $\lambda$ | Accuracy % | Precision % | G-means % | F-Measure % |
|---|---|---|---|---|---|---|---|
| Sit to Stand | 0.005 | 2s | 0.1 | 97.50 | 20 | 75 | 30 |
| | 0.005 | 0.5s | 0.2 | 99.60 | 35 | 70 | 35 |
| | 0.005 | 0.5s | 0.4 | 99.50 | 30 | 70 | 40 |
| | 0.005 | 0.5s | 0.5 | 99.30 | 50 | 70 | 50 |
| | 0.005 | 0.5s | 0.6 | 99.20 | 50 | 90 | 66.70 |
| | 0.01 | 0.5s | 0.7 | 99.90 | 66.70 | 100 | 80 |
| | 0.005 | 2.5s | 0.8 | 99.80 | 50 | 85 | 57 |
| | 0.01 | 0.5s | 0.9 | 99.70 | 50 | 90 | 65 |

### 3.8.2 Stand to Sit

The Stand to Sit activity results for accuracy were about 99.9% for optimal parameters and 99.60% for Standard MEWMA which is relatively higher than the Standard MEWMA as presented in Table 3.3 and Table 3.5 respectively. For Stand to Sit activity, the maximum precision, G-Means and F-measure for standard MEWMA were achieved is about 40%, 70% and 50% respectively with 0.005 significance value and window sizes 3s as presented in Table 3.3. However, for optimal parameters, the approach attained maximum values are about 50% ,100% and 66.7% with $\lambda$ value 0.5, significance value 0.005 and window size 2.5s as presented in Table 3.5. The higher precision, G-means and F-measure were achieved is about 10%,30% and 16.70% respectively than the standard MEWMA. The accurate change point detection for sit to stand activity using optimal parameter selection can be shown in Figure 3.18.

**Table 3.5:** MEWMA optimal parameter for Stand to Sit activity

| Activity | Significance value | Win Size | $\lambda$ | Accuracy % | Precision % | G-means % | F-Measure % |
|---|---|---|---|---|---|---|---|
| Stand to Sit | 0.005 | 2.5s | 0.1 | 99.30 | 30 | 80 | 44.40 |
| | 0.01 | 2.5s | 0.2 | 99.40 | 30 | 70 | 40 |
| | 0.005 | 1.5s | 0.4 | 99.60 | 50 | 75 | 40 |
| | 0.005 | 2.5s | 0.5 | 99.90 | 50 | 100 | 66.70 |
| | 0.01 | 2.5s | 0.6 | 99.40 | 40 | 85 | 57 |
| | 0.01 | 2.5s | 0.7 | 99.60 | 30 | 80 | 44.40 |
| | 0.01 | 0.5s | 0.8 | 99.50 | 40 | 75 | 40 |
| | 0.005 | 1s | 0.9 | 99.70 | 20 | 70 | 30 |

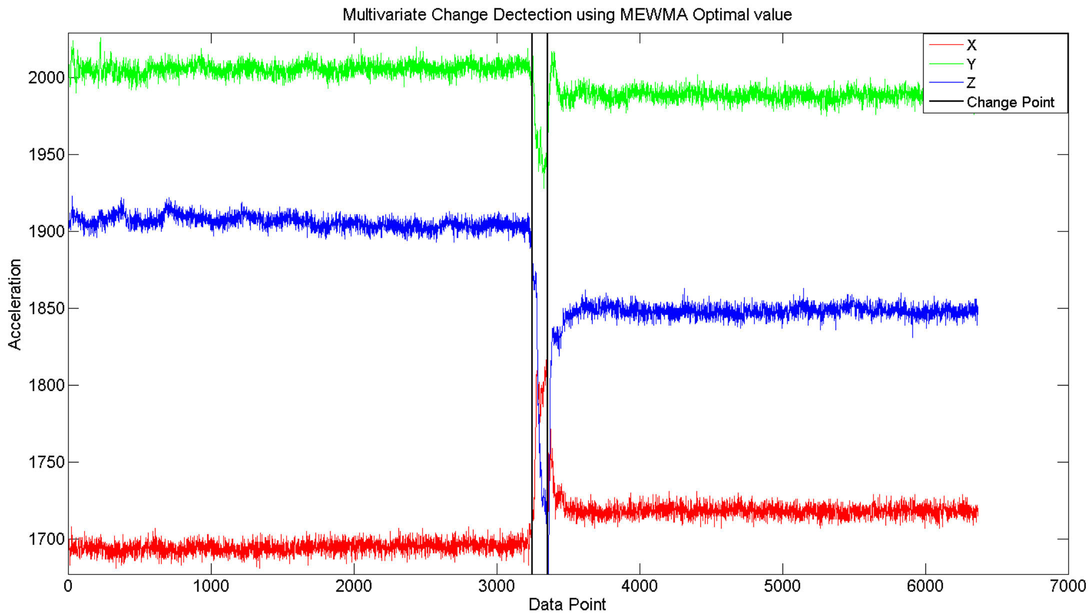



**Figure 3.18:** Stand to Sit

### 3.8.3 Stand - Walk Corridor - Stand

The accurate change point detection for Stand to Walk Corridor activity using optimal parameter selection can be shown in Figure 3.19. The stand to walk Corridor is relatively complex activity than the sit to stand and stand to sit activities. The complexity is because of consisting dynamic activity (walking) than the stationary activity (sit or stand). The Stand- Walk Corridor-Stand activity results for accuracy were about 99.9% for optimal parameters and 99.20% for Standard MEWMA which is relatively higher than the Standard MEWMA as presented in Table 3.3 and Table 3.6 respectively. The maximum precision, G-Means and F-measure for optimal parameters were achieved is about 50%, 70% and 50% respectively with $\lambda$ value 0.5, significance value 0.005 and window size 2s as presented in Table 3.6, whereas, for standard MEWMA approach the maximum values attained were about 40% ,70% and 40%, significance value 0.005 and window size 2.5s as presented in Table 3.3. The MEWMA with optimal parameter selection has achieved higher precision, G-means and F-measure is about 25%,30% and 25% respectively than the standard MEWMA.

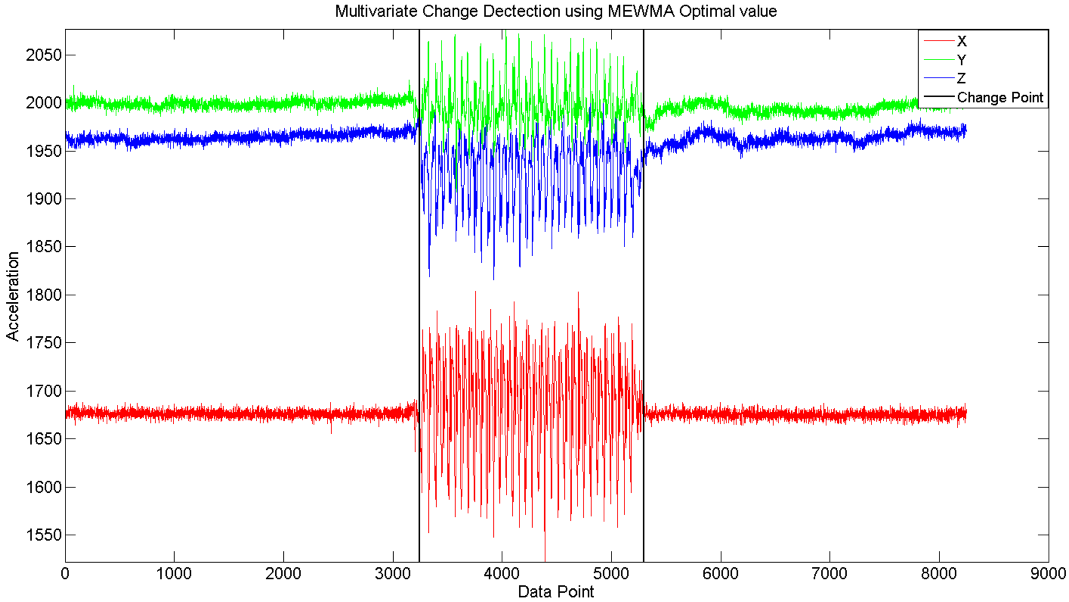



**Figure 3.19:** Stand WalkCorridor Stand

**Table 3.6:** MEWMA optimal parameter for Stand - WalkCorridor - Stand activity

| Activity | Significance value | Win Size | $\lambda$ | Accuracy % | Precision % | G-means % | F-Measure % |
|---|---|---|---|---|---|---|---|
| Stand - Walk Corridor - Stand | 0.01 | 3s | 0.1 | 99.20 | 10 | 65 | 20 |
| | 0.005 | 2.5s | 0.2 | 98.50 | 40 | 70 | 30 |
| | 0.01 | 2.5s | 0.4 | 99.10 | 30 | 66 | 40 |
| | 0.005 | 2s | 0.5 | 99.90 | 50 | 70 | 50 |
| | 0.01 | 1.5s | 0.6 | 98.80 | 40 | 68 | 40 |
| | 0.01 | 2s | 0.7 | 98.30 | 20 | 60 | 30 |
| | 0.01 | 2.5s | 0.8 | 99.10 | 30 | 68 | 40 |
| | 0.01 | 1s | 0.9 | 99.40 | 35 | 65 | 40 |

### 3.8.4 Stand - Walk Downstairs - Stand

Likewise, the walk corridor activity, the stand to walk downstairs activity is also complex because of non-stationary (walk downstairs) activity involved. The accurate change point detection for Stand-Walk Downstairs-Stand activity using optimal parameter selection can be shown in Figure 3.20. The Stand- Walk Downstairs-Stand activity results for accuracy were about 99.9% for optimal parameters and 98.50% for Standard MEWMA which is relatively higher than the Standard MEWMA as presented in Table 3.3 and Table 3.7 respectively. For Stand-Walk Downstairs-Stand activity, the maximum precision, G-Means and

F-measure for standard MEWMA were achieved is about 50%, 70% and 44.40% respectively with 0.005 significance value and window sizes 2s as presented in Table 3.3. However, for optimal parameters, the approach attained maximum values are about 50% ,90% and 66.7% with $\lambda$ value 0.7, significance value 0.01 and window size 1.5s as presented in Table 3.7.

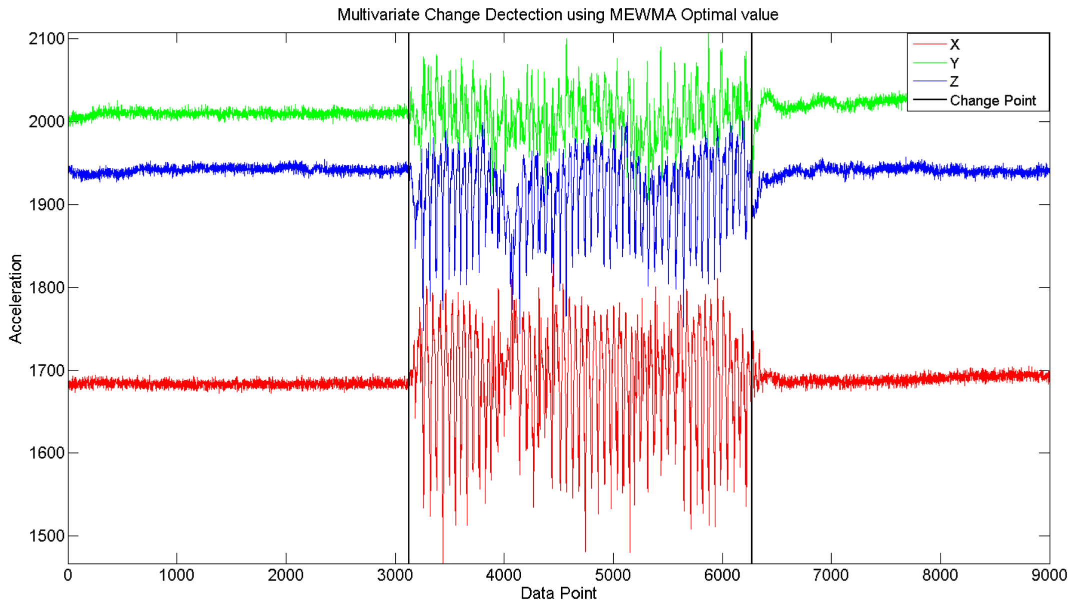



**Figure 3.20:** Stand - Walk Down Stairs - Stand

**Table 3.7:** MEWMA optimal parameter for Stand - Walk Down Stairs - Stand activity

| Activity | Significance value | Win Size | $\lambda$ | Accuracy % | Precision % | G-means % | F-Measure % |
|---|---|---|---|---|---|---|---|
| Stand- Walk Down Stairs- Stand | 0.005 | 2.5s | 0.1 | 97.70 | 30 | 65 | 40 |
| | 0.01 | 2.5s | 0.2 | 98.20 | 40 | 70 | 50 |
| | 0.01 | 1.5s | 0.4 | 98.50 | 20 | 75 | 30 |
| | 0.005 | 2s | 0.5 | 99.20 | 30 | 80 | 40 |
| | 0.025 | 1.5s | 06 | 99.30 | 40 | 70 | 44.40 |
| | 0.01 | 1.5s | 0.7 | 99.90 | 50 | 90 | 66.70 |
| | 0.01 | 1s | 0.8 | 99.50 | 20 | 90 | 40 |
| | 0.005 | 1s | 0.9 | 99 | 30 | 85 | 50 |

### 3.8.5 Stand - Walk Upstairs - Stand

The Stand-Walk Upstairs-Stand activity results for accuracy were about 99.9% for optimal parameters and 97.50% for Standard MEWMA which is relatively higher than the Standard MEWMA as presented in Table 3.3 and Table 3.8 respectively. For Stand-Walk Upstairs-Stand activity, the maximum precision, G-Means and F-measure for standard MEWMA were achieved is about 30%, 70% and 40% respectively with 0.005 significance value and window sizes 2s as presented in Table 3.3. However, for optimal parameters, the approach attained maximum values are about 50% ,70% and 50% with $\lambda$ value 0.6, significance value 0.005 and window size 1.5s as presented in Table 3.8. The higher precision and F-measure were achieved is about 20% and 10% respectively than the standard MEWMA. The accurate change point detection for sit to stand activity using optimal parameter selection can be shown in Figure 3.21.

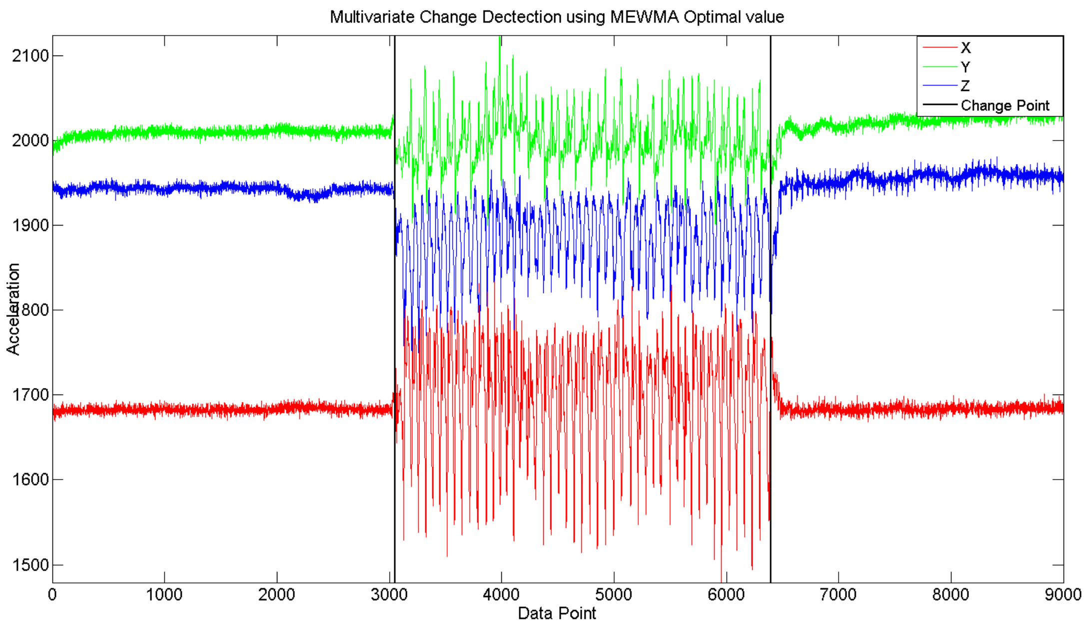



**Figure 3.21:** Stand - Walk Up Stairs - Stand

**Table 3.8:** MEWMA optimal parameter for Stand - Walk Up Stairs - Stand activity

| Activity | Significance value | Win Size | $\lambda$ | Accuracy % | Precision % | G-means % | F-Measure % |
|---|---|---|---|---|---|---|---|
| Stand- Walk Up Stairs- Stand | 0.005 | 2.5s | 0.1 | 97.50 | 10 | 65 | 20 |
| | | 2.5s | 0.2 | 98 | 30 | 60 | 44.40 |
| | | 1s | 0.4 | 99 | 20 | 65 | 30 |
| | | .5s | 0.5 | 98.50 | 40 | 66 | 44.40 |
| | | 1.5s | 0.6 | 99.90 | 50 | 70 | 50 |
| | | 2s | 0.7 | 97.90 | 20 | 60 | 30 |
| | | 3s | 0.8 | 99.50 | 30 | 65 | 40 |
| | | 1.5s | 0.9 | 98.80 | 20 | 60 | 30 |

A limitation is that the parameter selection is dependent on the activity.Moreover, the relationship between the x, y but not always z as the z-axis captures the forward movement of the leg and the y-axis captures the upward and downward motion. The x-axis captures horizontal movement of the user's leg. Figures 3.3 to 3.6 demonstrates the accelerometer data for a typical user, for all three axes of the five different activities. It is clear that standing and sitting (Figure 3.3) do not exhibit periodic behaviour but do have distinctive patterns, based on the relative magnitudes of the x, y, and z values, while the three other activities (Figures 3.4-3.6), which involve repetitive motions, do exhibit periodic behaviour. Note that for most activities the y values have the largest accelerations. This is a consequence of Earth's gravitational pull, which causes the accelerometer to measure a value of 9.8 $m/s^2$ in the direction of the Earth's centre. For all activities except sitting this direction corresponds to the y axis (see Figure 3.3). The periodic patterns for walking, Downstairs and Upstairs (Figure 3.4-3.6) can be described in terms of the time between peaks and by the relative magnitudes of the acceleration values. The plot for walking, shown in Figure 3.4, demonstrates a series of high peaks for the y-axis, spaced out at approximately $\frac{1}{4}$ second intervals. The peaks for the z-axis acceleration data echo these peaks but with a lower magnitude. The distance between the peaks of the z-axis and y-axis data represent the time of one stride. The x-axis values (side to side) have an even lower magnitude but nonetheless mimic the peaks associated with the other axes. For descending stairs, one observes a series of small peaks for y-axis acceleration that take place every $\sim \frac{1}{4}$ second. Each small peak represents movement down a single stair. The z-axis values show a similar trend with negative acceleration, reflecting the regular movement down each stair. The x-axis data shows a series of semi-regular small peaks, with acceleration vacillating again between positive and negative values. For ascending stairs, there are a series of regular peaks for the z-axis data and y-axis data as well; these are spaced approximately $\sim \frac{3}{4}$ seconds apart, reflecting the longer time it takes to climb up stairs.

## 3.9 Overall Results and Discussion

The optimal parameters of the MEWMA change point detection persistently attained substantially higher precision, G-means and F-measure than standard MEWMA. The results for accuracy were relatively high about 99.50% to 99.90% for optimal parameters and 97.50% to 99.60% for standard MEWMA. The accuracy is high because of considering each class equally important in the dataset even if exist the class imbalance problem in the dataset. The maximum precision for standard MEWMA was achieved in the range of 30% to 50% with 0.005 significance value and window sizes 2s to 3s. For optimal parameters, the approach attained figures in the range of 50% to 66.70% with $\lambda$ value (0.5 & 0.7), significance value (0.01 & 0.005) and window sizes (0.5s to 2.5s). The higher precision was attained due to the minimum number of false positive detections using optimal parameters.

Moreover, the highest G-means achieved for the optimal parameters ranged from 70% to 100% with the same $\lambda$, window sizes and significance values discussed earlier and a value of only 70% was attained for the standard MEWMA approach. The best F-measure obtained was between 50% and 80% for optimal parameters and 40% to 50% for the standard MEWMA approach.

The analysis of overall results suggests that optimal parameter selection provides better accuracy than standard parameter values and precision, G-means and F-measure improved by more than 20% for each activity with optimal parameters of MEWMA. The analysis of overall results suggests that optimal parameter selection provides better accuracy than standard parameter values. Moreover, the precision, G-means and F-measure were also improved by more than 20% for each activity with optimal parameters of MEWMA. In our experiments, low weight was assigned to historical data as compared to current data i.e. the changes are quite sudden. The empirical results achieved higher precision, G-means and F-measure because of the sudden transition occurred in the data for activities such as sit to stand and stand to sit as presented in Figure 3.17 and Figure 3.18 respectively. Moreover, as we are more interested in current data than the historical data and as $\lambda$ is the relative weight between historical and current data, however, in our application higher weight was assigned to the current data than the historical data for accurate change detection and get better results as presented in Table 3.4 and 3.5 respectively. Moreover, the empirical results were relatively higher because the transitions are more gradual for activities such as stand to walk, stand to walk Downstairs and stand to walk upstairs as presented in Figure 3.19, Figure 3.20 and Figure 3.21 respectively.

Furthermore, as we were not sure about the proper window size selection and probably similar point to first one about recent data being most important. Hence, a number of possible values for the window sizes $(1s, 1.5s, 2s, 2.5s, 3s, 4s)$, which are used to analyze the data using a sliding window with an increment of 1 data point to perform sequential analysis. The window sizes are used to evaluate the sequence from inside the window. These window sizes are chosen to combine some historical data with new data to balance the data and identify if the change happens. Additionally, we should be quite forgiving with the significance as fluctuation can occur at random, therefore, we consider a number of possible values for the significance values $h(0.05, 0.025, 0.01, 0.005)$, which are used to identify the confidence of the entire window for accurate change detection.

## 3.10 Chapter Summary

The multivariate exponentially weighted moving average (MEWMA) has been used to automatically detect change points corresponding to different transitions in user activity. The results evaluation shows that the standard MEWMA provides better accuracy and improved on the other metric measures such as precision, G-means and F-measure by more than 12%, 24% and 13% respectively than Univariate approach by (Jain and Wang, 2015). Moreover, the different parameters of MEWMA were evaluated to select the optimal parameter set. The standard MEWMA and optimal parameters were used to analyse the performance of MEWMA. The optimal parameters of MEWMA outperformed than standard values on real world accelerometer data for accuracy, precision, G-means and F-measure compared with the standard approach. Also, the MEWMA approach achieved low computation costs and can run in the online scenario. A key part of future work will be the automatic optimization of optimal parameter selection in terms of $\lambda$, window size and significance value. The synthetic dataset will also be used to determine the impact of parameter choices using MEWMA. Additionally, other multivariate algorithms for change point detection will be used from the state of the art to compare with MEWMA and analyse their performance.

# Chapter 4

# Automatic Parameter Optimization for Online Change Point Detection in Activity Monitoring Using Genetic Algorithm (GA)

## 4.1 Introduction

Chapter 3 has discussed that MEWMA is used with standard and tuned parameters such as $\lambda$, which weights the current versus historical data, window size and significance values with the aim of change-point detection. Also, the MEWMA approach tunes the different parameters to achieve better performance and accurate change-point detection. However, the limitation was that each parameter set needs to be evaluated manually to find the optimal values, which makes the approach computationally intense. However, chapter 4 provides detailed information about employing a genetic algorithm to automatically identify an optimal parameter set, using a fitness function for MEWMA, parameters such as the forgetting parameter $\lambda$, the window size, and significance value for each activity so as to maximize the Fitness Function. A genetic algorithm (GA) is used to mimic the process of evolution by taking a population of strings, which encodes possible solutions, and combining them based on the fitness function to produce solutions that are high performing. The fitness function is the core component of the GA. It evaluates each individual parameter set in the population to find the solution with an optimal fitness value. Moreover, within this chapter, the optimal parameter selection facilitates an algorithm to detect accurate change points and minimize false alarms. The performance of a real dataset and a synthetic dataset were evaluated based on data from an accelerometer collected for a set of different activities.

## 4.2 Genetic Algorithm

Arguably the most significant branch of computational intelligence is evolutionary algorithms (EAs), which have much potential to be used in many application areas. The basic concepts of EAs are inspired by observing the biological structure of nature; for instance, the principles of selection and genetic changes could be used to find the optimal solution for a given optimization problem (Holzinger et al., 2014). Moreover, the robust and adaptive characteristics of EAs are performing a global search instead of a local search to find the optimal solution in the search space.

The GA is a machine learning method which is inspired by the genetic and selection structure of nature (Goldberg and Holland, 1988). The GA is used to solve the optimization and search problems. Also, the predefined fitness function is optimized by performing a randomized and parallel search to find the optimal solution (McCall, 2005).

### 4.2.1 Optimization

Optimization is the process which is used to find the minimum or maximum value of a function. It modifies input characteristics of a system using a mathematical process to find the minimum or maximum output. This process can be used in various domains such as economics, chemistry, production, physics or any other measure. The optimization of a function can be either minimization or maximization and denoted by $f$. Hence, the maximization function $f$ is equivalent to the inverse of the minimization function e.g. $-f$ (Van Laerhoven and Schiele, 2009).

