# A Study On Fall Detection Using Feature Map Generated From Mobile Sensor Data

By


**Shameem Ahammed**

Roll: 1707117

**Department Of Computer Science and Engineering**

**Khulna University of Engineering & Technology**

**Khulna 9203, Bangladesh**


**March 2023**

# A Study on Fall Detection Using Feature Map Generated from Mobile Sensor Data

By

**Shameem Ahammed**

Roll: 1707117

A thesis submitted in partial fulfillment of the requirements for the degree of
"Bachelor of Science in Computer Science & Engineering"

**Supervisor:**

**Dr. Sk. Md. Masudul Ahsan**

Professor

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

________________________

Signature

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

March, 2023

# Acknowledgement

All the praise to the almighty Allah, whose blessing and mercy succeeded me to complete this thesis work fairly. I gratefully acknowledge the valuable suggestions, advice and sincere co-operation of Dr. Sk. Md. Masudul Ahsan, Professor, Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET). His prudence and keen supervision through this study have been a great inspiration for me.  I am also grateful to the people who has directly and indirectly helped me to complete this study.

Author

# Abstract

Falls refer to sudden unintentional move from higher level to lower level without any control. Falls can happen with people of any age, gender at any time. But the main sufferers of fall event are the elderly people. Elder population is growing rapidly day by day around the world. Most of the elderly people remains unsupervised major part of the day. If a fall occur with any person, the person may be laying down for a long period of time without any help. Thus, a fall can be costly in terms of health, money and lives. Therefore, a fall detection system is needed to automatically detect fall events which will play an important role in health care system of elderly people. Recently many researchers have devoted themselves to develop systems and methods for automatic fall detection. Fall detection techniques can be divided in four groups based on the source of data. They are namely: life information-based methods, radar-based methods, wearable/mobile sensor based and vision-based methods. Wearable/mobile sensor-based methods can be used in both inside and outside of the room. Hence it is preferred most now a days. Because of the scarcity of publicly available sensor dataset, a dataset was constructed. It contains six activities namely Falling, Standing, Walking, Sitting, Sitting on chair in the back side and Laying. Accelerometer and gyroscope data were collected using mobile phone sensor. These data were used for the classification purposes. Wearable/mobile sensor-based fall detection system has the ability to detect fall both in outdoor and indoor. In this work, feature map was generated from the mobile sensor data. Mobile sensor data are mainly accelerometer and gyroscope data. Some features are extracted from the sensor data and feature map was generated using the features. Then this feature map was used to classify the fall action with other activities.

# Contents

# List of Tables

# List of Figures

# Nomenclature

| | |
|---|---|
| 2D | Two Dimensional |
| 3D | Three Dimensional |
| ANN | Artificial Neural Network |
| BSC | Back Sitting on Chair |
| CNN | Convolutional Neural Network |
| DNN | Deep Neural Network |
| FN | False Negative |
| FP | False Positive |
| IBM | International Business Machines Corporation |
| kNN | kth Nearest Neighbor |
| LSM | Least Squares Method |
| LSTM | Long Short Term Memory |
| MI | Mutual Information |
| NMI | Normalized Mutual Information |
| PCCR | Pearson Correlation Coefficient |
| ReLU | Rectified Linear Unit |
| TN | True Negative |
| TP | True Positive |
| SVM | Support Vector Machine |
| WHO | World Health Organization |
| XCORR | Cross Correlation |

# CHAPTER 1

# Introduction

## 1.1 Introduction

Falls are unfortunate events aggravating the risk for mortality, morbidity, disability, and frailty among the elderly community. Fall events hampers the independent motion of elderly people. Fall detection is a challenging task for some certain reasons. Lack of appropriate data and techniques have made fall detection system challenging. This chapter describes the thesis's background, motivation, problem statement and objectives.

## 1.2 Background

Falls are a significant health risk [1] and barrier to independent life for the elderly. The estimated fall incidence occurs every year people aged over 75 is at least 30% [2, 3]. According to World Health Organization (WHO), each year an estimated 684000 individuals die from falls globally. Over 80% of them are from low and low-middle income countries like Bangladesh. WHO also suggest that people over 60 years of age are the greatest sufferer of the fall event [21]. The study [4] has shown how age causes fall.

Fall detection and Human Activity Detection or Human Activity Recognition is sometimes referred as the same technique. Fall is a kind of unwanted activity. Fall can happen with other activities. Fall is considered as an abnormal activity [21]. The process of detecting fall occurrences using relevant data of a person is known as fall detection. There are several fall detection methods. They can roughly be divided into four categories based on the source of data. They are namely: life information-based methods, radar-based methods, wearable/mobile sensor based methods, vision based methods. Each of these strategies has benefits and drawbacks.

Life information based methods uses a person's current and historical medical information to predict a fall. It is cost efficient but do not provide any information related to real-time fall detection. This information is used to fix the threshold value in some cases [5]. This method is used less in compare to other methods.

In radar-based methods, a radar is placed at a certain distance from a target person and the target person moves at a certain direction [6]. The wave from the radar is taken to consideration for classifying the activity. However, this method is costly and it is only applicable for only one person in the scene.

Vision based fall detection system takes a steam of images or videos. Identify person from other objects and then detect fall and other activities. Vision based fall detection system is cost efficient. But in outdoor situation it is tough to install the requirements. It is quite difficult to get video or image information of a person’s movement in all the area he/she moves in a normal day. Vision based system is thus useful for indoor situation only.

Using wearable sensors or mobile sensor is a convenient way to obtain reliable motion data related to falls. Wearable/mobile sensor based methods collects accelerometer, gyroscopic and rotational data available from the sensors or mobile device. Accelerometer provide triaxial acceleration of the subject. Gyroscope measures the triaxial angular acceleration. These two sensors are commonly used for fall detection or activity recognition purpose. Many threshold based algorithm use these data along with a predefined threshold to detect fall event. Threshold based algorithms have some difficulties of calculating proper threshold for people of various ages, gender, weights, heights etc. Deep learning and machine learning based algorithms eliminates the curse of threshold. It uses a train dataset to train a model in supervised manner. The model determines proper weights and bias value for itself. This method is challenging because it is sometime difficult to find the proper parameter for the model, which features to use and objectively evaluating the model. In this study, machine learning techniques are used for fall detection along with some other common activities.

## 1.3 Motivation

This study motivates in the way that it can be helpful for the public health. Nothing can be more pleasant than helping people using the acquired knowledge. Fall event can cost lives of elderly people. After a fall event the victim normally lay down on the floor without any help. The person can be senseless at that period. This can cause long term health hazard even death. It has been reported that emergency departments treat 2.8 million older people for fall injuries every year [7]. Injuries like broken bones and serious head damage occurs in 20% of fall event [7]. Fall event cost roughly $31 billion in a year [7].  In the socioeconomic structure of our country, monitoring an elderly people is seldom performed. Thus automatic detection of fall event has become very necessary.

Sensors are publicly available now a days [34]. Every mobile phone/smart phone have the required sensors for fall or activity detection system. The number of mobile users is increasing day by day. Therefore, it is quite easy to get sensor data from mobile and use them for real time fall detection system. In this study, mainly fall detection method was focused rather than developing a system. Once the detection method is fixed then it will be easier to integrate into a system.

## 1.4 Problem Statement

The accuracy and robustness of a fall detection method is very crucial because human lives depend on it. Detecting fall from various activities can be challenging. There are some actions like jumping, running which can be thought as a fall event. Again, fall can be happened in any direction of the body. Therefore, a fall event can be confusingly detected as non-fall and a non-fall event can be detected as a fall event. Hence, the fall detection method should minimize the miss detection and detect all types of falls. Moreover, in this study, a very newly created dataset was used to train the model. A mobile with accelerometer and gyroscope sensor will be used for fall detection purpose. The mobile phone should be placed in the thigh pocket of the actor.

## 1.5 Objectives

This study aims to accurately detect fall along with other activities. The intention of this work is to use sensor data from mobile sensor kept in a certain position of body. The specific objectives of this study are to:

i. Extract feature from sensor data and create feature maps from the sensor data

ii. Detect fall with other activities using the feature maps generated from the sensor data.

iii. Evaluate the performance using standard performance measuring methods

iv. Compare the output of this study with the existing study.

## 1.6 Organization

This thesis has six chapters. The Chapter 1 provides and introductory discussion on the thesis’s background, motivation, problem statement and objectives as well. The rest of the thesis are organized as follow.

Chapter 2 discusses the literature review. It contains the previous recent works on fall detection. This chapter views different techniques used in the previous works.

Chapter 3 explains the used models and the generated feature maps from the sensor data.

Chapter 4 explains the proposed methodology and the significance of the present study.

Chapter 5 describes the experimental setup, experimental result and the performance measurement and performance comparison of this work.

Chapter 6 contains the conclusion with some potential direction for additional research.

# CHAPTER 2

# Literature Review

## 2.1 Introduction

Fall detection system can be classified into different categories based on different parameters such as source of data, detection technique etc. In this study, only sensor based methods are considered. Fall detection methods using sensor data can be categorized into three groups [8]. They are: threshold based methods, machine learning based method and threshold and machine learning based method.

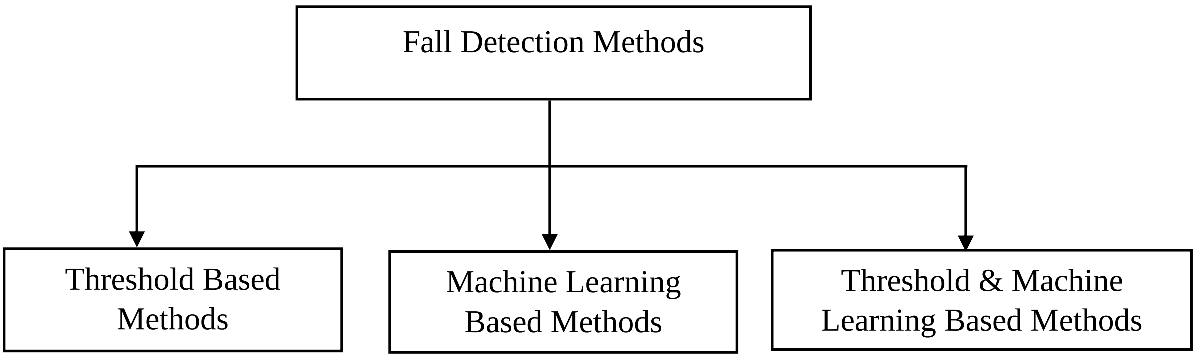



Figure 2.1: Fall Detection Methods Classification

## 2.2 Threshold Based Fall Detection

Threshold based fall detection methods use a predefined value for a specific event. It requires less computational effort. The accuracy of this type of methods largely depends on the threshold value. Figure 2.2 shows the block diagram of threshold based fall detection methods. It takes the sensor data, performs some calculation on the sensor data and compare with the predefined threshold value. The final result is based on the comparison of the threshold and input data.

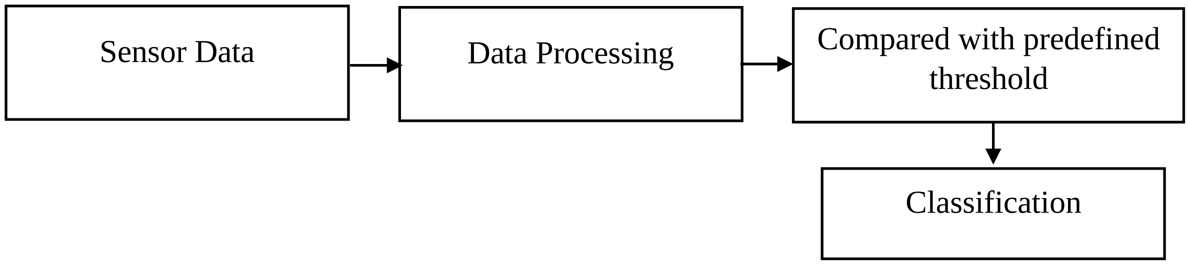



Figure 2.2: Threshold Based Fall Detection System Overview.

Luis N. et al. [9] uses accelerometer and gyroscope data to detect fall event. They select the location of the device. They give the freedom of carrying the device. The device can be placed in wrist, chest pocket, side pocket, pants' pocket, talking position, texting position etc. The x, y, z values of accelerometer and the magnitude of gyroscope were used to identify the position of the device. The location is selected based on some decision made using the sensor values. Depending on the location, a threshold is fixed. This threshold value is used for fall detection. They got different accuracies in different position of the device. The highest accuracy was 95.8% in texting position. The main drawback of their method is the threshold. Threshold should be different for people of different height. Jiangpeng Dai et al. [10] used mobile phone as a platform of pervasive fall detection. They developed PreFallD system. The system used accelerometer data to detect fall. If the data satisfies a preset condition, then the fall detection algorithm is triggered. They calculated the $|A_T|$ and $|A_V|$. The difference between the maximum and minimum value within the triggered time is compared with a threshold value. If $|A_T|$ and $|A_V|$ are less than the threshold then they classified it as fall. The performance measured as FP and FN. The FP was 8.7% and the FN is 2.67%. Arkham Zahri et al. [11] uses accelerometer and gyroscopic data to detect fall. They detect fall with other Activity Daily Living (ADL). They first found the maximum and minimum values from accelerometer and gyroscope. If the value overtakes a predefined threshold, then the angle of the device is considered for fall detection. They acquired relatively high accuracy compared to other studies. They got maximum of 96.67% and minimum of 86.67% for different kinds of fall. Falin Wu et al [12] used only accelerometer data to detect fall. They considered the square root sum of the triaxial value of accelerometer and the angle calculated from the accelerometer value. The rotation angle value of before and after fall is calculated for fall detection. The test sensitivity and specificity of the threshold based algorithm was 91.6% and 88.7% respectively. Jin-Shyan et al. [13] developed enhanced threshold based fall detection algorithm. Their method could detect four different types of falls. The device collect data by keeping the device in the front pocket of pant. They used three predefined thresholds. Different thresholds were used to detect different activities. Their method could detect other ADLs along with fall. They acquire 96% accuracy.

The main problem with the threshold based algorithm is defining the threshold value. The threshold value should vary from person to person and place to place. The accuracy of threshold based algorithm largely depends on proper threshold.

## 2.3 Machine Learning Based Fall Detection

According to IBM [14], machine learning is the branch of artificial intelligence and computer science which deals with data and algorithm to learn like human. Machine learning learns from previous data and predict on unseen data (supervised learning). In the study of fall detection, machine learning method eliminates the problems with threshold based method. It produces better result than threshold based method.

