Bending Of CNN Model With Genetic Algorithm To Recognize Human Activities From Sensor Data

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Engineering

in

Computer Science & Engineering

By

Apu Sarkar

Examination Roll No. - M4CSE22020

Registration No. - 154144 of 2020-2021

Session - 2020-2022

Under the guidance of

Prof. (Dr.) Ram Sarkar

Department of Computer Science and Engineering

Jadavpur University, Kolkata - 700032

India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that the dissertation entitled "Bending Of CNN Model With Genetic Algorithm To Recognize Human Activities From Sensor Data" has been carried out by Apu Sarkar (Examination Roll No - M4CSE22020, Registration No. - 154144 of 2020-2021, Session - 2020-2022) under my guidance and supervision and be accepted in partial fulfilment of the requirements for degree of Master of Engineering in Computer Science and Engineering from the Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other University or Institute.

.....

Dr. Ram Sarkar(Thesis Supervisor)
Prof., Department of Computer Science and Engineering
Jadavpur University, Kolkata - 700032

.....

Dr. Anupam Sinha
Prof. and HOD, Department of Computer Science and Engineering
Jadavpur University, Kolkata - 700032

.....

Prof. Chandan Mazumdar
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

Declaration Of Authorship

I, hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of his Master in Computer Science and Engineering studies.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials that are not original to this work.

Signature :

Name : Apu Sarkar

Examination Roll No. : M4CSE22020

Registration No. - 154144 of 2020-2021

Thesis Tittle : ***Bending Of CNN Model With Genetic Algorithm To Recognize Human Activities From Sensor Data.***

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval

This is to certify that the thesis entitled "Bending Of CNN Model With Genetic Algorithm To Recognize Human Activities From Sensor Data" is a bonafide record of work carried out by Apu Sarkar in partial fulfillment of the requirements for the award of the degree of Master of Engineering in Computer Science and Engineering of the Department of Computer Science and Engineering, Jadavpur University during the period of July 2020 to June 2022. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Approved by :

.....
Signature of Examiner 1

.....
Signature of Examiner 2

*Only in case the thesis is approved

Acknowledgement

This thesis has become a reality with the kind support and help of many individuals. I would like to express my sincere gratitude to all of them.

First and foremost I am extremely grateful to my esteemed supervisor, Dr. Ram Sarkar, Professor, Department of Computer Science and Engineering, Jadavpur University for his invaluable advice, continuous support, and patience during my masters study. This research work is mostly a product of his vision and suggestions. His immense knowledge and skilled expertise guided me throughout this study and motivated me to be on track.

I would also like to thank Mr. Sk Sabbir Hossian, B.E. 4th year student, Department of Computer Science and Engineering, Jadavpur University for imparting his knowledge and expertise in this study.

Finally, I would like to express my heartfelt gratitude to the constant people of my life. The list includes my family members: my parents for their unconditional love and support. Special thanks to my senior Neelotpal Chakraborty, Ph.D. Scholar, Department of Computer Science and Engineering, Jadavpur University for encouraging me to venture outside of my comfort zone. My appreciation also goes out to all of those with whom I have had the pleasure to meet and work with during my master's journey.

.....

Apu Sarkar
Registration No. - 154144 of 2020-2021
Examination Roll No. : M4CSE22020
Department of Computer Science and Engineering
Jadavpur University, Kolkata - 700032

Abstract

Human activity recognition (HAR) is a critical application on wearable devices for fitness tracking, healthcare, and elder care assistance. Inaccurate recognition results, on the other hand, may have a negative impact on users or even result in an unexpected accident. As a result, it is critical to improve the accuracy of human activity recognition. Recent advancements in deep learning (DL) techniques and their automatic feature extraction ability attract many researcher for HAR. However, the large size feature maps generated by these DL models affect the overall classification accuracy, and it also increases the computational cost, as there are many redundant features generated by the DL models. This thesis aims to provide effective and efficient HAR methods to address the major HAR challenges, which can be divided into three contributions. The first contribution is a novel wearable sensor signal to a corresponding activity image encoding algorithm that addresses the problem of capturing information in both the time and frequency domains. The second contribution is to propose a novel feature extractor based on a deep learning approach that addresses the extraction of quality features for activities. The third contribution is a novel feature selection framework that selects the most relevant features and addresses the negative effect of large feature size on the performance of a HAR model. Extensive experiments on publicly available datasets have been conducted for the proposed approaches. Experiments have shown that the proposed methods outperform the state-of-the-art methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Over the last few years, Human Activity Recognition (HAR) has emerged as an active area of research. It is a significant and challenging field that can support a wide range of novel ubiquitous applications. Smart homes, just-in-time information systems for office workers, surveillance and interactive game interfaces, and home healthcare are all examples of these applications. Activity recognition is a multidisciplinary research field that is linked to machine learning, artificial intelligence, machine perception, ubiquitous computing, human computer interaction, psychology, and sociology. As a result, it has piqued the interest of researchers from a wide range of disciplines.

An activity recognition system's goal is to recognize its users' actions or activities by unobtrusively observing people's behaviour and the characteristics of their environments and taking appropriate actions in response. Such systems, for example, could enable the development of just-in-time learning environments that educate and inform people by presenting information at the right time as they move through the environment by recognising activities in real time. Knowing what someone is doing allows you to determine the best time to interrupt them and present them with useful information or messages. A person preparing dinner provides an excellent opportunity for a teaching system to display words in a foreign language related to cooking.

Activity recognition systems in the home can monitor users' activities over long periods of time to remind them to perform forgotten activities or complete actions such as taking medicine, assist them in recalling information, or encourage them to act more safely [1, 2]. In a hospital setting[3], such systems can alert a doctor or nurse to run certain tests before performing surgery. A behaviour model can be developed in a surveillance system[4] using recognised activities, allowing the system to predict the intent and motive of people as they interact. Furthermore, in a manufacturing context, such systems can ensure product quality by monitoring a set of actions. Finally, by recommending tiny behavioural changes to their users, these systems can play an important part in supporting a healthy lifestyle. For example, people can be urged to take the stairs rather than the elevator, or to stand after a lengthy period of sitting.

We as human beings, are capable of comprehending and interpreting the activities of those in

our immediate environment. Our capacity to recognise activities looks to be straightforward and natural, but it is actually a complex effort of sensing, learning, and inference. Humans get knowledge from their previous experiences. All of these functions of perceiving the environment, learning from previous experience, and using information for inference, however, remain a significant barrier for machine. As a result, the goal of activity recognition research is to enable computers to recognise people's actions in the same way that humans do.

1.2 Approaches To HAR

The first step in achieving the aim of recognising daily activities is to provide sensing capabilities to activity identification systems. As illustrated in Figure 1.1, mainly two approaches have been used for this purpose: video-based, and wearable sensor-based.

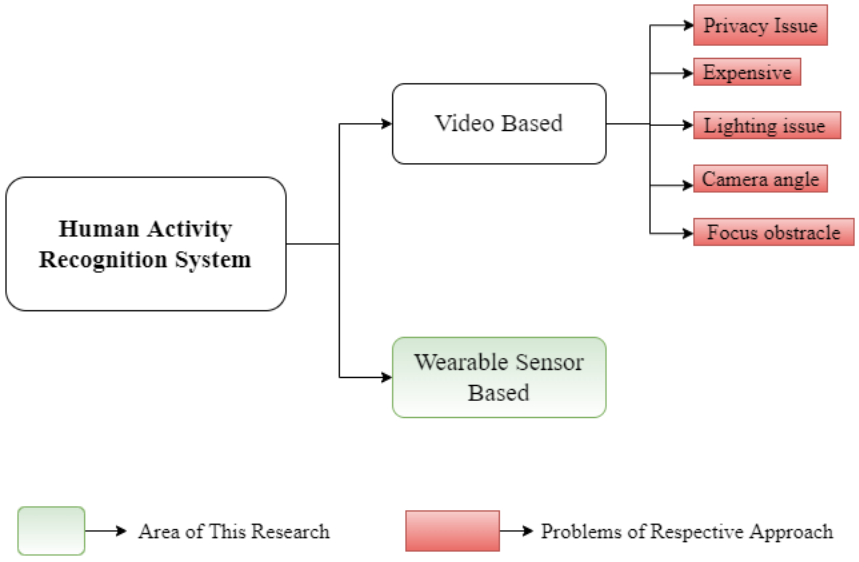


Figure 1.1: Two main approaches generally used for HAR.

1.2.1 Video based HAR

These approaches use a video camera to measure and recognise physical activity. Due to clutter, fluctuating lighting, and the wide range of activities that occur in actual situations, this approach often works well in the lab but fails to achieve the same accuracy in real-world settings. Additional issues arise from the complexity of dealing with changes in the scene, such as illumination, several individuals, and clutter. Furthermore, sensors such as microphones and cameras are typically costly. Finally, because these devices are frequently used as recording devices, some people may regard them as a threat to their privacy.

1.2.2 Wearable Sensor based HAR

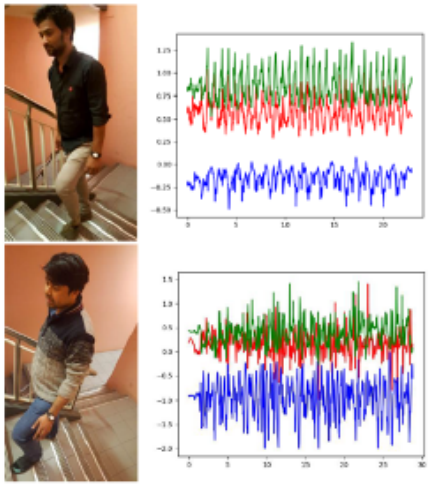
These approaches are meant to be worn during regular everyday activities to continuously capture bio-mechanical and physiological data regardless of subject location, making them a viable alternative for recognising daily human activities, particularly bodily or physical activities. Bodily functions necessitate repetitive motion of the human body, which is limited to a considerable

extent by the body's structure. Walking, running, and other activities are examples. Because wearable sensors can be integrated into clothing, worn as wearable devices, or can embedded with various other devices like smartphones, they are well suited to gathering data on daily physical activity patterns across time. Wearable sensors can measure physiological indicators that may not be measurable using ambient or video sensors since they are attached to the subjects they are monitoring and are independent of the infrastructure. Furthermore, unlike video sensors, such sensors are inexpensive and do not pose a threat to people's privacy.

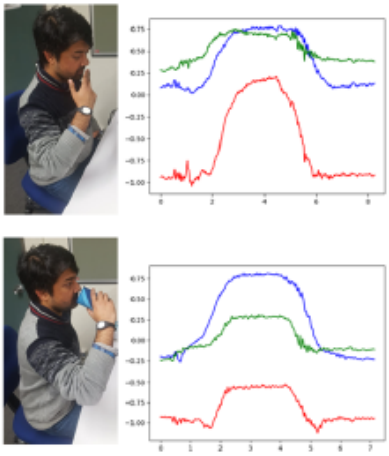
1.3 Main Challenges in Wearable Sensor based HAR

For the activity prediction tasks, generalizing any model for different activities and sensors is a very challenging task. The fundamental challenge in HAR is that human daily life activity is highly complex and diverse, which can impair HAR's reliability and accuracy.

1. One of the main challenges in wearable sensor based HAR is intraclass variability problem. This problem arises when same activity is performed by different individual or even same person, resulting different signal pattern for the same activity. For example, if we consider figure 1.2a both the person are performing walking upstairs activity but



(a) Intra-class Variability



(b) Inter-class Similarity

Figure 1.2: Two main challenges of wearable sensor based HAR. (a)Intra-class variability problem where same activity generates different signal pattern. (b) Inter-class similarity where different activities generates similar signal pattern.

corresponding signal pattern are different. These changes make HAR more challenging.

2. Another main challenge in sensor based HAR is when two different class of activities generates similar signal pattern. This is also known as the inter-class similarity problem. For example as shown in the figure 1.2b though same person is performing two different activity i.e., drinking water and smoking, the signal pattern is very similar due to the similar hand movement pattern.

3. Sometimes the problem arises due to the participating volunteers own habit resulting unstable classification result. This problem mainly occurs due to change of orientation of the sensors. Due to the lack of orientation independence, users are forced to place sensors such as the accelerometer and gyroscope in a specific orientation, limiting their ability to use these devices.

1.4 Research Aim and Objectives

The main aim and objectives of this thesis are mentioned below.

1.4.1 Aim

- There are plenty of researchers, trying to improve the recognition accuracy of wearable sensor based HAR model using various deep learning techniques considering either time domain or frequency domain information at a time, but not the both, and with a large number of features. The research in this thesis seeks how to improve the recognition accuracy of a HAR model considering both the time and frequency domain information of sensor signal with a reduced feature set.