The minimization and maximization function can be defined using Equation 4.1

and Equation 4.2 respectively.

$$\begin{aligned} &\text{Given} \quad f : \Re^n \;\rightarrow\; \Re \\ \text{Find } \hat{X} \in \Re^n \quad &\text{such that } f(\hat{X}) \leq f(X), \quad \forall X \in \Re^n \end{aligned} \tag{4.1}$$

$$\begin{aligned} &\text{Given} \quad f : \Re^n \;\rightarrow\; \Re \\ \text{Find } \hat{X} \in \Re^n \quad &\text{such that } f(\hat{X}) \geq f(X), \quad \forall X \in \Re^n \end{aligned} \tag{4.2}$$

The function $f$ is called objective function that maps the function space with the search space. Moreover, the $\Re^n$ of $f$ is denoted as search space or parameter space (Kennedy, 2006) and each element in $\Re^n$ signifies a solution while $\hat{X}$ represents an optimal solution in the search space. Moreover, the number of dimensions and parameters involved in search space is denoted by $n$. In this study, the objective function or fitness function is maximized to find the optimal solution to a system.

The GA starts with a random sample of variable sets and repeatedly modifies a population of individual solutions. Various criteria can be used for the selection process to obtain the desired solution through the evaluation of individual solutions. The best individual solution is selected as an input for the next generation. The GA is used for solving optimization problems based on natural selection, which is the process used in driving biological evolution (Malhotra et al., 2011). The optimization modifies input characteristics of a system using a mathematical process to find the minimum or maximum output. The maximization of the fitness function in the GA is used to find the optimal solution to a system.

Moreover, the population of "individuals" are used by GA, where each "individual" signifies a possible solution to a specific problem. The fitness score is assigned to each "individual" according to the problem in order to generate a good solution. Within the population, the best "individuals" are selected as a high-fit and are given more chances to reproduce by "cross breeding" with other individuals. Hence, new individuals are generated as "offspring" which take features obtained from each "parent". Likewise, the least-fit individual in the population has minimum chance for reproduction and hence is declined or dies out.

Furthermore, the best individuals from the current generation produce a new population from all possible solutions and generate new individuals by the cross-breeding process. The formation of a new generation consists of mainly different individuals, which have better characteristics than the previous generation. The process evolves and after successive generations, the individual with best characteristics disseminates throughout the population. This cross breeding of high-fit individuals helps to explore the most promising area of the search space.
Thus, the population should converge to an optimal solution to the problem, if the GA has been designed properly (Busetti, 2007).

### 4.2.2 Basic principles

A number of steps are carried out in GA to find an optimal solution of the problem. In the first step, the focus is on construction or forming a suitable coding (or representation) of the problem. In the second step, the objective function or fitness function is required to provide a figure of merit (fitness) for each solution. During execution, a parent will be selected for reproduction and recombined to make offspring. In the following sections, these aspects are explained in detail.

#### 4.2.2.1 Coding

The probability of potential solution to the problem may be based on a set of parameters such as the dimensions of the multivariate data. Such parameters are known as a gene and combined to structure a string of values referred to as a chromosome. For instance, for the problem to maximize a function that involves three variables, $F(x, y, z)$, the GA takes each variable as a 10 digit binary number; hence, the chromosome consists of three genes which each contain 10

```
BEGIN /* genetic algorithm */
    generate initial population
    compute fitness of each individual

    WHILE NOT finished DO
    BEGIN /* produce new generation */

        FOR population_size / 2 DO
        BEGIN /* reproductive cycle */
            select two individuals from old generation for mating
                /* biassed in favour of the fitter ones */
            recombine the two individuals to give two offspring
            compute fitness of the two offspring
            insert offspring in new generation
        END

        IF population has converged THEN
            finished := TRUE

    END

END
```

**Figure 4.1:** The basic GA Pseudo code (Basseville et al., 1993)

binary digits.

As the GA algorithm is inspired by the biological structure of nature, so in genetic terms, a chromosome that represents each set of parameters is described as a genotype (Beasley et al., 1993). Further, the genotype consists of information required to build up an organism that is known as phenotype. The same scenario is used in GAs, for example, in the change detection task in multivariate data, the specific set of parameters is identified as the genotype while the combination of these helps in developing solution and is referred as phenotype. The performance of a phenotype helps in identifying the fitness of an individual parameter. This can be calculated using the fitness function to minimize or maximize the fitness value. The basic pseudo code of GA is illustrated in Figure 4.1.

#### 4.2.2.2 Fitness Function

The most important part of the GA is the fitness function and it is used in GAs to solve optimization problem. The 'fitness' word is taken from evolutionary theory because the function evaluates and specifies how each potential solution is 'fit' for a given problem (Mitchell, 1995). The GA initiates with the array of chromosomes selected randomly which is considered as the initial population. The chromosome can be a numerical value or values that signifies a possible solution to the problem that the GA is trying to solve. For example, if a problem has $X_p$ dimensions, then each chromosome can be encoded as an array elements of $X_p$ .

$$Chromosome = [P_1, P_2, ..., P_{Xp}]$$

where each $p_i$ is the specific value of the $i^{th}$ parameter set (Haupt and Haupt, 2004). Moreover, the fitness function evaluates each specific chromosome and return a single value which is considered the fittest value corresponding to the ability of individuals that chromosome represents (Beasley et al., 1993).

#### 4.2.2.3 Selection

In the selection process, the GA uses the fitness value as a discriminator between the quality of solutions obtained from the chromosomes in a GA population. The chromosomes are selected for reproduction based on the best fitness value. The fitter the chromosome is, the higher chance it has to be selected as compared with lower fitness values. Hence, more emphasis on the selection of more highly fit solutions is created. The most common selection method is replacement in which the most highly fit chromosomes have greater chances to be selected more than once or recombined with themselves (McCall, 2005). A number of methods has been used for the selection process such as Uniform, Roulette Wheel, Stochastic uniform and Tournament.

In uniform selection method, the parents are selected randomly from a uniform distribution based on the expectation and number of parents. This method is not considered as useful for selection due to unidirectional search but can be used for testing GAs. The roulette wheel method is used by allocating a probability to each chromosome and select the chromosomes whose probability is proportional to its relative fitness. In further steps, the method randomly selects a chromosome whose probability is equal to the sum of finesses of all chromosomes in the population (Goldberg and Holland, 1988). Moreover, the stochastic uniform approach explicitly selects each chromosome randomly with probability in proportion to its expectation. The algorithm moves along at evenly space interval and gives a chance to weaker member of population to be selected, by mutation. In the Tournament selection method, the set of chromosomes are randomly selected with uniform probability. However, the selection is made from the set of chromosomes which has the highest fitness (McCall, 2005).

#### 4.2.2.4 Reproduction

The reproduction step is used after the selection of a parent with high fitness from the population. Further, the chromosome of these parents is recombined using the crossover and mutation mechanism. The basic mechanism of crossover and mutation are explained in the following section

#### 4.2.2.5 Crossover

Crossover is used to mix the genetic material of the two selected parents chromosomes and to generate one or two child chromosomes. Once the two parent's chromosomes are selected for reproduction, a random number is generated with uniform probability in the interval [0,1] and the comparison is performed against the pre-determined "crossover rate". Thus, if the random number is less than or equal to the pre-determined crossover then the crossover will be applied otherwise not. Different crossover functions are used in the literature e.g. single point, Scattered, Two point, heuristic and arithmetic. The most common crossover operator is single point as shown in Figure 4.2.

In single point crossover, the two individuals with their chromosomes are divided randomly at a certain position, which forms two head segments and two tail segments. Furthermore, the tail and head segments are swap over to produce two new chromosomes as shown in Figure 4.2. Each individual has obtained some gene data from each parent and form two offspring.

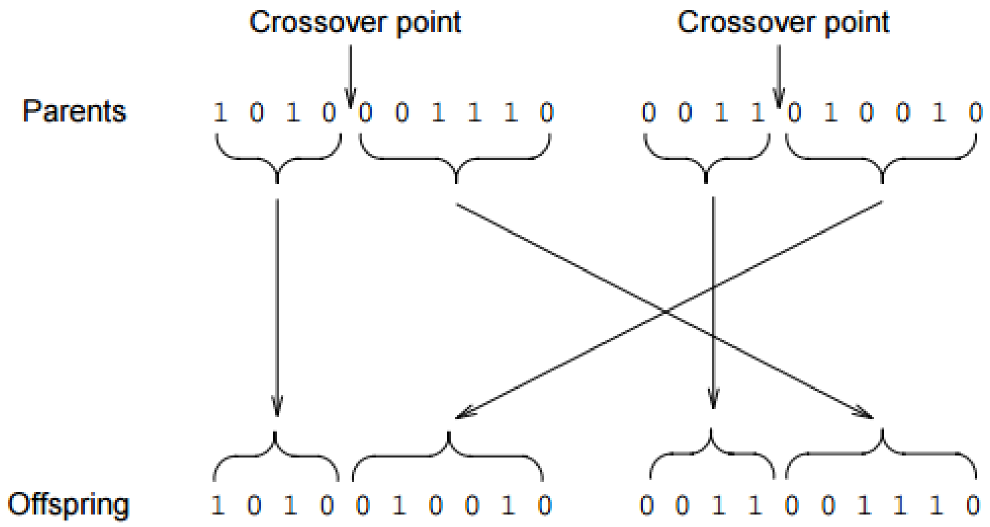



**Figure 4.2:** Single-point Crossover

The scatter crossover function is used to create a random binary vector for two parent chromosomes., The genes are selected from the first parent where the vector is 1 and from second parent, the genes are selected where vector is 0. Then, the genes are combined to form the child. The two-point crossover function, first randomly select the two integers such as $a$ and $b$ between 1 and $n$ where $n$ is the total number of variables. The genes are selected from the first parent, If the position of the chromosome is less than or equal to $a$. However, the genes are selected from the second parent form the position $a+1$ to $b$. Furthermore, again from first parent, the genes are selected, if the position of chromosome greater than b. Finally, we concatenate the selected genes to form a single gene.

Moreover, the heuristic function is used to create children randomly for two parents where each one has minimum distance in line from their parent. The function is based on the assumptions that offspring with a small distance from the parent has better fitness and a large distance is associated with worst fitness. The arithmetic function can be used to create children by calculating uniform random arithmetic mean from their parents.

#### 4.2.2.6 Mutation

After crossover, the mutation is applied individually to one or more child in the offspring. In this process, each gene is randomly altered with small probability typically 0.001. Mutation provides genetic diversity and enables the GA to search a broader space. For example, mutation is performed on the fifth gene of the chromosome as shown in Figure 4.3. Different functions are used to perform mutation such as Gaussian, Uniform and Adaptive. The Gaussian function adds a random number to each vector which is taken from the Gaussian distribution centered on zero. The two parameters such as scale and shrink are used to manage the standard deviation of this distribution. The scale parameter is used to induce the standard deviation at the first generation while the shrink parameter is used to decrease in standard deviation as the generations move on. Hence, if the shrink parameter is 0 the standard deviation is constant and if it is 1, the standard deviation shrinks to 0 linearly and indicate that the last generation is reached. The Uniform function for mutation involves two steps. First, the algorithm selects a fraction of the vector entries where each entry has the same probability. Secondly, the random number replaces each selected entry uniformly from the range of entries. Moreover, the adaptive feasible function is used to randomly generate directions that are adaptive with respect to the last successful or unsuccessful generation. The linear constraints and bounds can be satisfied by choosing proper step length for each direction.


**Figure 4.3:** Mutation

## 4.3 Parameter Optimization using a GA for online change point detection

The MEWMA approach is a statistical method that averages the input data within a data stream and assigns lower weights to earlier data points. The primary aim of using the MEWMA is to detect small shifts quickly in time-series data as discussed in detail in section 3.3.2.1. In the proposed solution, the MEWMA is used to analyze all the covarying time-series data at the same time thus taking into account the interrelationship among the variables. MEWMA is used with standard and tuned parameters such as $\lambda$, which weights the current data versus historical data, window size, and statistical significance values, with the aim of accurate change-point detection. In addition, we use the GA to automatically identify an optimal parameter set for the MEWMA including $\lambda$, window size, and significance value for each activity by evaluating the fitness function of the F-measure. The MEWMA can be calculated using Equation 3.5 and Equation 3.6 with the same parameter values discussed earlier in section 3.3.2.1 of chapter 3. Furthermore, Multivariate analysis is used to measure more than one characteristic of a system and also to evaluate the relationship among these characteristics. In multivariate analysis, we consider the data stream of length q consisting of specific data points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \ldots, \mathbf{X}_q$ (e.g., for accelerometer values $\mathbf{X}_i$ =(-1.858,-9.649,1.132) where the elements represent the $x, y$, and

$z$ values of the 3-dimensional accelerometer signal. In general, a sequence of data point $\mathbf{X}_1$ to $\mathbf{X}_q$ may contain different distributions. In particular, the two subsequences $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \ldots, \mathbf{X}_{i-1}$ and $\mathbf{X}_i, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_q$ may follow different distributions (say, for example, $D_1$ and $D_2$, where $D_1$ and $D_2$ can be equal or different). The aim of the algorithm is to determine and classify the position of change points $x_i$ in the data stream. In each data stream, MEWMA is used to evaluate the position of change points and calculate the exponentially weighted moving average of multivariate input vectors $\mathbf{X}_i$ to provide accurate change-point detection. We consider a number of possible values for the window sizes (1s, 1.5s, 2s, 2.5s, 3s), which are used to analyze the data using a sliding window with an increment of 1 data point to perform sequential analysis. The window sizes are used to evaluate the sequence from inside the window. These window sizes are chosen to combine some historical data with new data to balance the data and identify if the change happens. Also, these are reasonable sizes that are taken from experimentation. Likewise, the $\mathbf{Z}_i$ represents the MEWMA vector and is calculated by using the multivariate input vectors as shown in Equation 3.5. In addition, the variance-covariance matrix of $\mathbf{Z}_i$ is calculated recursively and represented by $\Sigma_i$ to find T-squared, as shown in Equation 3.6.

Once the T-squared statistic is calculated as shown in Equation 3.6, we consider a number of possible values for the significance values $h$ (0.05, 0.025, 0.01, 0.005), which are used to identify the confidence of the entire window. These values are used in the literature and define regions where the test statistics are unlikely to lie (Handbook, 2016). If the T-squared value is greater than $h$, then $x_i$ will be labeled as a change point within the data stream. The analysis of the accelerometer data identifies the actual values of the specific change points, which may represent an increase or decrease in the data. Thus, when executing a sliding window $k$ can be used to eliminate such adjacent change points. As discussed earlier in section 4.2, the GA can be used for solving optimization problems based on natural selection, which is the process used in driving biological evolution (Malhotra et al., 2011). The optimization modifies input characteristics of a system using a mathematical process to find the minimum or maximum output. The objective of the fitness function in GA is used to find the optimal solution to a system. In our case, each distinct combination of the three variables provides a single solution in the population, namely $\lambda_i$, the window size, and the significance. Over a number of generations, these solutions "evolve" towards the optimal solution (McCall, 2005). The fitness function is the core component of the GA. It evaluates each individual parameter set in the population to find the solution with an optimal fitness value. In our fitness function, we initialize the population of vectors whose elements contain the $\lambda_i$ values, the window sizes, and the significance values. Our fitness function then tries to find the solution with the maximum F-measure value given a range of input values. The F-measure is used as the measure to find the overall effectiveness of the change detection or activity recognition by combining the precision and recall. The fitness function can be defined as follows:

$$F - measure_{max} = max_{(\lambda_i, win\_size, sig\_value)}(F - measure_{MEWMA}) \tag{4.3}$$

For simplicity, we assume $\lambda_i$ is equal to $\lambda$ for $i = 1, \ldots, p$, where $\lambda_i$ ranges from 0.1 to 1 for each activity with the corresponding significance values of 0.05, 0.01, 0.025, 0.005 and window sizes of 1 s, 1.5 s, 2 s, 2.5 s and 3 s. Our proposed model uses Equation 4.3 as the fitness function by initializing upper and lower bounds of the three parameters to find the maximum F-measure with the optimal parameter set. After the exploration with different parameter settings, the optimal GA parameters, which maximize the fitness function of the F-measure, are shown in Table 4.1.

The selection function in the GA chooses the parents for the next generation based on their scale values by evaluating the fitness function. As we need to find the maximum value of the fitness function using Equation 4.3, the individual with the maximum value of the fitness function has greater chance for repro-

**Table 4.1:** Genetic algorithm (GA) Parameters.

| **Parameters** | **GA** |
|---|---|
| Population Size | 50 |
| Selection | Stochastic uniform |
| Reproduction | 0.8 |
| Crossover | Scattered |
| Mutation | Adaptive feasible |
| Generations | 100 |

duction and also for generation of offspring. Here we used a stochastic uniform distribution to build in randomness. The reproduction function helps to determine how the GA creates children at each new generation. Elite count or the crossover fraction can be used to create new children at each generation. The first method specifies the number of individuals that are guaranteed to survive in next generation. However, the later method specifies the fraction of the next generation which crossover produces; we here use reproduction probability 0.8 and mutation with probability 0.2 so as to allow some new values to take part in the optimization process.

The crossover combines two individuals or parents to form a new individual or child for the next generation. Different methods such as constraint dependent, scattered, heuristic, and arithmetic approaches can be used depending on the problem requirement. We choose the scatter method to make random selection. In the population, the mutation function makes small random changes in the individuals, which provide genetic diversity and enable the GA to search in a broader space. Different methods can be used for this, such as the Gaussian function, uniform function, and adaptive feasible function for random modification. We choose an adaptive feasible solution because it randomly generates directions that are adaptable with respect to the last successful generation.
The GA process, illustrated in Figure 4.4 with respect to the GA parameters

proposed in Table 4.1, is described as follows (McCall, 2005):

- Initialize the population size is with the number 50, which specifies how many individuals there are in each of the iterations. Usually, the number 50 is used for a problem with five or fewer variables, and the number of 200 is used otherwise.

- Check the termination condition of the algorithm to determine if the number of generations has exceeded the maximum value. If so, the GA algorithm is terminated, otherwise, continue with the following steps.

- Calculate the maximum value of the fitness function using Equation (4.3).

- The individuals are selected from the current population applying a stochastic uniform function. Each parent corresponds to a section proportional to its expectation. The algorithm moves along in steps of equal size. At each step, a parent is allocated from the section uniformly.

- The individuals are then reproduced randomly with a fraction using the crossover operation. The scatter function is used to select the genes where the vector is 1 from the first parent and 0 from the second parent before combining them to form a child.

- Mutation is then applied with the adaptive feasible method to randomly generate individuals in the population.

- Finally, a new generation is updated and the GA algorithm loops back to check the termination condition. The default value for the generations is 100 multiplied by the number of variables used, but we choose the best value for generation by experimentation with different values

## 4.4 Experimental Setup

The experiments were performed on two real datasets and one synthetic dataset for optimal parameter selection using a GA. The evaluation is performed using different metric measures and the GA is used to automatically identify an optimal parameter set, using a fitness function for MEWMA, and parameters such as the forgetting parameter $\lambda$, the window size, and significance value for each activity so as to maximize the fitness function i.e. the F-measure. The detailed explanation for all datasets are given in section 4.4.1 and section 4.4.2 respectively.

### 4.4.1 Real Dataset 1

In our experiments, we used a real dataset for evaluation. AlgoSnap uses the CrowdSignals platform to collect sample datasets to help and support researchers in academia. CrowdSignals.io is a non-profit research community. The CrowdSignals platform was created by AlgoSnap to build a large labelled mobile and sensor dataset for the research community. Our sample dataset is taken from the above platform and fed to the algorithm as a stream, to represent a real deployment. This sample dataset was collected from two participants who kept a smartphone inside the right-front pants pocket and wore a smartwatch on the dominant wrist (Algosanp., 2016). The data from each participant was captured continuously for 2.5 hous using 20 sensors with sample frequency of 74.4 Hz. Each participant performed eight different activities and also labelled these activities.

The eight different activities performed by each participant were eating, washing hands, smartphone kept on the table, sitting, standing, walking, running, and driving. The duration of an activity varied from 1 min to 5 min depending on the activity. A transition could be regarded as an activity itself, especially if takes a long time. However, here we focus on the core activities and primary change points. The time delay ranges from 5 ms to 12 ms. The participant used the smart phone Android app online to explicitly label the start and end times of each activity performed. However, the start and end time was not very precise which required manual correction of the data after collection.

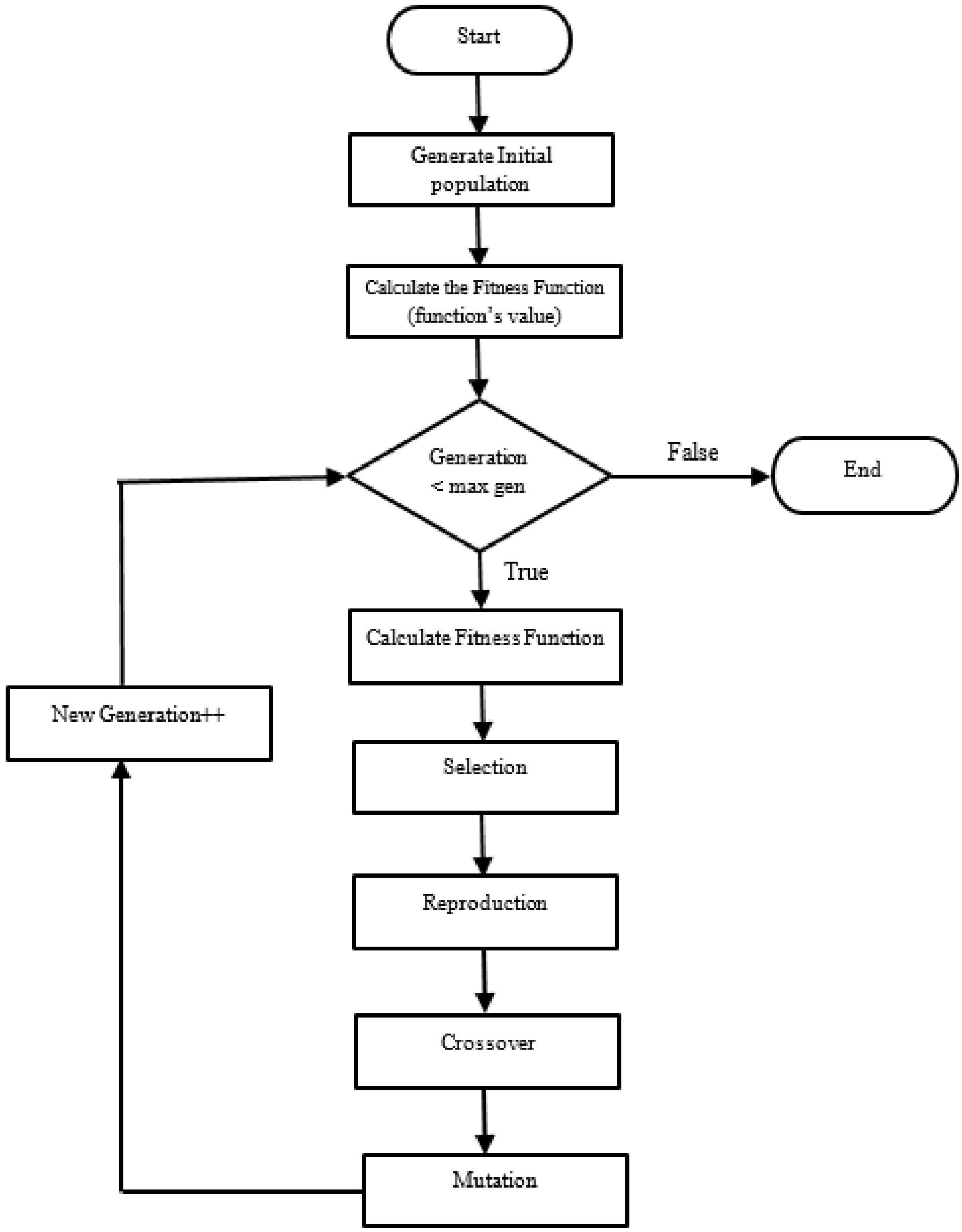



**Figure 4.4:** Flow chart of various stages to perform genetic algorithm (GA) optimization

Moreover, the labeled data is sent periodically to the server which runs the GA offline for optimization as shown in Figure 4.5. The start and the end time for each activity are denoted in the dataset as a truth table. In the sample dataset, various sensors were used to collect data, but only accelerometer data is used in our experiments. For illustrative purpose, only one accelerometer sensor, with three dimensions, was used but other authors have demonstrated how multimodal sensors can be used to increase activity recognition and enable the recognition of activities in various situations (Han et al., 2014). After the data collection, the activity execution of accelerometer data was wirelessly streamed to a receiving computer via the IEEE 802.15.1 Bluetooth communications protocol.

The study in (Ziefle et al., 2013) elaborates the high acceptance for telemedicine and usability of a telemedicine approach. The deployment of such an application is useful in emergency situations and achieves higher accuracy and quality of data for monitoring of patient vital parameters over time. A limitation could be the privacy issues, date security, and high probability of false alarms. In our work, we partly address the additional problem of low user acceptance due to excessive requirements to interact with the mobile phone.