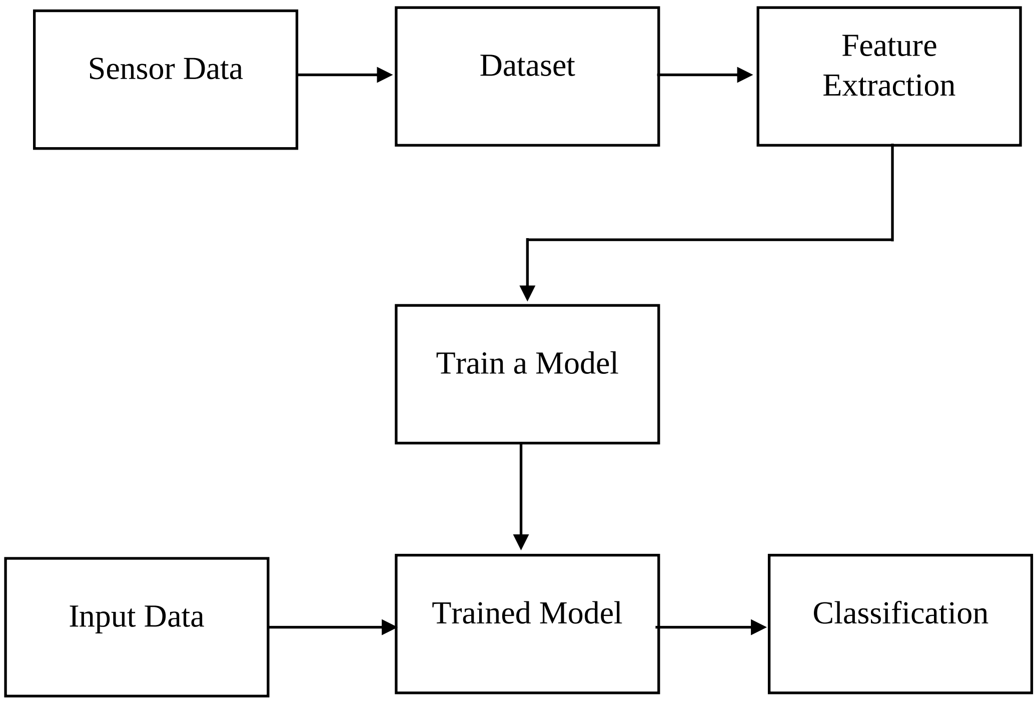



Figure 2.3: Machine Learning Based Fall Detection System Overview.

From the figure 2.3 it is shown that first a dataset is created using the sensor data. Then relevant features are extracted to train a model. Once the model is trained, it gets appropriate weights and bias value. Then it can classify activities on unseen data. The machine learning method is computationally expensive than the threshold based method.

John C. et al. [15] introduced a novel fall detection method in two steps. They collected five daily activities namely, walking, jogging, standing, sitting, lying. A fall was also performed after each activity. They collected five sensor data from the smartphone, accelerometer, gyroscope, magnetometer, gravity, and linear acceleration. In the first step, they identified the correct type of fall from multiclass classification. In the second step they produced binary classification using the multiclass perception. Five different machine learning classification models were used: the Support Vector Machine (SVM), k-Nearest Neighbor (kNN), Naïve Bayes, Decision Tree and Random Forest. Among all the classifier the SVM produced more accurate result. Their got maximum accuracy as 95.65%. Pranesh et al. [16] used MobiFall dataset to introduce a fall detection system using machine learning. They used feature selection method to reduce the dimensionality of the dataset. They extracted statistical features: mean, maximum, minimum, standard deviation etc. in both time domain and frequency domain. They extracted these features for each axis of the triaxial accelerometer data. They implemented five different algorithms: Support Vector Machine (SVM), k-Nearest Neighbor (kNN), Naïve Bayes, Artificial Neural Network (ANN) and Least Squares Method (LSM). They got relatively low accuracy than other related works. The best accuracy was 87.5% in case of kNN. Jian et al. [17] introduced a portable fall detection system using k-Nearest Neighbor (kNN). The magnitude of both accelerometer and gyroscope was extracted to fit in the kNN model. Their system discriminated falls from other ADLs with a sensitivity of 95% and a specificity of 96.67%. Ahmet et al. [18] used machine learning techniques to separate falls from ADLs. They used six machine learning classifiers: kNN, SVM, LSM, Bayesian decision making (BDM), Dynamic Time Warping (DTW) and ANN. They first extracted features from the sensor data. Then Principal Component Analysis was used to find the useful features. These features are used to the classifier to detect the fall. Among all the machine learning algorithms, the k NN produces the best result with the sensitivity, specificity and accuracy above 99%. Saminda et al. [19] introduced semi-automated feature extraction for fall detection. They implemented the k-means clustering algorithm as the semi-automated extractor of training example from motion data. Their used dataset contains four types of fall data and seven types of non-fall data. They proposed classification networks as the combination of neural networks and softmax regression. Their softmax layer identified 87.17% fall from non-fall events and the neural network layer identified 100% fall from non-fall events.

Deep learning based techniques are also used now a days. Transfer learning technique is also used. The main objective of transfer learning is to learn a model for a large dataset and then transfer the acquired knowledge to solve a specific problem. Frédéric Li et al. [20] used deep transfer learning technique for sensor modality classification. They used a large amount of time series data to firstly train a Deep Neural Network (DNN). This network can learn general characteristics of the used time series data. Then the acquired knowledge was used in another DNN designed to solve the target problem. Neeraj et al. [25] uses CNN and LSTM separately and then used CNN and LSTM combinedly. They showed that the combined architecture produces better result.

## 2.4 Machine Learning and Threshold Base Fall Detection

The combined machine learning and threshold based method both classify using machine learning models and compare with threshold value. The general block diagram is given below.

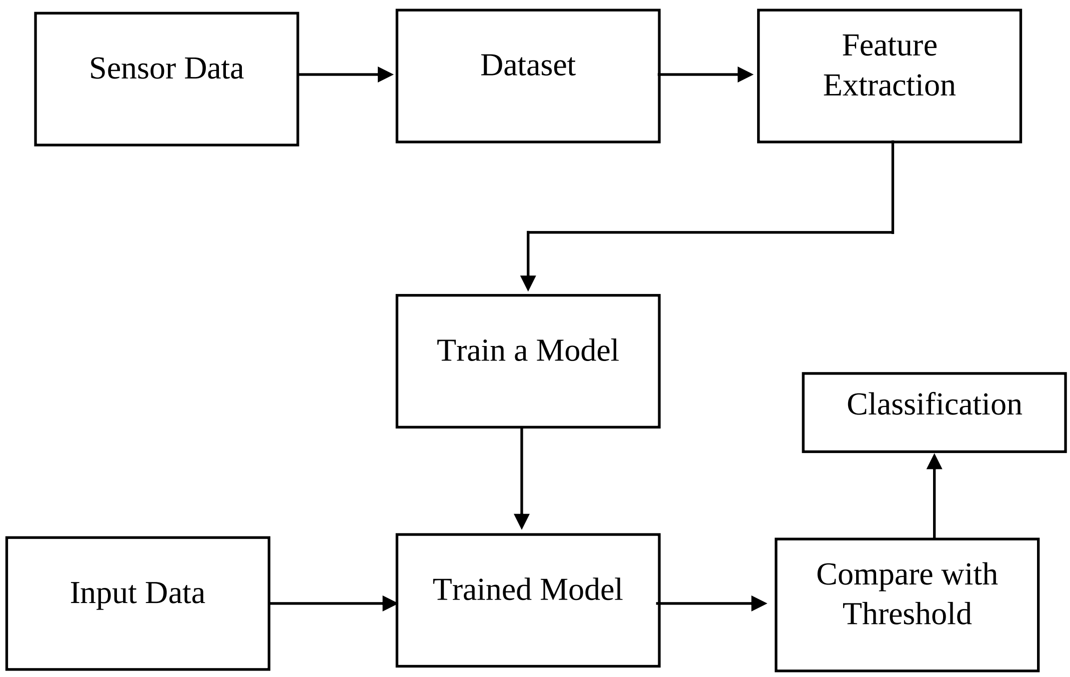



Figure 2.4: Machine Learning and Threshold Based Fall Detection System Overview.

Panagiotis et al. [27] used both threshold based and machine learning model combinedly for fall detection. They use kNN as the classifier. The sensitivity and specificity of the system were 97.53% and 94.89%, respectively. Stefan et al. [28] introduces fall detection system using both threshold and machine learning models. For machine learning model, they used fuzzy logic and AdaBoost. R. L. Victor et al. [38] proposed High-Level Fuzzy

Petri Net method on mobile sensor data for fall detection. They used z axis data of the G-sensor and acquired 80% accuracy.

## 2.5 Conclusion

The reviewed system mostly used kNN, SVM, LSTM models on different dataset. These models produced satisfactory results. Deep learning models like CNN are used less. In [29] CNN was used with LSTM and produces a comparative result. ConvLSTM produced better result in [30] rather than separate CNN and LSTM models. Moreover, there are a large number of previous studies where threshold based methods are used. In this study, mainly deep learning based method like CNN are used along with LSTM.

# CHAPTER 3

# Theoretical Consideration

## 3.1 Introduction

In this study, fall detection method is done by constructing feature map from the sensor data. In the recent studies, it has been shown that they used statistical features of the sensor data. But in this work correlations among different sensor values are considered. Five features are extracted from the sensor data. Two sensors, accelerometer and gyroscope are used to get the sensor data. Both of these sensors produce triaxial data. Accelerometer produces acceleration in x, y and z axes [35]. While gyroscope produces angular velocity in x, y and z axis. The axis distribution of mobile phone is shown in figure 3.1. The y axis is along the height of the mobile phone and the x axis is along the width of the mobile phone. The z axis is the perpendicular to the xy plane i.e., the perpendicular to the screen of the mobile phone.

The accelerometer thus measures the acceleration of the mobile phone’s triaxial direction. The gyroscope measures the angular velocity of the three axes. In this study, the correlation features between two axis value is calculated and a feature map is constructed using these values. Then the feature map is used to a classification model for classifying the fall and other activities. This chapter describes the feature maps and the classifier used in this study.

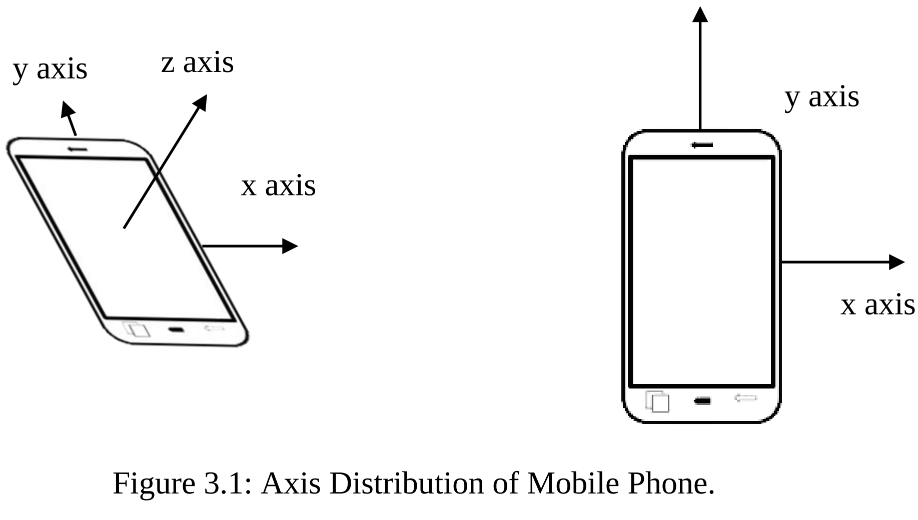



Figure 3.1: Axis Distribution of Mobile Phone.

## 3.2 Feature Extraction from Sensor Data

Feature extraction is an important part of fall detection. Sensor data follows a certain pattern. Recognition of this pattern can make the fall detection process easier. In the previous works, mainly statistical features were extracted. Statistical features are used for fall detection in [16]. Statistical features can be extracted from a single signal. In this study, some connectivity features are extracted.

Statistical features are time domain features (e.g., minimum, maximum, kurtosis, mean, skewness, range etc.) [24]. These features can detect fall but is not robust to anti-noise platform. There are some other features in frequency and time frequency domain. All of these features are measured for individual signal. These features do not imply any relationship among the signals.

The connectivity features are measured between two signals. There are some connectivity features: Cross Correlation (XCORR), Pearson Correlation Coefficient (PCCR), Mutual Information (MI), Normalized Mutual Information (NMI), Mean etc. In this study, these five connectivity features are used to generate the feature map and the feature map is used to detect fall. Only one statistical feature, mean is used in this work.

## 3.3 Fall Detection Using Features

Previous fall detection methods used statistical features. In [25] external features were combined with the sensor data for human activity recognition. The statistical features were used in a classification model including SVM, kNN, DNN, CNN etc. In this study, connectivity features are used for fall detection. The literature review section describes different fall detection methods with their accuracy. In this study, connectivity features were extracted and then used for classification purposes. Feature map was generated using the connectivity features and then passed to a classifier. The classifier detects the activity.

## 3.4 Feature Map

A feature map is constructed by calculating the values among two signals of the sensor data. For example, a cross correlation among the acceleration x axis and acceleration y axis, among x axis and z axis, among acceleration x axis and gyroscope y axis value and so on. The accelerometer and gyroscope sensors of the mobile phone produces triaxial values. So total 6 values are available from these two sensors. A magnitude value is calculated for both sensors using the following equations. Then the first derivation of the signals is calculated. Thus total 16 signals are found from the sensor data.

$$\text{Acceleration Magnitude, } A=\sqrt{Ax^2+Ay^2+Az^2} \qquad (3.1)$$

$$\text{Gyroscope Magnitude, } G=\sqrt{Gx^2+Gy^2+Gz^2} \qquad (3.2)$$

$$\text{First Derivation of the Signal, } Jx = \frac{dx}{dt},\ Jy = \frac{dy}{dt} \text{ and } Jz = \frac{dz}{dt} \qquad (3.3)$$

Here Ax, Ay and Az represents the x, y and z axis value of the accelerometer and Gx, Gy and Gz represents the gyroscopic value. The first derivative signal is calculated for both accelerometer and gyroscope.

In this study, five features are extracted: XCORR, PCCR, MI, NMI and MEAN.