1.4.2 Objectives

- To study how the time and information domain information helps to improve the deep learning based HAR models.
- To represent both the time and frequency domain information simultaneously from the sensor signal.
- To extract more discriminating features using deep learning techniques from the time and frequency domain information for recognition of human activities.
- To reduce the feature space by removing irrelevant features and recognize the human activities.
- To analyse and compare the results with other existing models.

1.5 Road-map of the Thesis

- Chapter 2 provides a comprehensive literature review of the state-of-the-art in HAR.
- Details of the research methodologies are provided in Chapter 3.
- Chapter 4 contains Experimental Details and Results.
- Chapter 5 concludes the work in this thesis

Chapter 2

Review of Literature

2.1 Traditional Machine Learning vs Deep Learning

After obtaining raw sensor signals from human activity data, the next step in the HAR process is to select the best method for properly analysing the data. Artificial intelligence has been used to solve recognition and classification problems for many years. Artificial intelligence (AI) is the concept of designing computers or machines to have the same characteristics as human intelligence. Traditional machine learning (TML) is one method for achieving AI. TML uses various algorithms to analyse data, learn from it, and then make a prediction or classification about something related to the data. The goal is to provide enough data to a machine so that it can learn enough from it to correctly predict or classify a new piece of data. In the recent past, researchers have introduced several handcrafted feature extraction methods to extract various spatiotemporal features from the raw sensor data. Then traditional supervised machine learning techniques Support Vector Machine (SVM) [5, 6, 7], K-Nearest Neighbors (KNN) [8, 9, 10], Decision Tree [11], Ensemble approach [12, 13] are used for classification. However, there are certain limitations of this approach like the requirement of domain expertise and rigorous data pre-processing. Also, failing to establish a proper spatial and temporal relationship among handcrafted features limits the flexibility of these approaches.

Despite the success of TML methods in sensor-based HAR, the process is not fully automated. The requirement of a human expert within the domain to manually extract features that the TML algorithm requires to make predictions is a critical step in the classification process for TML, as shown in Figure 2.1. This feature extraction requirement limits the flexibility of these methods.

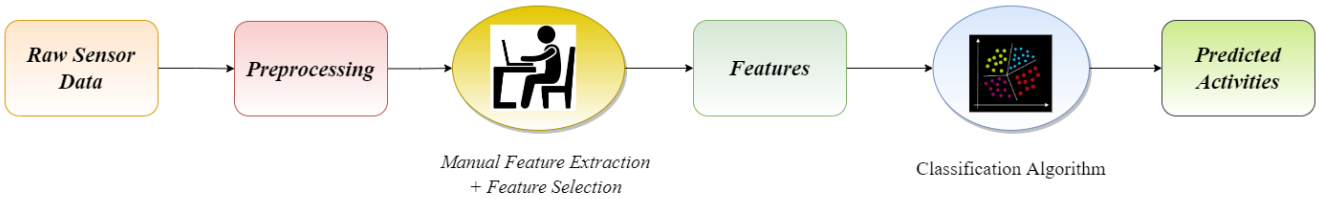


Figure 2.1: Step-by-step process used for HAR following a traditional Machine Learning based approach

Another disadvantage of TML is that its performance degrades as the amount of input data

increases. The goal of advances in technology and access to very large amounts of data is for algorithm performance to increase proportionally with the amount of data available. Unfortunately, research has shown that as the amount of input data for TML algorithms grows, the algorithms' performance plateaus. Because of this lack of progress, TML is unable to fully exploit the vast amounts of data available. The drawbacks of TML draw attention to a different subset of AI, deep learning, which introduces a more efficient approach to the HAR problem.

Deep learning is a machine learning method that uses artificial neural networks to perform automatic activity recognition and classification with little to no human intervention. Figure 2.2 depicts the process of activity recognition using deep learning algorithms, demonstrating that no human expert is required to complete feature extraction. Furthermore, deep learning algorithms have been shown to improve as the amount of data presented increases.

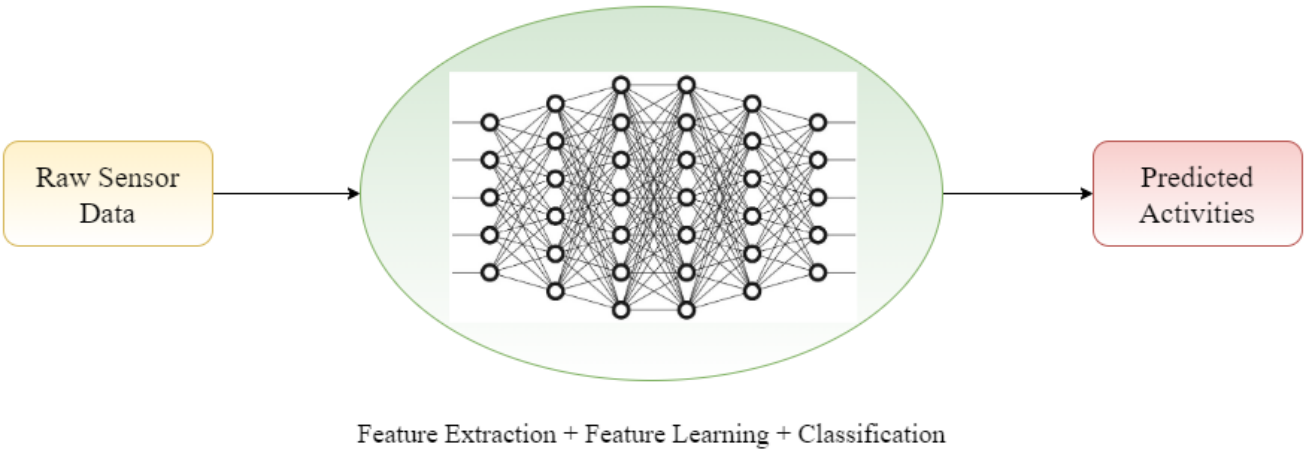


Figure 2.2: Step-by-step process used for HAR following a Deep Learning based approach

Since deep learning has achieved outstanding results in a variety of fields, developing a deep learning-based model for HAR gains more popularity among researchers. Many state-of-the-art models have been developed using various deep learning techniques like CNN, Recurrent Neural Network (RNN), etc.

CNN models showed lots of promise and achieved higher recognition accuracy than other state-of-the-art methods. Nair et al. [14] used the Temporal CNN architecture, a class of temporal models that used a hierarchy of temporal convolutions, which was able to take variable-length sequence data and learn long-term dependencies. Münzner et al. [15] proposed CNN-based sensor fusion technique to solve the problems of normalization and fusion of multimodal sensors. In [16, 17, 18, 19, 20] authors have used various CNN architectures to improve the recognition accuracy of HAR. Ensemble of CNN models found in [21, 22] which aim to achieve better performance than the individual models.

RNN, another deep learning technique, was also extensively used by many researchers for HAR. RNN has the special ability to learn sequences of spatial data. Like, long short-term memory (LSTM) based networks can learn long-term dependencies from any sequences of data which makes it more applicable in wearable/inertial sensor-based HAR. Preeti Agarwal and Mansaf Alam [23] developed a lightweight model using shallow RNN combined with LSTM for activity recognition. Authors in [24, 25, 26, 27] used LSTM based architectures to learn spatio-temporal features for the classification of human activities. Researchers also proposed various hybrid models like the

combination of CNN-RNN [28], CNN-LSTM [29, 30, 31, 32, 33], LSTM-CNN [34], CNN-GRU (Gated Recurrent Unit) [35] and achieved significant improvement in recognition accuracy.

2.2 Literature Review

Previous physical activity recognition schemes generated features using a wide range of techniques. These features are then used as inputs to classification schemes after they have been generated. This section provides a high-level overview of various feature generation techniques.

2.2.1 Time/Temporal Domain Features based Model

Some studies, typically of a statistical nature, derived time-domain features directly from a window of acceleration data. The mean, median, variance, skewness, and kurtosis are some examples [36, 37]. In other studies, high and low pass filters were used to separate accelerometer signals on a frequency basis. Means for the low frequency and rectified high frequency components are calculated separately and used as inputs to the classification schemes. Cross-correlation coefficients have also been used to quantify the similarity of acceleration signals from different axes on the same and different body segments [36].

The authors in [38] found that varied time series features are not evident in the temporal domain but present in the frequency domain. As an alternative graphical representation for time series classification, they investigated the use of recurrence plots and proposed a method capable of extracting texture features from that graphical representation, and used those features to classify time-series data. In their work, Garcia-Ceja et al. [39] proposed a similar approach. They modelled the physical activity as a set of recurrence plots distance matrices to capture temporal patterns in the signal. Afterward, a CNN was used to classify the distance matrices and obtain the final prediction. In [40], the authors experimentally found that image representation of time series data introduces different feature type that was not available in 1-dimensional sensor data. Hence, they first encoded the sensor signal as a 2-dimensional texture image using a recurrence plot to visualize the recurrent nature of a trajectory through phase space. Then they used a CNN model to learn different levels of features from the texture images. To address the variability in the distinctive region scale and sequence length, Zhang et al. [41] proposed two stages approach, where firstly they encoded the sensor data using Multi-scale Signed Recurrence Plots (MS-RP), an improvement of recurrence plot, and then applied a Fully Convolutional Networks and ResNet to handle these images.

2.2.2 Frequency/Spatial Domain Feature based Model

To derive frequency-domain features, the sensor data window must first be transformed into the frequency domain, typically with a Fast Fourier transform (FFT). A FFT typically produces a set of basis coefficients that represent the amplitudes of the signal's frequency components as well as the distribution of the signal's energy. From these coefficients, various methods can be used to characterise the spectral distribution. For example, In [42], the authors have implemented the idea of transforming the 1-dimensional signal into 2-dimensional using FFT. This frequency-domain image was called the spectrogram, which represents the composition of a signal from several frequencies over time, acts as an input to a three-layered CNN model for features extraction and classification. Lawal et al. [43] in their work encoded sensors signal into spectrogram using

Short Time Fourier Transformation (STFT). A simplified two-stream VGG-Net [44] like CNN architecture was proposed for activity and location recognition.

2.2.3 Co-ordinate Transformation and Probabilistic Features Based Model

Some studies represent the sensor data window in a polar coordinate system instead of the typical Cartesian coordinates, typically with a Gramian Angular Field (GAF). A GAF uses Gram Matrix, which preserves the temporal dependency. For example, Zhiguang Wang and Tim Oates [45] introduced two frameworks for encoding time series data as images known as GAF and Markov Transition Field (MTF). They used Tiled CNNs to classify the single GAF and MTF images as well as the compound GSF-MTF images. Qin et al. [46] introduced a novel method to encode time series data into two-channel GAF images by unifying global and local time-series features. Then they presented a fusion ResNet framework, which learned the generated GAF image pixels correspondences between acceleration and angular velocity features. Almost similar work was done by the authors in [47]. Contrary to the previous work, they used four different types of activity images and made each one multimodal by convolving it with two spatial domain filters: the Prewitt filter and the High-boost filter. ResNet-18 was used to extract the deep features from multi-modalities and fused by canonical correlation-based fusion. Finally, a multi-class SVM was used for activity recognition.
Inspired by the recent success of deep learning techniques especially CNN in computer vision, encoding time series data as images gains more acceptance among researchers. This method allows the machine to visually recognize and classify by learning visual patterns and structures. In [48], Hur et al. proposed a novel encoding technique for converting an inertial sensor signal into an image with minimal distortion, namely Iss2Image (Inertial sensor signal to Image). Iss2Image divided real-valued sensor reading into three parts: integers, first two decimal places, and the next two decimal places and then encoded as a three-channel image. Finally, a CNN model was used for image-based activity classification. Another similar encoding technique was proposed by Daniel et al. in [49]. The proposed INIM framework first encoded the sensors signal into 3D RGB images and then used a residual network trained on the ImageNet dataset [50] for activity recognition.

2.2.4 Feature Selection (FS) based Model

A few researchers have also tried to choose the relevant features utilizing various FS-based techniques for improving the overall accuracy in the field of activity recognition. Buenaventura et al. [51] proposed a HAR model based on sensor fusion in smartphones which used filter-based method to rank the features. An enhanced HAR method was proposed by Fan et al. [52] where Bee Swarm Optimization (BSO) with a deep-Q-network was used. Dewi et al. [53] performed a comparative study on HAR datasets using four classifiers namely Random Forest (RF), SVM, KNN, and Linear Discriminant Analysis (LDA) from which it was concluded that RF has the highest accuracy. Nguyen et al. [54] proposed a position-based FS method for body sensors for daily activity recognition. Filter based methods were used to reduce the feature set followed by a correlation-based optimization and a classifier to determine the overall accuracy of the proposed method.