#### 4.4.1.1 Results and Discussion

A real dataset, as described, has been used by the GA to identify the optimal set of parameters for the MEWMA approach in change-point detection. For the multivariate approach the $x, y$ and $z$ acceleration magnitude is calculated from the captured data and used as the input to the MEWMA algorithm. The MEWMA algorithm is initially used to analyze different parameters including $\lambda$ (0.1 to 1), the window size (1 s, 1.5 s, 2 s, 2.5 s, 3 s) and the significance values (0.05, 0.025, 0.01, and 0.005) to find the accurate change point. We considered all the values of $\lambda$ in the range varying from 0.1 to 1 to allow for some contribution from both historical data and current data.

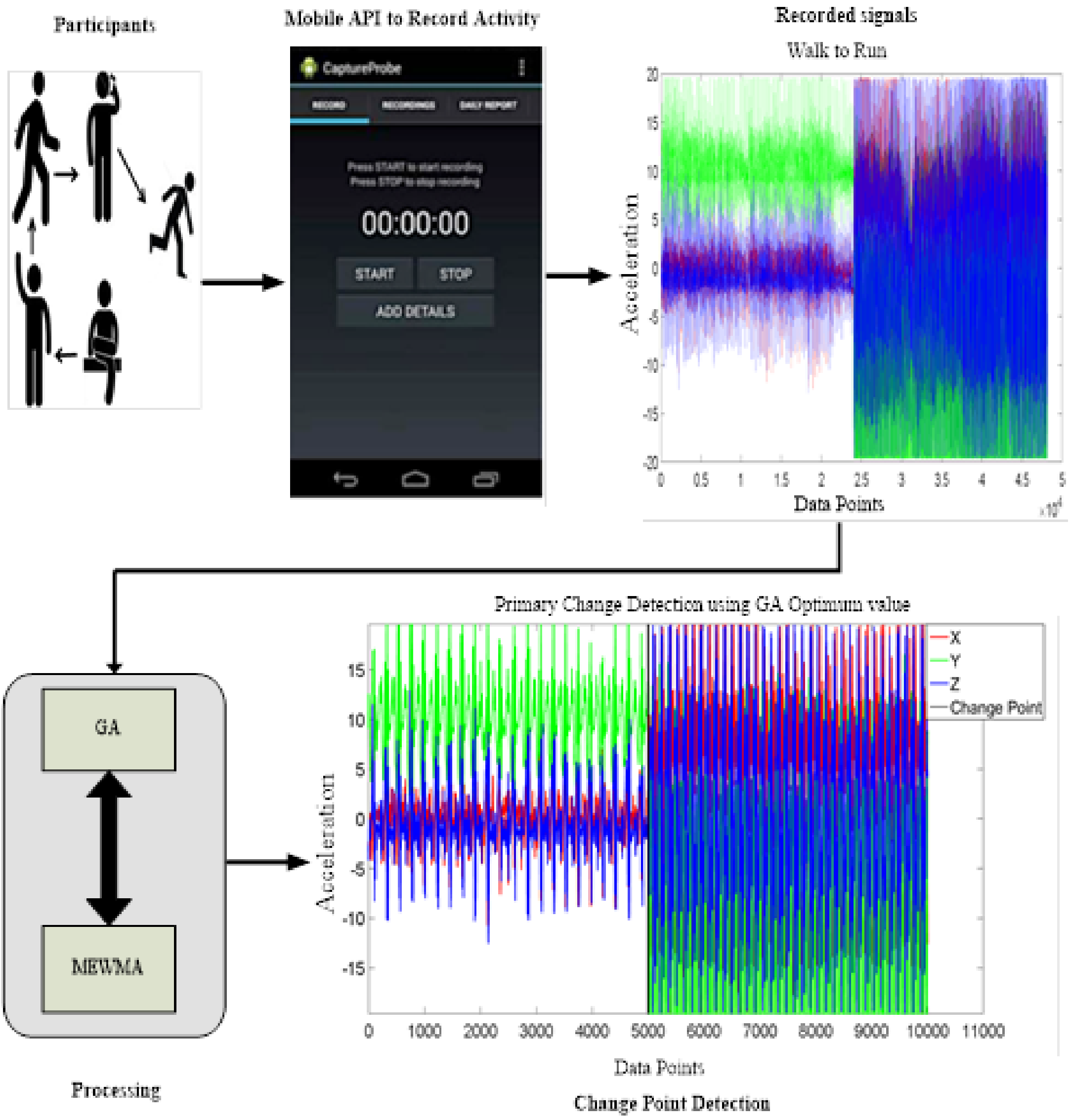



**Figure 4.5:** The System Model

Moreover, MEWMA also combines historical data and current data. Following this, the GA is used to identify the optimal set of parameters for the MEWMA algorithm. However, the GA implemented in Matlab 2014 typically takes a long time, where in our experiments it takes approximately between 10 min and 25 min to run on a system with processor 3.40 GHz and 8 GB RAM. The parameter values are not likely to change too frequently, so the GA could be run offline periodically. The F-measure metric was used to evaluate the optimal change point in the activity monitoring using the GA. In our experiments, when determining true positives a quarter second buffer was included at either side of the manually labelled change point to accommodate subjectivity errors inherent in manual labelling. Thus, a detected change point was considered true if its index in the data stream, $i$, $i \epsilon z - f/4, ..., z + f/4$ where $z$ is the index in the data stream of the manually labelled change point and $f$ is the sampling frequency in $Hz$. In our experiment, we formed a dataset containing activities such as walking to running, walking to driving, walking to washing hands, walking to standing, and walking to sitting.

The objective of our proposed technique is to identify the optimal set of MEWMA parameters using the GA for detecting change points in high-level activities such as walking to running and walking to driving, examples of which are shown in Figures 4.6 and 4.7 respectively. The sliding window with optimal change-point detection parameters for the activity "walking to running" has window size of 3s with significance value $p = 0.05$ and $\lambda = 0.7$. The optimal change-point detection parameters for the activity "walking to driving" are that the window size is 2.5s, significance value $p = 0.05$, and $\lambda = 0.6$.

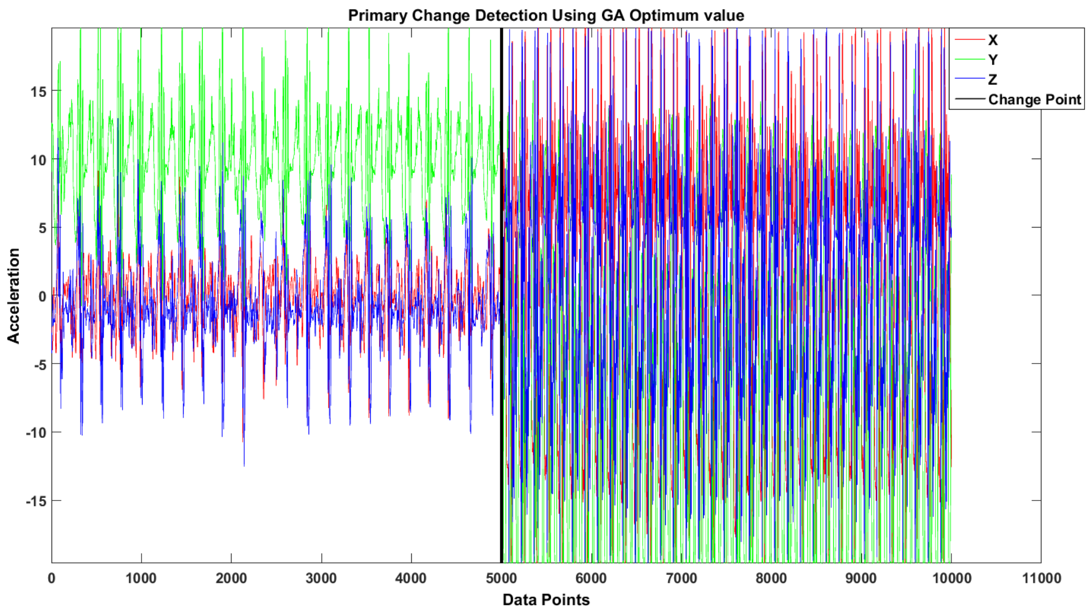



**Figure 4.6:** A real dataset example of a sliding window change-detection result for the activity "walking to running".

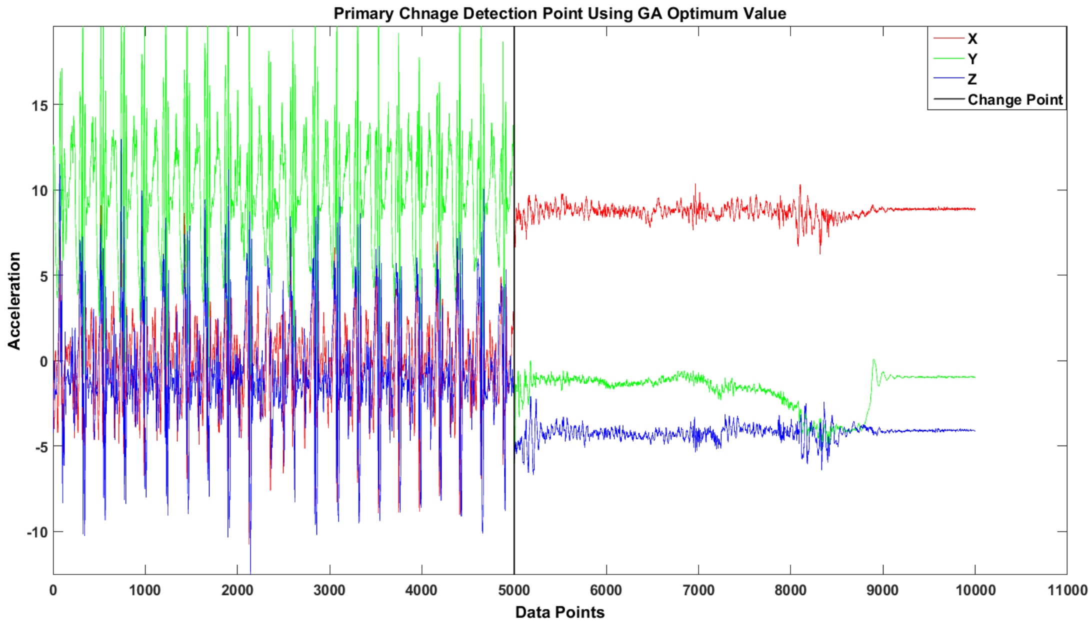



**Figure 4.7:** Real dataset example of sliding window change-detection results for the activity "walking to driving".

The experimental results for real dataset of five different activities are presented in Table 4.2. Moreover, the experimental results identify the changes between core activities are shown in Figure 4.6 and Figure 4.7 respectively. The detected transitions were all of the transitions that occur in the dataset. Here, the data points relating to the core activities are used to determine when the change points occur. In our experiments, we analyzed dynamic activities such as walking followed by another dynamic activity such as running or driving due to its complexity and varying characteristics.

**Table 4.2:** Non-optimized and optimized with GA parameter set for five different activities on a real dataset

| Activity | Sig Value | Non-Optimized | | | | Optimized with GA | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\lambda$ | Win Size | F-Measure | Accuracy | $\lambda$ | Win Size | F-Measure | Accuracy |
| Walk to Sit | 0.05 | 0.3 | 2s | 50% | 99.4% | 0.4 | 1.5s | 66.7% | 99.8% |
| Walk to Stand | | | 2s | 50% | 99.4% | 0.4 | 1.5s | 66.7% | 99.8% |
| Walk to wash hands | | | 2.5s | 50% | 99.4% | 0.5 | 2s | 66.7% | 99.8% |
| Walk to Driving | | | 3s | 40% | 98.5% | 0.6 | 2.5s | 50% | 99.4% |
| Walk to Running | | | 3s | 40% | 98.5% | 0.7 | 3s | 50% | 99.4% |

The proposed approach optimized the MEWMA parameters in order to find the best set of parameters for accurate change point detection for the different activities presented in Table 4.2. Furthermore, accuracy and F-measure metrics have been used to find the optimal parameters selection of the MEWMA algorithm. The accuracy is the ratio of the number of correctly classified data points to the total number of data points and can be calculated using Equation 3.6. Moreover, the F-measure is used to find the overall effectiveness of the activity recognition by combining precision and recall using Equation 3.8, Equation 3.9 and Equation 3.12 respectively. The Accuracy and F-measure are discussed in detail in chapter 3.

The non-optimized experimental results on the real dataset are presented in Table 4.2. The maximum F-measure and accuracy values are in the range of 40%–50% and 98.5%–99.4%, respectively among all the activities. The walking activity followed by a static activity achieved a maximum F-measure of about 50%, whereas subsequent dynamic activities have achieved 40%. However, the optimized experimental results on a real dataset that achieved the maximum accuracy and F-measure were in the range of 99.4%–99.8% and 50%–66.7%, respectively. The walking activity followed by static activity achieved a maximum F-measure of circa 66.7%, whereas subsequent dynamic activities achieved 50%. The highest accuracy and F-measure values in the experimental results on real dataset are achieved using the GA optimal parameter set of $\lambda$ (0.4–0.7), significance value $p = 0.05$ and window sizes (1.5 s, 2 s, 2.5 s and 3 s) as shown in Table 4.2. The highest F-measure values achieved are 50%–66.7% for all activities using the optimal parameter set with the real dataset. A dynamic activity such as walking followed by a static activity such as sitting, standing, and hand washing achieved the highest F-measure of 66.7% with an optimal parameter set of $\lambda$ (0.4 and 0.5), significance value $p = 0.05$, and window size 1.5s and 2s. However, the subsequent dynamic activities such as driving and running achieved the highest F-measure of 50% with an optimal parameter set of $\lambda$ (0.6 and 0.7), significance value $p = 0.05$, and window size 2.5s and 3s. Moreover, the accuracies achieved with optimal parameter set by the GA ranged from 99.4% to 99.8% as shown in Table 4.2.

The experimental results show that the F-measure values are higher using the optimal parameter set from the GA than the results with non-optimized parameters. Additionally, in Table 4.2, the accuracies are also improved from a range of 98.5% to 99.4% with non-optimized parameters to a range of 99.4% to 99.8% with the optimized parameters.However, as there is a little room for improvement hence, improvement on this data is very challenging. When we take out the inter-activity transition period and simulate data on this basis, the advantage of using the GA optimization is even more significant. The reason is that in the simulated data we ignored the transition data, which may be from a different distribution from the data relating to the core activities (Khan et al., 2016).

#### 4.4.1.2 Walking in the Wild

Generally, sensor data is collected in a laboratory setting and subjects perform the activities that are specified by experimenters. In the wild, however, behavior is not prescribed and the sensor data must be labeled during or after the sensor data is generated, as shown in Figure 4.8. This problem occurs in online change detection in real-time scenarios. In this situation, we can alert the reminding software that we would like to sample data more frequently to increase the accuracy of activity detection. Also, we would like to be able to identify and detect early on that a change seems to be happening and ask the user for some information on what activity is actually being performed in order to improve our algorithm. An alert about the change could be issued to get a response from the user on what activity is being performed. The alert and response thus provide more labeled data for learning. Periodically we rerun the GA algorithm offline using new data. The data is typically processed locally on a mobile phone or smart watch but a summary of the data is transferred to the server periodically.The inter-activity transition period was taken as an activity to identify a change point in different user activities.

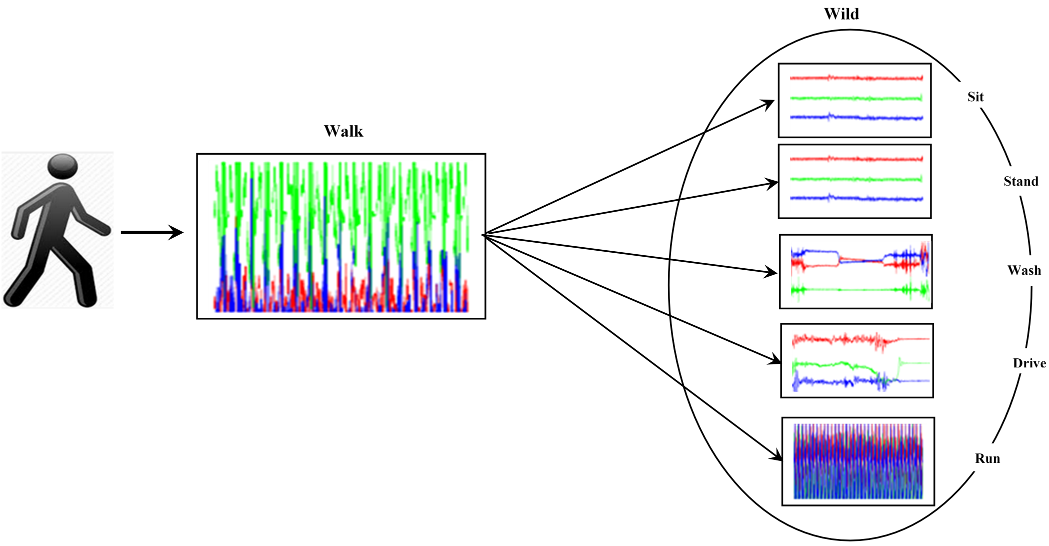



**Figure 4.8:** Walk in the Wild

When the person is walking or sitting for long time, the storing or handling of the data could drain the battery as a mobile device typically has limited battery capability. The assumption of this work is that we need a lightweight and early warning indicator when a change is about to happen. Walking in the Wild refers to transitions from walking to a state that is not prescribed in advance.We also performed experiments on walk to wild irrespective of the activity which is happening next, as presented in Table 4.3. The optimal parameter set is discovered for accurate change detection using the GA. The best F-measure and accuracy achieved was 66.7% and 99.8% respectively with the optimal parameter set of $\lambda = 0.7$, significance value $p = 0.05$, and window size 3s. The experimental results of walk to wild are presented in Table 4.3. In our experiments, the data was used from the existing dataset discussed in section 4.4.1 from walking into different user activities such as sit, stand, drive and run for accurate change point detection.

A class imbalance problem usually exists in datasets when the total number of instances of one class (the minority) is excessively low as compared with the number of instances of the other (majority) class (Ni et al., 2015). This highlights

**Table 4.3:** Optimized parameter set with GA for walking in the wild on real dataset

| Activity | $\lambda$ | Win Size | Sig Value | F-Measure | Accuracy |
|---|---|---|---|---|---|
| Walk to Wild | .7 | 3s | 0.05 | 66.7% | 99.8% |

the skewed distribution of classes within the dataset, and often the minority class is the class of interest (Galar et al., 2012). In our dataset, we have only one TP point (represents a correctly identified change point) and a high number of TN (the non-transitional points which are not labeled as change). We used the F-measure for evaluation because it is a combination of precision and recall, as presented in Equation 3.11. As the precision is the ratio of TP over the total number of TP and FP (the non-transition point which the algorithm highlighted as a change) therefore one or two FP detections reduced the F-measure to 66.7% and 50%, respectively, due to the imbalance class problem in our real dataset.

### 4.4.2 Real Dataset 2

Real and synthetic datasets of accelerometer data were used for evaluation. The real dataset consisted of two users' data generated by wearing the shimmer wireless sensing platform. The sensor placement positions on the subject's body enabled anterior-posterior and lateral movements to be captured effectively. The two users performed 8 different activity transitions consisting of sit to stand, stand to sit, stand to walk corridor, stand to walk downstairs ,stand to walk upstairs, walk corridor to stand, walk downstairs to stand and walk upstairs to stand (Zhang et al., 2011). For each activity, the participant remained in each state for 30 seconds and then transitioned to another activity like sit to stand. The activity execution of accelerometer data was wirelessly streamed to a receiving computer via the IEEE 802.15.1 Bluetooth communications protocol (Zhang et al., 2011).

#### 4.4.2.1 Synthetic dataset

We have generated synthetic dataset from the real dataset discussed in section 4.4.2 for the same set of activities as the real dataset using a multivariate normal random number generator (mvnrnd) Matlab function (Multivariate, 2016). The mean and covariance vector is chosen randomly for each activity using a multivariate normal random numbers distribution function. The ten instances (random vectors) are generated for each activity and these random vectors were used to generate a synthetic data set for 8 different activities. The different parameters for change detection algorithm are used to find the primary change points in these activities. After exploration with different GA parameter setting, the optimal GA parameter settings chosen as shown in Table 4.1 which is used to maximize our fitness function and return the optimal best parameters with maximum F-measure.

#### 4.4.2.2 Results and Discussions

This Section evaluates the genetic algorithm on a real and synthetic dataset that identifies the optimal set of parameters for a Multivariate Exponentially Weighted Moving Average approach to change point detection. For the multivariate approach the $x, y$ and $z$ acceleration magnitude is calculated from captured data and used as input to the MEWMA algorithm. The MEWMA vector is used to detect changes in the data stream. The MEWMA is initially used to analyze different parameters; $\lambda$ (0.1 to 1) window size (1s, 1.5s, 2s, 2.5s, 3s) and significance values (0.05, 0.025, 0.01, and 0.005) to find the accurate change point. Following this, the GA is used to identify the optimal set of parameters for the MEWMA algorithm. The F-measure metrics were used to evaluate the optimal change point detection in activity monitoring using GA. The detected change point is considered true if in the data stream the index $i$, $i \epsilon z - f/4, ..., z + f/4$ where $z$ indicates the index of manually label change in the data stream and $f$ denotes the sampling frequency in $Hz$.The objective of our proposed technique is to identify the optimal set of MEWMA parameters for detecting transitions in high level activities such as stand to walk downstairs and walk downstairs to stand as shown in Figure 4.9 and Figure 4.10 respectively. The following Figure 4.9 (a & b) and Figure 4.10 (a & b) are the examples of Real and synthetic datasets respectively.

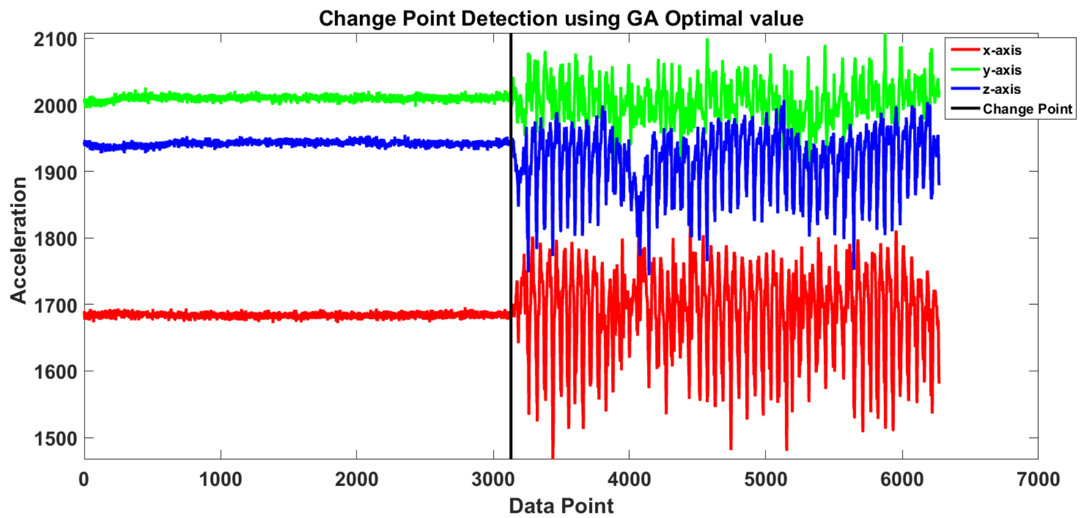
The real dataset example of activity 'stand still to walk downstairs' and 'walk downstairs to stand still' is shown in Figure 4.9. The sliding window change detection results was window size 1s second with significance $p$=0.05, $\lambda$=0.5 and 2s second win size, significance $p$=0.05 and $\lambda$=0.7 respectively as shown in Figure 4.9.

The synthetic dataset example of activity 'stand still to walk downstairs' and 'walk downstairs to stand still' is shown in Figure 4.10. The sliding window change detection results was window size 1s second with significance $p$=0.05, $\lambda$=0.5 and 2s second window size, significance $p$=0.05 and $\lambda$=0.7 respectively as shown in Figure 4.10.

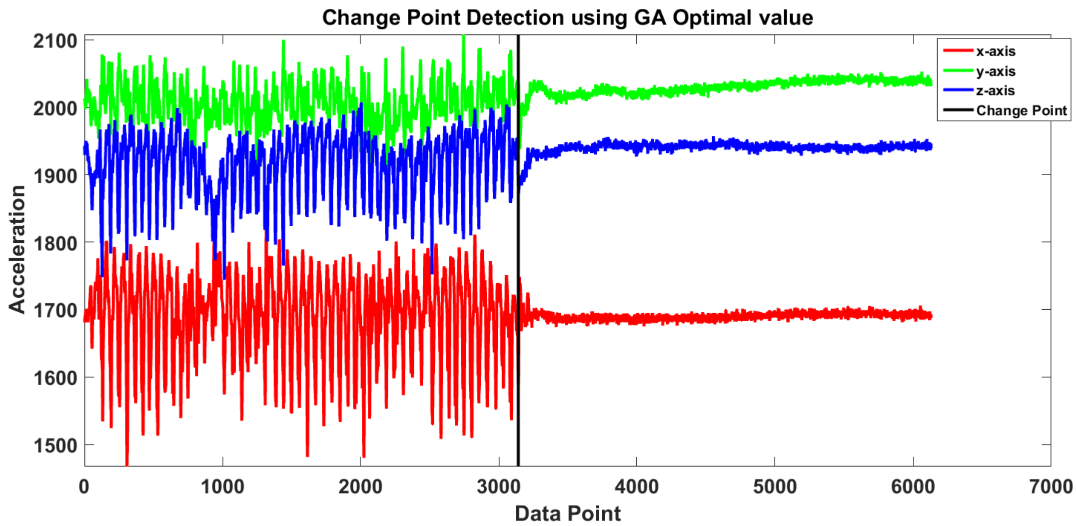
The results of our experiments on real and synthetic datasets for eight different activities are presented in Table 4.4 and 4.5. The proposed technique optimized the MEWMA parameters to find the optimal set of parameters for accurate change detection in each activity as presented in Table 4.4 and 4.5.