### 3.4.1 Cross Correlation (XCORR)

The cross correlation is the comparison of the two different time series to detect if there is a correlation between the two signals with the same maximum and minimum values. It calculates the linear correlation between two signals. In a word, it measures the similarity of the two signals. The equation for calculating cross correlation is written as:

$$XCORR_{XY}(t) = \begin{cases} \sum_{n=0}^{N-t-1} x_{n+t}\,\bar{y}_n & \text{if } t \geq 0 \\ XCORR_{YX}(-t) & \text{if } t<0 \end{cases} \qquad (3.4)$$

where t = (-N-1)……., -2, -1, 0, 1, 2, ……., (N-1). Here t denotes the lag or time shift parameter. The $XCORR_{XY}$ (t) denotes the cross correlation between the signal X and Y with lag t. The range of this measurement is -1 to 1.

### 3.4.2 Pearson Correlation Coefficient (PCCR)

For two signals X and Y Pearson Correlation Coefficient measures the linear correlation between the signals [32]. The range of this correlation coefficient value is – 1 to 1. This value also denotes the direction of correlation. If the value is greater than 0 then both signals are positively correlate and if the value is less than 0 then they are negatively correlate. If the value is 0 then there is no correlation between the two signals. The formula for calculating Pearson Correlation Coefficient is written as:

$$PCCR_{X,Y} = \frac{n\sum X*Y-\sum X*\sum Y}{\sqrt{n\sum X^2-(\sum X)^2}\sqrt{n\sum Y^2-(\sum Y)^2}} \qquad (3.5)$$

Here n is the number of samples. The X and Y denotes the individual component of the two signals.

### 3.4.3 Mutual Information (MI)

The amount of information about one random variable that can be learned by observing another variable is measured as mutual information [33]. The mutual information of two random variable X and Y is defined as:

$$MI\ (X, Y) = H(X) + H(Y) - H(X, Y) \qquad (3.6)$$

Here H denotes the Shannon entropy [23]. H(X) and H(Y) represents the marginal entropies of individual X and Y respectively. H (X, Y) represents the combined entropy. Mutual Information is nonnegative. The range of Mutual information is $0\leq MI\ (X, Y)< \infty$. A MI (X, Y) of 0 denotes that X and Y are independent. Mutual Information is a symmetric measurement that is MI (X, Y)= MI (Y, X).

### 3.4.4 Normalized Mutual Information (NMI)

The Mutual Information described above don't have the upper limit. The range of Mutual information is $0\leq MI\ (X, Y)< \infty$. Sometimes it is better to use the normalized the mutual information. The mutual information value is measured in the range of 0 to 1. The Normalized mutual information is defined as bellow:

$$NMI(X,Y)= \frac{MI(X,Y)}{H(X)+H(Y)} = \frac{H(X)+H(Y)-H(X,Y)}{H(X)+H(Y)} =1- \frac{H(X,Y)}{H(X)+H(Y)} \qquad (3.7)$$

The Normalized Mutual Information can be thought as the degree of dependence. The value 0 means independence and 1 means dependence (strong dependence).

### 3.4.5 Mean

Mean is a statistical measurement. It denotes the average of a series of data. The mean is calculated as the sum of all values divided by the number of values. In this study, the mean is treated is a different manner. Here mean of two different signal is calculated. The mean of X and Y is calculated as:

$$M\ (X, Y) = \frac{SUM(X)+SUM(Y)}{n+m} \qquad (3.8)$$

Here n and m denote the number of values X and Y have respectively.

## 3.5 Convolutional Neural Network

A Convolution Neural Network (CNN) is a Deep Learning algorithm with an input layer, several hidden layers, max-pooling layers, flatten layers and dense layer[31]. It takes inputs and assign learnable weights and bias to every aspect of the input and be able to differentiate one from another. The architecture of CNN is similar to the connectivity of the human brain. CNN extracts feature from the input and pass from one layer to another. In the recent years, CNN has become a widely used classifier. The first layer of a Convolutional network is the input layer. All the other layer in between the input and output layer is known as hidden layer. A layer consists of a certain number of neurons. The figure 3.2 shows a general architecture of a CNN network. Every layer of the CNN network has a kernel matrix. The kernel is a small array of number that is used across the input. For each cell in input the product of input and kernel is summed up. The output of convolution is called feature map. Figure 3.3 shows a convolution process with a dummy example. The convolution process is monitored by the size of kernel and the number in kernel. The convolution doesn’t allow to overlap the kernel to input. Thus, the dimension of the feature map is reduced. This problem is addressed by padding. Stride is defined as the distance between two successive kernel position. In moder CNN model padding is chosen as 0 and stride is chosen to 1. The process of training a CNN model with regard to the convolution layer is to identify the kernels that work best for a given task based on a given training dataset.

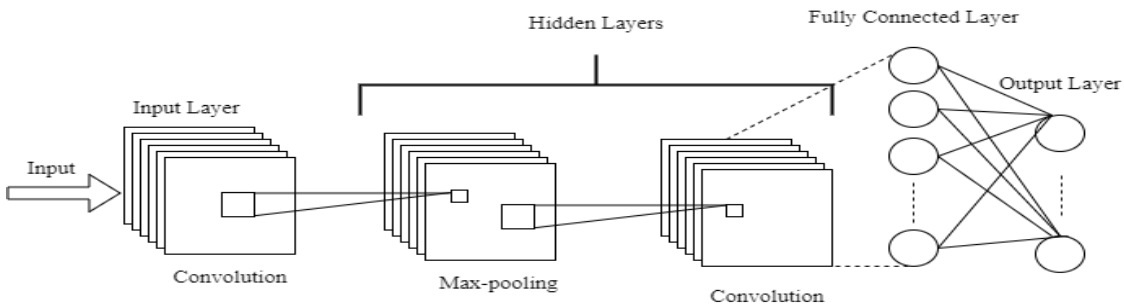



Figure 3.2: Building Blocks of CNN.

Batch size is also an important parameter for a CNN model. It has a great effect on the performance of the model. A large batch size allows more data to be calculated parallelly with more memory cost but saves computation time. A smaller batch size allows to train on a diverse set of samples. It helps in better generalization performance but requires more time.

Training of a CNN model refers to the process of finding proper kernel and weights based on the input data. Backpropagation is the process of reducing the loss and converge to the global minima. The effectiveness of a CNN model is measured by the loss function and the forward propagation. The layers of CNN and other necessary parts are described in the following sections. CNN can be one, two, three or multi-dimensional. The output of CNN model is passed through an activation function. There are several activation functions with different characteristics. As a summery classification using a CNN involves training the model on a labeled dataset, learning to extract relevant features from the dataset, and using these features to make predictions on new unseen data.

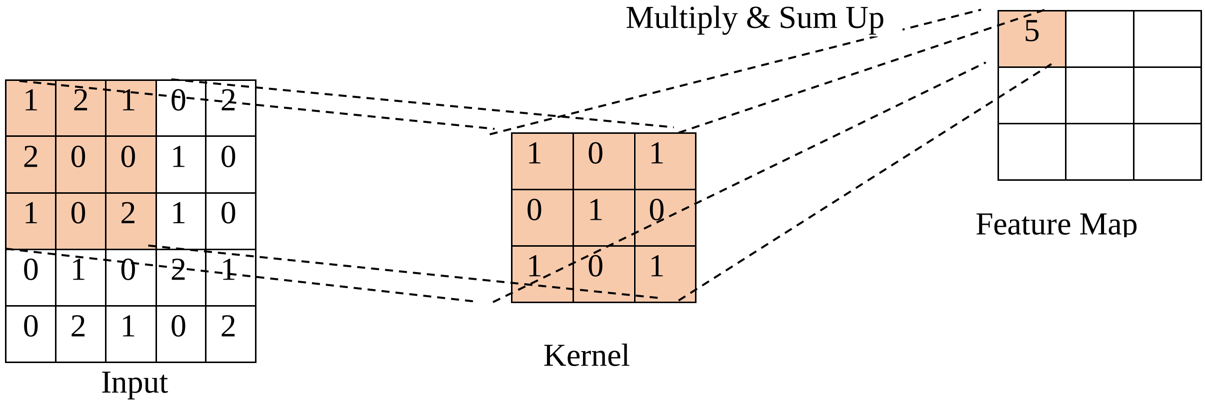



Figure 3.3: Example of Convolution Operation.

### 3.5.1 Input Layer

Input layer is the first layer of a CNN network. This layer is responsible for accepting the input. It passes the input to the subsequent layers. It acts as a placeholder for the input data. The input layer is defined by the dimension of input data.

### 3.5.2 Hidden Layer

Hidden layer is any layer that is not an input layer or an output layer. There may be one or more layers in between the input layer and output layer. These layers are known as hidden layer. Hidden layer extract important features and pass it to subsequent layers. Each neuron of a hidden layer is connected to the neurons of the previous layer. Each neuron applies a linear calculation of the input values and followed by a non-linear activation function. In each layer of the hidden layer the convolution operation is performed if the hidden layer is a convolution layer. The convolution operation is shown in figure 3.3. The number of hidden layer and the number of neurons in each hidden layer largely impact the accuracy of the model. Complex network requires more parameters. In figure 3.2 the hidden layer is marked. More hidden layers may cause the overfitting problem.

### 3.5.3 Dense Layer

Dense layer of a neural network is a fully connected layer. The term fully connected means that every neuron of a certain layer is connected to every neuron of the previous layer. The dense layer is used to make the prediction based on the processed input data from the previous layer. It allows the neural network a non-linear relationship between the input and output layer. The capacity of the model is increased by adding more neurons to the dense layer [40], but this might also raise the possibility of overfitting, which is when the model performs well on training data but badly on new, test data.

### 3.5.4 Max-Pooling Layer

Similar to convolutional layer, pooling layer is also responsible for reducing the spatial dimension of the convolved features. This layer help in decreasing computational power by reducing the dimension. There are two pooling layers: max pooling and average pooling. The max pooling layer outputs the maximum inside the kernel while average pooling outputs the average. In most cases the max pooling is used. Figure 3.4 shows the max pooling operation.

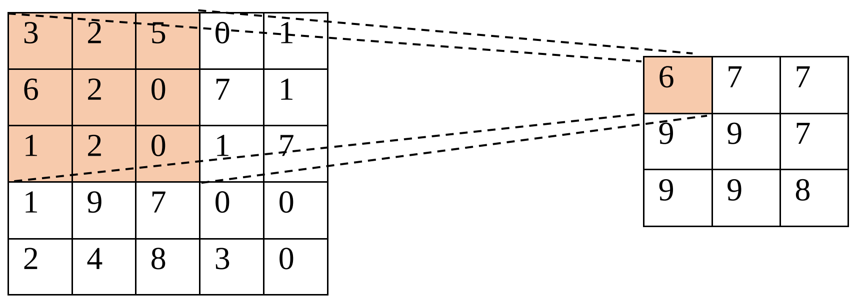



Figure 3.4: Max Pooling Operation.

Max-pooling is typically performed using a pooling window of fixed size and stride the pooling operation is performed independently for each feature map in the input. The pooling window is colored in the figure 3.4.

### 3.5.5 Dropout Layer

The dropout layer is a regularization technique used in Deep Neural Networks, including CNNs. The main purpose of dropout layer is to prevent the overfitting problem. In dropout layer, some neurons of the network are randomly dropped. This makes the neuron to depend on other neurons to make prediction and prevents any single neuron to from having too much influence on the final output.

### 3.5.6 Activation Function

The activation function decides whether a neuron should be activated or not on the basis of calculating the weighted sum and a bias value. In a word, activation function decides whether a neuron is important or not. The purpose of an activation function is to add non-linearity to the neural network. There are three types of activation functions: binary step, linear and non-linear activation function. Non-linear activation function is mostly used. The non-linear activation function also has different types. Some of them are: sigmoid/logistic, tan hyperbolic, rectified linear unit (ReLU) [36].

The sigmoid/logistic activation function converts the input into 0 to 1 range. Therefore, it is specially used in tasks related to probability. The mathematical expression of sigmoid/logistic activation function is given below:

$$f(x)= \frac{1}{1+ e^{-x}} \qquad (3.9)$$

The softmax function is a more generalized logistic function used for multiclass classification purposes.

The tanh or tan hyperbolic function converts the input in the range -1 to 1. The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph. The mathematical expression of tanh function is given below:

$$f(x)= \frac{e^{x}- e^{-x}}{e^{x}+e^{-x}} \qquad (3.10)$$

It performs better than sigmoid function. Tanh function mainly used for binary classification. Both sigmoid and tanh function are used in feed forward networks. The output of tanh function is zero centered.

The Rectified Linear Unit (ReLU) gives the impression of a linear function. The ReLU activation function is commonly used now a days. It is allowed to use for backpropagation and simultaneously make it computationally efficient. ReLU function can be expressed as below:

$$f(x) = \max(0, x) \qquad (3.11)$$

ReLU does not allow all the neurons to activate. Thus, ReLU is computationally efficient because it reduces the number of neurons.

### 3.5.7 Flatten Layer

In Convolutional Neural Networks (CNNs), the Flatten layer is a pre-processing layer that transforms the multidimensional output of a convolutional or pooling layer into a one-dimensional vector. This is required because convolutional or pooling layers in a neural network expect a multidimensional output, whereas fully connected layers in a neural network demand a one-dimensional input (height, width, and depth). The Flatten layer essentially transforms the data into a one-dimensional array from the output of a convolutional or pooling layer. The Flatten layer rearranges the elements so that they can be utilized as input to a fully linked layer rather than altering the number of items in the data.

### 3.5.8 Output Layer

The final layer of a neural network is the output layer, which offers predictions for a particular task. Depending on the kind of problem being solved and the neural network's architecture, the output layer can take many different shapes. The output layer of classification problems frequently consists of a single neuron with a softmax activation function. The output neuron's raw scores are translated by the softmax function to a probability distribution across all conceivable classes.

### 3.5.9 Optimizer

Optimizer is a function or an algorithm that adjusts the parameters of a neural network. It modifies attribute such as weights, bias. Optimizer helps in reducing overall loss and improve the accuracy. A loss function measures the difference between the actual value and the predicted output. The goal of an optimizer is to minimize the loss by adjusting the weights and bias [37]. The right choose of optimizer is problem specific and depends on the nature of data and the architecture of the neural network. There are some optimizers with different characteristics such as Adam, Adagard, AdaDelta, Gradient Descent etc.

The term Adam is derived from Adaptive Moment Estimation. It is one of the most popular and widely used optimizer in neural network.