2.3 Knowledge Gap

Over the years many researchers have proposed various state-of-the-art techniques to solve the activity recognition problem. Still there remains few facts which may improves the overall prediction accuracy such as:

1. In the case of wearable sensor signals, the majority of previous works capture either only time-domain information or only frequency domain information. Because these wearable sensor signals are continuous time-series data, they contain both time and correlated frequency information. Signal time and frequency domain information are both important for recognizing human activity using wearable sensors.

2. The growing popularity of deep learning techniques encourages researchers to use various deep learning algorithms to extract various important features for human activity classification. These deep learning algorithms generate a relatively large set of features, which may include some irrelevant features. These irrelevant features not only expand the feature space but also have a negative impact on performance.

Chapter 3

Methodology

This chapter provides an overview of the activity recognition system proposed in this thesis. It also describes the research methodology used to collect the necessary data for developing the data-model and evaluating the model and classification algorithms.

3.1 Overview of The Research Approach

The method used in this work for the development of the activity recognition system consists of various steps. (1) Firstly, the continuous raw sensor data are segmented using overlapping sliding window. (2) Secondly, each and every segmented activity is then encoded as 2-dimensional time and frequency domain representation. (3) Thirdly, important temporal and spatial features are extracted. (4) Fourthly, a FS framework is used to reduce the feature space by removing redundant features. (5) Finally, using this reduced feature set various activities are classified using the K-NN classifier.

3.2 Dataset Preparation

3.2.1 Used Sensor

Recent advancement in semiconductor technology improves the sensing technology. A range of wearable sensors have been used to assess daily mobility levels in free-living subjects.

1. **Accelerometer:** An accelerometer is a type of electronic sensor that measures the acceleration forces acting on an object to determine its position in space and monitor its movement. Figure 3.1 shows the tri-axial embedded accelerometer module.

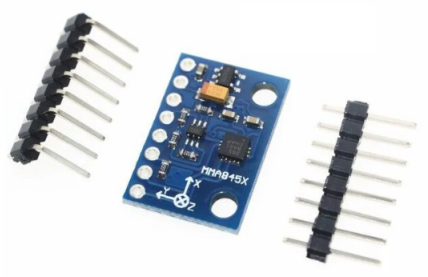


Figure 3.1: Tri-axial embedded accelerometer.

Accelerometers have emerged as the most useful tool for assessing mobility in both clinical and home settings. Transducers are used in accelerometers to measure acceleration. Most physical activity recognition systems have used accelerometers, which are capable of responding to both gravity and movement acceleration. The output of such accelerometers at any point in time is a linear combination of these two components, the acceleration component due to gravity (GA) and the acceleration component due to bodily motion. These two components cannot be easily separated because they are linearly combined and overlap in both time and frequency.

2. **Gyroscope:** Gyro sensors are devices that detect orientation and angular velocity. They are also known as angular rate sensors or angular velocity sensors. A triple axis MEMS gyroscope is capable of measuring rotation around three axes: x, y, and z. Figure 3.2 shows the MEMS gyro sensor.

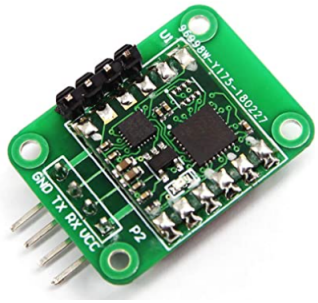


Figure 3.2: Tri-axial embedded MEMS gyro sensor.

Some gyros are single or dual axis, but triple axis gyros in a single chip are becoming smaller, less expensive, and more popular. When the gyro is rotated, the angular velocity of a small resonating mass changes. This movement is converted into very low-current electrical signals that a host microcontroller can amplify and read.

3. **Magnetometer:** A magnetometer, also known as a compass, is a type of navigation device that measures the strength of the magnetic field or the magnetic dipole moment. A magnetometer is a device with a sensor that measures the density of magnetic flux. Magnetometers measure the direction, strength, or relative variation of a magnetic field at a specific location. Figure 3.3 shows the embedded tiny magnetometer sensor.

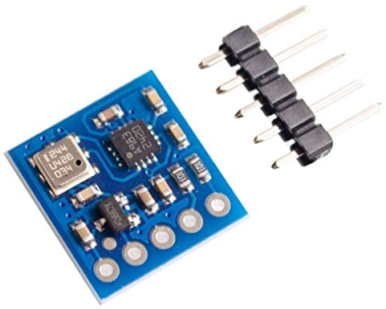


Figure 3.3: Embedded magnetometer sensor.

A magnetometer signal helps to understand human physical behaviour by magnetic induction signal. This system represents the motion of human body parts through variations in magnetic signal signals transmitted from transmitter to receiver during physical action.

3.2.2 Activity Segmentation

In activity classification, as in any other pattern recognition problem, the sensor signal is first divided into smaller time segments known as windows. Each window's features are computed separately and fed into the classification algorithms. Windows are defined concurrently with data collection in real-time applications, and a continuous real-time activity profile is produced. When processing sensor data offline, the windows are defined first, and classification algorithms are applied sequentially to each window. This data is then combined to create an activity profile for the entire signal.

In activity monitoring, three different windowing techniques have been used: event-defined windows, activity-defined windows and sliding windows.

In order to use event-defined windowing, pre-processing is required to locate specific events, such as heel strike or toe-off. These events are then used to define the next set of windows. The size of these windows is not fixed because such events may not be evenly spaced in time. A variety of methods for detecting heel strike and toe-off from body-worn sensor signals have been proposed. For example, search windows can be defined using either a low pass filtered version of the original signal [55, 56] or segmental angles [57], with maxima and minima corresponding to gait events. Another method is to determine when the anterio-posterior component of trunk acceleration changes sign.

The use of activity-defined windows is dependent on knowing when the activity changes. These points are then used to define sensor data windows, each corresponding to a different activity. Prior to explicitly identifying the specific activities, a number of methods for identifying activity-transition points have been proposed.Once defined, classification for each window is performed, sometimes using only a subset of the data contained within the window.

On the other hand, sliding window method divided the signal into fixed-length windows with no inter-window gaps or with specific percentage of overlapping. In no inter-window gap the window are placed one after another to sample activities. Figure 3.4 shows the no inter-window gap sampling process.

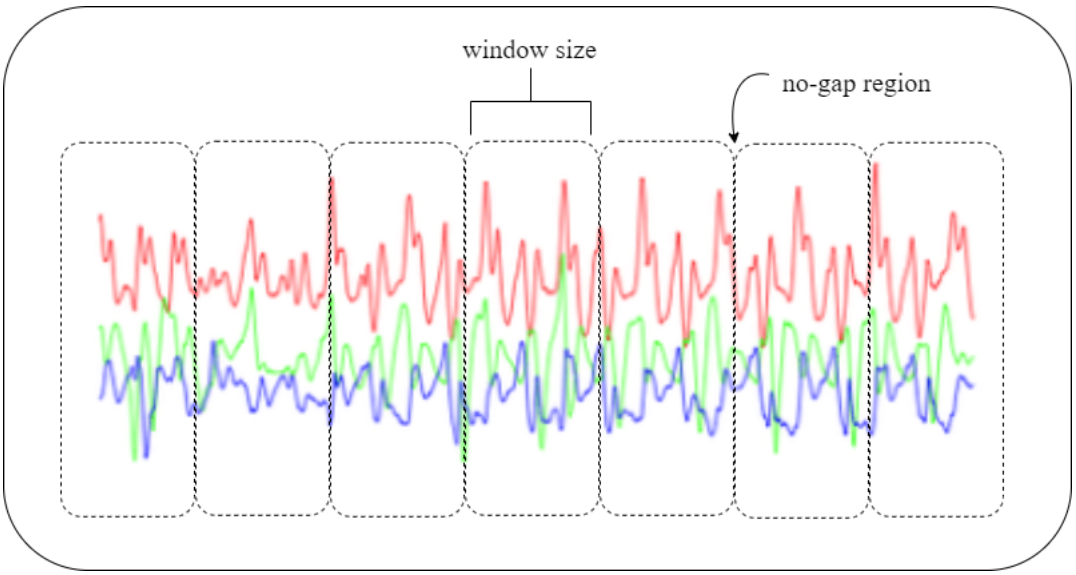


Figure 3.4: No inter-window gap based activity segmentation from tri-axial sensor data for wearable sensor based HAR.

In the case of overlapping sliding window, specific percentage of overlapping is provided to capture the transition or inter-dependencies between activities. Figure 3.5 shows the overlapping sliding window based activity sampling process.

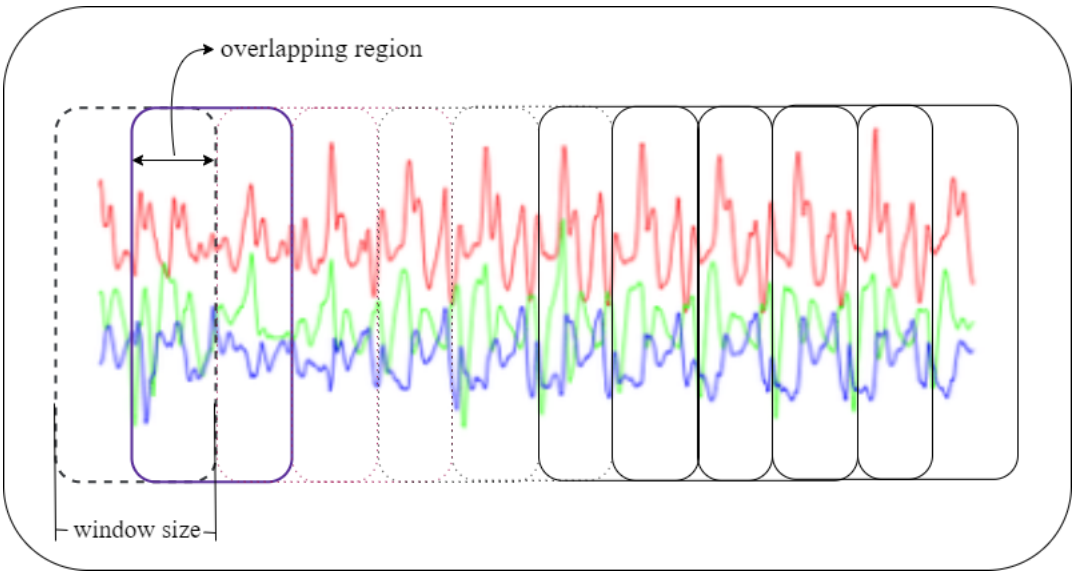


Figure 3.5: Overlapping window based activity segmentation from tri-axial sensor data for wearable sensor based HAR.

Previous studies used window sizes ranging from 0.25 s to 6.7 s, with some allowing for some overlap between adjacent windows[58, 36]. The sliding window method is ideal for real-time applications because it does not require pre-processing of the sensor signal. Because of its simplicity, this approach has been used in the majority of activity classification studies.

Since the goal of this study is to implement a model capable of recognizing activities in real-time, the fixed length overlapping sliding window based segmentation process is used. More specifically,

the overlapping sliding window with 50% overlap is considered. The performance of various window lengths for different activities across multiple subjects are analysed in order to select one that provided good estimates of the selected features while using the fewest samples in a given window. One limitation of this approach is that the appropriate window-length is determined by the training data. It does, however, provide a reasonable approximation of the study objectives.

3.3 Proposed Method

Here in this section, first the proposed activity image encoding technique is discussed briefly. Then the feature extraction process from the encoded images is described. The discussion of the feature extraction framework and activity classification process concludes this section. Figure 3.6 shows the working procedure of the proposed framework.

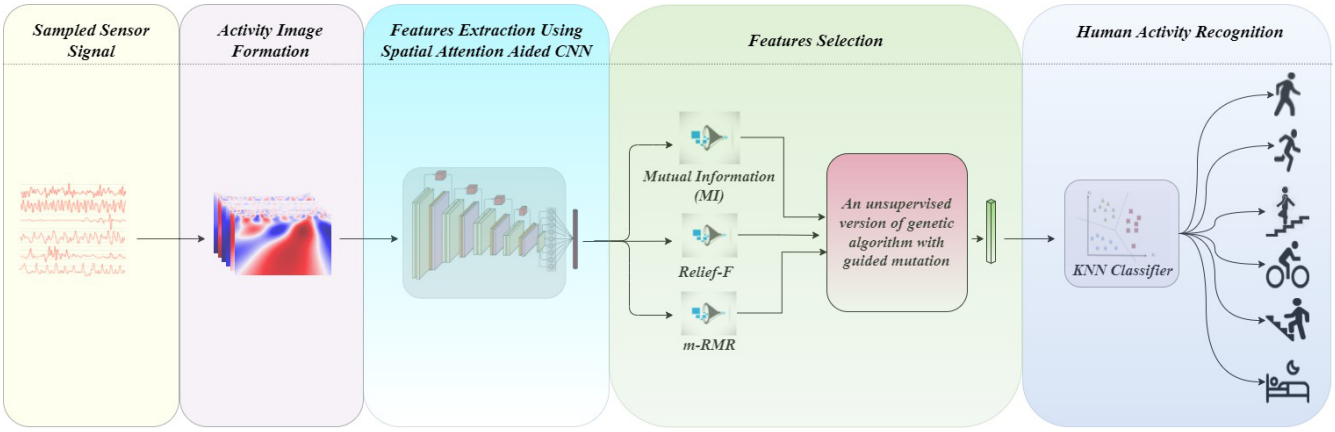


Figure 3.6: Overall workflow of the proposed HAR framework.