As discussed earlier accuracy and F- measure metrics were used for evaluation of optimal parameter selection for the MEWMA algorithm. The experimental results on the real dataset for best F-measure with optimal parameter set of $\lambda$ 0.5 & 0.7, significance value $p$=0.05 and window sizes (0.5s, 1s and 2s) is achieved for each activity as shown in Table 4.4.

The highest F-measure achieved is 66.7% and 50% for optimal parameter set using GA in all activities from the real dataset. The activities which initially start with static activities such as sit and stand achieved highest F-measure of 66.7% with optimal parameter set of $\lambda$ =0.5, significance value $p$=0.05 and window size 0.5s & 1s. The activities which start with dynamic behaviour such as walk downstairs, walk upstairs or walk corridor achieved the highest F-measure of 50% with an optimal parameter set of $\lambda$=0.7, significance value $p$=0.05 and

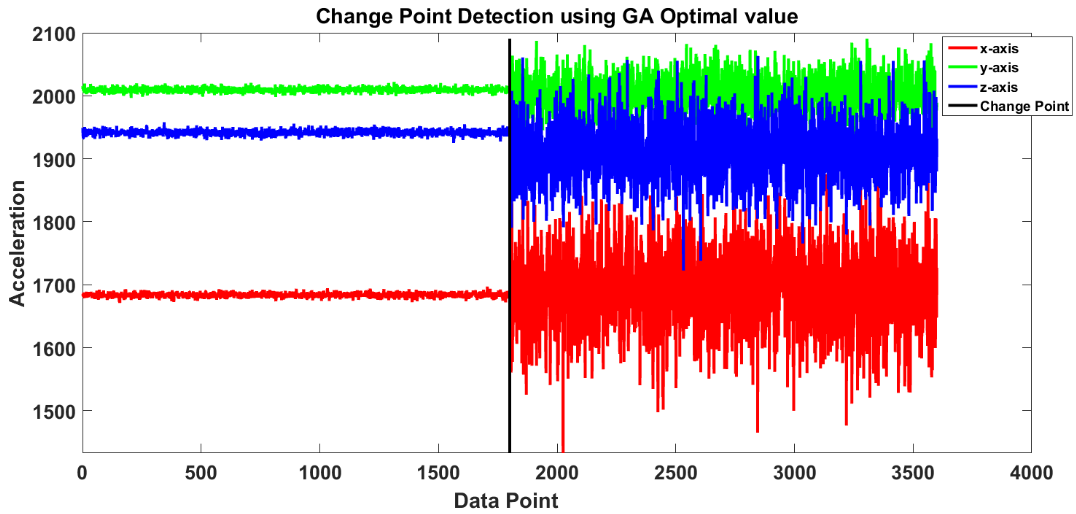



(a) Stand to Walk Downstairs

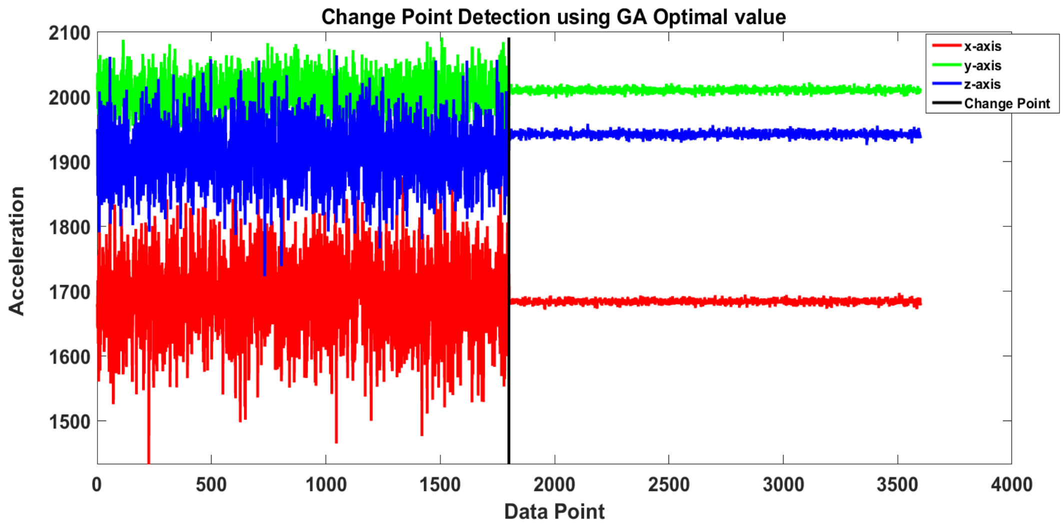



(b) Walk Downstairs to Stand

**Figure 4.9:** (a and b): Real dataset example of sliding window change detection results for the activity 'stand still to walk downstairs' and 'walk downstairs to stand still'




(a) Stand to Walk Downstairs




(b) Walk Downstairs to Stand

**Figure 4.10:** (a and b): Synthetic dataset example of sliding window change detection results for the activity 'stand still – walk downstairs' and 'walk downstairs to stand still'

window size 2s. The accuracy achieved for optimal parameter set with GA range from 99.4% to 99.8% as shown in Table 4.4.

**Table 4.4:** Non-Optimized and Optimised with GA parameter set for 8 different activities on Real dataset

| Activity | Sig Value | Non-Optimized | | | Optimized with GA | | |
|---|---|---|---|---|---|---|---|
| | | $\lambda$ | Win Size | F-Measure & (Accuracy) % | $\lambda$ | Win Size | F-Measure & (Accuracy) % |
| Sit to Stand | 0.05 | 0.3 | 1s | 50(99.4) | 0.5 | 0.5s | 66.7(99.8) |
| Stand to Sit | | | 1s | 50(99.4) | 0.5 | 0.5s | 66.7(99.8) |
| Stand to Walk Corridor | | | 1.5s | 40(98.5) | 0.5 | 1s | 66.7(99.8) |
| Stand to Walk Downstairs | | | 2s | 50(99.4) | 0.5 | 1s | 66.7(99.8) |
| Stand to walk Upstairs | | | 1.5s | 40(98.5) | 0.5 | 1s | 66.7(99.8) |
| Walk Downstairs to Stand | | | 2s | 40(98.5) | 0.7 | 2s | 50(99.4) |
| Walk Corridor to Stand | | | 2.5s | 40(98.5) | 0.7 | 2s | 50(99.4) |
| walk Upstairs to Stand | | | 1.5s | 40(98.5) | 0.7 | 2s | 50(99.4) |

The experiments were also performed on a real dataset without using an optimization technique as presented in Table 4.4. The maximum F-measure achieved was in the range of 40% to 50% for all activities. The activities with static activity initially achieved a maximum F-measure of circa 50% whereas dynamic activities initially achieved 40%. The accuracy achieved for the non-optimized results ranged from 98.5% to 99.4%. The evaluation of the results shows that the F-measure was higher about 50% to 66.7% for optimal set using GA than the 40% to 50% for non-optimized results. Moreover, as presented in Table 4.3, the accuracy is also improved from 99.4% to 99.8% with optimization when compared with the non-optimized accuracy of 98.5% as opposed to 99.4%.

We also used a synthetic dataset for our experiments; results are presented in Table 4.5. For synthetic data, we repeatedly generate randomized vectors 10 times for each activity. The experiments were performed on each random vector (10 times) for each activity and calculated the average F-measure. The achieved repetitions are very useful and have a good basis for evaluation of synthetic data. The highest average F-measure achieved for activities starting with a static activity such as sit and stand was in the range of 88.34% to 96.76% with optimal parameter set $\lambda$=0.4, significance value $p$=0.05 and window sizes 0.5s. The activities which started with a dynamic activity achieved the highest F-measure; 80.5% to 86.67% with optimal parameter set $\lambda$=0.6, significance value $p$=0.05 and window sizes 0.1s as shown in Table 4.5. In contrast to other results, the F-measure for synthetic dataset is high because of taking the average of the 10 repeated experiments for each activity.

**Table 4.5:** Optimal parameter set for 8 different activities using GA on Synthetic Dataset (Repeat 10 Times)

| Activity | $\lambda$ | Win Size | Sig Value | Average Mean ($\mu$) | Average Covariance ($COV$) | Average F-Measure % |
|---|---|---|---|---|---|---|
| Sit to Stand | 0.4 | 0.5s | 0.05 | 1843.75 | 15.71 | 96.67 |
| Stand to Sit | 0.4 | 0.5s | | 1872.30 | 617.87 | 93.34 |
| Stand to Walk Corridor | 0.4 | 0.5s | | 1869.26 | 1481.92 | 91.67 |
| Stand to Walk Downstairs | 0.4 | 0.5s | | 1870.34 | 2358.89 | 90.67 |
| Stand to walk Upstairs | 0.4 | 0.5s | | 1863.51 | 2267.31 | 88.34 |
| Walk Downstairs to Stand | 0.6 | 1s | | 1871.322 | 1488.38 | 86.67 |
| Walk Corridor to Stand | 0.6 | 1s | | 1870.787 | 1049.896 | 85.13 |
| walk Upstairs to Stand | 0.6 | 1s | | 1867.908 | 1442.591 | 80.50 |

The results indicate that the GA optimized algorithm provides an improvement, however, when we take out the inter-activity transition period and simulate data on this basis, the improvement is much more significant.

The decline of the F-measure is also partly due to the imbalance class problem (Galar et al., 2012) in our dataset. This problem happens when the total number of a class data (positive) is less than then the total size of other classes of data (negative). This highlights the skewed distribution of classes within the dataset and identifies that the minority class is the class of interest. In our dataset, we have only one true positive (TP) (which represents a correctly identified change point) and a high number of true negatives (TN) (the non-transitional points which are not labelled as change). We used the F-measure for evaluation because it is a combination of precision and recall, as presented in Equation. 3.12. Likewise, the precision is the ratio of TP over the total number of TP and false positives (FP) (the non-transition point which the algorithm is highlighted as a change) so that the one or two FP detections reduced the F-measure to 66.7% and 50% due to the imbalanced class problem in our real dataset.

## 4.5 Chapter Summary

The genetic algorithm is used to identify the optimal set of parameters for the MEWMA approach and automatically detect change points corresponding to different transitions in the user activities. The different parameters of the MEWMA are analyzed and evaluated to identify the optimal set of parameters for each activity using the GA. The optimal set of parameters selected using the GA outperformed on real world accelerometer data in terms of the accuracy and the F-measure. In this study the automatic optimization of the optimal parameter set was considered within the context of activity monitoring. Moreover, the MEWMA is a lightweight algorithm and can be incorporated into real world systems such as mobile-based applications for the collection and active sampling of labeled data.The change points in the data can be used to identify changes in activities and recognize and monitor good behaviour such as healthy exercise patterns based on these activities. The limitations of this work are the datasets that have the class imbalance problem and proper selection of lambda value for a specific activity. As the parameter set is dependent on activities so a parameter set can be generalized for a specific set of activities by clustering activities.Another limitation is using the same lambda value across all variants.

# Chapter 5

# Evaluation Framework to Analyze Different Multivariate Approaches and Optimization Techniques

## 5.1 Introduction

Within this chapter, multivariate approaches have been used to analyze and evaluate multivariate data for automatic change point detection. In multivariate data analysis, multiple characteristics of a system are evaluated simultaneously and the relationship among these characteristics are identified. Chapter 4 has provided detailed information about the genetic algorithm that was used to automatically identify an optimal parameters set, using a fitness function for MEWMA, including parameters such as the forgetting parameter $\lambda$, the window size, and significance value for each activity so as to maximize the fitness function. Chapter 5 provides detailed information about Multivariate Cumulative Sum Control Chart (MCUSUM) to automatically detect change points in user activities. Also, the Particle Swarm Optimization (PSO) is discussed in detail and used to identify optimal parameter set for MCUSUM and MEWMA for accurate change point detection. The MCUSUM is also used as a benchmark to our proposed technique MEWMA.

Moreover, MEWMA and MCUSUM approaches are used with GA and PSO to automatically identify an optimal parameter set using different parameters for

MEWMA and MCUSUM, so as to maximize the objective function namely the F-measure. The evaluation is performed using different metric measures and the experimental results shows that the proposed approach MEWMA performs better than the benchmark approach MCUSUM.

## 5.2 Cumulative Sum Control Chart (CUSUM)

This chapter introduces the traditional change detection used in process control and also discusses the control charts which are used to study process changes over time and determine whether the process is in control statistical state. Therefore, the univariate and multivariate CUSUM control chart are discussed in detail in this section for change detection in multivariate data.

Statistical process control (SPC) is the collection of procedures and methods for identifying specific causes of variations and monitoring the process behaviour to take in control of a target value. The main objective of SPC is to detect a change in the process mean as soon as possible after it has happened (Golosnoy et al., 2009). The control chart is the most important tool of SPC and developed by Walter Shewart in the early 1920s. The control chart contain the recorded data and helps in identifying any unusual event happens in the data such as a very low or high observation happened compared with the "typical" process performance (Golosnoy et al., 2009).

### 5.2.1 Control Chart

The statistical basis of a control chart typically consists of a centre line that represents the target value or the average value. The two horizontal lines are identified as the lower control limit (LCL) and upper control limit (UCL). The selected control limits are used to analyse if the process is in control. The process will be considered to be in control if the points falls within the control limits and therefore no action will be required. However, if a point falls outside of the control limits then the process is out of control and appropriate action is required to inquire and eliminate the causes liable for such practice (Hongcheng, 2007). An example of a typical control chart is presented in Figure 5.1, which comprises the graphical display of a quality characteristic that has been computed from a sample versus the sample number.

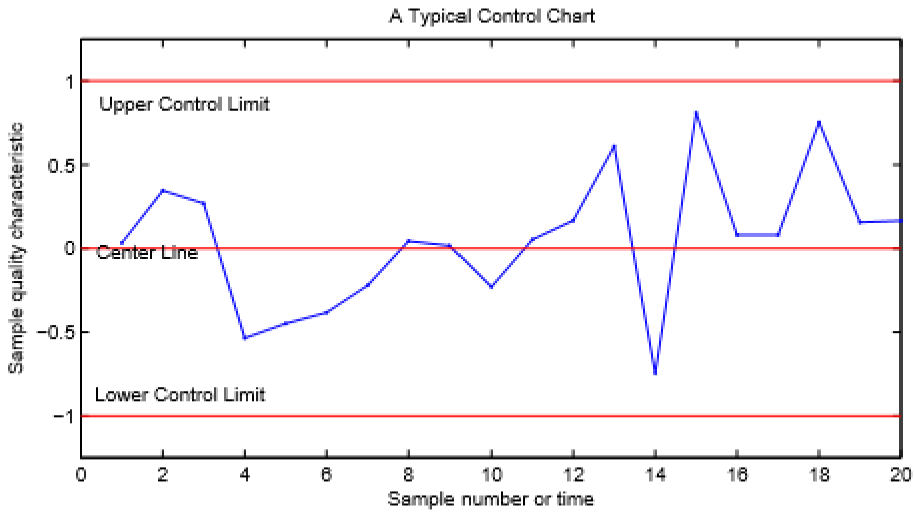



**Figure 5.1:** Control Chart (Hongcheng, 2007)

A number of control charts such as the X and S chart (Chen and Cheng, 1998), EWMA (Lucas and Saccucci, 1990), CUSUM (Cho et al., 2016) and Shewart individuals control chart (Kuncheva, 2009) have been used for various data structure such as variable or attribute data types (Bodnar and Schmid, 2007) that were subsequently developed to use for univariate data to monitor the centre and the variability of a process (Bodnar and Schmid, 2007). However, the CUSUM control chart is very efficient and has been widely used in the literature (Taylor, 2008), (Cho et al., 2016), (De Oca et al., 2010) and (Koepcke and Kretzberg, 2013) for detecting a small shift of the process mean.

### 5.2.2 Cumulative Sum (CUSUM) Chart

The CUSUM was initially developed by (Page, 1954) with the aim of detecting persistent shifts in the data stream of a process. In CUSUM, each plotted data point signifies the algebraic sum of the earlier data points and identifies recent deviation from the target. The CUSUM charts have received attraction earlier for detecting small but persistent shifts in the data. In Univariate CUSUM, the data are recorded individually at regular intervals and the results are based on the uncertain deterioration in the output after each observation. The univariate CUSUM can be defined using Equation 5.1.

$$\begin{aligned} \mathbf{C}_i &= \max\{\mathbf{C_{i-1}} + \mathbf{X}_i - \boldsymbol{\mu},\ 0\}, \quad i \geq 1 \\ \mathbf{C}_0 &= 0 \end{aligned} \tag{5.1}$$

where $\mathbf{C}_i$ is the $i^{th}$ CUSUM vector, $\mathbf{X}_i$ is the $i^{th}$ input observation vector, $i = 1, 2, 3, ..., n$ and $\mu$ is the mean of the input vector. Initially CUSUM starts with $\mathbf{C}_0 = 0$ and then sequentially calculates the CUSUM vector. The example of univariate CUSUM is represented in Figure 5.2. The x-axis represents the number of input data and the y- axis represents the CUSUM values. In the process, the observation after time 60 has shifted upward as the CUSUM values shown in the Figure 5.2.

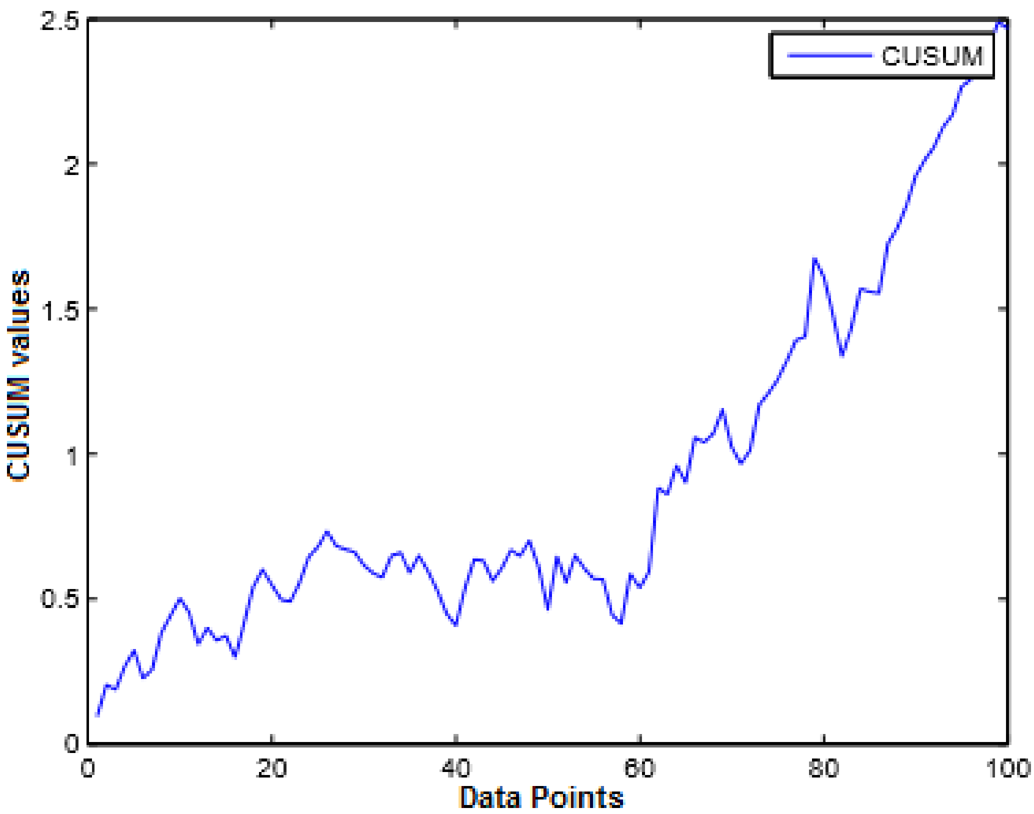



**Figure 5.2:** Univariate CUSUM

The limitation of univariate CUSUM is that it only analyzes one variable and also did not present much information about the data. Unlike Univariate data, the multivariate data involves more than one variable and presents more information about the data. Also, the analysis of multivariate data is complex to evaluate than univariate data. Therefore, an extension of CUSUM is Multivariate Cumulative SUM (MCUSUM) which is used to simultaneously monitor two or more related process of the input observation. The following section give details of the MCUSUM.

### 5.2.3 Multivariate Cumulative Sum (MCUSUM) Chart change point detection algorithm

The cumulative sum control chart is often used when small changes is more important in the data. The multivariate data processing is more complicated than univariate due to the simultaneous multidimensional data processing and evaluation. The Multivariate Cumulative SUM Control Chart (MCUSUM) is a statistical method that is used to simultaneously monitor two or more related process of the input observations to find the smaller and persistent shifts in the process data. The MCUSUM approach proposed by (Crosier, 1988) replaced the scalar quantity of univariate cumulative sum into vectors. The MCUSUM can be defined using Equation 5.2 and 5.3.

$$C_i = \sqrt{(\mathbf{S}_{i-1} + \mathbf{X}_i - \boldsymbol{\mu})' \boldsymbol{\Sigma}_{\mathbf{S}_i}^{-1} (\mathbf{S}_{i-1} + \mathbf{X}_i - \boldsymbol{\mu})} \tag{5.2}$$

$$\mathbf{S}_i = \begin{cases} 0 & if \;\; C_i \leq k \\ (\mathbf{S}_{i-1} + \mathbf{X}_i - \boldsymbol{\mu})(1 - k/C_i) & if \;\; C_i > k \end{cases} \tag{5.3}$$

where $\mathbf{X}_i$ is the input vector of $p$-dimensional set of observations for $i = 1, \ldots, p$ and $\boldsymbol{\mu}$ is the target vector represents the mean of the input observations while $k(> 0)$ is the reference value and optimal value for $k$ is 0.5 (Crosier, 1988), which is used for tuning a specific shift (Hongcheng, 2007) (Hameed et al., 2016). The $C_i$ is the generalized length of the CUSUM vector. Initially, MSUCUM starts with $\mathbf{S}_0 = 0$ and then sequentially calculates the MCUSUM vector. The $\Sigma$ is the covariance matrix of the input observations and $\mathbf{S}_i$ is the multivariate CUSUM vectors. The MCUSUM out of control vector is calculated using Equation 5.4.

$$\mathbf{Y}_i = \sqrt{\mathbf{S}_i' \mathbf{\Sigma}_{\mathbf{S}_i}^{-1} \mathbf{S}_i} < h \tag{5.4}$$

where $\mathbf{S}_i$ is the MCUSUM vector and $\mathbf{S}_i^{'}$ is its transpose. $\mathbf{\Sigma}_i$ is the covariance matrix of $\mathbf{S}_i$ and $h(> 0)$ , is chosen to achieve specified in-control signal. The Figure 5.3 represents the $x, y$ and $z$ input observations of the nine different activities collected using 3-axis accelerometer sensor. The details about the data collection is discussed later in section 5.6.1.

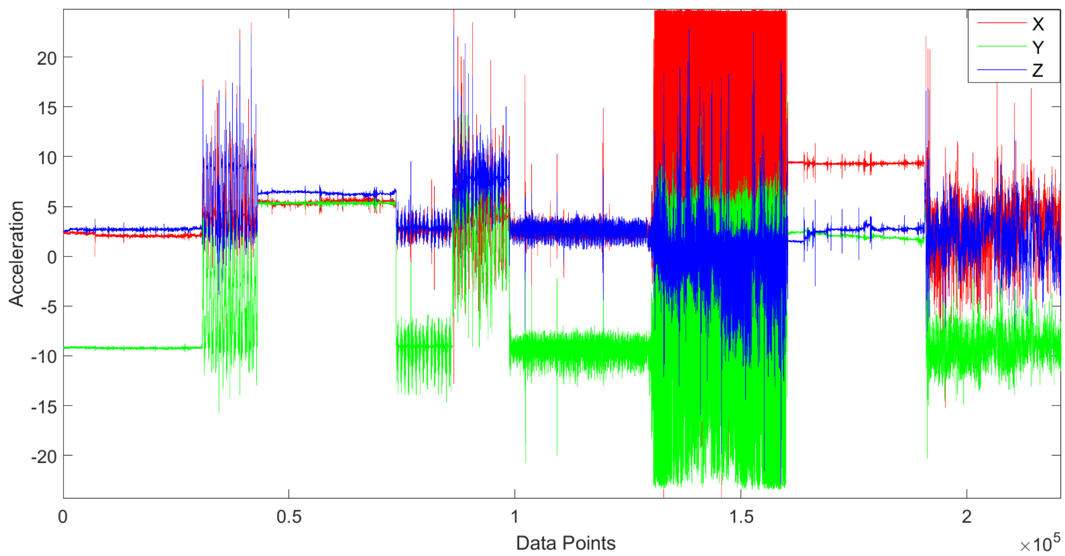



**Figure 5.3:** Accelerometer Signal

However, for clear understanding and visualization, two activities have been extracted from Figure 5.3. The two activities stand to walk corridor is shown in Figure 5.4 where the x-axis represents the number of data points of the input observation and the y-axis shows the acceleration. The activity has been changed from stand to walk corridor at approximately just after data point at 1900 as activity change from one activity to another as shown in Figure 5.4 which identify the accurate change point in the data.