### 3.5.10 Loss Function

In neural network, the loss function measures the difference between the actual and predicted labels [39]. Loss function mainly reduces the loss. Some of the loss functions are: binary cross-entropy, categorical cross-entropy, sparse categorical cross-entropy. In this study, sparse categorical cross-entropy is used mostly.

## 3.6 Conclusion

Convolution Neural Network is a widely used classifier in the modern time. Recent research on fall detection or human activity detection used statistical features on different machine learning models. In this study, feature map generated from sensor data was used for fall and other activity detection.

# CHAPTER 4

# Proposed Methodology

## 4.1 Introduction

In this study, a feature map based fall and other activity detection methodology is proposed. The feature map is generated from the sensor data (accelerometer and gyroscope). Five features are extracted between the signals and each of size 16 x 16. The feature map is then used in a classifier for classifying the activity.

## 4.2 Working Procedure

In this section, the working procedure of fall and other activity detection technique is discussed. Firstly, the sensor data of human activity is collected using mobile phone sensors. Only accelerometer and gyroscopic data are collected. The mobile phone is kept in the pants pocket of the actor.

Then magnitude signal and jerk signal are generated. The process of generating the magnitude and jerk signal is described in section 3.2. The accelerometer and gyroscope both produce triaxial data. Thus, there are six data from these two sensors. Magnitude signal is generated for both accelerometer and gyroscope. Hence, two more signal is added. Now the number of signals is eight. Then the jerk signal is calculated for each of the eight signals. Therefore, total 16 signals are there.

The accelerometer and gyroscope don't produce same number of signals. For finding the desired features, the number of samples in two considered signal should be same. For this reason, some data processing is performed. The data processing part make the accelerometer and gyroscope signal equal size.

Then the 5 features are generated for each of the 16 signals. This process generates 5 such 16 x 16 feature maps. This feature maps are then used in the classifier for classification. Figure 4.1 describes the working procedure of this study is summary.

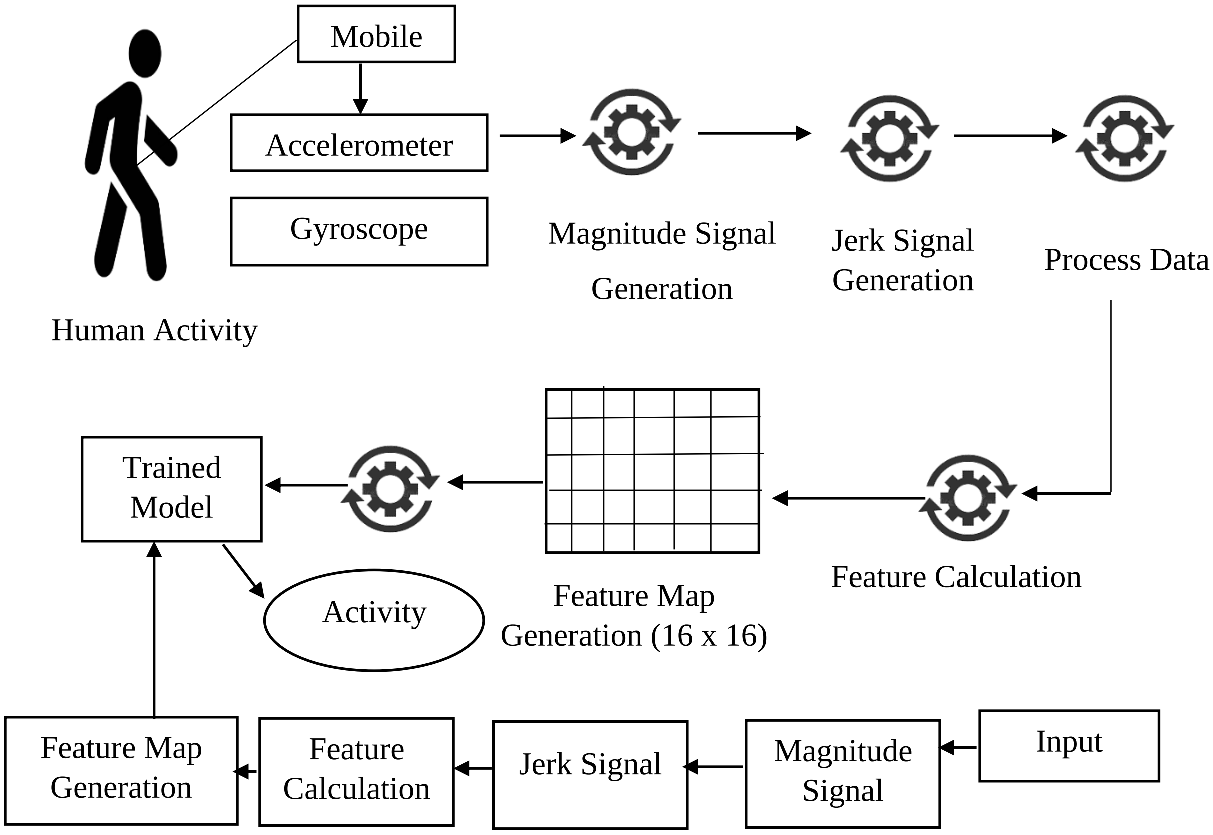



Figure 4.1: Working Procedure of This Study.

## 4.3 Fall Detection Using Feature Map

The generated feature map was used individually and combinedly in different architecture of classifiers. The major difference of these classifiers is in the input size. The feature maps are grouped in two, three, four and five.

Among the Deep Learning methods, CNN is the most successful classifier for two-dimensional data. The feature maps generated in this study are all two-dimensional data. That's why CNN was chosen as suitable classifier. The CNN architecture consists of an input layer, sever hidden layer, flatten layer, fully connected dense layer and output layer. Three convolution layers, one max-pooling layers, a flatten layer, a dense layer and an output layer constructs the CNN architecture for individual feature map. Figure 4.1 shows the CNN architecture of used in this study. The input contains 16 x 16 feature map. The first layer of the CNN is the input convolution layer. The feature size of the input layer is 16. The kernel size of the input layer is 7 x 7. There is a dropout of 0.25 for the second and third convolution layer. The input shape for the input layer is (16, 16, 1).

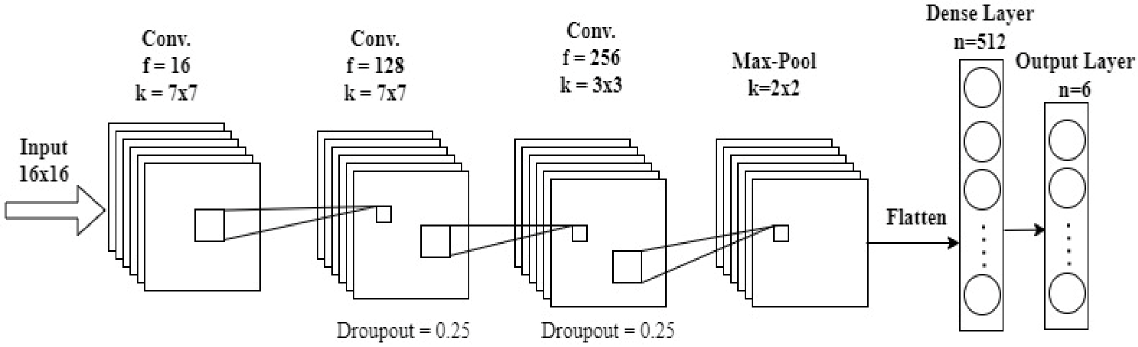



Figure 4.2: Configuration of the CNN model for fall detection in this study.

The padding and stride are kept as normal for each convolution layer. Rectified Linear Unit (ReLU) was used as activation function. The max-pool layer consists of kernel size of 2 x 2 with stride=2. The next or second convolution layer contains 128 filters. The kernel size is as same as the input layer. The third convolution layer contains filter size of 256 with kernel size 3x3. Then there is a flatten layer and a dense layer. The flatten layer converts the convolution matrix into a single column. The dense layer contains 512 nodes. Finally, the output layer contains 6 nodes for 6 classes. For the output layer sigmoid softmax activation function was used with l2 regularization of 0.01. The architecture in figure 4.2 was used for individual feature map classification. A 3D CNN architecture of the same with two convolution layers, one max-pooling layer, one flatten layer and a dense layer was built to classify using the combined feature map. Figure 4.4 shows the architecture of the 3D CNN model for combined feature map.

**LSTM Model**

A bidirectional LSTM [30] model architecture shown in figure 4.3 was used for individual feature maps only. It consists of a bidirectional LSTM layer, a dense layer and an output layer.

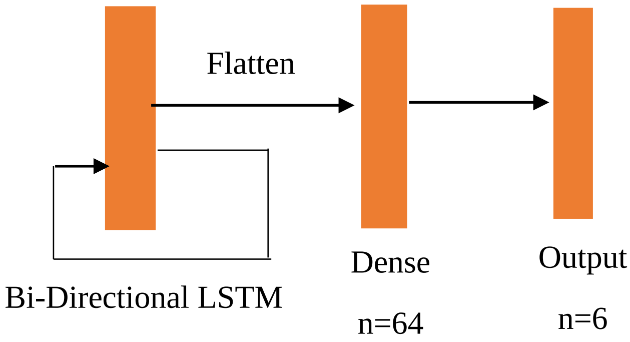



Figure 4.3: Bidirectional LSTM used in this study.

## 4.4 Fall Detection Using Individual Feature Map

In this study, five feature maps are generated which has been described in the previous chapter. The feature maps were used as the input of the classifier. The feature maps were used individually. For individual feature map different accuracies were found. Figure 4.2 and figure 4.3 shows the building block of the used classifier.

## 4.5 Fall Detection Using Combined Feature Map

Feature maps are combined in two, three, four and five. For example, mutual information and normalized mutual information are combined into on. Thus two, three, four and five feature maps are grouped into one. The five feature maps were used individually and by combining them in a single one. The combined feature map increases the dimension from 2D to 3D. A For combined feature map a 3D CNN architecture was used. It consists of two convolution layers, a max pooling layer, a flatten layer and a dense layer. Figure 4.4 shows the architecture of the 3D CNN. The input shape varies for different combination. For a combination of two feature map the input size is 2 x 16 x 16 again for combination of three feature map 3 x 16 x 16.

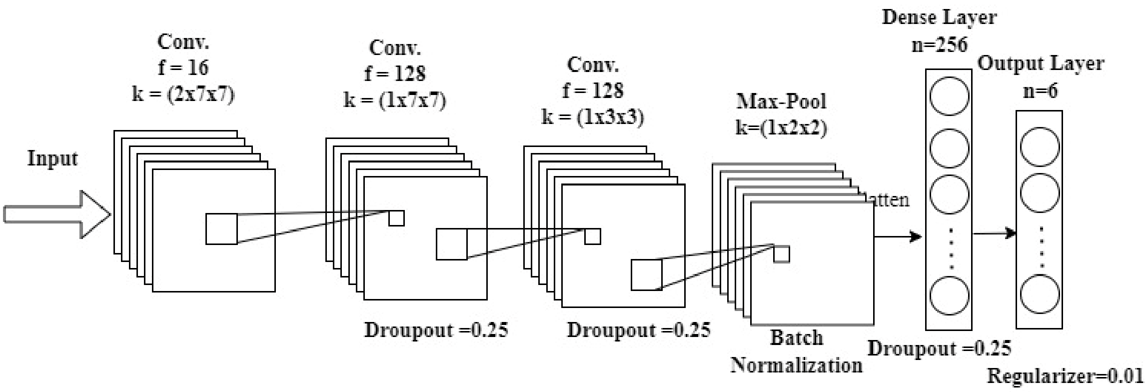



Figure 4.4: 3D CNN model used in this study.

## 4.6 Conclusion

In this study, connectivity features were used in terms of feature map for classification process. This type of feature map is first to use for activity detection purpose. Previous fall detection methods used classical machine learning models and statistical features. In this chapter, the methodology of this study is described.

# CHAPTER 5

# Experiments and Results

## 5.1 Introduction

This chapter describes the experimental outcomes of the fall detection system for the feature maps created with XCORR, PCCR, MI, NMI and Mean individually generated from mobile sensor data. The experimental analysis is shown for the new dataset created for fall detection purpose. The feature maps were used individually as the input of the CNN for classification. Finally, the obtained results of this work were compared with other acknowledged approaches in the literature review section to validate the present study.

## 5.2 Experimental Setup

The models were trained by the Adam optimizer and sparse categorical cross-entropy was used as the loss function. Learning rate, batch size and epochs for the classifiers were set to 0.00001, 32 and 5000. From the available data 20% data was reserved as testing set. Two deep learning frameworks, Keras and Tensorflow, available in Python, were used for training the model. The device configuration is:

### 5.2.1 Hardware

For fall detection system:

- CPU: Intel (R) Core (TM) i5-8250U CPU @ 1.60GHz   1.80 GHz
- RAM: 8.00 GB
- GPU Nvidia Geforce MX130 2 GB
- Operating System: 64-bit windows operating system

For dataset creation:

- Mobile Phone Model: Redmi Note 5
- UI Version: MIUI Global 11.0.3
- Android Version: 9 PKQ1.180904.001
- CPU: Octa-core Max 1.80 GHz
- RAM: 4 GB

### 5.2.2 Software

In this study, Python was used as programming language. Google Colab was used as code editor and online code executor. Some Python libraries such as Numpy, Sklearn, Scikit-learn, Pandas etc. were used. The Keras and Tensorflow was used to implement the neural networks. Android Studio was used to develop an android application for data collection process. Android Studio was used on the same hardware configuration used for fall detection system.

## 5.3 Dataset

In this study, a new dataset was created. There are six actions: Fall, Sitting, Laying, Standing, Walking. The purpose of creating the new dataset was to clearly detect the pre-fall and post fall activities from a fall event. Five volunteers of different age, weight and height performed the six actions. During the data set collection phase, the mobile device was kept in the pants' pocket of the actor. The reason behind keeping the mobile phone into the pants' pocket is keeping in mind that human is usually used to keep their phone inside the pants' pocket. Moreover, keeping the mobile device inside the pant pocket reduces the extra pressure for carrying the device. The actor acts as normal human for every activity. The mobile device was kept in different orientation inside the pocket for creating variety of data.

The dataset collection process consists of three steps. First the actor has to go to the application developed for data collection purpose. Then select an action he/she going to perform and start action. There is a waiting phase around 3 seconds. In the meantime, the actor has time to keep the mobile device inside the pocket. After 3 seconds, the data will be automatically collected using the accelerometer and gyroscope of the mobile device with a vibration. Then the actor will perform the certain action. The data is saved as a CSV file inside the mobile device storage.