3.3.1 Time-Frequency Domain Transformation of Activity Signal

This section describes the process of transforming the sensor signal to corresponding time and frequency domain representation. This work utilizes the wavelet transformation, specifically continuous wavelet transformation to obtain spatial-temporal information as a 2-dimensional image. This images are called as activity images. This section starts with introducing the continuous wavelet transformation and then describes the required algorithm.

Continuous Wavelet Transform

Wavelet transform has been applied in time-frequency analysis and spatial domain signal analysis over the years and this is one of the most effective mathematical tools used for signal processing. A wavelet transform is a signal convolution with a set of functions derived from translation and dilation of a primary function. The primary function is known to as the mother wavelet, and the translated or dilated functions are referred to as wavelets.

A wavelet is a rapidly decaying wave-like oscillation defined as function $\psi(t)\epsilon L^2(R)$ with a zero mean and exists for a finite duration, localized both time and frequency. By scaling and translating this wavelet $\psi(t)$, we can produce a family of wavelets by using eq (3.1) as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{3.1}$$

where, $a, b \epsilon R$ and $a > 0$. a is known as the scaling parameter and b is the transitional value. The wavelet transform of a continuous signal with respect to wavelet function $\psi(t)$ is defined as eq (3.2)

$$W_x(a,b) = \int_{-\infty}^{+\infty} x(t)\psi^*_{a,b}(t)\,dt \tag{3.2}$$

where, $x(t)$ is a time-domain signal, $\psi^*_{a,b}(t)$ is the complex conjugate of mother wavelet. From eq(3.1) and eq(3.2) we get eq (3.3), which defines the CWT as

$$X_w(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} x(t)\psi^*\left(\frac{t-b}{a}\right)dt \tag{3.3}$$

CWT is nothing but the inner product of signal $x(t)$ with a continuous wavelet $\psi(t)$ scaled by parameter a and translated by value b. The pseudo code for the CWT is shown in Algorithm-1.

Algorithm 1: Pseudo code for Continuous Wavelet Transform (CWT)

Input: $1D\ time-series$: *list of fixed timestamps*
wavelet : *a function*
scale : *list of positive numbers*

Output: *2D coefficient matrix of size $s \times s$, where s is the length of the scale parameter*

Procedure:

1. Take the wavelet and compare it to a section at the starting of the original time-series signal.
2. Compute the inner product of the wavelet and the signal.
3. Shift the wavelet to the right and repeat steps 1-2 until the signal is processed.
4. Scale (stretch/shrink), the wavelet, and repeat steps 13.
5. Repeat steps 1 to 4 for all available values present in scale.

The outputs of the CWT are CWT coefficients, which reflect the similarity between the analyzed signal and the wavelet. These coefficients can be represented as a 2D image equivalent to the power spectrum, where time and scale/frequency are the 2 dimensions. However, the CWT coefficients depend on the choice of the mother wavelet.

One of the main advantages of wavelet transform is the presence of a wide variety of wavelets to choose from that best match the shape. This work uses the Gaussian Derivative Wavelets, specifically $5th$ order derivatives of the function given in eq (3.4)

$$\psi(t) = C\exp^{-t2} \tag{3.4}$$

where, C is the order-dependent normalization constant.

The $5th$ order Gaussian Derivative wavelet is a real-valued odd function, which is anti-symmetric around zero. The shapes of the $5th$ order Gaussian Derivative wavelet and various scaled wavelets are shown in Figure 3.7.

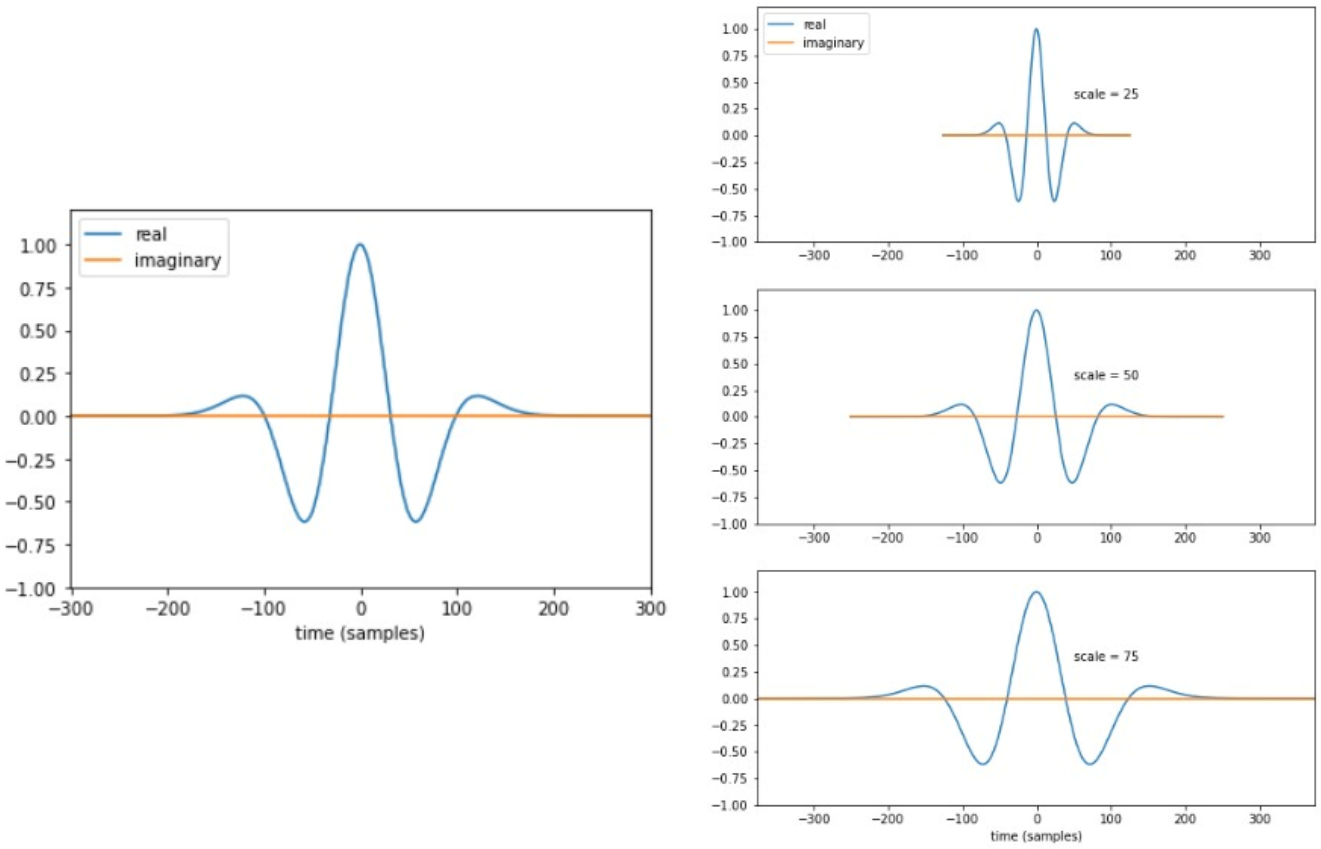


Figure 3.7: Anti-symmetric $5th$ order Gaussian Derivative wavelet and its various scaled version at scale 25, 50 and 75

As the wavelet is a real-valued function hence the imaginary part of the wavelet is zero. The $5th$ order Gaussian Derivative wavelet's structure is particularly effective for extracting more meaningful coefficients, since the wavelet may make a good alignment with the original sensor signal due to the scaling and translating parameters.

Inertial Sensor to Image Encoding Using CWT

In order to encode the raw sensor time-series data into an image form, this research utilizes the benefits of the 1-dimensional CWT, which takes 1-dimensional time-series as input and generates a 2-dimensional frequency-time domain scalogram. This scalogram is nothing but the CWT coefficients. Figure 3.8 depicts the image encoding process.

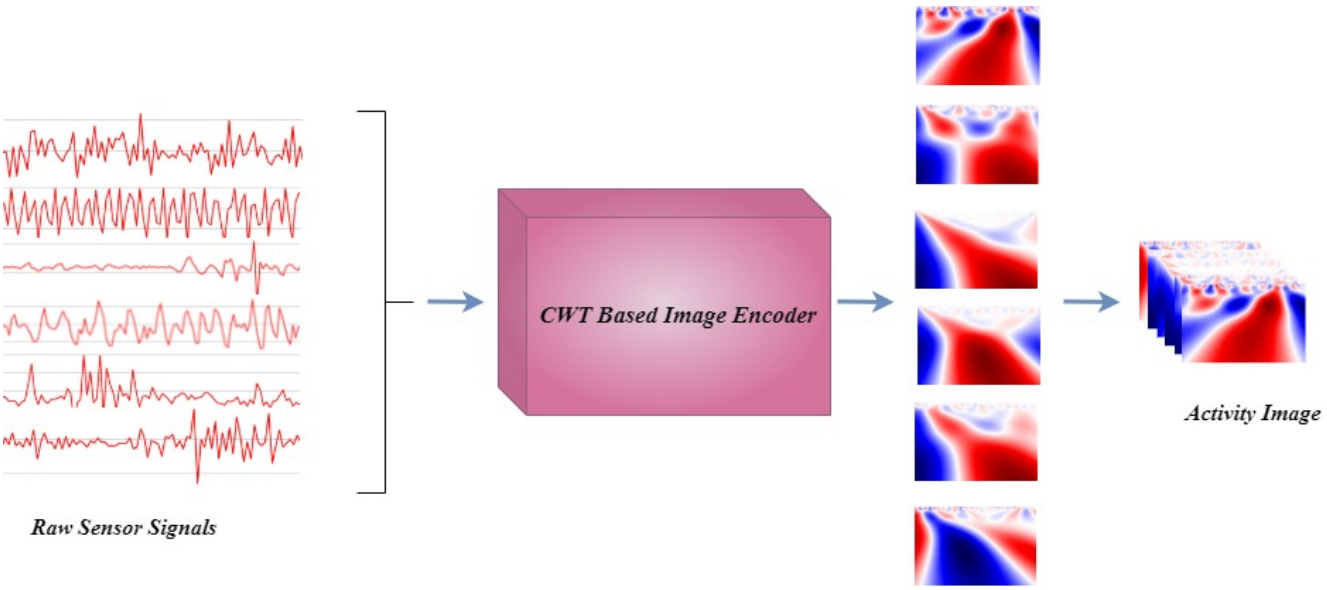


Figure 3.8: Activity image encoding process from the sampled raw sensor signals using CWT

Performing CWT on the entire time-series dataset is practically infeasible. Hence instead, CWT is applied on each sample of size $t \times c$, where t is the number of timestamps and c is the total number of sensor channels. The pseudo code for CWT based image encoding technique is given in Algorithm-2. The values of t and c vary from dataset to dataset. Each of the channels in c is a 1-dimensional time-series and act as the input to the CWT. t is used as the scale parameter. For each such sensor channel, a $t \times t$ scalogram is obtained as the output. Hence for one sample, a c dimensional $t \times t$ scalogram is found, where each dimension corresponds to each sensor channel.

Algorithm 2: Pseudo code for image encoding

Input: *A sampled dataset D, where each activity is sampled using a fixed-size overlapping window*
Output: *Images, collections of the corresponding encoded images as 4-dimensional array*
Start
$n \leftarrow$ total no. of samples/activities in the dataset D.
$s \leftarrow$[1, 2, 3, ., t] ▷ t is no. of timestamps in each activity sample.
$c \leftarrow$ no. of sensor channels in each activity sample.
images $\leftarrow$array of shape $n \times t \times t \times c$.
wavelet $\leftarrow$5*th* order Gaussian Derivative wavelet.
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** c **do**
    signal $\leftarrow$ D[i , : , j] ▷ extracting each channel
    coeff $\leftarrow$ **CWT(signal, wavelet, s)** ▷ finding CWT coefficients
    images[i , : , : , j] $\leftarrow$ coeff[: , : t] ▷ storing the images as an array
  end
end
End

Each and every activity in a dataset is encoded as a $t \times t \times c$ dimensional image using the method described above.

3.3.2 Feature Extraction Process

One of the most important phases in any classification task is Feature Extraction. The classifier's performance is determined by the quality of the features extracted by the Features Extractor. To extract features, this work employs a convolution neural network, a deep learning-based network known for image-related tasks. An attention mechanism, specifically spatial attention, is used to improve the quality of the extracted features. This section provides a detailed explanation of each component used in the Feature Extraction process.