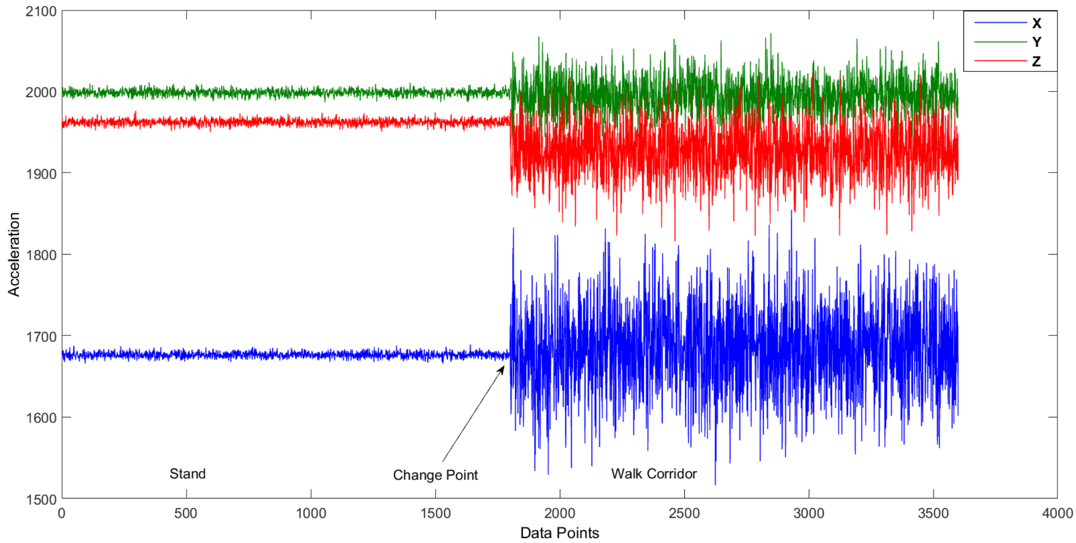



**Figure 5.4:** Stand to Walk Corridor

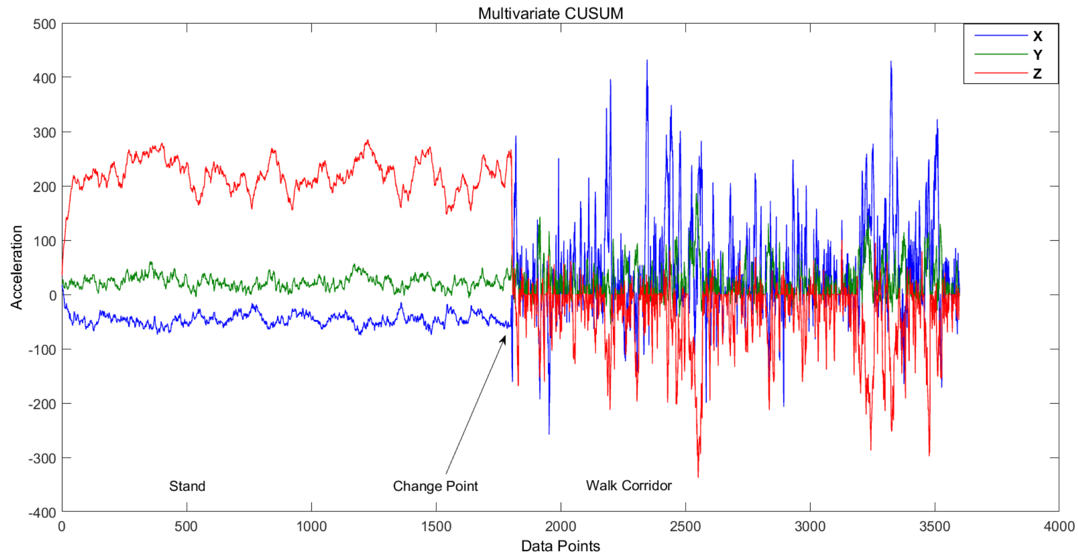



**Figure 5.5:** MCUSUM Vectors for Stand to Walk Corridor activity

Further, the example of Multivariate CUSUM is represented in Figure 5.5. The $x, y$ and $z$ represents MCUSUM vectors of the input observations for the stand to walk corridor activity of an accelerometer signal. The x-axis represents the number of data points of the input observation and the y-axis shows the acceleration. In the process, the mean of the observations has been changed at approximately the 1900 data point as shown in Figure 5.5 which can be identified as the change point in the data from stand to walk corridor.

## 5.3 Particle Swarm Optimization (PSO)

This section evaluates the effectiveness of the optimization approach which is used to find the minimum or maximum value of a function. The PSO is a relatively recent heuristic search algorithm (Hassan et al., 2004) used for solving optimization problem. The Particle Swarm Optimization (PSO) is a population based stochastic optimization technique inspired by the swarming or social behaviour of bird flocking or fish schooling. A "swarm" can be defined as a chaotic collection of moving individuals in the population that lean to form cluster where each individual seems to be moving in a random direction.

Particle Swarm Optimization (PSO) was first introduced in 1995 by James Kennedy and Russell C. Eberhart (Eberhart and Kennedy, 1995) with the aim to analyse the search space of a problem and also to find the required parameters setting to minimize or maximize a particular objective or fitness function. In the field of evolutionary computation, the concept of swarm intelligence was originated from the observation of swarming habits of certain types of animals such as bird and fish (Blondin, 2009).
The algorithm is initialized with a population of random solutions in order to find and search the optima by updating generations. In PSO, each particle represents a solution and the population of solutions is called a swarm of particles. Each particle keeps track of its coordinates in problem space which is associated with the best possible solution achieved so far and is called personal best (pbest). Moreover, another best value is also tracked which is obtained in the neighbours of the particle and is called local best (lbest). Once each particle takes all the population as its topological neighbours, the best value is a global best and is called (gbest). The best position of the particle is selected by calculating the velocity. Once a new position is reached, the best position of each particle and the best position of the swarm are updated as needed. The velocity of each particle

is then adjusted based on the experiences of the particle (Hu et al., 2004). The PSO have similar functionality to the GA. The PSO algorithm is initialized with a population of random solutions in order to find and search the optima by updating generations. However, Unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions are called particles which fly through the problem space by following the current optimum particles.

The PSO consists of two main operators: position update and velocity update. During the execution of PSO algorithm, in each generation, every particle attempts to accelerate towards the particles of the previous best position and global best position. In each iteration, based on the current velocity of the particle, a new velocity value is calculated for each particle which updates the velocity for the current particle. The updated velocity is then further used to calculate the new position of each particle in the search space. This process is repeated for a given number of times till the end of the process calculated by the number of iterations in the process.

Equation 5.5 and Equation 5.6 are used respectively in PSO to calculate the updated velocity ($v$) and position ($x$) of the new particle as the process moves on.

$$v_{i,j}^{k+1} = v_{i,j}^{k} + c_1 r_1 (xbest_{i,j}^{k} - x_{i,j}^{k}) + c_2 r_2 (xgbest_{j}^{k} - x_{i,j}^{k}) \tag{5.5}$$

$$x_{i,j}^{k+1} = x_{i,j}^{k} + v_{i,j}^{k+1} \tag{5.6}$$

Where $v_{i,j}^{k+1}$ and $x_{i,j}^{k+1}$ represents the $j^{th}$ element of the $i^{th}$ particle's velocity and position vector respectively at the $k+1^{th}$ iteration . Moreover,$r_1$ and $r_2$ are the random numbers which are uniformly distributed and in the range of $(0, 1)$. The $xbest$ and $xgbest$ represents the best positions in the $i^{th}$ particles. $c_1$ and $c_2$ are the learning factors and represent the particle confidence in its cognition. The values for the two parameters are set to 2 as presented initially by (Eberhart and

Kennedy, 1995). The reason of both parameters having the same value is to keep balance between the local and global best position convergence because a high value of $c_1$ or $c_2$ will encourage the faster convergence towards specific position or direction (Kaveh, 2014).

Figure 5.6 indicates the schematic movement of a particle for updated velocity and position of a new particle using Equation 5.5 and Equation 5.6 respectively.

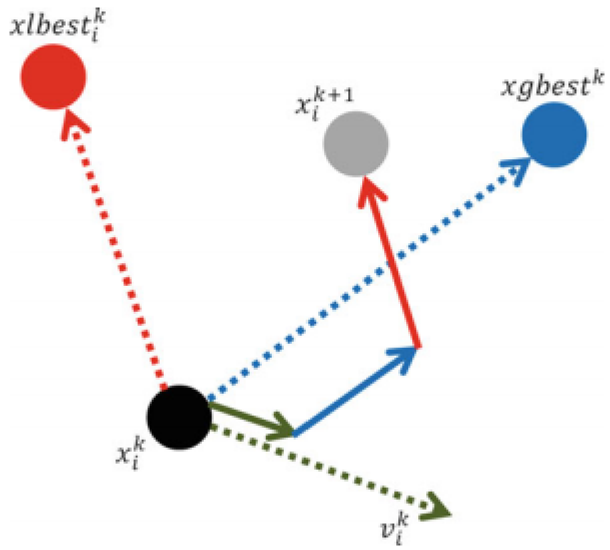



**Figure 5.6:** Movement of velocity and position updates in PSO algorithm (Kaveh, 2014)

The PSO algorithm maintains several candidate solutions simultaneously in the search space. In each iteration, the objective function is used to evaluate the candidate solutions and optimize its fitness value. Also, each candidate solution is represented as a particle and can be used to find minimum or maximum of the objective function.The pseudo code of PSO is illustrated in Figure 5.7.

```
Begin /* PSO Algorithm */

For each particle
   Initialize particle
END

Do
   For each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pBest) in history
         set current value as the new pBest
   End

   Choose the particle with the best fitness value of all the particles as the gBest
   For each particle
      Calculate particle velocity according Equation (5.5)
      Update particle position according Equation (5.6)
   End
While maximum iterations or minimum error criteria is not attained
```

**Figure 5.7:** Basic PSO Pseudo code

In the initial step, PSO randomly selects the candidate solutions in the search space. As shown in Figure 5.8, The PSO initially choose four particles as candidate solutions and try to find the global maximum in the search space. The x-axis shows all the possible solutions while the curve represents the objective function in the search space. The PSO algorithm does not have information about the objective function and therefore does not know how far or near the candidate solution is to the local or global maximum. The objective function is used by PSO algorithm to evaluate the candidate solution based on its fitness value.

In PSO, each particle manages and keep record of its position, velocity and fitness value. In addition, it also maintains the best fitness value achieved during the algorithm process and is called the individual best position and the candidate solution that achieves this best fitness value is called the individual best candidate solution (Blondin, 2009).

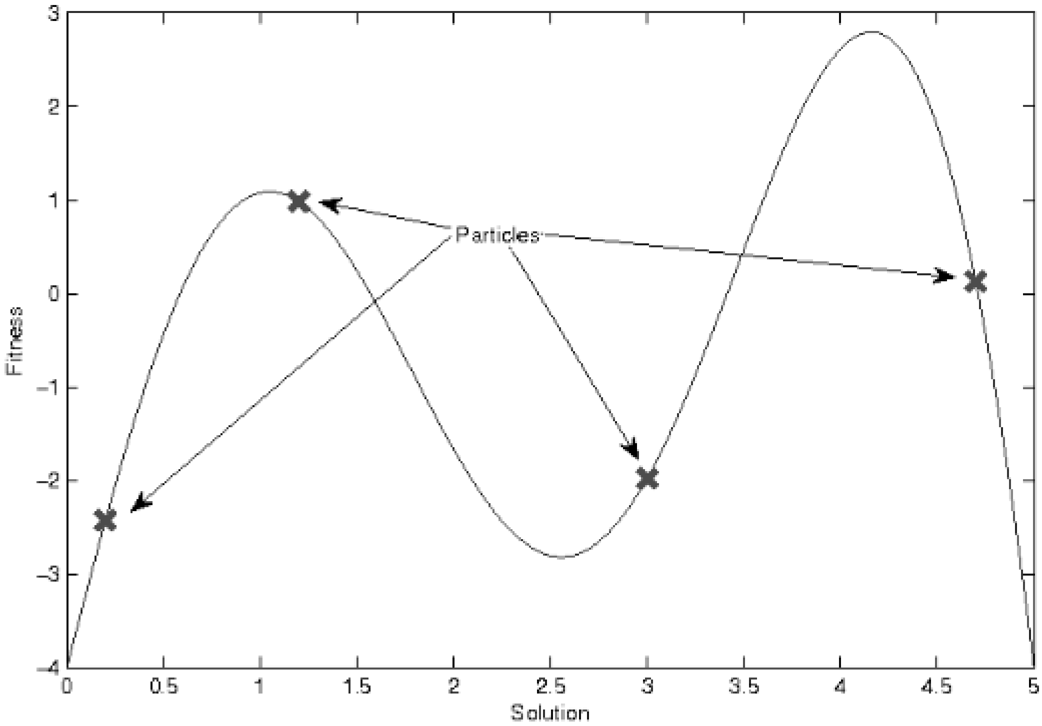



**Figure 5.8:** Initial PSO state (Blondin, 2009)

Likewise, the PSO algorithm among all possible particle in the swarm achieves the best fitness value which is known as global best fitness and the candidate solution that achieves this fitness is known as global best position or global best candidate solution (Blondin, 2009).

## 5.4 Change detection using MEWMA and MCUSUM in multivariate data

As discussed earlier that MEWMA and MCUSUM are used for change detection in multivariate data analysis. The details of MEWMA for change detection in multivariate data analysis has been explained and discussed in section 4.3. This section gives a detailed explanation about MCUSUM for change detection in multivariate analysis. we consider the data stream of length $q$ consisting of specific data points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \ldots \mathbf{X}_q$ e.g. for accelerometer value, $\mathbf{X}_i = (1.253, -9.382, 2.542)$ where the elements represent the $x, y$ and $z$ values of 3-dimensional accelerometer signal. In general, a sequence of data points $\mathbf{X}_i$ to $\mathbf{X}_q$ may contain different distributions. In particular, the two subsequence $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, ..., \mathbf{X}_{i-1}$ and $\mathbf{X}_i, \mathbf{X}_{i+1}, ..., \mathbf{X}_q$ may follow different distributions such as $D_1$ and $D_2$. The $D_1$ and $D_2$ can be the same or different. In each data stream, MCUSUM is used to calculate the Cumulative sum for each data point $\mathbf{X}_i$ to identify the position and detection of accurate change points in the data stream. In MCUSUM $\mathbf{S}_i$ is the MCUSUM vector calculated by using the multivariate input as shown in Equation 5.3 and the covariance matrix of $\mathbf{S}_i$ is calculated and represented by $\Sigma_i$ to find as shown in Equation 5.4.

In our experiments different window sizes (1s,2s,3s) are used to analyze the input data using sliding window with an increment of 1 data point to perform sequential analysis as disused earlier in section 3.3.3. The window sizes are used to evaluate the sequence of data points in the window. These window sizes are chosen to combine some historical data with the new data to identify any change. These are reasonable sizes that are taken from experimentation. Moreover, once $\mathbf{Y}_i$ for MCUSUM is calculated as shown in Equation 5.4, we consider a number of possible values $h$ (0.05, 0.025, 0.01 0.005) in order to evaluate the confidence of the entire window. The condition is verified if $\mathbf{Y}_i$ is greater than $h$, then $x_i$ will be labelled as a change point within the data stream otherwise not. The significance values are used in literature to define regions where the test statistics are unlikely to lie (Handbook, 2016).

## 5.5 Parameter optimization using GA and PSO

The GA has been discussed in detail for parameter optimization in section 4.2. Therefore, detailed information about PSO will be discussed in this section for optimal parameter selection.

The objective function in the GA and PSO is used to find the optimal solution to a system. In our case, each distinct combination of the three variables provides a single solution in the population, namely $\lambda_i$, the window size, and the significance for MEWMA and $k$, the window size, and the significance variable for MCUSUM. Over a number of generations, these solutions "evolve" towards the optimal solution.

Our objective function then tries to find the solution with the maximum F-measure value given a range of input values for both algorithms. The F-measure is used as the measure to find the overall effectiveness of the activity recognition or change detection by combining the precision and recall. The objective function for GA and PSO using MEWMA and MCUSUM can be defined as follows in Equation 5.7 and Equation 5.8 respectively.

$$F - measure_{max} = max_{(\lambda_i, win\_size, sig\_value)}(F - measure_{MEWMA}) \tag{5.7}$$

$$F - measure_{max} = max_{(k, win\_size, sig\_value)}(F - measure_{MCUSUM}) \tag{5.8}$$

Both algorithms MEWMA and MCUSUM use the three variables as input where window size ranges from 1s, 2s and 3s and significance values of 0.05, 0.01, 0.025, 0.005 are same for both algorithms. However, MEWMA used $\lambda_i$ ranges from 0.1 to 1 and MCUSUM used $k$=0.5 as a standard value presented in (Crosier, 1988) as shown in Equation 5.7 and Equation 5.8 respectively.

The objective function defined in Equation 5.7 and Equation 5.8 are initialized by upper and lower bounds of the three parameters to find the maximum F-measure with the optimal parameter set. After the exploration with different parameter settings, the optimal GA and PSO parameters, which maximize the fitness function of the F-measure, are shown in Table 4.1 and 5.1. The GA parameter has shown in Table 4.1 and discussed in detail in section 4.3 of chapter 4. The PSO parameters is shown in Table 5.1 as follows.

**Table 5.1:** Particle Swarm optimization (PSO) Parameters

| **Parameters** | **PSO** |
|---|---|
| Swarm Size | 50 |
| Initial Swarm | *pswcreationuniform* |
| HybridFcn | *fmincon* |
| Max Iterations | 100 |

The Matlab 2015b global optimization tool box (Matlab-Toolbox, 2015) was used for experiments and the PSO parameters are set according to our experimental setup as shown in Table 5.1.

Initially, the PSO creates particles at random with uniform distribution using *pswcreationuniform* function within the defined lower bound and upper bound given in Equation 5.7 and Equation 5.8. The Hybrid function is used to perform constrained or unconstrained minimization or maximization. In our experiments, we used *fmincon* function which provide constrained maximization for our objective function. The rest of the options *MaxStallIterations*, *MaxStallTime*, *ObjectiveLimit* etc are kept Matlab default for PSO, detailed information can be found on (Matlab-Toolbox, 2015).

The PSO process, illustrated in Figure 5.9 with respect to the PSO parameters proposed in Table 5.1, is described as follows (Chavan and Adgokar, 2015).

- Initialize the population size with the number 50, which specifies how many individuals there are in each of the iterations. Usually, the number 50 is used for a problem with five or fewer variables, and the number of 200 is used otherwise.
- Initialize swarm and each particle randomly with initial position and velocity with the search space.

- Calculate the maximum value of the objective function using Equation (5.8).
- Initially, the first objective values and positions are inevitably considered as personal best values and personal best positions. Further, the global best value and position are chosen based on the best fitness value among all particles and that the particle value and position are selected as global best value and position in the whole swarm population.
- If the stopping criteria becomes false, then the velocity and position of the particles are updated using Equation 5.5 and Equation 5.6 respectively.
- Finally, a new generation is updated and the PSO algorithm loops back to calculate the fitness value and updated position for each particle. The updated personal best value and position is compared to the previous personal best value and position. If the new fitness value is better than previous one then the personal best value and position are updated. The same process is carried out for updating the global best value and position.
- The process is continued till the termination condition is satisfied. The default value for the generations is 100 multiplied by the number of variables used, but we choose the best value for generation by experimentation with different values.

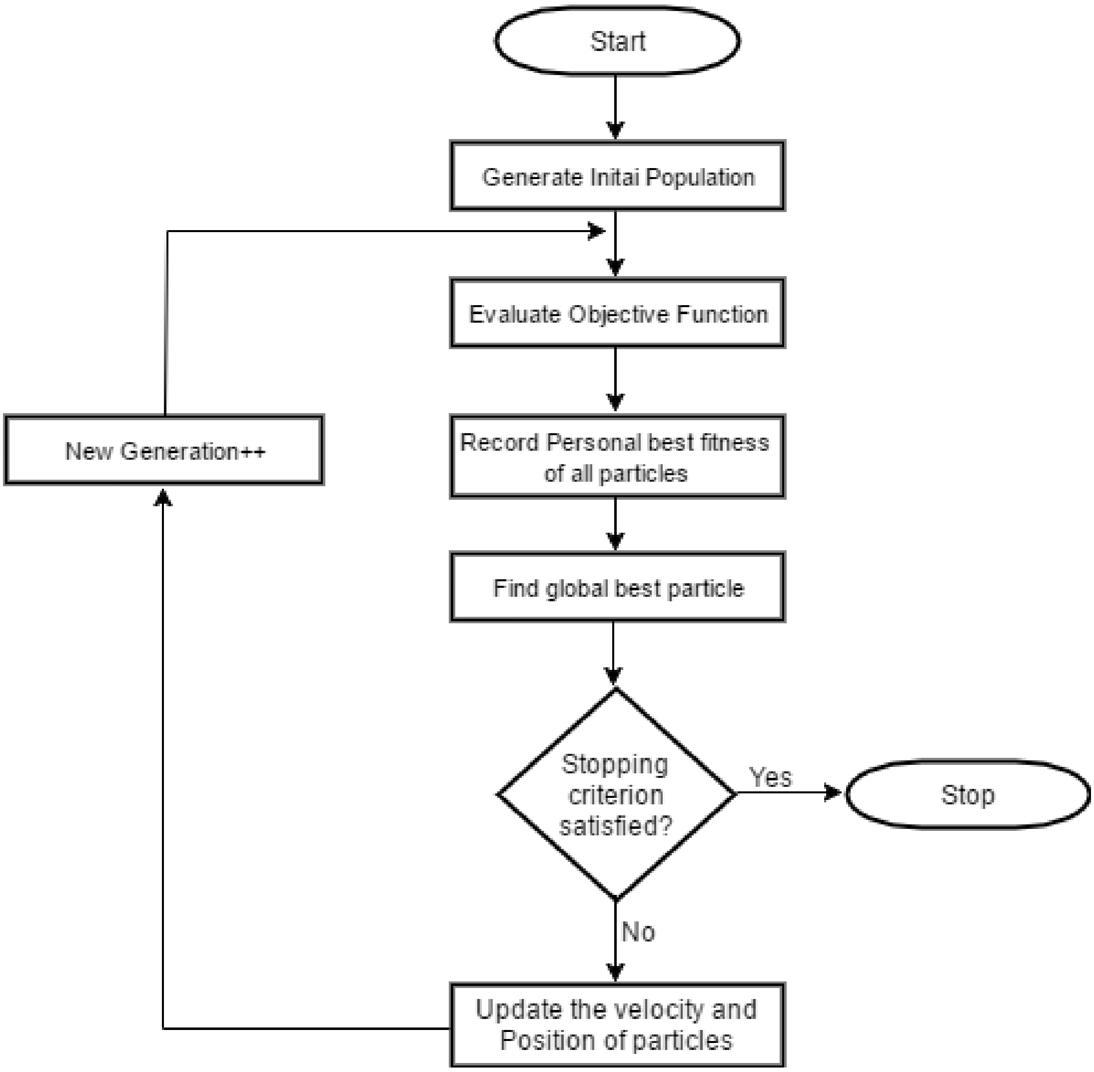



**Figure 5.9:** Flow chart of various stages to perform Particle Swarm Optimization (PSO)

## 5.6 Experimental Setup

This section evaluates the performance of two multivariate change point algorithms e.g. MEWMA & MCUSUM with optimization facet using GA & PSO to automatically identify optimal parameter set for accurate change point detection. The different combination like MEWMA with GA & PSO and MCUSUM with GA & PSO has been used for evaluating the performance of the change point detection algorithms. The evaluation is performed on a real dataset using different metric measures such as accuracy, precision, sensitivity, G-means and F-measures. As in this study, we used multivariate approaches to analyze and evaluate multivariate data for automatic change point detection. In multivariate data analysis, more than one characteristics of a system are evaluated simultaneously and the relation among these characteristics are identified. The MEWMA approach was proposed in section 3.3.2.1 which tunes the different parameters including lambda, window size and significance value with the aim to achieve better performance and accurate change point detection. MCUSUM, discussed in section 5.2.3, has also been implemented as a multivariate approach from literature to be used as a bench mark to our proposed technique MEWMA. Additionaly, the GA and PSO have been discussed in section 4.2 and 5.5,which are used to automatically identify optimal parameters set used for MEWMA and MCUSUM, so as to maximize the objective function i.e. the F-measure. The evaluation is performed using different metric measures and the experimental results show that the proposed scheme performs better than the benchmark scheme. The detailed explanation about the real dataset is given in the following section 5.6.1.

### 5.6.1 Real Dataset

The dataset used here was collected by (Patterson et al., 2017) from ten healthy participants using 3-axis accelerometer sensor in order to evaluate the change point detection algorithm. The participants consist of five females and five males wearing shimmer sensing platform (Burns et al., 2010) (Patterson et al., 2017) placed on their chest, right wrist ankle. The data for different activities was collected and captured with a sample frequency of 102.4 *Hz*. The nine various activities performed by each participant are presented in Table 5.2. The different activities were classified as static, transitional and dynamic. In static activities, the participant was asked to remain comfortably still such stand, sit while in transitional activities, the data captures the transition between two activities such as stand to walk, sit to lie. Moreover, the dynamic activities imply that the activity inherently contains meaningful human movements such as walking, running and vacuuming. The change points in the dataset were labelled manually based on the recorded time a participant was to change an activity. For each participant, the resultant dataset contains a continuous data stream of approximately 35 minutes activities carried out according to the sequence given in Table 5.2. The 95 labelled transitions recorded for each participant, which in total becomes 950 in total for 10 participants (Patterson et al., 2017). In the dataset, most of the transitions are from static to dynamic activities and vice versa. However, the dataset also contains transitions form dynamic to dynamic activity like waking to running. After the data collection, the activity execution of accelerometer data was wirelessly streamed to a received computer via Bluetooth communication protocol. We have used the dataset 2 in my paper entitle "Using genetic algorithms for optimal change point detection in activity monitoring" which is the part of Chapter 4. By the time, the dataset used in chapter 5 was not publicly available.