The data was collected a normal sampling rate of the sensors. The data was saved with a timestamp. The timestamp is a dummy time and it is continuous for a particular action. The timestamp if further used for calculating the jerk signal. The accelerometer data file and gyroscope data file are saved with separate name. The name format is Action_ac_number_of_file.csv for accelerometer data. And for the gyroscopic sensor data the format was Action_gy_number_of_file.csv. Both files have 4 columns: time, ac_x/gy_x, ac_y/gy_y and ac_z/gy_z. The ac denotes that it is an accelerometer data and the gy denotes that it is a gyroscopic data. The x, y and z denote the axis.

The dataset contains 6 different actions. The amount of data for each action is mention below:

Table 5.1: Number of data for each activity.

| **Activity** | **Number of data** |
|---|---|
| Fall | 206 |
| Walk | 222 |
| Standing | 218 |
| Laying | 247 |
| Sitting | 250 |
| Back Chair Sitting (BSC) | 253 |

The data set was created being inspired by [26].

## 5.4 Performance Evaluation Terms

The performance is measured in terms of accuracy, precision, recall and F1 score. Four statistical measurements: true positive (TP), false positive (FP), true negative (TN) and false negative (FN) were used for this measurement.

**Ture Positive (TP):** True positive is the measurement of the how many predicted values are accurate i.e.; the predicted output is equal to corresponding actual value and the predicted class is positive and the actual class is also positive.

**False Positive (FP):** False positive is the number of predicted classes as positive but the actual classes are negative.

**True Negative (TN):** Measures how many negative classes are measured as negative

**False Negative (FN):** Measures how many positive classes are predicted as negative.

The above 4 statistical measurements are used to create the confusion matrix. The confusion matrix is constructed as follow:

Table 5.2: Confusion Matrix.

| Labels | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positive (TP) | False Positive (FP) |
| Predicted Negative | False Negative (FN) | True Negative (TN) |

Accuracy, Precision, Recall and F1 score can be calculated using this matrix.

**Accuracy:** Accuracy measures the correctness of the model's predicted class. Accuracy can be calculated using the four statistical measurements describe above:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (5.1)$$

**Precision:** Precision measures the amount of data that are correctly classified as positive. The mathematical expression for precision is given below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \qquad (5.2)$$

**Recall:** Recall is the measurement how accurately the actual positive data are classified. The expression for recall is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \qquad (5.3)$$

**F1 Score:** F1 score is defined as the harmonic mean of precision and recall. The formula for calculating the F1 score is given below:

$$\text{F1} = 2 * \frac{\text{Precision*Recall}}{\text{Precision + Recall}} \qquad (5.4)$$

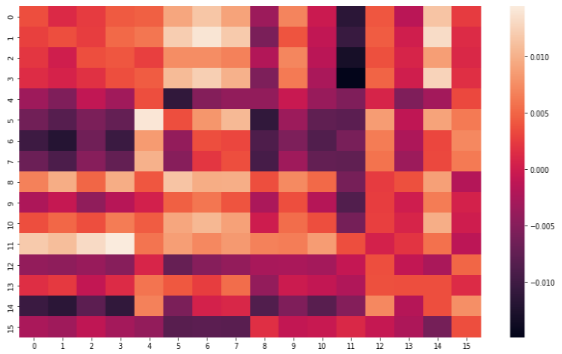
Precision, recall and f1 score was measured for each class and the accuracy was measured for overall model performance.

## 5.5 Performance Evaluation

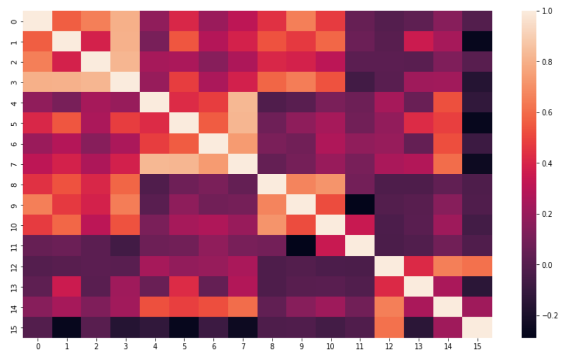
In this section, the experimental results will be discussed. As the feature map were trained in two ways: individually and combinedly. The performance was measured for both of the cases.

### 5.5.1 Generated Feature Maps

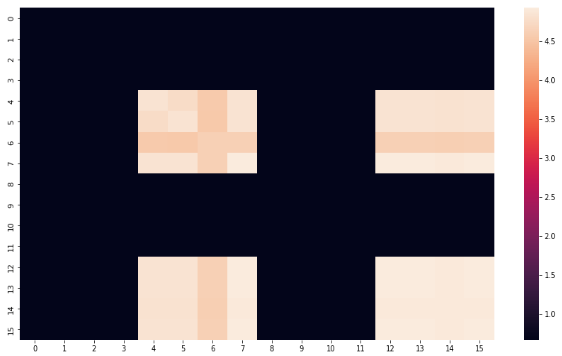
The generated feature maps for different activities are shown below:



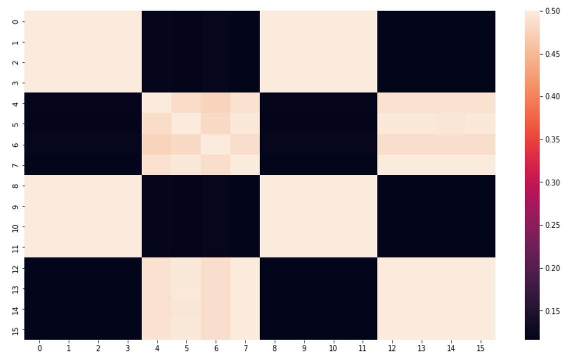
(a) Cross Correlation



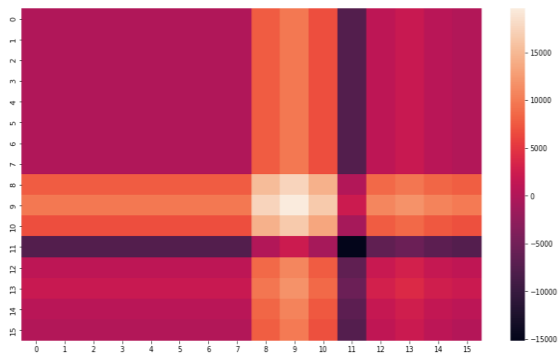
(b) Pearson Cross Correlation



(c) Mutual Information



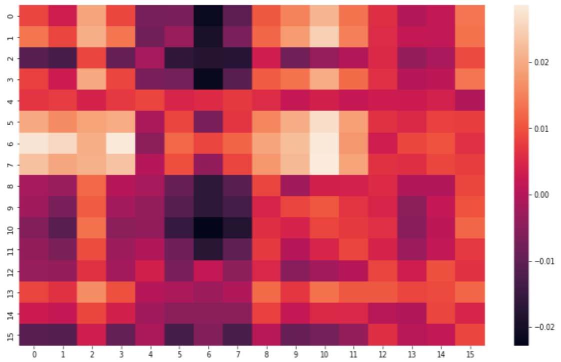
(d) Partial Mutual Information



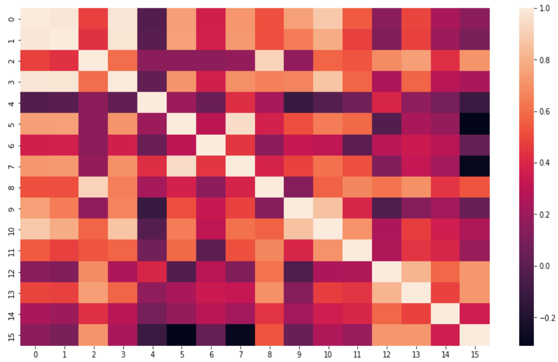
(e) Mean

Figure 5.1: Feature maps of fall activity.

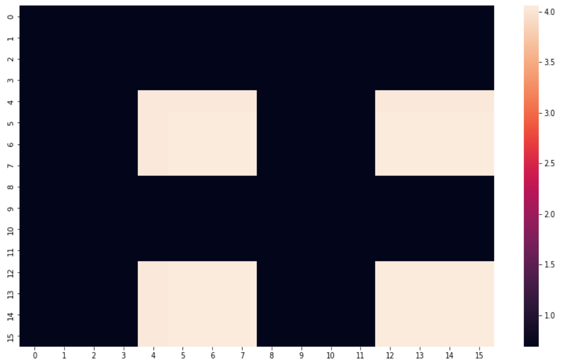
Figure 5.1 shows the feature maps generated for fall activity. Different feature maps is different is values.



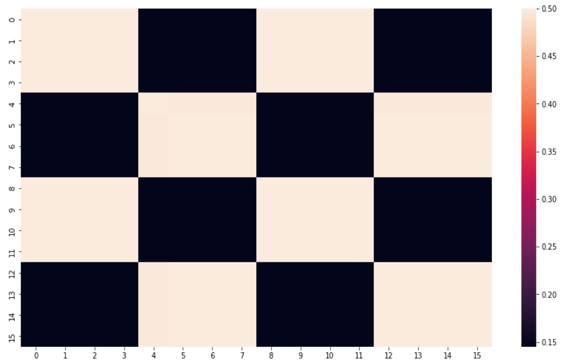
(a) Cross Correlation



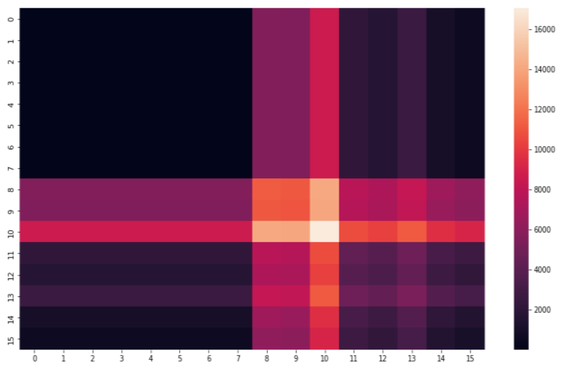
(b) Pearson Cross Correlation



(c) Mutual Information



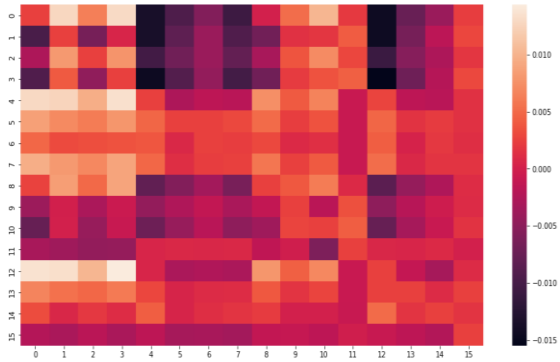
(d) Partial Mutual Information



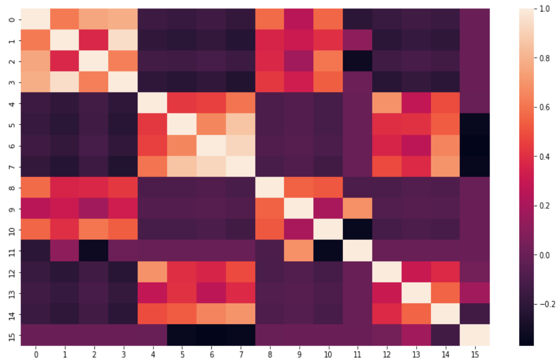
(e) Mean

Figure 5.2: Feature Map of walk activity.

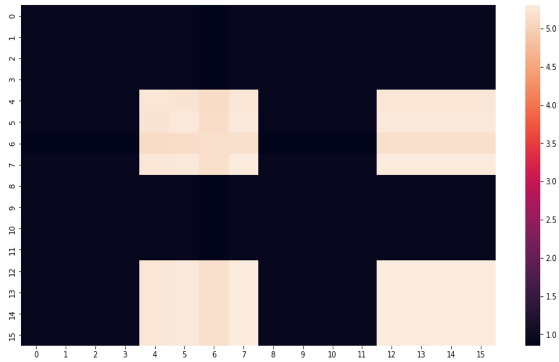
Here the figure 5.2 shows some generated feature maps for walking activity. Each of these feature maps are different from one another. The value is different for different feature map. There are similarities between the two same feature maps.