Convolution Neural Network

CNN, or convolutional neural network, is a type of neural network model designed for working with two-dimensional image data, though it can also be used with one-dimensional and three-dimensional data. The convolutional layer, which gives the network its name, is central to the convolutional neural network. This layer performs a "convolution", which is a linear operation that involves multiplying a set of weights with the input, similar to a traditional neural network. This set of weights is referred to as a filter or kernel, and it is typically smaller than the input size. From left to right, top to bottom, the filter is applied systematically to each overlapping part or filter-sized patch of the input data to generate the features map. Figure 3.9 depicts the features map creation process in 2-dimensional space.

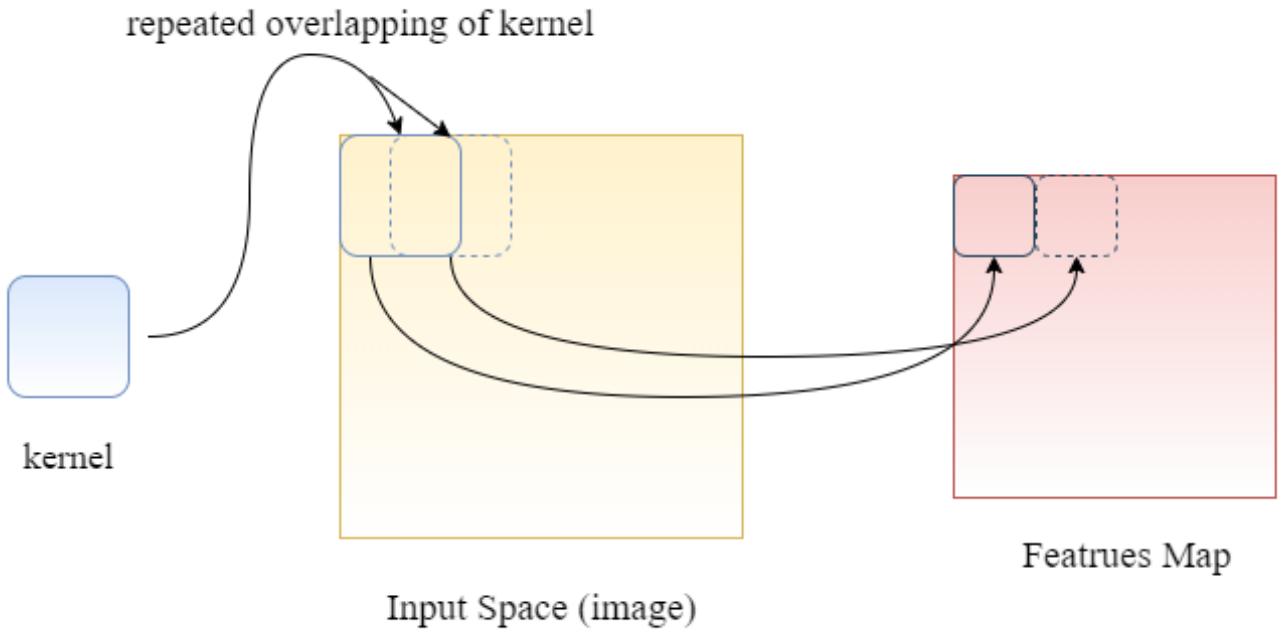


Figure 3.9: Example of a 2-dimensional filter/kernel applied to a 2-dimensional input to create a 2-dimensional features map

After creating a feature map, each value in the feature map can be subjected to a non-linearity, such as a ReLU, followed by a sub-sampling process to reduce the feature map dimensions. Layers, which perform these functions are known as activation and pooling layers respectively.

Spatial Attention

Attention is a technique in neural networks that mimics cognitive attention. The effect improves some aspects of the input data while decreasing others; the motivation is that the network should devote more attention to the small but important aspects of the data. Learning which part of the data is more important than another is context dependent. In 2014, Bahdanau et al. [59] proposed the attention mechanism to address the bottleneck problem that arises when using a fixed-length encoding vector, as the decoder would have limited access to the information provided by the input. Recently, the attention mechanisms attract more and more researchers interest and have been widely used with the CNN and RNN models in many domains like computer vision and image processing [60].

Spatial attention represents the attention mechanism/attention mask on the feature map, or a single cross-sectional slice of the tensor.

(a) Normal bird image

(b) Bird image with Spatial Attention Mask

Figure 3.10: Example of Spatial Attention Mechanism. The attention mask helps to focus just on the bird. - Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

For instance, in the image 3.10a, the object of interest is a bird, thus the Spatial Attention will generate a mask (visible in the image 3.10b) which will enhance the features that define that bird. Figure 3.10 shows the effect of spatial attention mask. By refining the feature maps with Spatial Attention, the input to the subsequent convolutional layers is improved, which eventually enhances the model performance. The spatial attention mechanism is primarily used in this study to focus on a specific spatial-temporal region in the activity image, where changes in spatial and temporal information are very sharp.

The Spatial Attention Modules (SAM), which are used in this work, are collection of three convolution layers with different kernel size. Figure 3.11 shows the structure of the SAM block.

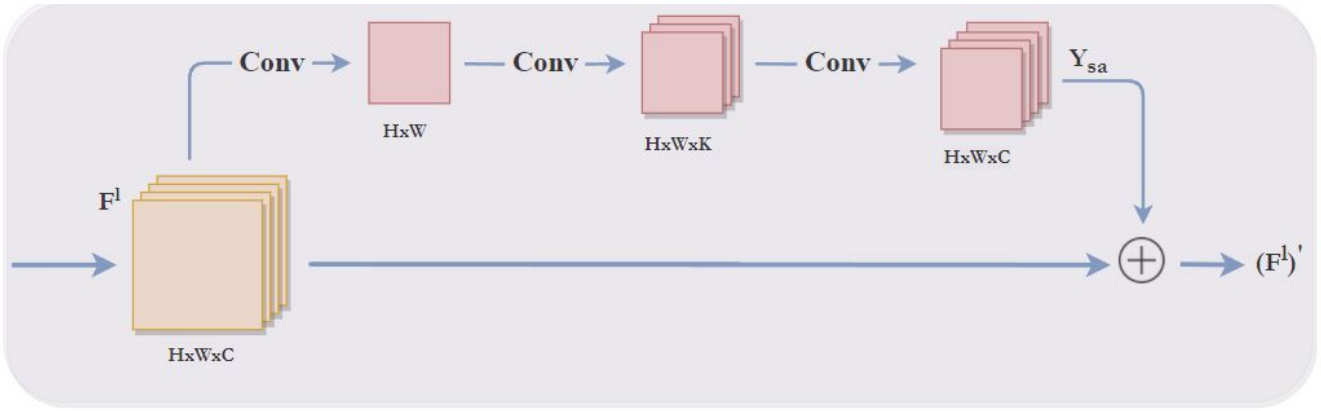


Figure 3.11: Illustration of the Spatial Attention Module used in this work

The SAM block takes features map of previous convolution layer as input and then a convolution layer with 1×1 kernel is first used to fuse the information along the channels, generating a 2-dimensional feature map $Y \epsilon R^{H \times W}$. Then, two 2-dimensional convolution layers are used to generate the spatial attention features map $Y_{sa} \epsilon R^{H \times W \times C}$. For these two 2-dimensional convolution layers, the number of convolution filters varies from module to module. Four such SAM blocks are used with the base CNN features extractor. The details of spatial attention module architecture are shown in Table 3.1.

Table 3.1: Details of Spatial Attention Module architecture used in the present work.

Module	Convolution Layer - 1		Convolution Layer - 2		Convolution Layer - 3	
	Filter Size	No of Filters	Filter Size	No of Filter	Filter Size	No of Filters
SAM - 1	1x1	1	3x3	16	3x3	32
SAM - 2	1x1	1	3x3	32	3x3	64
SAM - 3	1x1	1	3x3	32	3x3	64
SAM - 4	1x1	1	3x3	64	3x3	128

ReLU is used as the activation function for the convolution layers and padding operator to avoid the change in spatial size. Finally, the output of the SAM block Y_{sa} is used to re-calibrate F^l using eq (3.5).

$$\left(F^l\right)' = Y_{sa} + F^l \tag{3.5}$$

Where, F^l is the feature map from the previous convolution layer. This $\left(F^l\right)'$ acts as the input for the next CNN layer in the base network.

Feature Extractor Architecture

This section discusses the details about the feature extractor architecture. In this work, a four layered CNN has been used as the features extractor. To improve the feature extraction quality, Spatial Attention Module are also used. Figure 3.12 shows the architecture of the proposed

feature extractor. It mainly consists of a CNN having four convolution layers and spatial attention sub-networks. The spatial attention sub-networks, which are variants of widely used CNNs, use attention modules to fine-tune the feature maps in each convolution layer, thereby enhancing CNN's learning ability.

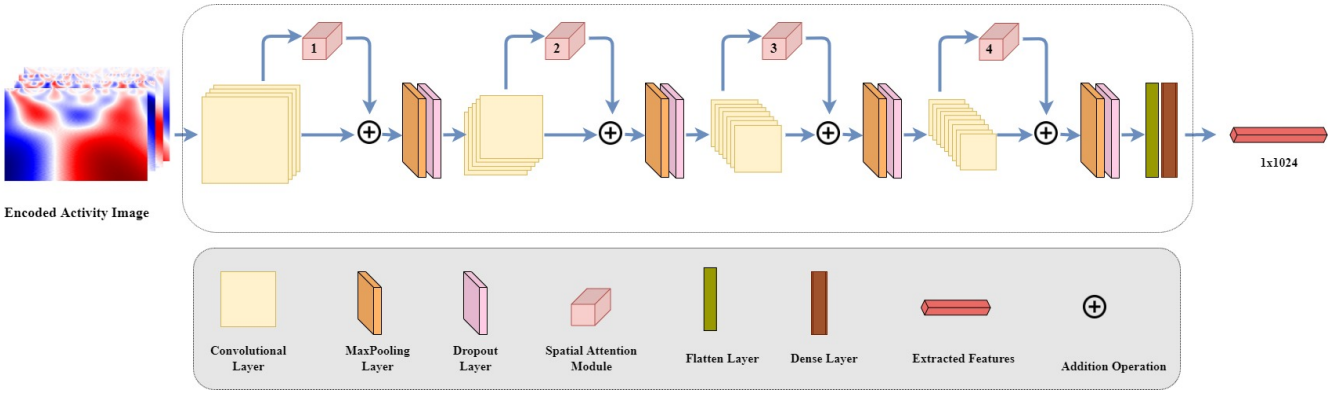


Figure 3.12: The architecture of the proposed CNN based feature extractor

Following each convolution layer, a max-pooling layer is used to lessen data variance, and a dropout layer to avoid over-fitting. Before the max-pooling layer, the attention feature maps from the spatial attention sub-network is added to re-calibrate the original features. This layering scheme is repeated three times with a different number of 3×3 filters. All neurons of these convolution layers have Re-LU (Rectified Linear Unit) as an activation function to learn the non-linear representation. The details of the network architecture are given in Table 3.2.

Table 3.2: Details of the CNN architecture used for the purpose of feature extraction. SAM here refers to Spatial Attention Module

Layer	Type	Filter Size	No of Filters	Strider
Input	128x128x6 for UCI-HAR 128x128x12 for MHEALTH 80x80x6 for WISDM	-	-	-
conv2D_1	Conv2D + ReLU	3x3	32	1x1
SAM_1	-	-	-	-
max_pooling2D_1	MaxPooling2D	2x2	-	-
dropout_1	Dropout (20%)	-	-	-
conv2D_2	Conv2D + ReLU	3x3	64	1x1
SAM_2	-	-	-	-
max_pooling2D_2	MaxPooling2D	2x2		
dropout_2	Dropout (20%)	-	-	-
conv2D_3	Conv2D + ReLU	3x3	64	1x1
SAM_3	-	-	-	-
max_pooling2D_3	MaxPooling2D	2x2	-	-
dropout_3	Dropout (20%)			
conv2D_4	Conv2D + ReLU	3x3	128	1x1
SAM_4	-	-	-	-
max_pooling2D_4	MaxPooling2D	2x2	-	-
dropout_4	Dropout (20%)	-	-	-
flatten	Flatten()	-	-	-
output	Fully Connected Layer (1024 units) + ReLU	-	-	-

At last, the output features are first flattened and then passed through a fully connected layer, which generates a 1024-dimensional feature vector from the input image.

3.3.3 Feature Selection

FS is the process of reducing the number of input variables when developing a predictive model. This is directly linked with the dimensionality reduction. Inclusion all of the redundant and irrelevant features may have a negative impact on the model's overall performance and accuracy. As a result, it is critical to identify and select the most relevant features from the data while removing irrelevant or less important features, which is accomplished through FS mechanism. There are basically three main types of FS techniques: wrapper methods, embedded methods and filter methods. Figure 3.13 shows different categories of FS techniques.