**Table 5.2:** The nine various activities performed by each participant

| Activity Seq. | Label | Type | Description |
|---|---|---|---|
| 1 | Standing | Static | Stand for 5 minutes (min) |
| 2 | Stand-sit | Transitional | Stand for 10 second (s), Sit for 10s (15 repetitions) |
| 3 | Sleeping | Static | Lie on sofa for 5 min |
| 4 | Stand-walk | Transitional | Stand for 10s, Walk for 20s (15 repetitions) |
| 5 | Sit-Lie | Transitional | Sit for 10s, Lie for 10s (15 repetitions) |
| 6 | Walking | Dynamic | Walk on treadmill at constant speed of 5 min |
| 7 | Running | Dynamic | Run on treadmill at constant speed of 5 min |
| 8 | Watching TV | Static | Sit on sofa for 5 min |
| 9 | Vacuum | Dynamic | Vacuum for 5 min |

## 5.7 Results and Discussion

The real dataset as described earlier has been used in the evaluation of MEWMA and MCUSUM approaches in change-point detection using GA and PSO to find the optimal parameter set. As we are evaluating the multivariate data, the

$x, y$ and $z$ acceleration magnitudes are captured and used as input to MEWMA and MCUSUM approaches. Initially, both approaches used different parameters including $\lambda$(0.1 to 1) for MEWMA and $k$=0.5 for MCUSUM, the significance values (0.05,0.025,0.01,0.005) and the window sizes (1s,2s,3s) to find accurate change point. Hence, GA and PSO was used to find and identify the optimal set of parameters for MEWMA and MCUSUM. The F-measure metrics were used as an objective function to analyze and evaluate the optimal change point in activities using GA and PSO. A detected changes point is considered true, if its index lies in the data stream,$l \ \epsilon \ z - f/4, ..., z + f/4$ where $z$ is the index of manually labelled change point and f is the sampling frequency in $Hz$. Moreover, the data with detected change points is sent periodically to the server which runs the GA and PSO for optimization as shown in Figure 5.10.

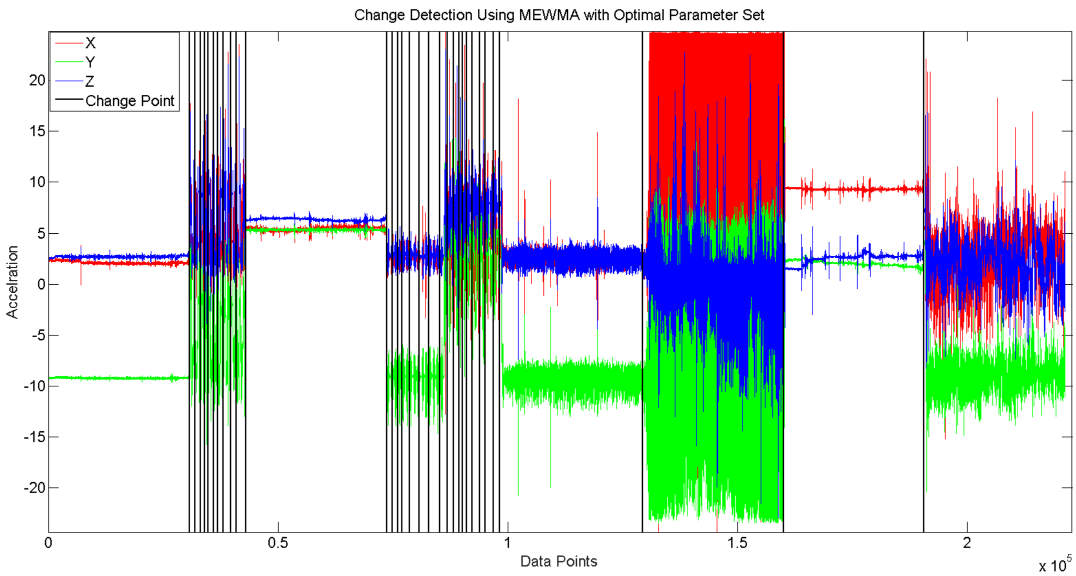



**Figure 5.11:** Real dataset example of change point detection using MEWMA for different activities. The x, y and z axis represent the MEWMA vectors of the input observation of the accelerometer signal while the vertical lines presents the change detection points detected by the MEWMA algorithm.

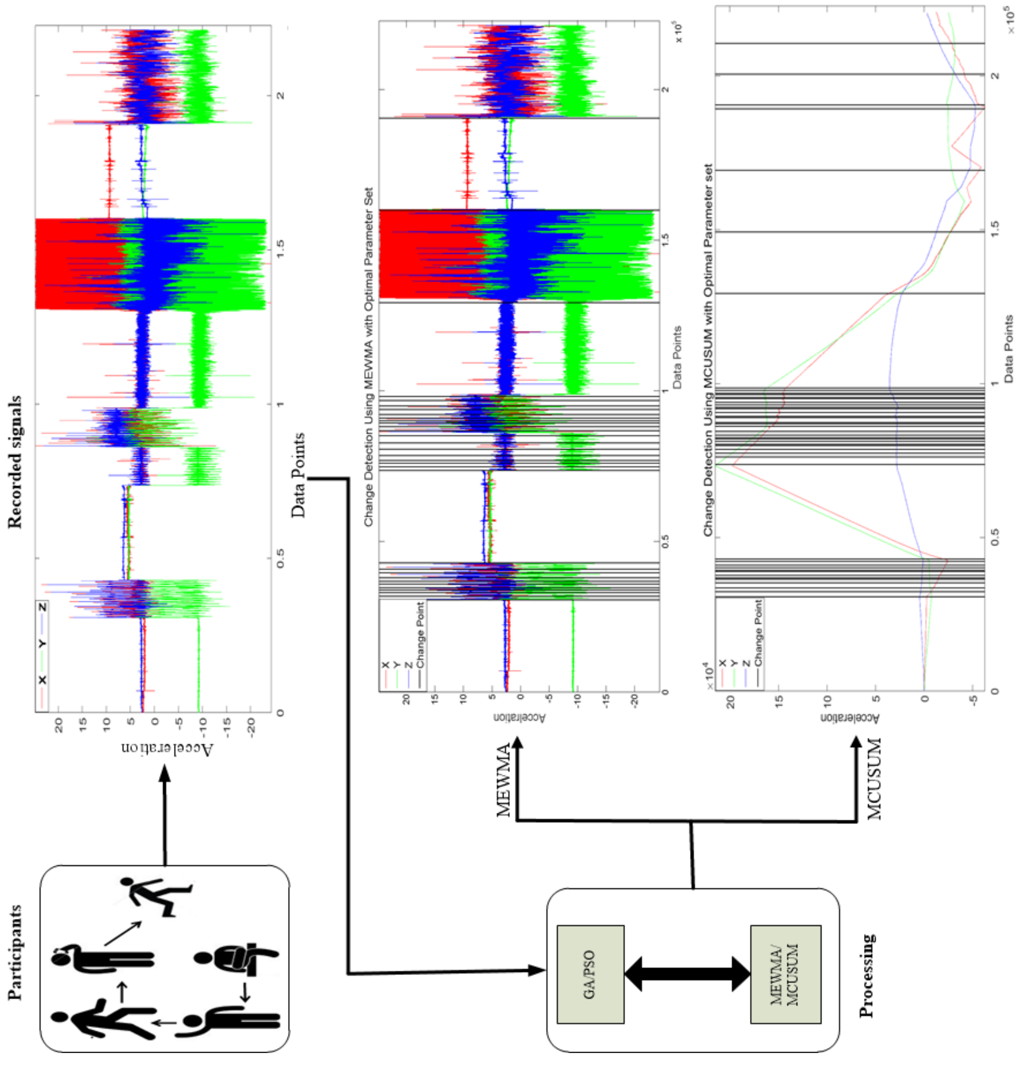



**Figure 5.10:** The System Model

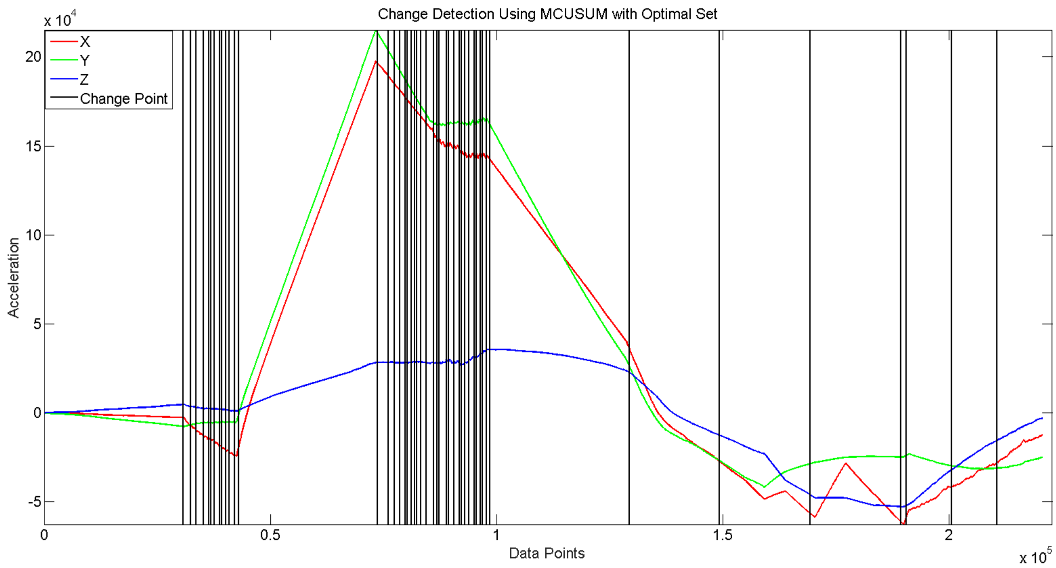



**Figure 5.12:** Real dataset example of change point detection using MCUSUM for different activities. The x, y and z axis represent the MCUSUM vectors of the input observation of the accelerometer signal while the vertical lines presents the change detection points detected by the MCUSUM algorithm.

The positive and negative detection is defined as true positive (TP), false positive (FP), true negative (TN) and false negative(FN) which have been explained in detail in section 3.5. The positive and negative detection about change points are classified for the purpose of different evaluation metrics measures. The real dataset example of change detection using MEWMA and MCUSUM for different activities are shown in Figure 5.11. and 5.12 respectively.

The accuracy, precision, specificity, sensitivity and G-means metrics were used for evaluation of optimal parameter selection for MEWMA and MCUSUM algorithm. The GA and PSO are used for optimal parameters selection for both change detection algorithms. The F-measure metrics were used as an objective function to analyze and evaluate the optimal change point in activity monitoring using GA and PSO. Moreover, the evaluation metrics such as accuracy, precision, sensitivity, G-Means and F-measure are discussed in detail in section 3.6. The limitation of this work is that we are using the same lambda value across all variants, however, there is a possibility of using a set of lambda values simultaneously (one for each variant) that could be referred as fully Multivariate approach, and which will be addressed in our future work.

### 5.7.1 Accuracy

The MEWMA with PSO achieved highest accuracies of 99.9%, 99.7% and 99.3% for window sizes (1s ,2s and 3s), $\lambda$ (0.5, 0.6 & 0.7) and $p$=0.05 for the optimal parameter set for 9 different activities. Correspondingly, the MEWMA with GA achieved highest accuracies of 99.7%, 99.5% and 99% for window sizes (1s ,2s & 3s), $\lambda$ (0.5, 0.6 & 0.7) and $p$=0.05 for the optimal parameter set of 9 different activities as shown in Figure 5.13.

The MCUSUM with PSO achieved highest accuracies 99.5%, 99.4% and 99% for window size (1s ,2s & 3s), $k$=0.5 and $p$=0.05 for the optimal parameter set of 9 different activities. Correspondingly, the MCUSUM with GA achieved highest accuracies of 99.3%, 99.2% and 98.8% for window size (1s ,2s & 3s), $k$=0.5 and $p$=0.05 for the optimal parameter set of 9 different activities as shown in Figure 5.14.

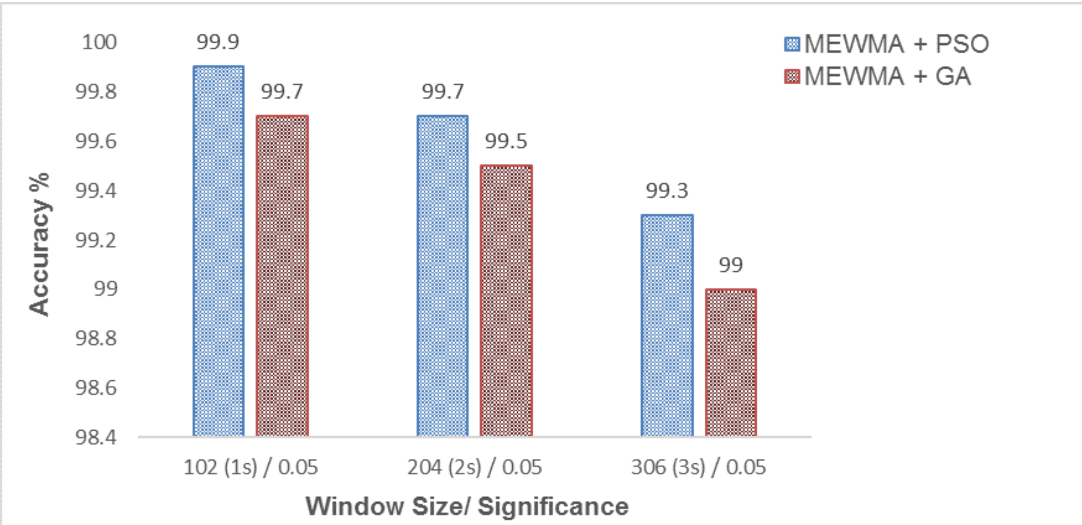



**Figure 5.13:** Comparison of Accuracy between MEWMA (PSO and GA)

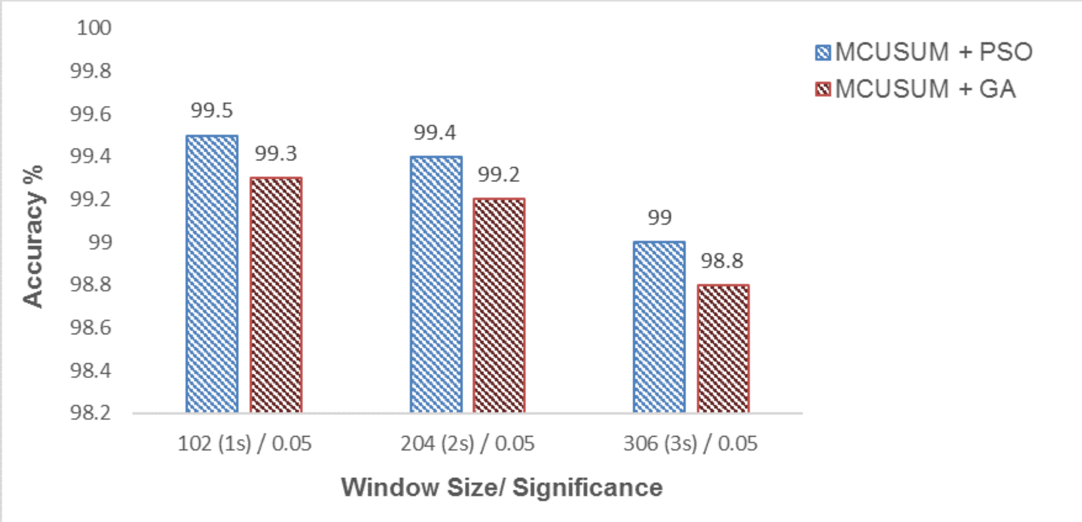



**Figure 5.14:** Comparison of Accuracy between MCUSUM (PSO and GA)

The accuracy is relatively high for both MEWMA (PSO & GA) and MCUSUM (PSO & GA) because of the relatively high disproportionate number of TNs in the data. The reason is the class imbalance problem (Galar et al., 2012) in our dataset discussed earlier in section 4.3.3. The MEWMA and MCUSUM with PSO achieved highest accuracy as compared to MEWMA and MCUSUM with GA.

A one-sided $t$-test is performed to find the statistical significance for the accuracy metric for 10 experiments repeatedly performed for each approach i.e. MEWMA with PSO and MCUSUM with PSO. The results of the one-sided $t$-test evaluate that the MEWMA with PSO is statistically significant by achieving the significance 0.0207 which is less than the standard $p$-value=0.05. Therefore, MEWMA with PSO outperformed than MCUSUM with PSO by achieving higher accuracy for accurate change point detection as shown in Figure 5.13 and 5.14 respectively.

### 5.7.2 Precision

The maximum precisions attained for MEWMA with PSO are 60.78%, 50% and 45.45% while for MEWMA with GA are 57.50%,48% and 43% for the optimal set of parameters using the same window sizes, lambda values and significance value as discussed in the above section. The precision of MEWMA (PSO & GA) is represented in Figure 5.15.

Likewise, the MCUSUM (PSO & GA) has achieved maximum precision of about 55%, 45.98%, 40% while for MCUSUM with GA is about 52%,43% ,38% for the same window sizes, $k$=0.5 and significance values for the optimal set of parameters as discussed earlier. The precision of MCUSUM with PSO and GA is represented in Figure 5.16.

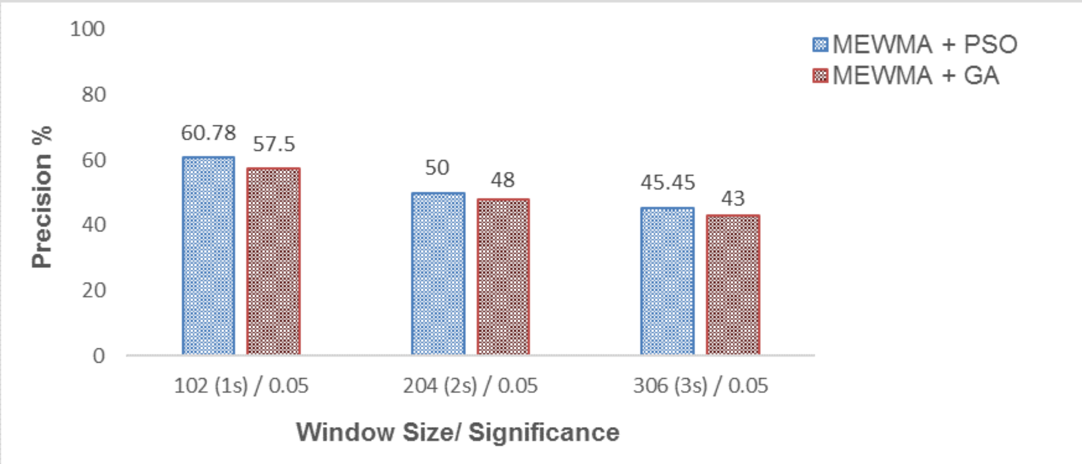



**Figure 5.15:** Comparison of Precision between MEWMA (PSO and GA)

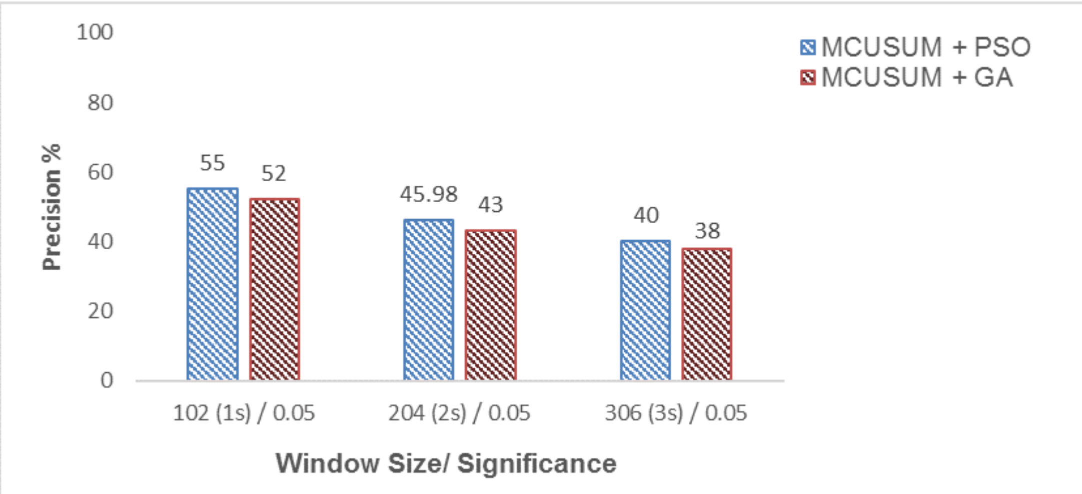



**Figure 5.16:** Comparison of Precision between MCUSUM (PSO and GA)

The higher precision is achieved for MEWMA (PSO & GA) than MCUSUM (PSO & GA) as shown in Figure 5.15 and 5.16. However, the MEWMA with PSO improved than MCUSUM with PSO approximately 5.60% for each window size for accurate change point detection using optimal parameter set. The reason for low precision is due to the high number of occurrences of false alarms as our algorithm is very sensitive and detects possible change points even if they are small. A one-sided $t$-test is performed to find the statistical significance for the precision metric for 10 experiments repeatedly performed for each approach i.e. MEWMA with PSO and MCUSUM with PSO. The results of the $t$-test evaluate that the MEWMA with PSO is statistically significant by achieving the significance 0.0388 which is less than the standard $p$-value.

### 5.7.3 Sensitivity

The maximum sensitivity values achieved by MEWMA with PSO are 65.26%, 35.79% and 25% while 60.5%, 31.50% and 23.50% for MEWMA with GA using the same optimal parameter set with window sizes (1s,2s,3s), $\lambda$ (0.5, 0.6 0.7) and $p$=0.05 as shown in Figure 5.17. The MEWMA with PSO has approximately 4.5% higher sensitive value on average for each window size than MEWMA with GA. Likewise, the highest sensitivity was achieved for MCUSUM with PSO is about 29.47%, 26.37% and 20% while 27.5%, 25% and 18.50% for MCUSUM with GA using optimal parameter set with window sizes (1s,2s,3s), $k$=0.5 and $p$=0.05 as shown in Figure 5.18.

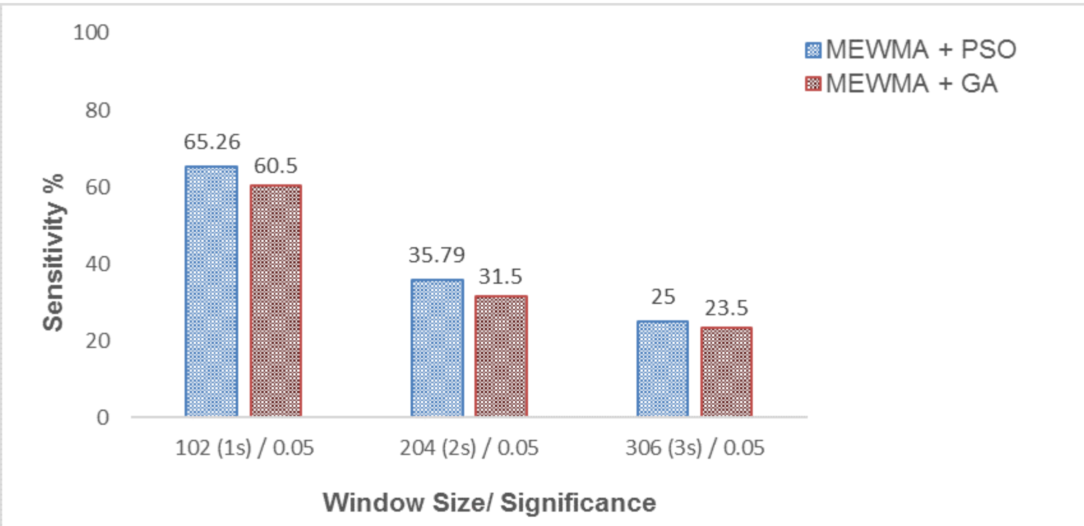



**Figure 5.17:** Comparison of Sensitivity between MEWMA (PSO and GA)

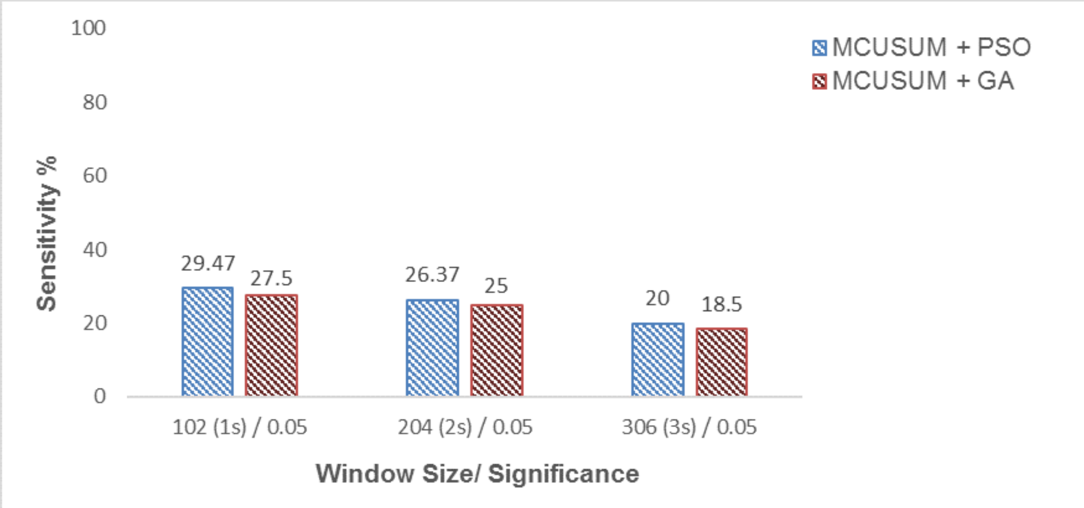



**Figure 5.18:** Comparison of Sensitivity between MCUSUM (PSO and GA)

The MCUSUM with PSO is improved approximately 1.5% on average for each window size than MCUSUM with GA.