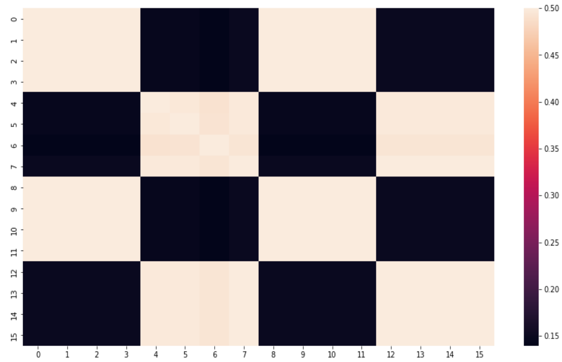
(a) Cross Correlation



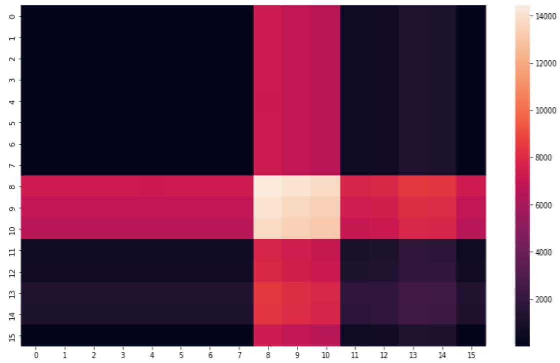
(b) Pearson Cross Correlation



(c) Mutual Information



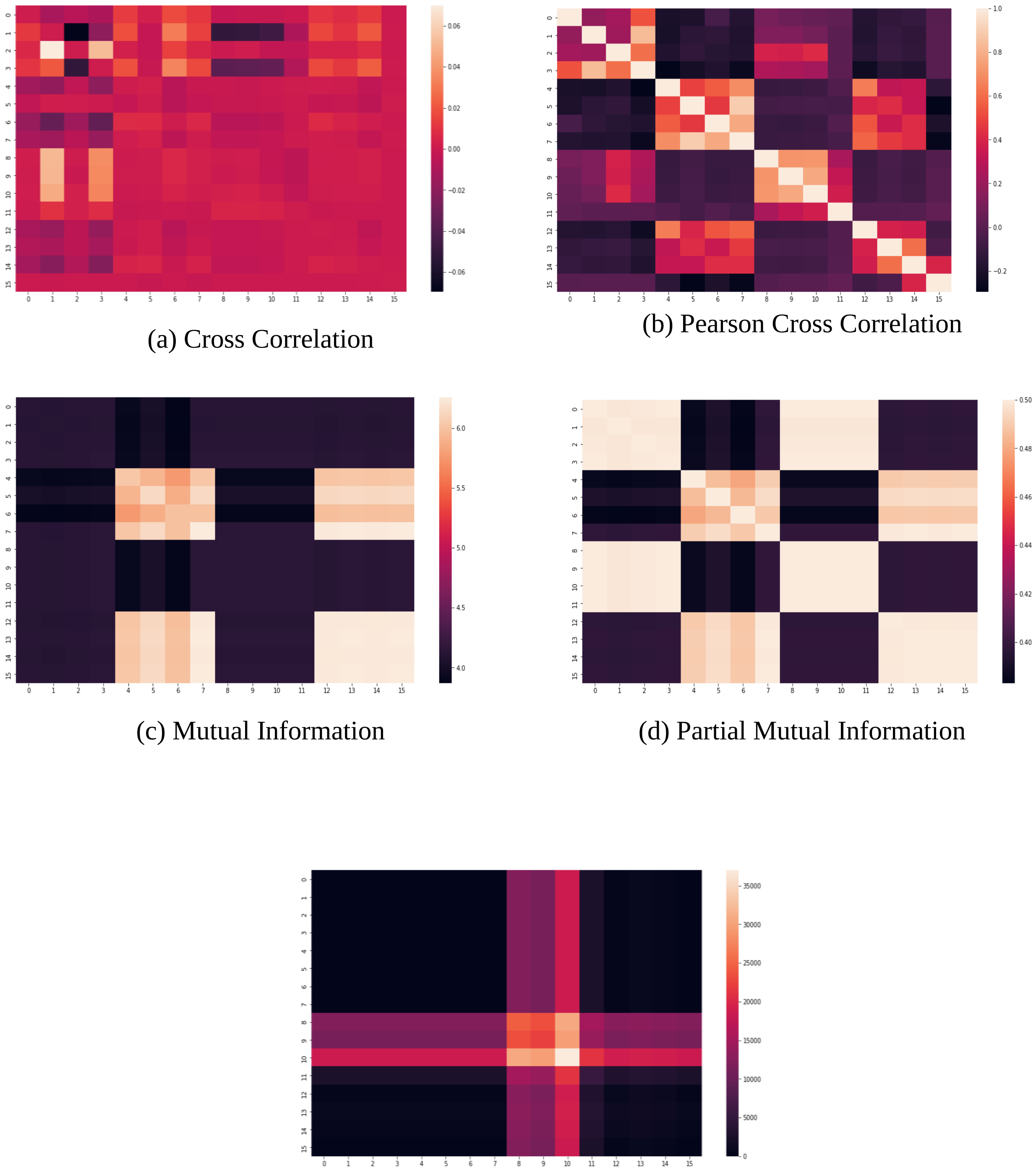
(d) Partial Mutual Information



(e) Mean

Figure 5.3: Feature map of standing activity.

Here the figure 5.3 shows some generated feature maps for standing activity. These feature maps are significantly different from the others. The color intensity indicates different values.



(a) Cross Correlation

(b) Pearson Cross Correlation

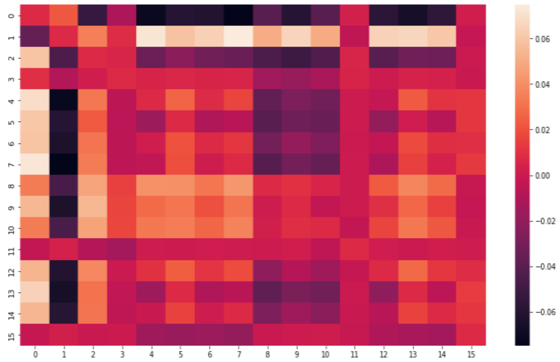
(c) Mutual Information

(d) Partial Mutual Information

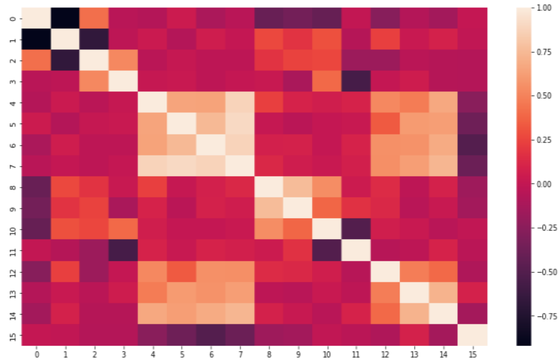
(e) Mean

Figure 5.4: Feature map of back sitting on chair (BSC) activity.

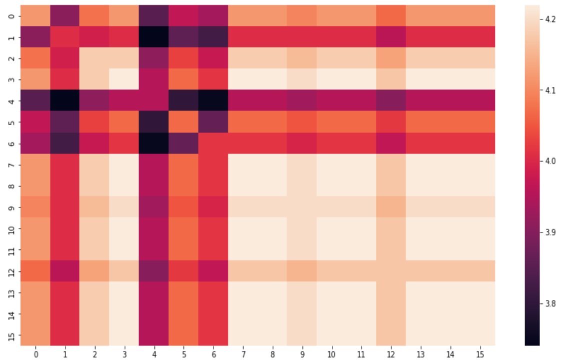
Figure 5.4 shows some of the generated feature maps for back sitting on chair activity. For this activity, the mutual information and normalized mutual information is significantly different from others.



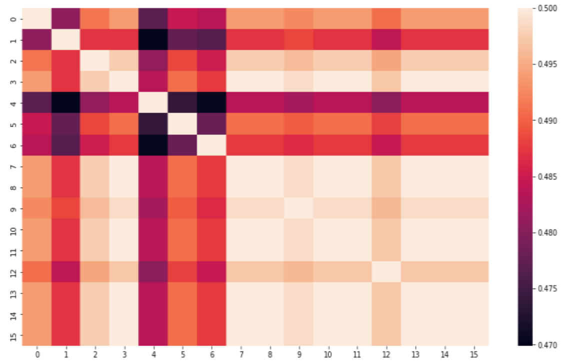
(a) Cross Correlation



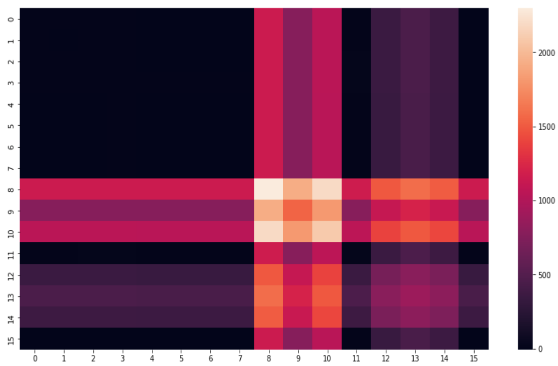
(b) Pearson Cross Correlation



(c) Mutual Information



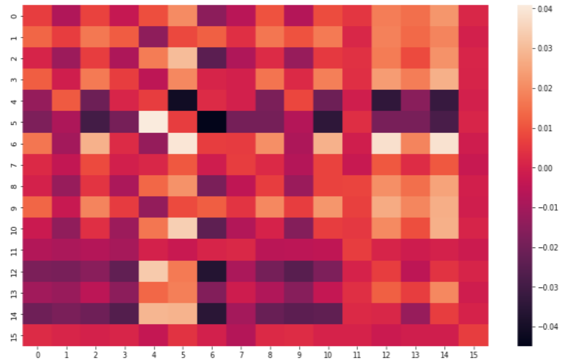
(d) Partial Mutual Information



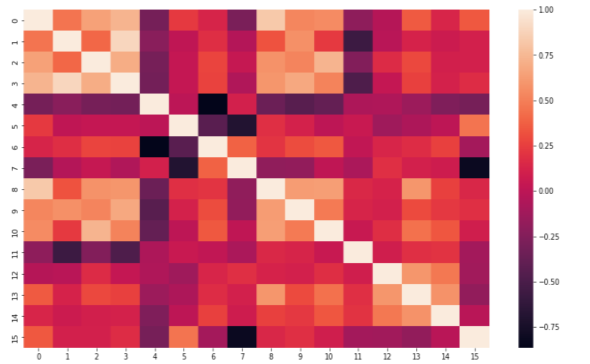
(e) Mean

Figure 5.5: Feature map of sitting activity.

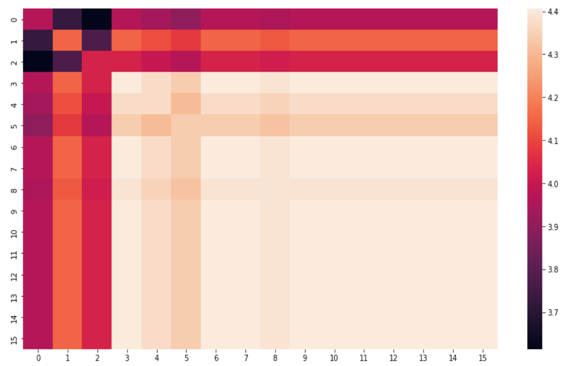
Figure 5.5 shows some of the generated feature maps for sitting activity. The mutual information and normalized mutual information are totally different from others and it is clearly visible.



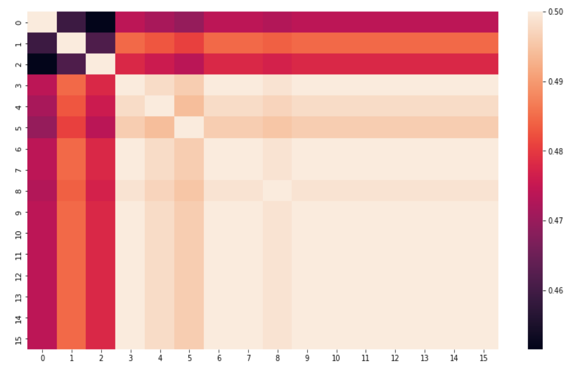
(a) Cross Correlation



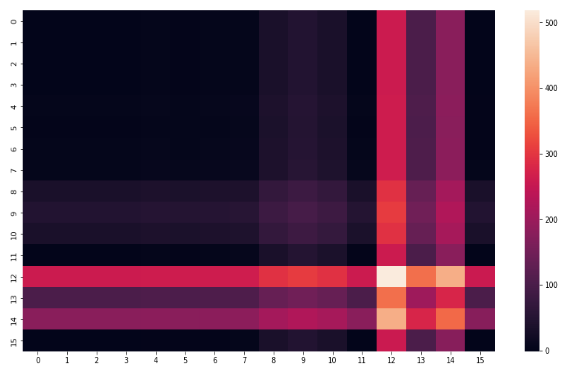
(b) Pearson Cross Correlation



(c) Mutual Information



(d) Partial Mutual Information



(e) Mean

Figure 5.6: Feature map of laying activity.

Figure 5.6 shows some of the generated feature maps for laying activity. The mean feature map here is somewhat different to look at from others.

### 5.5.2 Result of Classification

In this study, individual feature map was used for classification and the performance was measured for each feature map individually. A CNN and a LSTM model was trained using the individual feature map. For combined feature map two, three, four and five feature maps were grouped together and used as a single one. The input shape for two combined feature map was (2 x 16 x 16), for three combined feature map it was (3 x 16 x 16), for four (4 x 16 x 16) and for five (5 x 16 x 16). The other parameters were kept as usual. The figure 5.7 shows the accuracy for different combination of feature map for CNN model. All the results are evaluated on 80% train data and 20% test data of the created dataset.

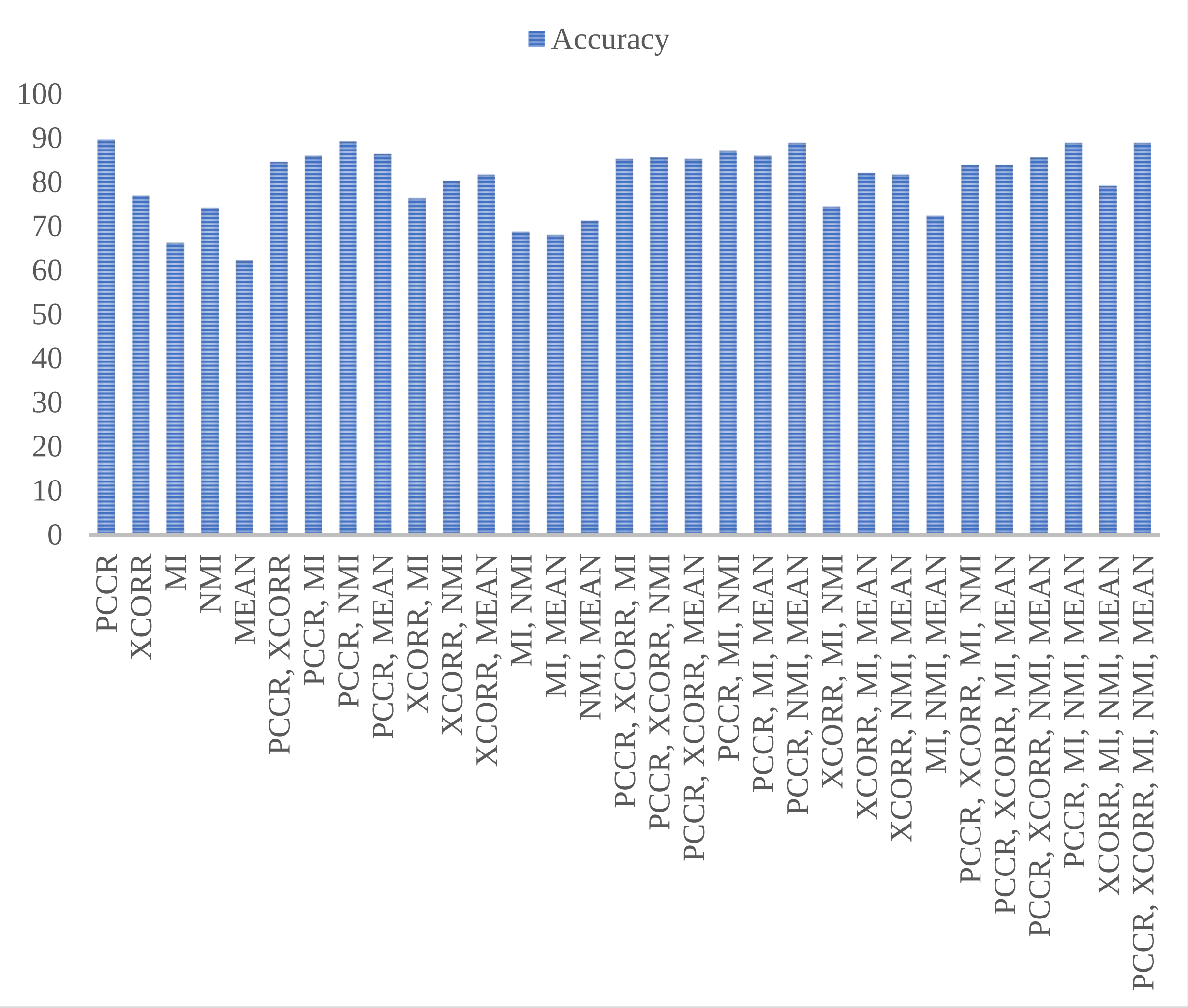



Figure 5.7: Accuracy for different combination of feature map.