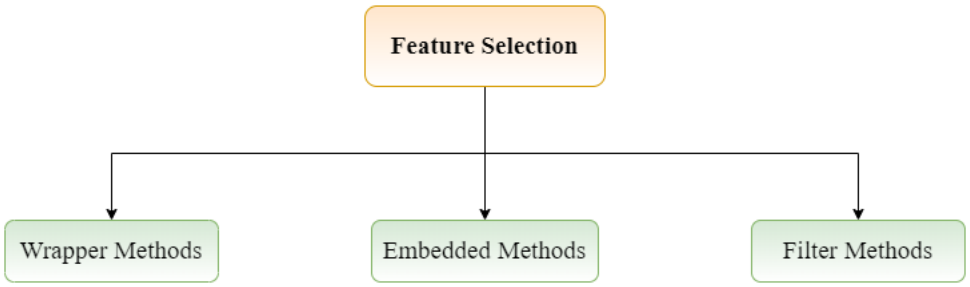


Figure 3.13: Overview of FS techniques

A wrapper method employs a classifier to compute the effectiveness of each candidate solution (i.e., a subset of features) and select the subset of features with the highest effectiveness score. Filter-based methods, on the other hand, rank the features in order of importance and eliminate the less important features. Because filter methods do not require a learning algorithm, they are generally faster than wrapper methods.

In the proposed method, Genetic Algorithm (GA) is used as the unsupervised FS algorithm, and three different filter methods are used to calculate the fitness of each chromosome in the population of GA. This section describes the details of FS techniques used in this work.

Filter Methods

This study used three filter-based approaches, MI, ReliefF, and mRMR, to determine the fitness of each individual chromosome.

1. **Mutual Information**: MI [61]is used to measure the non-linear relations between two random variables. It is used to quantify the quantity of data obtained from a random variable by observing the other random variable. It can be referred to as the reduction of uncertainty of a random variable when the other variable is known. Hence, a high MI value suggests a large reduction in uncertainty while a low value suggests less reduction. It can be calculated using equation 3.6:

$$I(X;Y) = \sum_{y \epsilon Y} \sum_{x \epsilon X} P_{X,Y}(x,y) \cdot \log \left(\frac{P_{X,Y}(x,y)}{P_X(x) \cdot P_Y(y)} \right) \tag{3.6}$$

where $P_{X,Y}(x,y)$ denotes the joint probability density function of X and Y, and the marginal density functions are denoted by $P_X(x)$ and $P_Y(y)$. The similarity the joint distribution

$P_{X,Y}(x, y)$ to the product of the factored marginal distributions is determined by MI. It equals zero if and only if two random variables are independent, and higher values indicate greater dependency.

2. **Relief-F**: Relief was proposed by Kira and Rendell [62] for binary class problems by using the Euclidean distance measure. Relief-F algorithm is based on the Relief algorithm, a filter method used in FS. Relief was designed primarily for use in the problems of binary classification with discrete or numerical features. Relief assigns a relative weight/score to each feature and acts as a filter method by eliminating the low-ranked features. The feature score changes according to the detection of feature value differences between neighbouring instance pairs. If a difference in feature value is discovered with the same class (a hit) in a neighbouring instance pair, the feature score falls. On the other hand, if a feature value difference is observed with different class values (a miss) in a neighbouring instance pair, the feature score climbs. However, it is limited to only two class problems. An extension of the Relief-F algorithm can be used to solve multi-class problems by searching for k closest misses in each class and averaging their contributions for updating W, weighted by each class's prior probability. In the contribution of weights to each feature, it takes the average of k nearest hits and misses. This k can be adjusted and set based on the dataset in question. Furthermore, Relief-F can handle missing data by employing a conditional probability of feature weights. It is defined by the formula given in equation 3.7.

$$W(X_j) = \frac{1}{nK} \sum_{i=1}^{n} \sum_{l=1}^{K} \left(|x_{i,j} - x_{i,j}^{M_l}| - |x_{i,j} - x_{i,j}^{H_l}| \right) \tag{3.7}$$

where $x_{i,j}$,$x_{i,j}^{M_l}$ or $x_{i,j}^{H_l}$ denotes the j-th component of sample x_i, its l-th closest Miss $x_i^{M_l}$ or its l-th closest Hit $x_i^{H_l}$ respectively. n is the total number of samples and K is the number of Misses or Hits considered for each samples.

3. **Minimum Redundancy maximum relevance**: mRMR [63] is a filter ranking approach in FS that ranks features according to correlation to the class and itself. Preferably, features with a high correlation with the class (output) and a low correlation between themselves are chosen. For continuous features, correlation with the class (relevance) can be evaluated by the F-statistic values and the correlation between features (redundancy) can be determined using Pearson Correlation Coefficient (PCC) values. A greedy search is applied to select the features one by one as the final goal is to maximize the objective function, which is determined by relevance and redundancy. MID (Mutual Information Difference) and MIQ (Mutual Information Quotient) criteria are the two commonly used types of the objective function which represent the difference between relevance and redundancy, or the quotient of relevance and redundancy. It is calculated using the formula given in equation 3.8

$$score_i(f) = \frac{F(f, target)}{\sum_{s \epsilon f'(i-1)} |corr(f, s)| / (i-1)} \tag{3.8}$$

where i is the i-th iteration, f is the feature that is being evaluated, F is the F-static, $f'(i-1)$ denotes the features selected until $i-1$ iterations and $corr$ is Pearson correlation.

Genetic Algorithm: An overview

GA is a popular meta-heuristic evolutionary algorithm which is used for solving complex optimization problems. It is a nature-inspired algorithm with biological features like selection, crossover, and mutation. GA comprises the following steps initial population creation, parent selection, crossover, mutation, and generation of child chromosomes. Initially, a random population is generated with a finite number of chromosomes, each filled with some random values of fixed length. Parent chromosomes are selected from this set of chromosomes which are further used to create the child chromosomes after performing crossover and mutation. A fitness function is defined to evaluate the fitness of each chromosome. If the fitness values of the child chromosomes surpass the fitness of some existing chromosomes in the current population, they replace the chromosomes having low fitness values. The fitness measures the quality of the represented solution obtained at each iteration. These processes are repeated until the generation of the next set of chromosomes that go through the same selection, crossover, and mutation process and eventually the subsequent generations are generated through this method. Individuals with the least fitness die as new generations form, making room for new offspring. This leads to a near optimal solution after a fixed number of iterations. A binary version of GA is used in FS, with each chromosome represented as a vector of '0's and '1's. A '0' indicates that the corresponding feature is not selected, whereas a '1' indicates that the corresponding feature is selected.

Proposed GA Variant

GA is one of the oldest and classical evolutionary algorithms, inspired by nature. Over the years, various researchers have utilized this algorithm in the field of FS and optimization. It is proved to be one of the best-known algorithms which provides a near-optimal subset of features from the whole feature space. Exploration and exploitation are being performed by the key operators i.e., crossover and mutation. Numerous modifications have been suggested by various researchers to improve GA and reach the near optimal solution. The mutation in GA is decided by a mutation probability which is quite random in nature. Moreover, the fitness of each candidate solution is determined by a learning algorithm (i.e., a classifier) which is often very time-consuming. Keeping the above facts in mind, this study proposes a modified version of GA which estimates the fitness of the candidate solutions by calculating the aggregate of three filter-based methods, thereby improving the computational time significantly. Also, instead of random mutation, a different mutation method is proposed which improves the fitness of the individual candidate solution. A multi-point crossover is used and for parent selection is done using Roulette wheel for better exploitation. The pseudo code of the mutation technique is described in Algorithm-3.

Fitness Function

Wrapper based FS methods generally use a learning algorithm (i.e., a classifier) to evaluate the fitness of the chromosomes. Since GA is commonly used a wrapper-based method, it follows the same logic, however it increases the computational time. To overcome this problem, the usage of classifier is replaced by determining the score of each feature vector (i.e., a chromosome) by the help of filter methods, which aids in assessing the strength of each chromosome in an unsupervised way.

A chromosome is a binary vector with 0 indicating that the feature is to be not taken and 1 indicating that the feature is to be taken. By using the three filter methods, a filter-value (i.e., a score) corresponding to each feature is obtained. The filter-value of each feature is the average

Algorithm 3: Pseudo code of the mutation technique

Input: *A binary feature vector (F) of size M*
A score vector of length M containing the average of the three filter methods (score).

Output: *An improved binary feature vector (F') of size M.*

Start
$F' = F$
$total = 0$ ▷ sum of scores of non-zero valued features
$n = 0$ ▷ count of non-zero valued features
for $i \leftarrow 1$ **to** M **do**
  if $F(i) \neq 0$ **then**
    $total = total + score(i), n = n + 1$
  end
end
$avgscore = \frac{total}{n}$
for $i \leftarrow 1$ **to** M **do**
  if $rand(0, 1) > mutation\_probability$ **then**
    continue
  end
  if $score(i) > avgscore$ *and* $F(i) = 0$ **then**
    $F'(i) = 1$, update avgscore ▷ Add Feature
  end
  if $score(i) < avgscore$ *and* $F(i) = 1$ **then**
    $F'(i) = 0$, update avgscore ▷ Remove Feature
  end
end
End

of the value of the three filter methods. We can say that the feature column with the maximum filter-value is most important while the feature with the minimum filter-value is least important. Hence to calculate the score of each individual chromosome, the mean of the filter-values of all the features which are currently 1 are taken. The pseudo-code of the fitness value calculation is described in the Algorithm-4.

Algorithm 4: Pseudo code of the fitness value calculation

Input: *A chromosome (binary vector) of size M.*
Scores of ReliefF, MI and mRMR.
Output: *The fitness value of the chromosome.*
Start
for $i \leftarrow 1$ **to** M **do**
  $score(i) = \frac{ReliefF(i)+MI(i)+mRMR(i)}{3}$
end
$total = 0$ ▷ sum of scores of non-zero valued features.
$n = 0$ ▷ count of non-zero valued features
for $i \leftarrow 1$ **to** M **do**
  if $chromosome(i) \neq 0$ **then**
    $total = total + score(i), n = n + 1$
  end
end
$fitness = \frac{total}{n}$
End

In FS, this study intend to increase the classification accuracy of the problem under consideration and decrease the number of features selected simultaneously. In order to do so, a single objective function is defined, which estimates the overall fitness of each chromosome (feature subset). This objective function is defined in equation 3.9.

$$Fitness_{overall} = \alpha \times F + (1 - \alpha) \times \frac{|F| - |f|}{|F|} \tag{3.9}$$

Where F is the fitness of the chromosome, $\alpha \, \epsilon \, [0, 1]$ represents the relative weightage between the fitness value and number of features not selected, $|F|$ is the number of features in the given dataset and $|f|$ is the number of features in the feature subset.

Since this study aims to increase the fitness value and reduce the number of features in the feature subset, the research's objective is to increase the Fitness_overall value.

3.3.4 Classification Of Human Activities

Once features have been derived to characterize a window of sensor data, they are used as input to a classification algorithm. There are many classifiers are out there for the classification task. In this research, K-Nearest Neighbour(K-NN) is used as the classifier. It is a distance based supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The goal of the KNN algorithm is to find all of a new unknown data point's nearest neighbours in order to determine what class it belongs to. The pseudo code for K-NN is given in algorithm-5

Algorithm 5: Pseudo code for K-NN

Input: *Unlabelled data point*
Output: *Assigned label for that data point*
Procedure:

1. Select the number of 'K' value.
2. Calculate Euclidean distance between the new data point and all data points.
3. Sort all the data points in the ascending order of distance from the new data point and select starring 'K' data point from the sorted list.
4. Count the class label for each of the 'K' nearest point.
5. Assign the new data points to that class label for which the number of the neighbor is maximum.

K-NN is also called as the distance based classifier as it uses the euclidean distance to find the nearest neighbour. Euclidean distance is a Cartesian geometry based distance metric. The euclidean distance between two point $A(x1, y1)$ and $B(x2, y2)$ are calculated using equation 3.10

$$dist_{(A,B)} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)} \tag{3.10}$$

K-NN is one of the most used classifier as it is vary simple to implement, it is very robust to noise and it's effectiveness with large training dataset.

The K-NN was trained utilising the optimal features set provided by the FS approaches in this study. The overall classification of K-NN was good due to low feature dimensionality and a large dataset.

Chapter 4

Experiments And Results

This chapter discusses the improvements found in HAR accuracy with the application of the feature selection approach, as well as the effect of various other key parameters on overall accuracy. This chapter begins with a detailed description of the dataset used in this work, then discusses the implementation details, various performance metrics used to measure performance, and finally discusses the experimental results.