However, the analysis of MEWMA with PSO results in about 35.79%, 9.42% and 5% higher sensitivity values in each window size respectively as compared to MCUSUM(PSO). Also, MEWMA(GA) is improved about 33%, 6.5% and 5% for each window size respectively compared to MCUSUM(GA) as shown in Figure 5.17 and 5.18. A one-sided $t$-test is performed to find the statistical significance for the sensitivity metric for 10 experiments repeatedly performed for each approach i.e. MEWMA with PSO and MCUSUM with PSO. The results of the $t$-test evaluate that the MEWMA with PSO is highly statistically significant by achieving the significance 0.0069 which is less than the standard $p$-value.

### 5.7.4 G-Means

The MEWMA with PSO achieved highest G-means is about 80.78%, 60.93% and 39.73% for window size (1s ,2s and 3s), $\lambda$ (0.5, 0.6 0.7) and $p$=0.05 for the optimal parameter set for 9 different activities. On the other hand, the MEWMA with GA achieved highest G-means is about 75.5%, 57.5% and 37% for window size (1s ,2s and 3s), $\lambda$ (0.5, 0.6 & 0.7) and $p$=0.05 for the optimal parameter set of 9 different activities as shown in Figure 5.19. The MEWMA with PSO is improved approximately 3% on average for each window size than MEWMA with GA.

Similarly, the MCUSUM with PSO achieved highest G-means for about 54.29%, 51.31% and 30% for window size (1s ,2s & 3s), $k$=0.5 and $p$=0.05 for the optimal parameter set of 9 different activities. On the other hand, the MCUSUM with GA achieved highest accuracy is about 52.5%, 48.5% and 27.5% for window size (1s ,2s and 3s), $k$=0.5 and $p$=0.05 for the optimal parameter set of 9 different activities as shown in Figure 5.20.

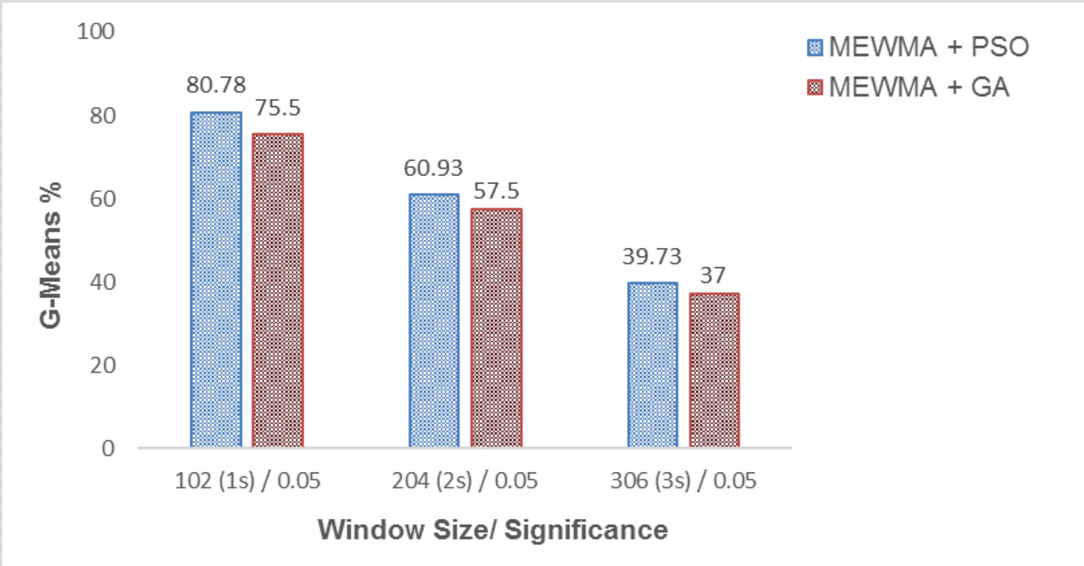



**Figure 5.19:** Comparison of G-Means between MEWMA (PSO and GA)

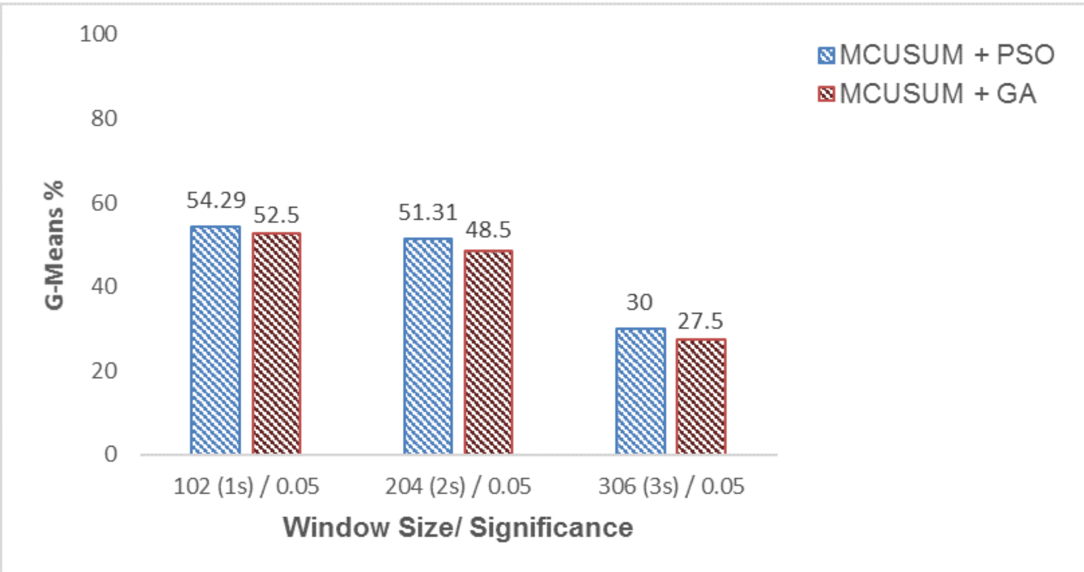



**Figure 5.20:** Comparison of G-Means between MCUSUM (PSO and GA)

The MCUSUM with PSO is improved approximately 2.5% on average for each window size compared with MCUSUM with GA.

However, the G-mean analysis of MEWMA (PSO & GA) and MCUSUM (PSO & GA) was improved comapred with MEWMA (PSO) with about 26.5%, 9.5% and 9.7% for each window size respectively as compared to MCUSUM(PSO). Also, MEWMA(GA) is improved about 23%, 9% and 9.5% for each window size respectively compared to MCUSUM(GA) as shown in Figure 5.19 and 5.20. A one-sided $t$-test is performed to find the statistical significance for the accuracy metric for 10 experiments repeatedly performed for each approach i.e. MEWMA with PSO and MCUSUM with PSO. The results of the $t$-test evaluate that the MEWMA with PSO is statistically significant by achieving the significance 0.0431 which is less than the standard $p$-value.

### 5.7.5 F-Measure

The maximum F-Measure was achieved for MEWMA with PSO for about 62.94%, 41.72% and 30.44% compared with 60.5%, 39% and 27.5% for MEWMA with GA using optimal parameter set with window sizes (1s,2s,3s), $\lambda$ (0.5, 0.6 & 0.7) and p=0.05 as shown in Figure 5.21.

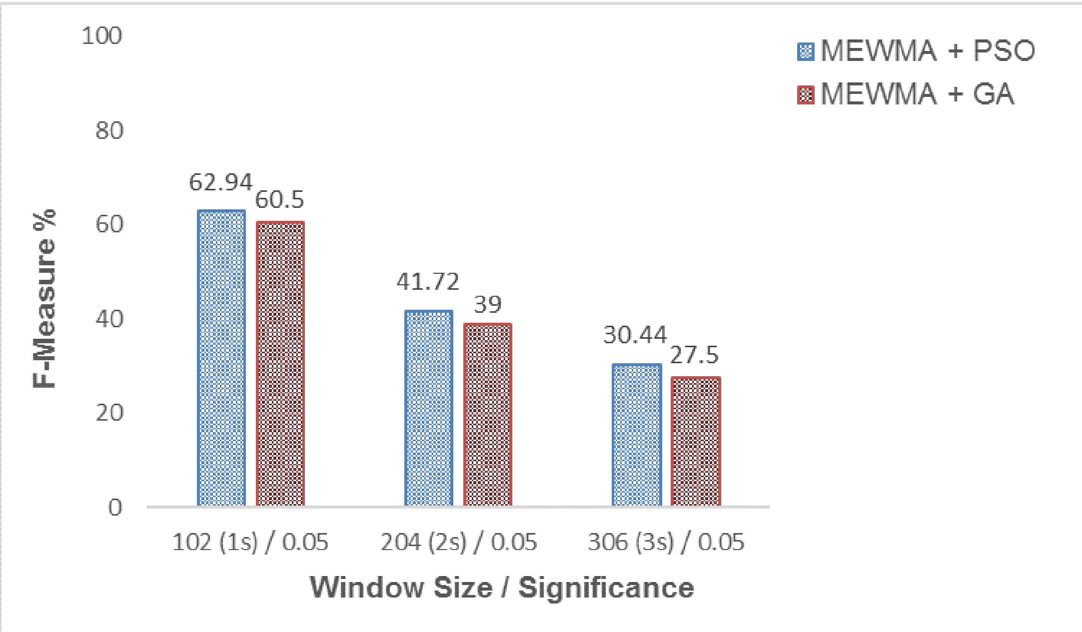



**Figure 5.21:** Comparison of F-Measure between MEWMA (PSO and GA)

The MEWMA with PSO is improved approximately 2.7% on average for each window size than MEWMA with GA.

Likewise, the highest F-Measure that was achieved for MCUSUM with PSO is about 40.29%, 35.62% and 22.94% while 37.5%, 33.5% and 20.5% for MCUSUM with GA using optimal parameter set with window sizes (1s,2s,3s), $k$=0.5 and $p$=0.05 as shown in Figure 5.22. The MCUSUM with PSO is improved approximately 2.4% on average for each window size than MCUSUM with GA. A one-sided $t$-test is performed to find the statistical significance for the accuracy metric for 10 experiments repeatedly performed for each approach i.e. MEWMA with PSO and MCUSUM with PSO. The results of the $t$-test suggest that the MEWMA with PSO is statistically significant by achieving the significance 0.0246 which is less than the standard $p$-value.

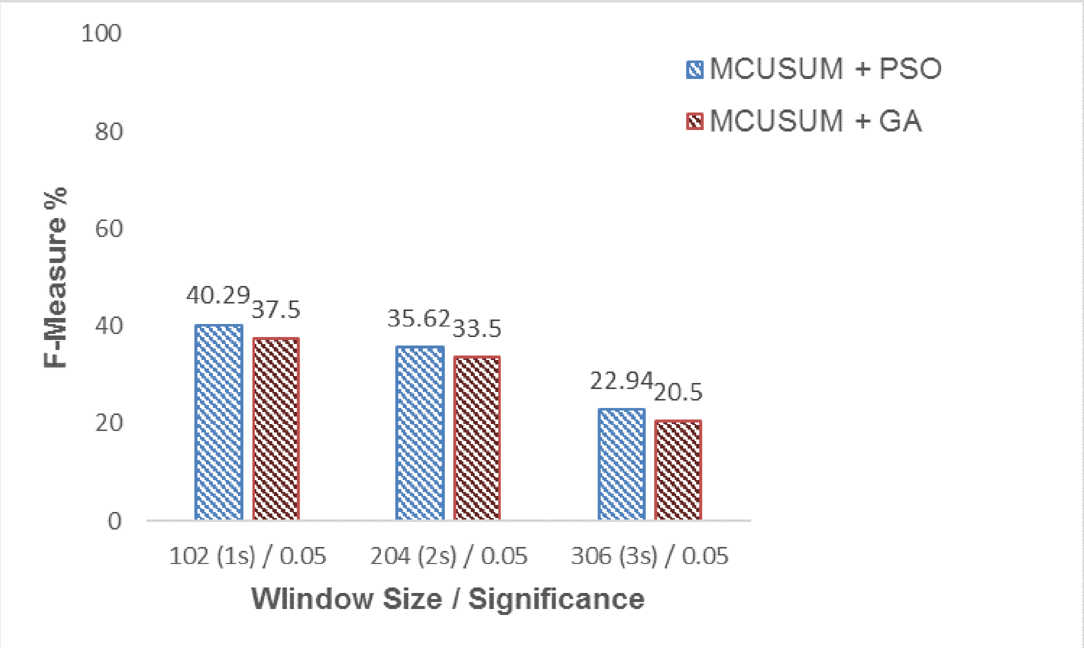



**Figure 5.22:** Comparison of F-Measure between MCUSUM (PSO and GA)

However, the F-Measure analysis of MEWMA (PSO & GA) and MCUSUM (PSO & GA) was improved with MEWMA (PSO) by about 22.65%, 6.1% and 7.5% for each window size respectively as compared to MCUSUM(PSO). Also, MEWMA(GA) is improved about 23%, 6.5% and 7% for each window size respectively compared to MCUSUM(GA) as shown in Figure 5.21 and 5.22 respectively.

### 5.7.6 Computational Cost or Time Complexity

This section presents the empirical timings of both algorithm MEWMA (PSO & GA) and MCUSUM (PSO & GA) for accurate change detection using optimal parameter selection. The techniques are implemented in Matlab 2015b and experiments are performed on a system with processor 3.40 *GHz* and 8GB RAM. The Matlab *tic toc* function is used to calculate the time for optimal parameter set with accurate change and high metric measures.

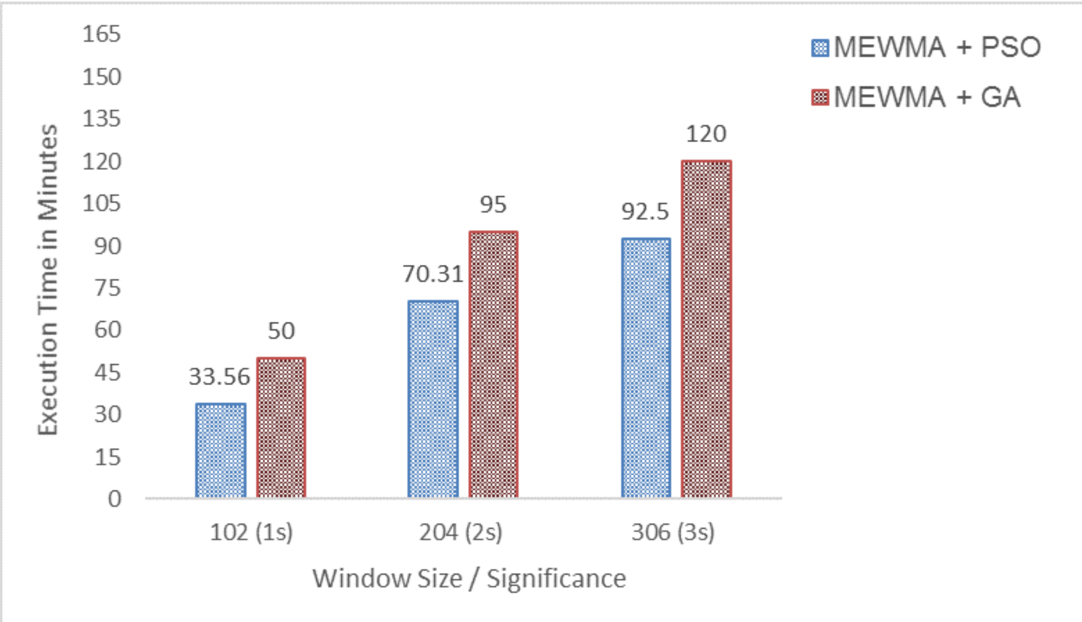



**Figure 5.23:** Comparison of Computational Cost between MEWMA (PSO and GA)

The results in Figure 5.23 presents that MEWMA (PSO) took less time at about 17.56 min, 24.69 min and 27.5 min respectively for each window size compared with MEWMA (GA) for optimal solution of accurate change detection. Likewise, MCUSUM (PSO) also took less time about 23.6 min, 35.27 min and 40.20 min respectively for each window size toward optimal solution of accurate change detection as shown in Figure 5.24.

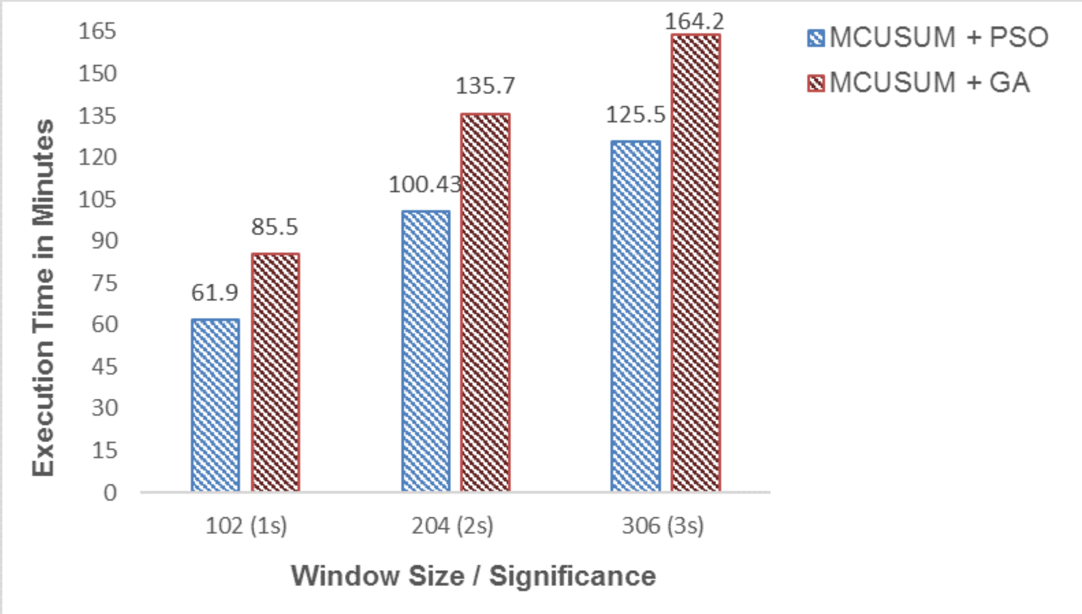



**Figure 5.24:** Comparison of Computational Cost between MCUSUM (PSO and GA)

Furthermore, the MEWMA (PSO) outperforms than MCUSUM (PSO) in achieving low computational cost of about 28.34 min, 30.12 min and 33 min for the same window sizes towards optimal solution. Similarly, the MEWMA (GA) also performed better than MCUSUM(GA) by using minimal computational cost of about 35.5 min, 40.70 min and 44.20 min for the same window sizes towards optimal solution. Also, the *t*-test results justify computational efficiency of MEWMA with PSO over MCUSUM with PSO by proving statistical significance with 95% confidence achieved after 10 repeated experiments were investigated. The PSO and GA are both population based algorithm, however, PSO is a relatively recent heuristic search algorithm compared with GA. PSO is computationally efficient because it uses less number of functions than GA for evaluation towards optimal solutions (Hassan et al., 2004). Hence, as we are more inclined towards online activity monitoring which require lightweight algorithm for evaluation of data. Therefore, the analysis of current results reflects that the MEWMA is a good choice for online implementation for accurate change point detection. The comparison of MEWMA with PSO & GA and MCUSUM with PSO & GA is also presented in Table 5.3 and 5.4 respectively.

**Table 5.3:** MEWMA with PSO & GA

| Win size | Sig Value | $\lambda$ | Accuracy | | Precision | | Sensitivity | | G-Means | | F-Measure | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MEWMA+PSO | MEWMA+GA | MEWMA+PSO | MEWMA+GA | MEWMA+PSO | MEWMA+GA | MEWMA+PSO | MEWMA+GA | MEWMA+PSO | MEWMA+GA |
| 1s | 0.05 | 0.5 | 99.9 % | 99.7 % | 60.78 % | 57.50 % | 65.26 % | 60.50 % | 80.78 % | 75.50 % | 62.94 % | 60.50 % |
| 2s | | 0.6 | 99.70 % | 99.50 % | 50 % | 48 % | 35.97 % | 31.50 % | 60.93 % | 57.50 % | 41.72 % | 39 % |
| 3s | | 0.7 | 99.3 % | 99 % | 45.45 % | 43 % | 25 % | 23.5 % | 39.73 % | 37 % | 30.44 % | 27.50 % |

**Table 5.4:** MCUSUM with PSO & GA

| Win size | Sig Value | $k$ | Accuracy | | Precision | | Sensitivity | | G-Means | | F-Measure | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MCUSUM+PSO | MCUSUM+GA | MCUSUM+PSO | MCUSUM+GA | MCUSUM+PSO | MCUSUM+GA | MCUSUM+PSO | MCUSUM+GA | MCUSUM+PSO | MCUSUM+GA |
| 1s | 0.05 | 0.5 | 99.5 % | 99.3 % | 55 % | 52 % | 29.47 % | 27.50 % | 54.29 % | 52.50 % | 40.29 % | 37.50 % |
| 2s | | | 99.40 % | 99.20 % | 45.98 % | 43 % | 26.37 % | 25 % | 51.31 % | 48.50 % | 35.62 % | 33.50 % |
| 3s | | | 99 % | 98.80 % | 40 % | 38 % | 20 % | 18.5 % | 30 % | 27.50 % | 22.94 % | 20.50 % |

## 5.8 Chapter Summary

The multivariate approaches are used to analyze and evaluate multivariate data for automatic change point detection. In multivariate data analysis, more than one characteristics of a system evaluated simultaneously and the approach also identify the relationship among these characteristics.

The proposed MEWMA approach which tunes the different parameters such as lambda, which weights the current versus historical data, window size and significance value with the aim of achieving better performance and accurate change point detection. Also, we implement MCUSUM a multivariate approach to use as a bench mark for our proposed technique. Moreover, the GA and PSO are used to automatically identify an optimal parameter set using different parameters for MEWMA and MCUSUM, so as to maximize the objective function i.e. the F-measure. The evaluation is performed using different metric measures and the experimental results show that the proposed scheme outperforms than the bench mark scheme. Also, the computation cost is less than the benchmark approach.Moreover, t-tests were also performed for each evaluation metric and the results show that the proposed approach is statistically significantly better than the benchmark technique.

# Chapter 6

# Conclusion and Future Work

## 6.1 Introduction

In this thesis, previous work has been improved by analyzing automatic and online change detection in multivariate data in activity monitoring. The aim of this research is to detect and identify specific transition that can be used in various scenarios such as to identify patient vital sign in medical domain or generating activity labels for the purposes of annotating real-world datasets. In Chapter 1, Research Objectives have been highlighted which has been achieved through the completion of a number of studies as follows.

### Objective 1: Review of wearable sensors that can be used for human activity monitoring.

Chapter 2 provided a review of wearable sensors that have been used for human activity monitoring and detection. The information about different kind of activities monitored using these sensors were also provided. Moreover, the analysis of different segmentation and feature extraction approaches were also discussed. Further, chapter 2 discussed the different approaches for change point detection in time series data that identify the key challenge of multivariate data analysis, which was the focus of ensuing work.

### Objective 2: To evaluate a multivariate approach for online change and identify the optimal parameter set for accurate change-point detection in activity monitoring with high metric measures.

Data was collected from tri-axle accelerometer (Zhang et al., 2011) for different activities which consist of static and dynamic activities. The x, y and z axis of an accelerometer signal is divided into multiple windows of different sizes. The MEWMA with sliding window algorithm was applied for accurate change point detection, which was developed in Matlab. Also, the univariate algorithm was presented by (Jain and Wang, 2015) implemented in Matlab and used as a benchmark to our proposed approach. The different metric measures were also calculated for both approaches to evaluate results for accurate change point detection.

### Objective 3: To implement and evaluate optimization algorithms for the multivariate approach to automatically identify optimal parameter set for accurate change-point detection.

Work described in Chapter 4, provides detailed information about employing a genetic algorithm to automatically identify an optimal parameter set, using a fitness function for MEWMA, parameters such as the forgetting parameter $\lambda$, the window size, and a significance value for each activity so as to maximize the Fitness Function. Moreover, within this chapter, optimal parameter selection facilitates an algorithm to detect accurate change points and minimize false alarms. The performance of a real dataset and a synthetic dataset were evaluated based on accelerometer data collected for a set of different activities.