From figure 5.7 it is shown that PCCR has the highest accuracy as an individual feature map. Any combination of PCCR produce higher accuracy. On the contrary, MI and mean has produced lower accuracy. Any combination of these two also produced comparatively lower accuracy. The highest accuracy was 89.57% for the test data while the lowest was 62.23%. The accuracy of the proposed LSTM model was almost similar to the proposed CNN model. LSTM model was used for individual feature map only. Figure 5.8 shows the result of CNN and LSTM models for individual feature map.

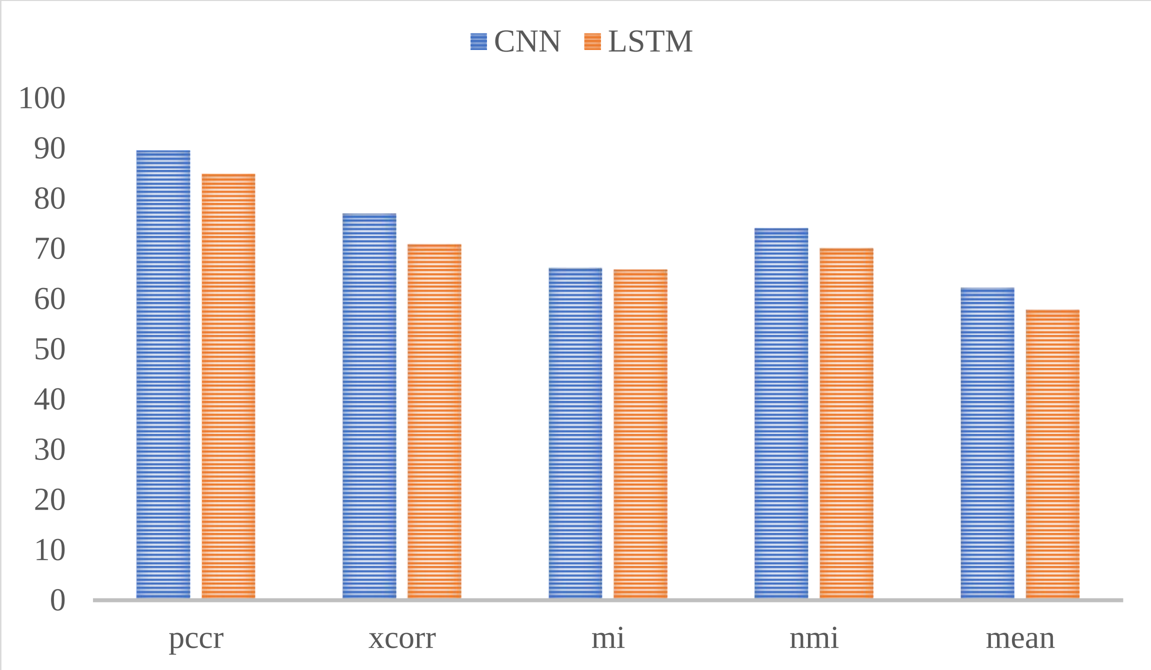



Figure 5.8: Result for individual feature map using CNN and LSTM.

Figure 5.8 suggests that the proposed CNN model has performed better than the proposed LSTM model. The best accuracy using the LSTM model was 84.89%. For LSTM model the PCCR feature map has performed well as like for CNN model. Table 5.3 to 5.5 shows the precision, recall and f1 score respectively for all the combination using CNN model.

Table 5.3: Precision for different combination of feature map using CNN.

| Feature Map Combination | Precision | | | | | |
|---|---|---|---|---|---|---|
| | BSC | Fall | Laying | Sitting | Standing | Walk |
| PCCR | **0.94** | 0.85 | 0.92 | **0.90** | 0.83 | 0.83 |
| XCORR | 0.65 | 0.76 | 0.84 | 0.79 | 0.79 | 0.81 |
| MI | 0.61 | 0.44 | 0.83 | 0.58 | 0.64 | 0.82 |
| NMI | 0.63 | 0.69 | 0.92 | 0.66 | 0.66 | 0.92 |
| MEAN | 0.54 | 0.33 | 0.94 | 0.58 | 0.68 | 0.75 |

Table 5.3: Precision for different combination of feature map using CNN (continued).

| Feature Map Combination | Precision | | | | | |
|---|---|---|---|---|---|---|
| | BSC | Fall | Laying | Sitting | Standing | Walk |
| PCCR, XCORR | 0.84 | 0.73 | 0.89 | 0.86 | 0.87 | 0.87 |
| PCCR, MI | 0.86 | 0.85 | 0.88 | 0.78 | 0.86 | 0.92 |
| PCCR, NMI | 0.90 | **0.90** | 0.94 | 0.80 | 0.88 | 0.94 |
| PCCR, MEAN | 0.88 | 0.79 | 0.92 | 0.81 | 0.86 | 0.90 |
| XCORR, MI | 0.68 | 0.74 | 0.93 | 0.69 | 0.71 | 0.86 |
| XCORR, NMI | 0.69 | 0.71 | 0.90 | 0.85 | 0.86 | 0.84 |
| XCORR, MEAN | 0.87 | 0.74 | 0.90 | 0.82 | 0.78 | 0.78 |
| MI, NMI | 0.58 | 0.50 | 0.80 | 0.63 | 0.68 | 0.88 |
| MI, MEAN | 0.45 | 0.55 | 0.92 | 0.72 | 0.69 | 0.82 |
| NMI, MEAN | 0.56 | 0.53 | 0.92 | 0.64 | 0.76 | 0.84 |
| PCCR, XCORR, MI | 0.79 | 0.77 | 0.96 | 0.87 | 0.80 | 0.90 |
| PCCR, XCORR, NMI | 0.83 | 0.75 | 0.92 | 0.88 | 0.84 | 0.90 |
| PCCR, XCORR, MEAN | 0.83 | 0.77 | 0.92 | 0.86 | 0.86 | 0.87 |
| PCCR, MI, NMI | 0.91 | 0.85 | 0.89 | 0.83 | 0.81 | 0.92 |
| PCCR, MI, MEAN | 0.88 | 0.76 | 0.90 | 0.87 | 0.79 | 0.94 |
| PCCR, NMI, MEAN | 0.88 | 0.87 | 0.90 | 0.86 | 0.87 | 0.94 |
| XCORR, MI, NMI | 0.71 | 0.70 | 0.80 | 0.69 | 0.71 | 0.82 |
| XCORR, MI, MEAN | 0.76 | 0.67 | 0.92 | 0.87 | 0.82 | 0.88 |
| XCORR, NMI, MEAN | 0.76 | 0.75 | 0.96 | 0.78 | 0.82 | 0.84 |
| MI, NMI, MEAN | 0.65 | 0.53 | 0.85 | 0.69 | 0.72 | 0.83 |
| PCCR, XCORR, MI, NMI | 0.81 | 0.68 | 0.94 | 0.89 | 0.85 | 0.85 |
| PCCR, XCORR, MI, MEAN | 0.87 | 0.73 | 0.94 | 0.82 | 0.91 | 0.87 |
| PCCR, XCORR, NMI, MEAN | 0.81 | 0.74 | 0.96 | 0.79 | 0.88 | 0.85 |
| PCCR, MI, NMI, MEAN | 0.90 | 0.76 | **1.00** | 0.82 | 0.88 | **0.96** |
| XCORR, MI, NMI, MEAN | 0.75 | 0.69 | 0.85 | 0.76 | 0.83 | 0.86 |
| PCCR, XCORR, MI, NMI, MEAN | 0.90 | 0.79 | 0.89 | **0.90** | **0.93** | 0.90 |

The precision was comparatively higher for laying and walking activity. Any combination of the feature map produces better result for these two activities. The precision for fall activity was best for the combination of PCCR and NMI.

Table 5.4: Recall for different combination of feature map using CNN.

| Feature Map Combination | Recall | | | | | |
|---|---|---|---|---|---|---|
| | BSC | Fall | Laying | Sitting | Standing | Walk |
| PCCR | **0.86** | 0.92 | 0.94 | 0.81 | **0.87** | 0.98 |
| XCORR | 0.76 | 0.76 | 0.86 | 0.72 | 0.59 | 0.91 |
| MI | 0.59 | 0.54 | 0.70 | 0.49 | 0.65 | 0.98 |
| NMI | 0.71 | 0.49 | 0.72 | 0.66 | 0.83 | **1.00** |
| MEAN | 0.41 | 0.57 | 0.88 | 0.53 | 0.46 | 0.87 |
| PCCR, XCORR | 0.82 | 0.89 | 0.84 | 0.81 | 0.72 | **1.00** |
| PCCR, MI | **0.86** | 0.89 | 0.88 | 0.74 | 0.78 | **1.00** |
| PCCR, NMI | **0.86** | **0.95** | 0.90 | 0.85 | 0.83 | 0.98 |
| PCCR, MEAN | 0.82 | 0.84 | 0.88 | 0.81 | 0.83 | **1.00** |
| XCORR, MI | 0.76 | 0.76 | 0.78 | 0.79 | 0.59 | 0.89 |
| XCORR, NMI | 0.80 | 0.86 | 0.86 | 0.72 | 0.67 | 0.89 |
| XCORR, MEAN | 0.78 | 0.78 | 0.86 | 0.70 | 0.76 | **1.00** |
| MI, NMI | 0.63 | 0.51 | 0.80 | 0.57 | 0.59 | 0.98 |
| MI, MEAN | 0.65 | 0.49 | 0.92 | 0.62 | 0.48 | 0.87 |
| NMI, MEAN | 0.61 | 0.51 | 0.88 | 0.64 | 0.61 | 0.98 |
| PCCR, XCORR, MI | 0.80 | 0.81 | 0.94 | 0.85 | 0.72 | 0.98 |
| PCCR, XCORR, NMI | 0.76 | 0.89 | 0.92 | 0.77 | 0.80 | **1.00** |
| PCCR, XCORR, MEAN | 0.84 | 0.81 | 0.88 | 0.81 | 0.78 | 0.98 |
| PCCR, MI, NMI | 0.82 | 0.89 | 0.94 | 0.72 | 0.85 | **1.00** |
| PCCR, MI, MEAN | **0.86** | 0.84 | 0.94 | 0.70 | 0.80 | **1.00** |
| PCCR, NMI, MEAN | 0.84 | 0.92 | 0.94 | 0.77 | **0.87** | **1.00** |
| XCORR, MI, NMI | 0.73 | 0.76 | 0.88 | 0.62 | 0.59 | 0.89 |
| XCORR, MI, MEAN | 0.82 | 0.81 | **0.98** | 0.70 | 0.67 | 0.91 |
| XCORR, NMI, MEAN | 0.80 | 0.89 | 0.88 | 0.74 | 0.67 | 0.91 |

Table 5.4: Recall for different combination of feature map using CNN (continued).

| Feature Map Combination | Recall | | | | | |
|---|---|---|---|---|---|---|
| | BSC | Fall | Laying | Sitting | Standing | Walk |
| MI, NMI, MEAN | 0.59 | 0.57 | 0.92 | 0.66 | 0.61 | 0.96 |
| PCCR, XCORR, MI, NMI | 0.75 | 0.86 | 0.94 | 0.72 | 0.76 | **1.00** |
| PCCR, XCORR, MI, MEAN | 0.80 | 0.89 | 0.90 | 0.85 | 0.70 | **1.00** |
| PCCR, XCORR, NMI, MEAN | 0.82 | 0.84 | 0.90 | 0.81 | 0.65 | **1.00** |
| PCCR, MI, NMI, MEAN | **0.86** | 0.86 | **0.98** | **0.87** | 0.76 | **1.00** |
| XCORR, MI, NMI, MEAN | 0.76 | 0.84 | 0.88 | 0.68 | 0.65 | 0.94 |
| PCCR, XCORR, MI, NMI, MEAN | 0.86 | 0.84 | 0.93 | 0.85 | **0.87** | **1.00** |

The recall was also good for walking activity. For fall activity the combination of PCCR and NMI produced the best recall.