4.1 Dataset Description

In this work, three publicly available wearable sensor datasets have been used. The implemented model is rigorously tested on this three datasets.

4.1.1 UCI-HAR Dataset

The UCI-HAR [64] is a publicly available benchmark dataset for HAR. The dataset was created by recording activities of daily living (ADL) using the embedded inertial sensors of a waist-mounted smartphone. Each participant in a group of 30 volunteers ranging in age from 19-48 years performed six activities: Walking, Walking, Upstairs, Walking, Downstairs, Sitting, Standing, and Laying wearing a Samsung Galaxy S II smartphone on their waist. Figure 4.1 and 4.2 show images and graphs of the walking upstairs and walking actions and their corresponding acceleration data.

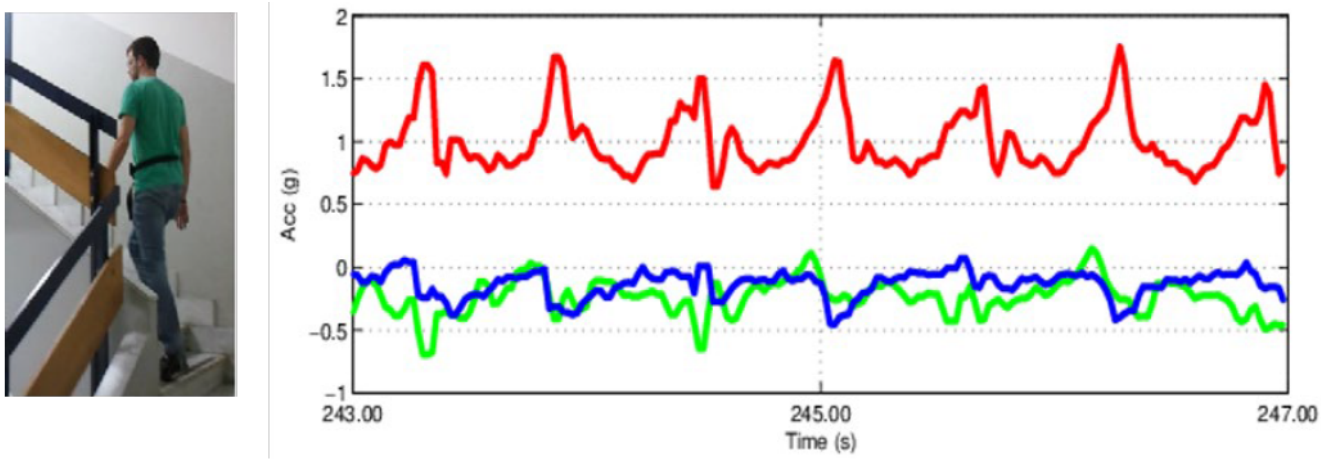


Figure 4.1: Image and Acceleration Signal for activity Walking Upstairs of UCI-HAR dataset-D. Anguita, A. Ghio, L. Oneto, X.Parra and J. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones, 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

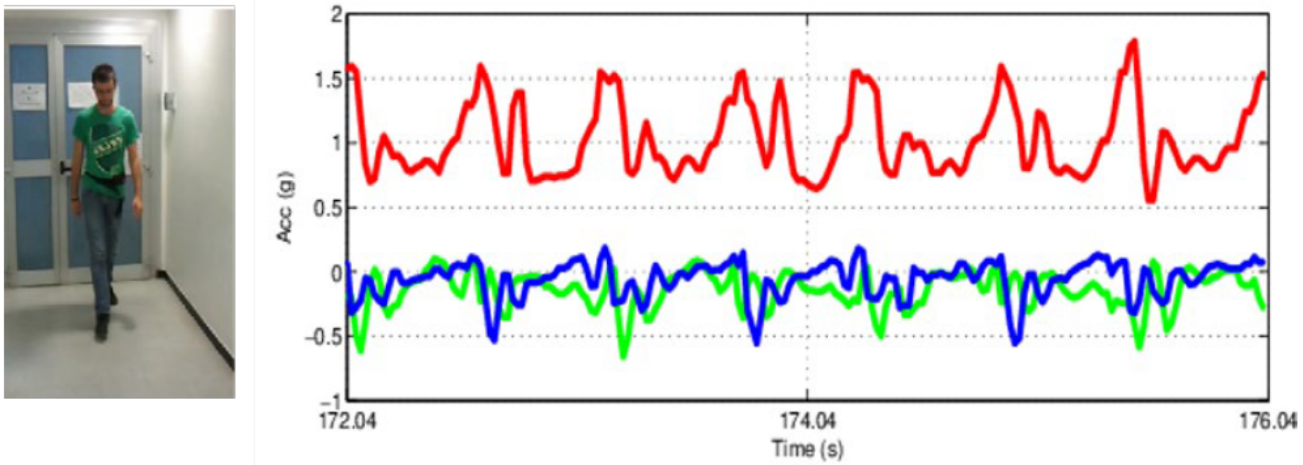
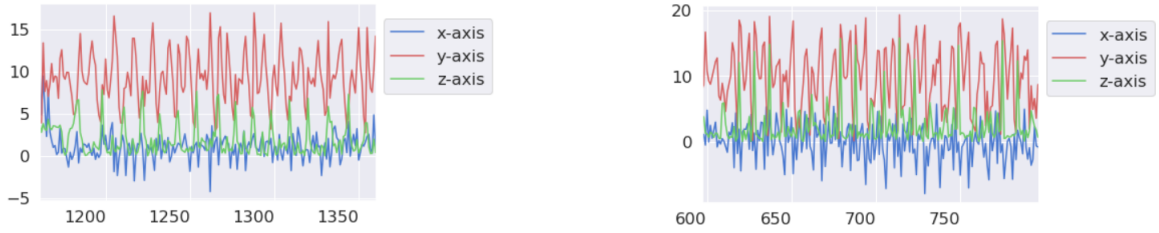


Figure 4.2: Image and Acceleration Signal for activity Walking of UCI-HAR dataset-D. Anguita, A. Ghio, L. Oneto, X.Parra and J. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones, 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

Total nine features (body acceleration, total acceleration, and angular velocity signals in all X, Y, Z-axis) were captured using the embedded accelerometer and gyroscope at a constant sampling rate of 50Hz. The raw signals were first pre-processed by applying noise filters and then sampled using a fixed-length overlapping sliding window of 2.56 sec and 50% overlap (128 readings per window). The dataset is randomly partitioned into two parts, where 70% were used for training and the rest 30% for testing.

4.1.2 WISDM Dataset

WISDM dataset [65] contains data collected through controlled laboratory conditions in Fordham Universitys Wireless Sensor Data Mining lab. The samples were captured using a smartphone embedded accelerometer and the data collection process was controlled using an application that was executed on an android smartphone. Figure 4.3 shows the data of the accelerometer sensor used in WISDM dataset.



(a) "Walking Upstairs" (b) "Walking"

Figure 4.3: Tri-axial acceleration signal for two different action of WISDM dataser

The experiment was carried out on 36 people and each performed six activities - Walking, Jogging, Sitting, Standing, Upstairs, and Downstairs with an Android phone in their front leg pocket. The 3-axial accelerometer signals were collected at a constant sampling rate of 20Hz i.e., each reading at every 50ms and a total of 20 readings per second. For our proposed work, the raw signals are sampled using a fixed-length overlapping window of 4 sec and 50% overlap (80 readings per window).

4.1.3 MHEALTH Dataset

The Mobile Health (MHEALTH)[66, 67] dataset is a multi-modal wearable sensor dataset. The dataset contained body motion and vital signs recordings for 10 volunteers of diverse profiles. Each volunteer performed 12 different physical activities (Standing Still, Lying Down, Walking, Climbing Stairs, Cycling, Jogging, Running, etc.) wearing three wearable sensors (accelerometer, gyroscope, and 2-led electrocardiogram). The sensors were attached to the chest, right wrist, and left ankle with elastic straps. Using these sensors various motions like acceleration, angular velocity, magnetic field orientation were measured for better body dynamics while performing different activities. All the sensor modalities were recorded at a constant sampling rate of 50Hz. For our proposed work, we consider only the accelerometer and gyroscope sensors readings placed on different body parts. Similar to WISDM dataset, the raw signals are sampled using a fixed-length overlapping window of 2.56 sec and 50% overlap (128 readings per window).

Table 4.1 provides the information about the three datasets. UCI-HAR and WISDM both the datasets contain 6 activities but the number of sensors is different. The MHEALTH dataset contains the 12 activities with more additional sensors. UCI-HAR contains the largest number of training and testing data. Whereas MHEALTH contains more additional sensors compared to the rest of two datasets.

Table 4.1: Details of the datasets used.

Dataset	No. of Activities	Sensors	Sampling rate (in Hz)	No. of Training Samples	No of Testing Samples
UCI-HAR	6	Accelerometer, Gyroscope	50	7352	2947
WISDM	6	Accelerometer	20	5806	1452
MHEALTH	12	Accelerometer, Gyroscope, Magnetometer, 2-led ECG	50	4288	1073

4.2 Performance Metrics

To measure the performances, in this work, mainly accuracy, precision, recall, F1- score, and confusion matrix have been used as the performance measures. Besides, the micro-averaging scores for calculating precision, recall and F1- scores have been used.

4.2.1 Accuracy

Accuracy is defined as the proportion of correctly predicted samples to the total number of samples. A True Positive (TP) outcome is one in which the model correctly predicts the positive class. A True Negative (TN), on the other hand, is an outcome in which the model correctly predicts the negative class. Similarly, a False Positive (FP) is an outcome in which the model predicts the positive class incorrectly and a False Negative (FN) is an outcome in which the model predicts the negative class incorrectly. The accuracy can be calculated in terms of TP, TN, FN, and FP using eq (4.1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

4.2.2 Precision

Precision is defined as the percentage of positive samples identified correctly, based on the total number of samples identified as positive. Precision can be calculated using eq (4.2).

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

4.2.3 Recall

Recall is the proportion of positive samples that are accurately identified out of all positive trials. We can calculate the recall using eq (4.3).

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

4.2.4 F1-score

F1-score is a comprehensive approximation of the model's accuracy and it is nothing but the harmonic mean of precision and recall. It can the calculated using eq (4.4).

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.4}$$

4.2.5 Confusion Matrix

Confusion matrix is a square matrix that represents the overall performance of a classification model. The rows of the confusion matrix represent true class label instances, while the columns represent predicted class label instances. This matrix's diagonal elements count the number of trials where the predicted label equals the true label. The confusion matrix is an important metric for visualizing the model's classification performance.

4.3 Model Implementation

The models are built using the Python programming language, the Keras application programming interface (API), and the Tensorflow framework. Deep learning algorithms require a significant amount of computing power, and running these algorithms on a system with only a Central Processing Unit (CPU) can take hours to days. When running deep learning algorithms, GPUs, which are more powerful than CPUs, are preferred because they can quickly compute the complex mathematical operations required by deep learning neural networks. The experiments are performed on a laptop with having AMD Ryzen 7 4800H (2.90 GHz) processor with 8 GB of RAM and NVIDIA GeForce GTX 1660 Ti GPU with 4 GB of VRAM. The PC is powered by a 64-bit Windows 10 operating system.

For the activity image encoding process, more specifically for CWT based transformation, PyWavelets [68], an open-source python wavelet transform library, has been used.

The feature extractor model is trained under supervised learning methodology. All the weight and bias used for different layers have been initialized randomly. Adam optimizer is used and the sparse categorical cross entropy losses are minimized. The CNN model is trained for 150 epochs with a batch size of 32. Table 4.2 provides the hyper-parameter details used to tuned the model.

Table 4.2: Details about the different hyper-parameters used during experimentation.

Stage	Hyper-parameter	Value
Feature Extraction	Optimizer	Adam
	Learning Rate	0.001
	Number of Epochs	150
	Batch Size	32
Feature Selection	Population Size	10
	Crossover Probability	0.6
	No of Iteration	20
Classification	K value for K-NN	5

For FS techniques, different values of various hyper-parameters have been used for experimentation. Finally, for the proposed method with FS, 10 as the population size has been used, and the value of crossover probability has been set to 0.6. For the KNN classifier, the k value is set to 5.

4.4 Results and Discussion

4.4.1 Without Feature Selection vs With Feature Selection

Initially, the experiments have been carried out without the use of FS. The CNN feature extractor has been used to extract features for activity recognition. The model's performance on three different datasets is summarized in Table 4.3 before FS.

Table 4.3: Performance details of the proposed method without FS.

Dataset	Extracted Features	Accuracy (in %)	Precision	Recall	F-1 Score
UCI-HAR	1024	98.74	0.9874	0.9874	0.9874
WISDM	1024	98.34	0.9834	0.9834	0.9834
MHEALTH	1024	99.72	0.9972	0.9972	0.9972

The feature extractor, extract three different 1024×1 dimensional features map. These extracted features were then directly used for the activity classification.