**Objective 4: To develop an evaluation framework to compare different multivariate and optimization approaches for change-point detection. The fusion of such approaches could empower a system to automatically identify optimal parameter set for accurate change detection.**

Chapter 5 investigated and provided information about using the Multivariate Cumulative Sum Control Chart (MCUSUM) to automatically detect change points in user activities. Also, the Particle Swarm Optimization (PSO) is discussed in detail and used to identify optimal parameter settings for MCUSUM and MEWMA for accurate change point detection. The MCUSUM is also used as a benchmark to our proposed technique MEWMA. Moreover, in Chapter 5, MEWMA and MCUSUM approaches are used with GA and PSO to automatically identify an optimal parameter set using different parameters for MEWMA and MCUSUM, so as to maximize the objective function, namely the F-measure. Data was collected using tri-axle accelerometers sensors (Patterson et al., 2017) placed on the chest, wrist and ankle of the participants. The data for different nine activities was collected and captured with a sample frequency of 102.4 Hz. The different metric measures were also calculated for both approaches to evaluate results on a real dataset for accurate change point detection.

Furthermore, the following section will provide detail information relating to the contributions to knowledge from this work and possible areas for future work.

## 6.2 Summary of contribution

This thesis contributes to the area of change point detection in online activity monitoring. Multivariate data has been analyzed and investigated using existing sensors, which have been used for collecting data for various activities

### 6.2.1 Taxonomy of change point detection in activity monitoring (Objective 1)

Literature contained in chapter 2, presented a taxonomy of wearable sensors and monitoring of different activities using these sensors. The synthesis and integration of this information determine the current key challenge of online change point detection in multivariate data as a key issue to be addressed.

### 6.2.2 Multivariate approach for online change in activity monitoring (Objective 2)

Chapter 3 is the first to investigate the analysis of univariate and multivariate data obtained from tri-axle accelerometer (Zhang et al., 2011) for accurate change point detection. The analysis concluded that multivariate data gives a richer picture of the process than univariate data to identify accurate change point detection. The evaluation was performed using different metric measures such as accuracy, precision, G-means and F-measure.

Therefore, the multivariate exponentially weighted moving average (MEWMA) has been used to identify detect change points corresponding to different transitions in user activity. The results evaluation shows that the proposed standard MEWMA provides better accuracy and improved on the other metric measures such as precision, G-means and F-measure by more than 12%, 24% and 13% respectively as compared with the univariate approach presented by (Jain and Wang, 2015)

Moreover, within this thesis, the different parameters of MEWMA were evaluated to select the optimal parameters. The standard MEWMA and optimal parameters were used to analyze the performance of MEWMA. The optimal parameters of MEWMA outperformed standard values for real world accelerometer data for accuracy, precision, G-means and F-measure compared with the standard approach. Also, the MEWMA approach achieved low computation costs and can run in the online scenario.

### 6.2.3 Optimization algorithm for the multivariate approach to automatically identify optimal parameter set for accurate change-point detection (Objective 3)

Chapter 4 delineated genetic algorithm to identify the optimal set of parameters for the MEWMA approach and automatically detect change points corresponding to different transitions in the user activities. Within this chapter, the genetic algorithm (GA) is used to mimic the process of evolution by taking a population of strings, which encodes possible solutions, and combining them based on the fitness function to produce solutions that are high performing. The fitness function is the core component of the GA. It evaluates each individual parameter set in the population to find the solution with an optimal fitness value. The F-measure is used as an objective function for evaluation in GA because it is a combination of precision and recall.

Within this chapter, the experiments were performed on real and synthetic datasets. The evaluations results of real dataset reflected that the F-measure was higher about 50% to 66.7% for optimal set using GA than the 40% to 50% for non-optimized results. Moreover, the accuracy was also improved from 99.4% to 99.8% with optimization when compared with the non-optimized accuracy of 98.5% as opposed to 99.4%.

Based on the results and findings in this thesis, it can be concluded that the optimal set of parameters selected using the GA outperformed on real world accelerometer data in terms of the accuracy and the F-measure. In this thesis, the automatic optimization of the optimal parameter set was considered within the context of activity monitoring. Moreover, the MEWMA is a lightweight algorithm and can be incorporated into real world systems such as mobile-based applications for the collection and active sampling of labeled data. The change points in the data can be used to identify changes in activities and recognize and monitor good behavior such as healthy exercise patterns based on these activities.

### 6.2.4 Evaluation framework to compare different multivariate and optimization approaches for change-point detection. (Objective 4)

Chapter 5 is first to explore the evaluation results from comparison of Multivariate approaches such as MEWMA and MCUSUM to automatically detect change points in user activities. In addition, the GA and PSO were used to automatically identify an optimal parameter set using different parameters for MEWMA and MCUSUM, so as to maximize the objective function that is F-measure.

Within this thesis, the evaluation was performed using different metric measures for MEWMA (PSO & GA) and MCUSUM (PSO & GA). The experimentation of different combination evaluation reflected that the MEWMA with PSO outperform than the rest in terms achieving high accuracy, precision, sensitivity, G-means and F-measure. Hence, a *t*-test was also performed to assess statistical significance for all evaluation metrics and the results justified that MEWMA with PSO is statistical significance with 95% confidence achieved for all metrics measures.

Moreover, the change detection in sensor-based time series data is valuable when monitoring human behaviour to detect and analyse changes. Such analysis can be used to detect patient vital signs like heart beat against various activities performed. Analysing sensor-based time series data can also be used to recognize and monitor good behaviour such as healthy exercise patterns based on performed activities.

Furthermore, in this thesis, we mainly focused on online activity monitoring which require lightweight algorithm for evaluation of data. Therefore, the analysis of current results reflects that the MEWMA with PSO is a good choice for online implementation for accurate change point detection.

# Future Work

In this thesis, different metrics were used for evaluation such as Accuracy, Precision, Sensitivity, Specificity etc. However, based on the datasets used, metrics such as accuracy and specificity are very high while metrics such as precision and sensitivity are very low. The increase and decrease occurs because of the class imbalance problem as the class distributions are highly imbalanced in the data set.

# Class imbalance problem

The class-imbalance problem is a real-world problem and exist in many real world applications (Galar et al., 2012). The problem happens when the total number of a class data (positive) is less than the total size of other classes of data (negative). This highlights the skewed distribution of classes within the dataset and identifies that the minority class is the class of interest. The class imbalance problem becomes more challenging in real time scenario, when the data streams arrive continuously and the class distribution is imbalanced. Future work could be therefore to explore and investigate the different online class imbalance learning approaches that can be used to balance the minority class in the dataset and possibly improve the classification results. Different approaches such as Oversampling based Online Bagging (OOB) and Undersampling based Online Bagging (UOB) (Wang et al., 2013) could be used to address the online class imbalance problem in the dataset.

## Accelerometer placement

The effectiveness of accelerometer location is also very important to get more detailed information about the different activities. For example, for walking activity, the accelerometer placement on ankle give more detailed information about the activity rather than placed on chest (Cleland et al., 2013). However, this could also be important for mobile sensing application when the accelerometer does not having fixed location. Hence, the future work would be to explore the accelerometer location for different activities to obtain more and detailed information related to each activity which can be used for analysis of change detection in different user activities.

## Sensor Data Fusion

Another promising direction of our research could be the data fusion obtained from multiple sensors such as accelerometer, Gyroscope and GPS, which can improve the evaluation metric measures. The data fusion from such sensors could provide a more unified picture and global view of the data that can help in identifying and analyzing the data for accurate change detection. The number of algorithms like Kalman filter (KF), Particle Filtering (PF) and Weighted Averages have been used for data fusion in literature. The Kalman Filter (KF) (Al-Jawad et al., 2013) is a popular statistical state estimation method that can be used to fuse dynamic signal level data. The state estimates of the system are determined based on a recursively applied prediction and update algorithm and assumes the state of a system at the current time is based on the state of the system at the previous time interval. One of the main advantages of the KF is that it is computationally efficient Luo et al. (2011). The Particle filtering (PF) (Kotecha and Djuric, 2003) is a stochastic method to estimate moments of a target probability density, when they can't be computed analytically. The principle is to generate random numbers called particles, from an "importance" distribution that can be easily sampled. Then, each particle is associated a weight that corrects the dissimilarity between the target and the importance probabilities. The weighted averages is a simple signal level fusion method for combining commensurate information by taking an average of all the sensor readings (Luo and Kay, 1990). The contribution of the "worst" sensor's error will be alleviated in the final estimate, although not eliminate it completely. To reduce the impact of large erroneous sensor readings weighted averages can be used (Yang and Yang, 2006). However, the data fusion of numerous sensor data is quit challenging and required more comprehensive analysis to evaluate the data.

Of the issues raised for future work, the imbalanced class problem is highly significant, as change points are rare events in the context of the frequency of data collection and we plan to address this issue in our future work.

# REFERENCES


ActivityRecognition. Activity recognition api, 2018. URL https://developers.google.com/location-context/activity-recognition/. (Accessed: 2018-01-30).

A. Al-Jawad, A. Barlit, M. Romanovas, M. Traechtler, and Y. Manoli. The use of an orientation kalman filter for the static postural sway analysis. *APCBEE procedia*, 7:93–102, 2013.

H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.

Algosanp. Critical values and p values, 2016. URL http://www.algosnap.com/.

U. Appel and A. V. Brandt. Adaptive sequential segmentation of piecewise stationary time series. *Information sciences*, 29(1):27–56, 1983.

M. Atallah, R. Gwadera, and W. Szpankowski. *Detection of sets of episodes in event sequences: Algorithms, analysis and experiments*. PhD thesis, Thesis, 2003.

A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *Architecture of computing systems (ARCS), 2010 23rd international conference on*, pages 1–10. VDE, 2010.

M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno. Early drift detection method. 2006.

L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive computing*, pages 1–17, 2004.

M. Basseville, I. V. Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993.

D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms: Part 1, fundamentals. *University computing*, 15(2):56–69, 1993.

S. Bersimis, S. Psarakis, and J. Panaretos. Multivariate statistical process control charts: an overview. *Quality and Reliability engineering international*, 23(5): 517–543, 2007.

A. Bifet and R. Gavalda. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 443–448. SIAM, 2007.

J. Blondin. Particle swarm optimization: A tutorial. *Availaible from: http://cs. armstrong. edu/saad/csci8100/pso tutorial. pdf*, 2009.

O. Bodnar and W. Schmid. Surveillance of the mean behavior of multivariate time series. *Statistica Neerlandica*, 61(4):383–406, 2007.

E. Borenstein and S. Ullman. Combined top-down/bottom-up segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 30(12):2109–2125, 2008.

J. Brusey, R. Rednic, and E. Gaura. Classifying transition behaviour in postural activity monitoring. *Sensors & Transducers*, 7:213, 2009.

A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca. Shimmer™–a wireless sensor platform for non-invasive biomedical research. *IEEE Sensors Journal*, 10(9):1527–1534, 2010.

F. Busetti. Genetic algorithms overview. *Retrieved on December*, 1, 2007.

A. Calzada, J. Liu, C. D. Nugent, H. Wang, and L. Martinez. Sensor-based activity recognition using extended belief rule-based inference methodology. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2694–2697, Aug 2014. doi: 10.1109/EMBC.2014.6944178.

F. Camci. Change point detection in time series data using support vectors. *International Journal of Pattern Recognition and Artificial Intelligence*, 24 (01):73–95, 2010.

C. J. Caspersen, K. E. Powell, and G. M. Christenson. Physical activity, exercise, and physical fitness: definitions and distinctions for health-related research. *Public health reports*, 100(2):126, 1985.

S. Chavan and N. P. Adgokar. An overview on particle swarm optimization: basic concepts and modified variants. *Int. J. Sci. Res*, 4(5):255–260, 2015.

G. Chen and S. W. Cheng. Max chart: combining x-bar chart and s chart. *Statistica Sinica*, pages 263–271, 1998.

L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.

H. Cho et al. Change-point detection in panel data via double cusum statistic. *Electronic Journal of Statistics*, 10(2):2000–2038, 2016.

I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay. Optimal placement of accelerometers for the detection of everyday activities. *Sensors*, 13(7):9183–9200, 2013.

I. Cleland, M. Han, C. Nugent, H. Lee, S. McClean, S. Zhang, and S. Lee. Evaluation of prompted annotation of activity data recorded from a smart phone. *Sensors*, 14(9):15861–15879, 2014.

D. J. Cook and N. C. Krishnan. *Activity learning: discovering, recognizing, and predicting human behavior from sensor data.* John Wiley & Sons, 2015.

J. Corge and F. Puech. Analyse du rythme cardiaque foetal par des méthodes de détection de ruptures. *Analysis and Optimization of Systems*, pages 853–866, 1986.

R. B. Crosier. Multivariate generalizations of cumulative sum quality-control schemes. *Technometrics*, 30(3):291–303, 1988.

M. F. S. D'Angelo, R. M. Palhares, R. H. Takahashi, and R. H Loschi. A fuzzy/bayesian approach for the time series change point detection problem. *Pesquisa Operacional*, 31(2):217–234, 2011.

W. Dargie. Analysis of time and frequency domain features of accelerometer measurements. In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pages 1–6. IEEE, 2009.

T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications.* Citeseer, 2006.

V. M. De Oca, D. R. Jeske, Q. Zhang, C. Rendon, and M. Marvasti. A cusum change-point detection algorithm for non-stationary sequences with application to data network surveillance. *Journal of Systems and Software*, 83(7): 1288–1297, 2010.

R. J. Di Francesco. Real-time speech segmentation using pitch and convexity jump models: Application to variable rate speech coding. *IEEE transactions on acoustics, speech, and signal processing*, 38(5):741–748, 1990.

W. O. Eaton. Measuring activity level with actometers: Reliability, validity, and arm length. *Child Development*, pages 720–726, 1983.

R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

I. A. Eckley, P. Fearnhead, and R. Killick. Analysis of changepoint models. *Bayesian Time Series Models*, pages 205–224, 2011.

G. Francis, C. E. Rash, and M. B. Russo. The human-machine interface challenge. *Helmet-Mounted Displays: Sensory, Perceptual and Cognitive Issues (TL CE Rash, MB Russo and E. Schmeisser, eds.)*, pages 29–44, 2009.

M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2012.

D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

V. Golosnoy, S. Ragulin, and W. Schmid. Multivariate cusum chart: properties and enhancements. *AStA Advances in Statistical Analysis*, 93(3):263–279, 2009.

M. Hameed, M. M. MANSOUR, and E. M. A. ELRAZIK. Mcusum control chart procedure: Monitoring the process mean with application. *Journal of Statistics: Advances in Theory and Applications*, 16(1):105–132, 2016.

M. Han, J. H. Bang, C. Nugent, S. McClean, and S. Lee. A lightweight hierarchical activity recognition framework using smartphone sensors. *Sensors*, 14 (9):16181–16195, 2014.

E. S. Handbook. Critical values and p values, 2016. URL http://www.itl.nist.gov/div898/handbook/prc/section1/prc131.htm.

R. Hassan, B. Cohanim, O. de Weck, and G. Venter. A copmarison of particle swarm optimization and the genetic algorithm. *American Institute of Aeronautics and Astronautics*, 2004.

R. L. Haupt and S. E. Haupt. *Practical genetic algorithms*. John Wiley & Sons, 2004.

J. Havskov and L. Ottemöller. Magnitude. In *Routine Data Processing in Earthquake Seismology*, pages 151–191. Springer, 2010.

D. M. Hawkins and Q. Deng. A nonparametric change-point control chart. *Journal of Quality Technology*, 42(2):165, 2010.

K. Holzinger, V. Palade, R. Rabadan, and A. Holzinger. Darwin or lamarck? future challenges in evolutionary algorithms for knowledge discovery and data mining. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, pages 35–56. Springer, 2014.

L. Hongcheng. *Multivariate extensions of CUSUM procedure*. PhD thesis, Kent State University, 2007.

X. Hu, Y. Shi, and R. Eberhart. Recent advances in particle swarm. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 90–97. IEEE, 2004.

X. Hu, H. Qiu, and N. Iyer. Multivariate change detection for time series data in aircraft engine fault diagnostics. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 2484–2489. IEEE, 2007.

J. Hunter and N. McIntosh. Knowledge-based event detection in complex time series data. In *Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, pages 271–280. Springer, 1999.

D. T. G. Huynh. *Human activity recognition with wearable sensors*. PhD thesis, Technische Universität, 2008.

A. Jain and Y. Wang. A new framework for on-line change detection, 2015. URL http://excelsior.cs.ucsb.edu/papers/as06.pdf.

T. Jeon, B. Kim, M. Jeon, and B.-G. Lee. Implementation of a portable device for real-time ecg signal analysis. *Biomedical engineering online*, 13(1):160, 2014.

A. Kaveh. *Particle Swarm Optimization*, pages 9–40. Advances in Metaheuristic Algorithms for Optimal Design of Structures, 2014.

Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 389–400. SIAM, 2009.

J. Kennedy. Swarm intelligence. In *Handbook of nature-inspired and innovative computing*, pages 187–219. Springer, 2006.

E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296. IEEE, 2001.

A. M. Khan. Human activity recognition using a single tri-axial accelerometer. *Department of Computer Engineering, Graduate School, Kyung Hee University, Seoul, Korea*, 2011.

N. Khan, S. McClean, S. Zhang, and C. Nugent. Using genetic algorithms for optimal change point detection in activity monitoring. In *Computer-Based Medical Systems (CBMS), 2016 IEEE 29th International Symposium on*, pages 318–323. IEEE, 2016.

M. B. Khoo. An extension for the univariate exponentially weighted moving average control chart. *Matematika*, 20:43–48, 2004.

D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 180–191. VLDB Endowment, 2004.

L. Koepcke and J. Kretzberg. Change point detection for neuronal data with cusum and classification methods. *BMC Neuroscience*, 14(1):P223, 2013.

J. H. Kotecha and P. M. Djuric. Gaussian sum particle filtering. *IEEE Transactions on signal processing*, 51(10):2602–2612, 2003.

L. I. Kuncheva. Using control charts for detecting concept change in streaming data. *Bangor University*, 2009.

L. I. Kuncheva and W. J. Faithfull. Pca feature extraction for change detection in multidimensional unlabeled data. *IEEE transactions on neural networks and learning systems*, 25(1):69–80, 2014.

B. Kuris and T. Dishongh. Shimmer-sensing health with intelligence, modularity, mobility, and experimental reusability. *Intel Corp*, 23:104–107, 2006.

S. Laxman, P. Sastry, and K. Unnikrishnan. Discovering frequent episodes and learning hidden markov models: A formal connection. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1505–1517, 2005.

S. Laxman, P. Sastry, and K. Unnikrishnan. Discovering frequent generalized episodes when events persist for different durations. *IEEE Transactions on Knowledge and Data Engineering*, 19(9), 2007a.

S. Laxman, P. Sastry, and K. Unnikrishnan. A fast algorithm for finding frequent episodes in event streams. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 410–419. ACM, 2007b.

C.-S. Li, P. S. Yu, and V. Castelli. Malm: A framework for mining sequence database at multiple abstraction levels. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 267–272. ACM, 1998.

M. Lovrić, M. Milanović, and M. Stamenković. Algoritmic methods for segmentation of time series: An overview. *Journal of Contemporary Economic and Business Issues*, 1(1):31–53, 2014.

C. A. Lowry, W. H. Woodall, C. W. Champ, and S. E. Rigdon. A multivariate exponentially weighted moving average control chart. *Technometrics*, 34(1): 46–53, 1992.

J. M. Lucas and M. S. Saccucci. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, 32(1):1–12, 1990.

R. C. Luo and M. G. Kay. A tutorial on multisensor integration and fusion. In *Industrial Electronics Society, 1990. IECON'90., 16th Annual Conference of IEEE*, pages 707–722. IEEE, 1990.

R. C. Luo, C. C. Chang, and C. C. Lai. Multisensor fusion and integration: Theories, applications, and its perspectives. *IEEE Sensors Journal*, 11(12): 3122–3138, 2011.

R. Malhotra, N. Singh, and Y. Singh. Genetic algorithms: Concepts, design for optimization of process controllers. *Computer and Information Science*, 4(2): 39, 2011.

S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

M. J. Mathie, A. C. Coster, N. H. Lovell, and B. G. Celler. Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiological measurement*, 25(2):R1, 2004.

M. T. Matlab-Toolbox. Particle swarm optimization (global optimization toolbox), 2015. URL https://uk.mathworks.com/help/gads/particle-swarm.html.

J. McCall. Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 184(1):205–222, 2005.

M. Mitchell. Genetic algorithms: An overview. *Complexity*, 1(1):31–39, 1995.

D. C. Montgomery. *Introduction to statistical quality control.* John Wiley & Sons (New York), 2009.

Multivariate. Multivariate normal random numbers, 2016. URL http://uk.mathworks.com/help/stats/mvnrnd.html. (Accessed: 2018-01-01).

K. Murao and T. Terada. A recognition method for combined activities with accelerometers. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 787–796. ACM, 2014.

Q. Ni, A. B. García Hernando, and I. P. de la Cruz. The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development. *Sensors*, 15(5):11312–11362, 2015.

M. Nyan, F. Tay, K. Seah, and Y. Sitoh. Classification of gait patterns in the time–frequency domain. *Journal of biomechanics*, 39(14):2647–2656, 2006.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

X. Pan and J. E. Jarrett. The multivariate ewma model and health care monitoring. 2014.

A. Pantelopoulos and N. G. Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2010.

T. Patterson, N. Khan, S. McClean, C. Nugent, S. Zhang, I. Cleland, and Q. Ni. Sensor-based change detection for timely solicitation of user engagement. *IEEE Transactions on Mobile Computing*, 16(10):2889–2900, 2017.

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*, 56(3): 871–879, 2009.

G. J. Ross and N. M. Adams. Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology*, 44(2):102, 2012.

M. Sekine, T. Tamura, T. Togawa, and Y. Fukui. Classification of waist-acceleration signals in a continuous walking record. *Medical engineering & physics*, 22(4):285–291, 2000.

P. Sharkey and R. Killick. Nonparametric methods for online changepoint detection. *STOR601 Research Topic II*, 2014.

H. Shatkay and S. B. Zdonik. Approximate queries and representations for large data sequences. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*, pages 536–545. IEEE, 1996.

X. Song, M. Wu, C. Jermaine, and S. Ranka. Statistical change detection for multi-dimensional data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 667–676. ACM, 2007.

M. Stikic, D. Larlus, S. Ebert, and B. Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2521–2537, 2011.

M. A. Sturza. Navigation system integrity monitoring using redundant measurements. *Navigation*, 35(4):483–501, 1988.

M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008.

J.-i. Takeuchi and K. Yamanishi. A unifying framework for detecting outliers and change points from time series. *IEEE transactions on Knowledge and Data Engineering*, 18(4):482–492, 2006.

N. Tatti and J. Vreeken. The long and the short of it: summarising event sequences with serial episodes. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2012.

W. A. Taylor. Change-point analysis: A powerful new tool for detecting changes, 2000. *Preprint http://www. variation. com/cpa/tech/changepoint. html*, 2008.

M. F. Triola. *Elementary statistics*. Pearson/Addison-Wesley Reading, MA, 2006.

A. Ukil and R. Zivanovic. Automatic signal segmentation based on abrupt change detection for power systems applications. In *Power India Conference, 2006 IEEE*, pages 8–pp. IEEE, 2006.

K. Van Laerhoven and B. Schiele. An empirical study of time series approximation algorithms for wearable accelerometers. 2009.

R. Vlasveld. Temporal segmentation using support vector machines in the context of human activity recognition. Master's thesis, 2014.

S. Wang, L. L. Minku, and X. Yao. A learning framework for online class imbalance learning. In *Computational Intelligence and Ensemble Learning (CIEL), 2013 IEEE Symposium on*, pages 36–45. IEEE, 2013.

J. Welkowitz, B. H. Cohen, and R. B. Lea. *Introductory statistics for the behavioral sciences*. John Wiley & Sons, 2011.

G. Wetherill and D. Brown. Statistical process control: Theory and practice chapman and hall. *London, UK*, 1991.

A. Wirsch. *Analysis of a top-down bottom-up data analysis framework and software architecture design*. PhD thesis, Massachusetts Institute of Technology, 2014.

Y. B. Yahmed, A. A. Bakar, A. R. Hamdan, A. Ahmed, and S. M. S. Abdullah. Adaptive sliding window algorithm for weather data segmentation. *Journal of Theoretical and Applied Information Technology*, 80(2):322, 2015.

M. Yamada, A. Kimura, F. Naya, and H. Sawada. Change-point detection with feature selection in high-dimensional time-series data. In *IJCAI*, pages 1827–1833, 2013.

C.-C. Yang and Y.-L. Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10(8):7772–7788, 2010.

G.-Z. Yang and G. Yang. *Body sensor networks*, volume 1. Springer, 2006.

Y. Yu, Y. Zhu, S. Li, and D. Wan. Time series outlier detection based on sliding window prediction. *Mathematical problems in Engineering*, 2014.

S. Zhang, S. McClean, B. Scotney, L. Galway, and C. Nugent. A framework for context-aware online physiological monitoring. In *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, pages 1–6. IEEE, 2011.

M. Zhong, J. Wen, P. Hu, and J. Indulska. Advancing android activity recognition service with markov smoother. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, pages 38–43. IEEE, 2015.

M. Ziefle, L. Klack, W. Wilkowska, and A. Holzinger. Acceptance of telemedical treatments–a medical professional point of view. In *International Conference on Human Interface and the Management of Information*, pages 325–334. Springer, 2013.