Table 5.5: F1 score for different combination of feature map using CNN.

| Feature Map Combination | F1 Score | | | | | |
|---|---|---|---|---|---|---|
| | BSC | Fall | Laying | Sitting | Standing | Walk |
| PCCR | **0.90** | 0.88 | 0.93 | 0.85 | 0.85 | 0.95 |
| XCORR | 0.70 | 0. 76 | 0.85 | 0.76 | 0.68 | 0.86 |
| MI | 0.60 | 0.49 | 0.76 | 0.53 | 0.65 | 0.89 |
| NMI | 0.67 | 0.57 | 0.81 | 0.66 | 0.73 | 0.96 |
| MEAN | 0.47 | 0.42 | 0.91 | 0.56 | 0.55 | 0.80 |
| PCCR, XCORR | 0.83 | 0.80 | 0.87 | 0.84 | 0.70 | 0.93 |
| PCCR, MI | 0.86 | 0.87 | 0.88 | 0.76 | 0.82 | 0.96 |
| PCCR, NMI | 0.88 | **0.92** | 0.92 | 0.82 | 0.85 | 0.96 |
| PCCR, MEAN | 0.85 | 0.82 | 0.90 | 0.81 | 0.84 | 0.95 |
| XCORR, MI | 0.72 | 0.75 | 0.85 | 0.73 | 0.64 | 0.88 |
| XCORR, NMI | 0.75 | 0.78 | 0.88 | 0.78 | 0.76 | 0.87 |
| XCORR, MEAN | 0.82 | 0.76 | 0.88 | 0.76 | 0.77 | 0.88 |
| MI, NMI | 0.60 | 0.51 | 0.80 | 0.68 | 0.63 | 0.93 |

Table 5.5: F1 score for different combination of feature map using CNN (continued).

| Feature Map Combination | F1 Score | | | | | |
|---|---|---|---|---|---|---|
| | BSC | Fall | Laying | Sitting | Standing | Walk |
| MI, MEAN | 0.53 | 0.51 | 0.92 | 0.67 | 0.56 | 0.85 |
| NMI, MEAN | 0.58 | 0.52 | 0.90 | 0.64 | 0.67 | 0.90 |
| PCCR, XCORR, MI | 0.80 | 0.79 | 0.95 | 0.86 | 0.76 | 0.94 |
| PCCR, XCORR, NMI | 0.80 | 0.81 | 0.92 | 0.82 | 0.82 | 0.95 |
| PCCR, XCORR, MEAN | 0.83 | 0.79 | 0.90 | 0.84 | 0.82 | 0.92 |
| PCCR, MI, NMI | 0.87 | 0.87 | 0.91 | 0.77 | 0.83 | 0.96 |
| PCCR, MI, MEAN | 0.87 | 0.79 | 0.92 | 0.78 | 0.80 | 0.97 |
| PCCR, NMI, MEAN | 0.86 | 0.89 | 0.92 | **0.87** | **0.87** | 0.97 |
| XCORR, MI, NMI | 0.72 | 0.73 | 0.84 | 0.65 | 0.64 | 0.86 |
| XCORR, MI, MEAN | 0.79 | 0.73 | 0.95 | 0.78 | 0.74 | 0.90 |
| XCORR, NMI, MEAN | 0.78 | 0.81 | 0.92 | 0.76 | 0.74 | 0.88 |
| MI, NMI, MEAN | 0.62 | 0.55 | 0.88 | 0.67 | 0.66 | 0.89 |
| PCCR, XCORR, MI, NMI | 0.78 | 0.76 | 0.94 | 0.80 | 0.80 | 0.92 |
| PCCR, XCORR, MI, MEAN | 0.84 | 0.80 | 0.92 | 0.83 | 0.79 | 0.93 |
| PCCR, XCORR, NMI, MEAN | 0.82 | 0.78 | 0.93 | 0.80 | 0.75 | 0.92 |
| PCCR, MI, NMI, MEAN | 0.88 | 0.79 | **0.99** | 0.85 | 0.81 | **0.98** |
| XCORR, MI, NMI, MEAN | 0.76 | 0.76 | 0.86 | 0.72 | 0.73 | 0.90 |
| PCCR, XCORR, MI, NMI, MEAN | 0.88 | 0.82 | 0.93 | 0.85 | **0.87** | 0.95 |

The laying and walking activity have the highest f1 score. The combination of PCCR and NMI has again produced best f1 score.

From table 5.3, 5.4 and 5.5 it is observed that for laying and walking activity the performance of any combination of the feature maps is better. For fall activity, the combination of PCCR and NMI performs better. PCCR individually detects BSC activity with higher precision, recall and f1 score. The combination of PCCR, MI, NMI and MEAN produced best precision, recall and f1 score for laying and sitting activity. All combination of feature maps shown competitive performance for walking activity. The standing activity was detected well using all the feature maps together.

The validation loss decreases as the epoch increases. After some iteration the loss saturates. The accuracy also increases over iteration. Table 5.6 represents some comparison among several previous approaches on different data set with the proposed method.

Table 5.6: Comparison of the proposed method with previous works.

| Authors | Dataset | Accuracy |
|---|---|---|
| Pranesh et al. [16] | MobiFall[26] | 87.5% |
| John C. et al. [15] | Self | 95.65% |
| Ahmet et al. [18] | Self | 99% |
| Proposed Method | Self | 89.57% |

## 5.6 Conclusion

The performance was evaluated on a new dataset. Pearson Correlation Coefficient (PCCR) feature map individually or combinedly produce best result for detecting all activities. From this study it is observed that PCCR alone is enough for activity detection from sensor data. Because using PCCR as an individual feature map, the best outcome was found. The proposed CNN performs slide better than the Bidirectional LSTM model. The use of feature maps for detecting activities form sensor data is the first. The new techniques or methods may produce better result on the newly created dataset. The newly created dataset contains some similar activities like Sitting, Standing, Laying. Again, the Back Chair Sitting (BSC) is somewhat similar to the Fall event. These all makes the detection process more complex.

# CHAPTER 6

# Conclusion

## 6.1 Summary

The fall detection method using feature maps from sensor data is first to use. In the recent works, mainly statistical features were used. But in this work, connectivity features of the sensors were extracted to generate feature map. The feature maps were used for classification. Some of the feature maps produces satisfactory results. The features actually measure the relation between two signals. The relation among two signals for a certain activity may be identical. This identical characteristic is studied for fall detection purposes. In this study, the dataset contains some similar activities like Sitting, Standing and Laying indicates the inactive mode of the actor. These activities are confusing to detect from the sensor data because the sensor data don't vary too much as the actor remains inactive for these activities. Again, fall can be thought as similar to back sitting on chair activity.

## 6.2 Limitations

The proposed feature map based fall detection using sensor data has the following limitations:

- Computation cost: In this study, several new signals are calculated along with five features from the sensor data. This requires computational power and time. For real time system, the fall or activity detection process may be slower for the reason mentioned.
- Device/Sensor position: The dataset used in this study contains sensor data for device keeping in the thing pocket of the actor. For different position of the device carrying sensor may produce different result.
- Dataset: The dataset may contain for sensor data per activity. More variety of data for different orientation of the device may be collected.

- Feature Map: Only five feature maps were generated for this study. More such feature maps can be considered for better performance.
- False positives and negatives: The proposed methodology may detect some activity falsely. The false call may lead to incorrect decision for a real time system.

The proposed methods have several advantages. But the limitations such as computation cost, sensor position etc. cannot be ignored. Specially for real time system, these limitations should be resolved for the robustness of the system.

## 6.3 Future Works

The sensor data are cheaply available. Activity detection along with fall event from sensor data has the potential to be used on a large scale. The limitations of this study should be addressed properly and increase the robustness of the system irrespective of all environment. This work has the following scope to works:

- Integration with vision data: The sensor based method can be integrated with vision data for more precise detection of activities.
- Using more feature maps: More feature maps from different domains such as frequency domain can be studied.
- Real time system: A real time system can be developed using the proposed methodology. The real time system should be more accurate.
- Data: More variety of data of different activities with different types can be collect for activity or fall detection.

## 6.4 Conclusion

A fall detection system has been developed. A new dataset is created. Connectivity features were used instead of statistical features. Tough the result of the proposed methodology is low but it is satisfactory as a new feature extraction technique on new dataset. As CNN and LSTM performs like similar, more classical methods can be used using this feature maps for better result. The Pearson correlation coefficient feature clearly produce better result than others. The performance was measured in terms of precision, recall, f1 score and accuracy.

# REFERENCES


[1] S. Giovannini *et al.*, "Falls among older adults: Screening, identification, rehabilitation, and management," *Appl. Sci. (Basel)*, vol. 12, no. 15, p. 7934, 2022.

[2] H. K. Edelberg, K. Lyman, and J. Y. Wei, "Notation of previous falls in admission record of hospitalized elderly," *Aging (Milano)*, vol. 10, no. 1, pp. 67–70, 1998.

[3] M. C. Hornbrook, V. J. Stevens, D. J. Wingfield, J. F. Hollis, M. R. Greenlick, and M. G. Ory, "Preventing falls among community-dwelling older persons: results from a randomized trial," *Gerontologist*, vol. 34, no. 1, pp. 16–23, 1994.

[4] C. B. Brown *et al.*, "Step edge highlighters and illuminance changes influence stair descent in a real-world setting," *Ergonomics*, pp. 1–10, 2022.

[5] S.-G. Miaou, P.-H. Sung, and C.-Y. Huang, "A customized human fall detection system using Omni-camera images and personal information," in *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, 2006. D2H2*, 2006.

[6] L. R. Rivera, E. Ulmer, Y. D. Zhang, W. Tao, and M. G. Amin, "Radar-based fall detection exploiting time-frequency features," in *2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, 2014.

[7] CDC, "Centers for disease control and prevention," *Centers for Disease Control and Prevention*, 14-Mar-2023. [Online]. Available: https://www.cdc.gov/. [Accessed: 16-June-2022].

[8] M. Islam, N. Neom, M. Imtiaz, S. Nooruddin, M. Islam, and M. Islam, "A review on fall detection systems using data from smartphone sensors," *Ing. Syst. D Inf.*, vol. 24, no. 6, pp. 569–576, 2019.

[9] L. N. V. Colon, Y. DeLaHoz, and M. Labrador, "Human fall detection with smartphones," in *2014 IEEE Latin-America Conference on Communications (LATINCOM)*, 2014.

[10] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010.

[11] A. Z. Rakhman, L. E. Nugroho, Widyawan, and Kurnianingsih, "Fall detection system using accelerometer and gyroscope based on smartphone," in *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering*, 2014.

[12] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," *Int. J. Telemed. Appl.*, vol. 2015, p. 576364, 2015.

[13] M. Saleh and R. L. B. Jeannes, "Elderly fall detection using wearable sensors: A low cost highly accurate algorithm," *IEEE Sens. J.*, vol. 19, no. 8, pp. 3156–3164, 2019.

[14] "What are Convolutional Neural Networks?" *Ibm.com*. [Online]. Available: https://www.ibm.com/topics/convolutional-neural-networks. [Accessed: 1-Sep-2022].

[15] J. C. Dogan and M. S. Hossain, "A novel two-step fall detection method using smartphone sensors," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2019, pp. 434–438.

[16] P. Vallabh, R. Malekian, N. Ye, and D. C. Bogatinoska, "Fall detection using machine learning algorithms," in *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2016.

[17] H. Jian and H. Chen, "A portable fall detection and alerting system based on k-NN algorithm and remote medicine," *China Commun.*, vol. 12, no. 4, pp. 23–31, 2015.

[18] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors (Basel)*, vol. 14, no. 6, pp. 10691–10708, 2014.

[19] S. W. Abeyruwan, D. Sarkar, F. Sikder, and U. Visser, "Semi-automatic extraction of training examples from sensor readings for fall detection and posture monitoring," *IEEE Sens. J.*, vol. 16, no. 13, pp. 5406–5415, 2016.

[20] F. Li, K. Shirahama, M. A. Nisar, X. Huang, and M. Grzegorzek, "Deep transfer learning for time series data based on sensor modality classification," *Sensors (Basel)*, vol. 20, no. 15, p. 4271, 2020.

[21] "Falls," *Who.int*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls. [Accessed: 25-Jan-2023].

[22] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1082–1090, 2008.

[23] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol.

27, no. 3, pp. 379–423, 1948.

[24] R. C. King, E. Villeneuve, R. J. White, R. S. Sherratt, W. Holderbaum, and W. S. Harwin, "Application of data fusion techniques and technologies for wearable health monitoring," *Med. Eng. Phys.*, vol. 42, pp. 1–12, 2017.

[25] N. Varshney, B. Bakariya, A. K. S. Kushwaha, and M. Khare, "Human activity recognition by combining external features with accelerometer sensor data using deep learning network model," *Multimed. Tools Appl.*, 2022.

[26] G. Vavoulas, M. Pediaditis, E. G. Spanakis, and M. Tsiknakis, "The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones," in *13th IEEE International Conference on BioInformatics and BioEngineering*, 2013.

[27] P. Tsinganos and A. Skodras, "A smartphone-based fall detection system for the elderly," in *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, 2017.

[28] S. Madansingh, T. A. Thrasher, C. S. Layne, and B.-C. Lee, "Smartphone based fall detection system," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, 2015.

[29] C. Li, M. Liu, X. Yan, and G. Teng, "Research on CNN-BiLSTM fall detection algorithm based on improved attention mechanism," *Appl. Sci. (Basel)*, vol. 12, no. 19, p. 9671, 2022.

[30] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 799–804.

[31] R. Yamashita, M. Nishio, R.K.G. Do, et al., "Convolutional neural networks: an overview and application in radiology," Insights Imaging, vol. 9, pp. 611-629, 2018. doi: 10.1007/s13244-018-0639-9.

[32] "Pearson correlation coefficient," Wikipedia, Sep. 3, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient. [Accessed: Sep. 21, 2022]

[33] J. Walters-Williams and Y. Li, "Estimation of Mutual Information: A Survey," in Rough Sets and Knowledge Technology, P. Wen et al. Eds., vol. 5589, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2009, pp. 498-509. doi: 10.1007/978-3-642-02962-2_49.

[34] S. Hussain, J.H. Bang, M. Han, M.I. Ahmed, M.B. Amin, S. Lee, C. Nugent, S. McClean, B. Scotney, and G. Parr, "Behavior life style analysis for mobile sensory data in cloud computing through MapReduce," Sensors, vol. 14, no. 11, pp. 22001-22020, 2014. doi: 10.3390/s141122001.

[35] M. Liu, "A Study of Mobile Sensing Using Smartphones," Int. J. Distrib. Sens. Netw., vol. 9, no. 3, 2013, doi: 10.1155/2013/272916.

[36] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," Towards Data Sci, vol. 6, no. 12, pp. 310-316, 2017.

[37] J. Christ, E. Page, and G. Tagliarini, "Optimization Using Neural Networks," IEEE Trans. Comput., vol. 40, no. 12, pp. 1347-1358, Dec. 1991, doi: 10.1109/12.106220.

[38] V.R.L. Shen, H.-Y. Lai, and A.-F. Lai, "Application of High-Level Fuzzy Petri Nets to Fall Detection System Using Smartphone," in 2013 International Conference on Machine Learning and Cybernetics, Tianjin, China, 2013, pp. 1429-1435, doi: 10.1109/ICMLC.2013.6890807.

[39] K. Janocha and W. M. Czarnecki, "On Loss Functions for Deep Neural Networks in Classification," in arXiv preprint arXiv:1702.05659, Feb. 2017.

[40] M. C. Bhoyar, R. R. Manza and V. M. Thakare, "Impact of Hidden Dense Layers in Convolutional Neural Network to enhance Performance of Classification Model," IOP Conference Series: Materials Science and Engineering, vol. 1131, no. 1, Apr. 2021, doi: 10.1088/1757-899X/1131/1/012007.