Following that, the same experiment is repeated using the FS technique. The features extracted by the CNN-based feature extractor have been passed through the FS framework first to select only the best features. The best reduced feature set is then used for activity recognition. Table 4.4 summarizes the performance with FS.

Table 4.4: Performance details of the proposed method with FS.

Dataset	Reduced Features	Accuracy (in %)	Precision	Recall	F-1 Score
UCI-HAR	242	99.45	0.9945	0.9945	0.9945
WISDM	380	99.38	0.9938	0.9938	0.9938
MHEALTH	307	99.90	0.9990	0.9990	0.9990

If Tables 4.3 and 4.4 are compared, it is clear that application of FS technique reduces the feature space significantly. For all the three datasets, the reductions in size of the feature map are 76.36% for UCI-HAR, 62.89% for WISDM and 70.02% for MHEALTH. At the same time, application of the FS technique increases the accuracy by 0.72% for UCI-HAR, 1.01% for WISDM and 0.18% for MHEALTH dataset.

For better understanding of the training process of the CNN based features extractor, Figure 4.4 depicts the training history of the feature extractor network for UCI-HAR dataset, while the same for the WISDM and MHEALTH datasets are shown in Figures 4.5 and 4.6, respectively.

Figure 4.4: (a) Accuracy Plots for Training and Testing (b) Loss Plots for Training and Testing obtained using feature extractor model for UCI-HAR dataset

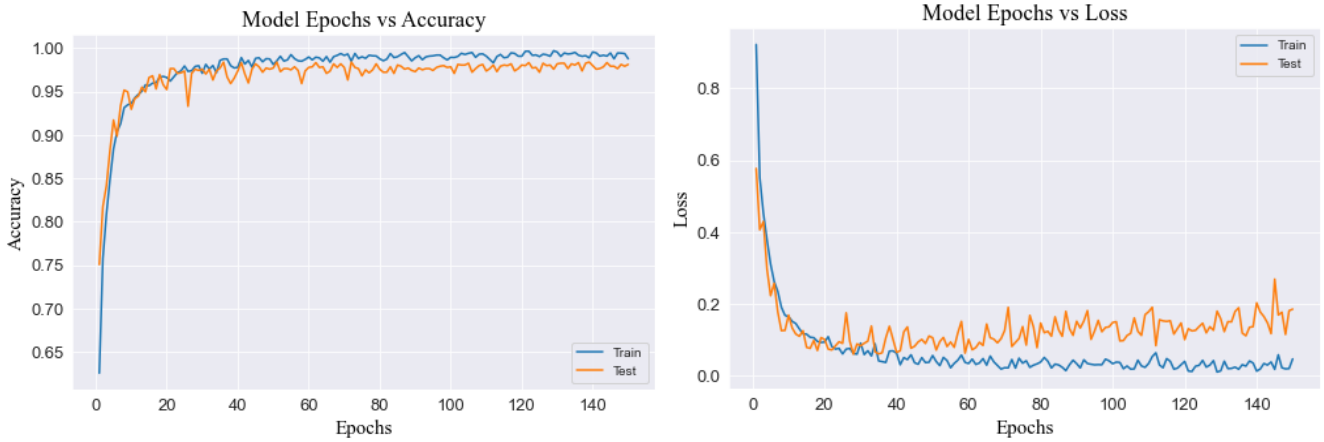


Figure 4.5: (a) Accuracy Plots for Training and Testing (b) Loss Plots for Training and Testing obtained using feature extractor model for WISDM dataset

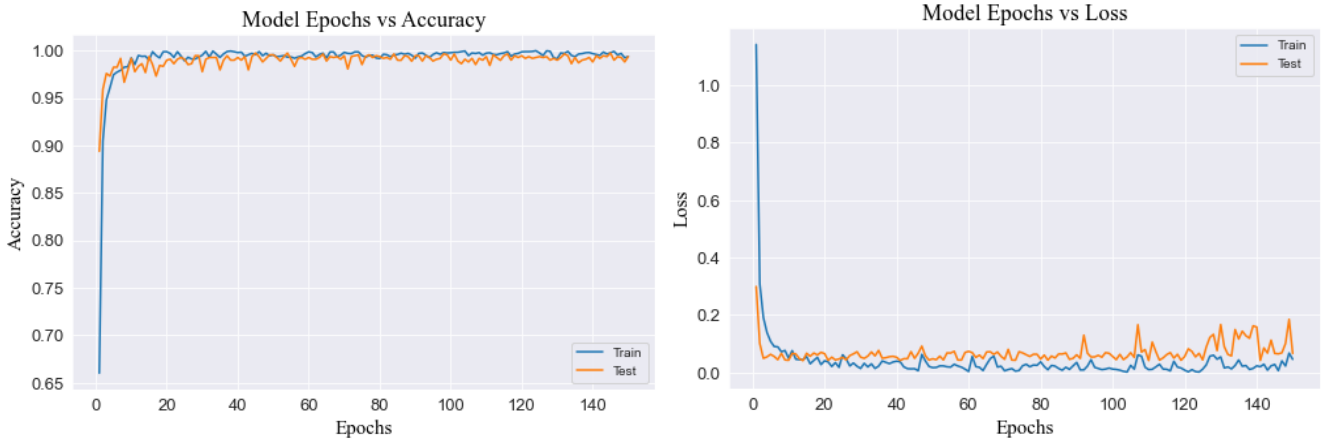


Figure 4.6: (a) Accuracy Plots for Training and Testing (b) Loss Plots for Training and Testing obtained using feature extractor model for MHEALTH dataset

4.4.2 Detailed Evaluation on UCI-HAR Dataset

Figure 4.7 shows the confusion matrices of the proposed method without FS and with FS side by side.

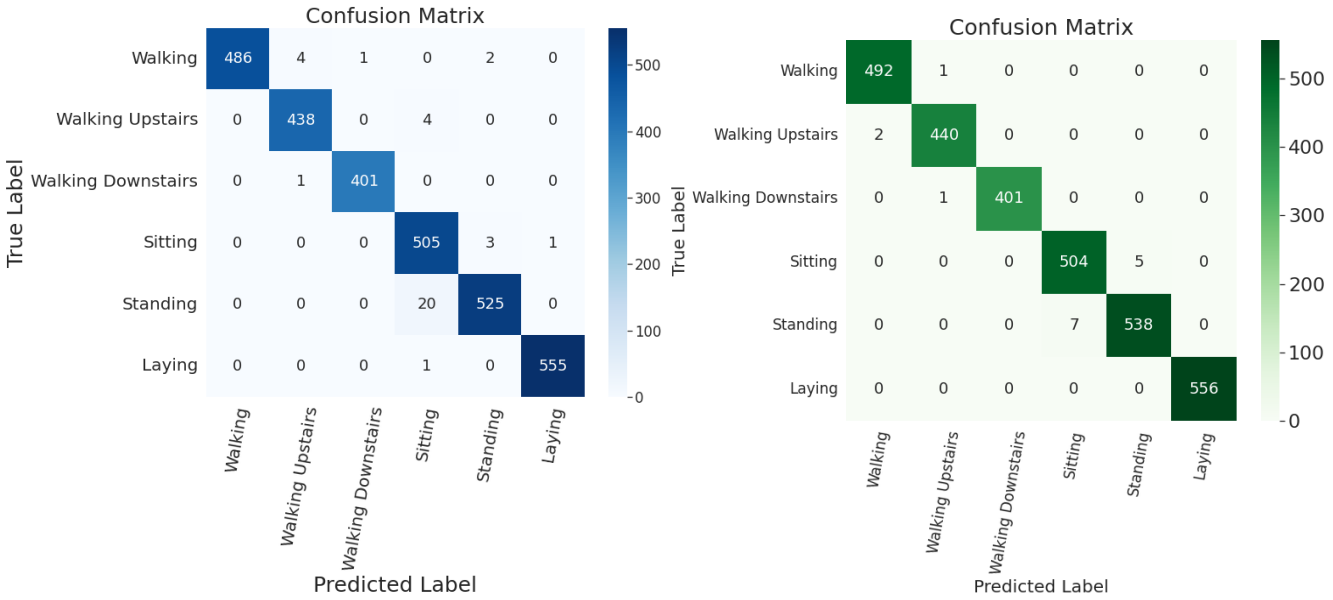


Figure 4.7: Confusion matrices for UCI-HAR on the model (a) without FS, and (b) with FS

On the UCI-HAR dataset before applying FS, out of 2947 test samples, a total of 2910 samples are correctly classified by the proposed model. After applying the FS technique, the total number of correctly classified samples increases to 2931, and overall, the accuracy is improved from 98.74% to 99. 45%. If we compare Figure 4.7(a) and Figure 4.7(b), we can see that FS technique improves the discrimination between Standing and Sitting. It also improves the recognition accuracy of the Walking activity class. Even after applying the FS there are still confusion between Sitting and Standing. The main reason could be that the two exercises are comparable from the perspective of movement sensors. Data from accelerometers and gyroscopes alone are insufficient for mining dipper discriminative information.

4.4.3 Detailed Evaluation on WISDM Dataset

When the trained model is tested on the WISDM dataset, the FS techniques improve the overall recognition accuracy from 98.34% to 99.38%. Figure 4.8 represents confusion matrices of the proposed method without and with FS. If the confusion matrices of Figure 4.8 have been compared, it is clear that the reduced optimal features map generated by the FS technique helps the classifier to recognize each activity more accurately as the classifier makes less confusion. In the case of WISDM, when the trained model has been tested with 1452 number of new instances, FS techniques increase the number of correctly classified samples from 1428 to 1443.

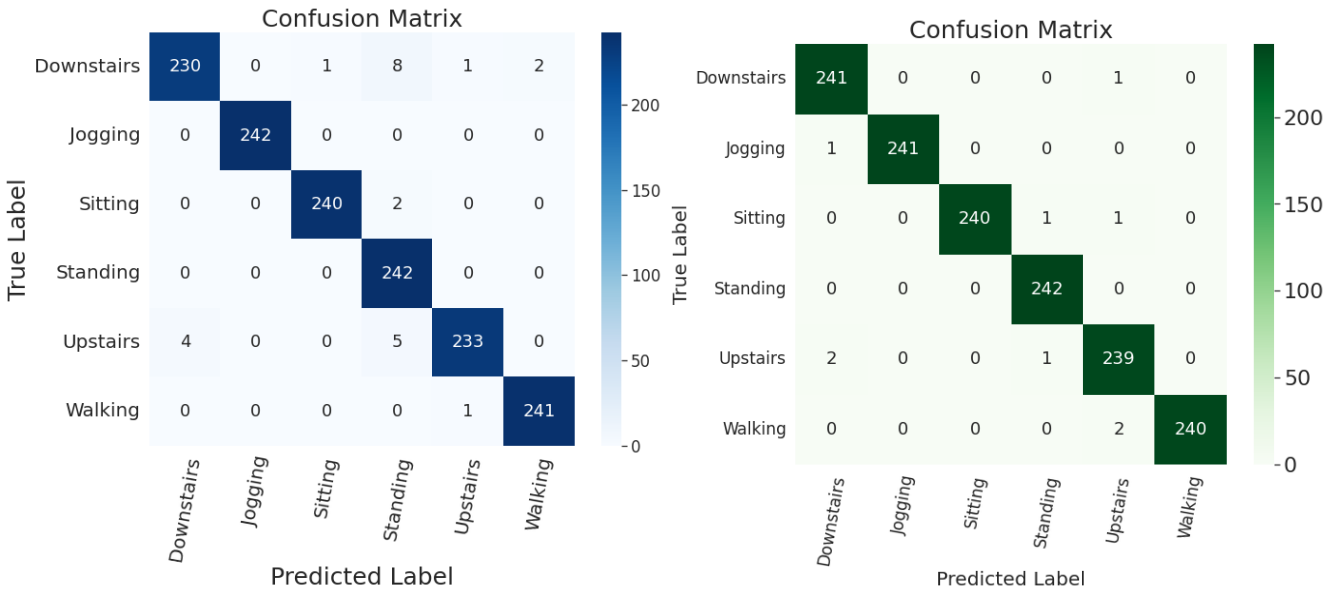


Figure 4.8: Confusion matrices for WISDM on the model (a) without FS, and (b) with FS

4.4.4 Detailed Evaluation on MHEALTH Dataset

The proposed methods are examined using a total of 1052 new samples in the context of the MHEALTH dataset. Figure 4.9 depicts the confusion matrices of the proposed method without FS and with FS. The confusion matrices present in Figure 4.9 show that though the model without FS performed well, the model gets a little confused while recognizing complex activities like Knees bending and Waist bends forward. But FS technique reduces the number of confusions and increases the overall accuracy from 99.72% to 99.90%.

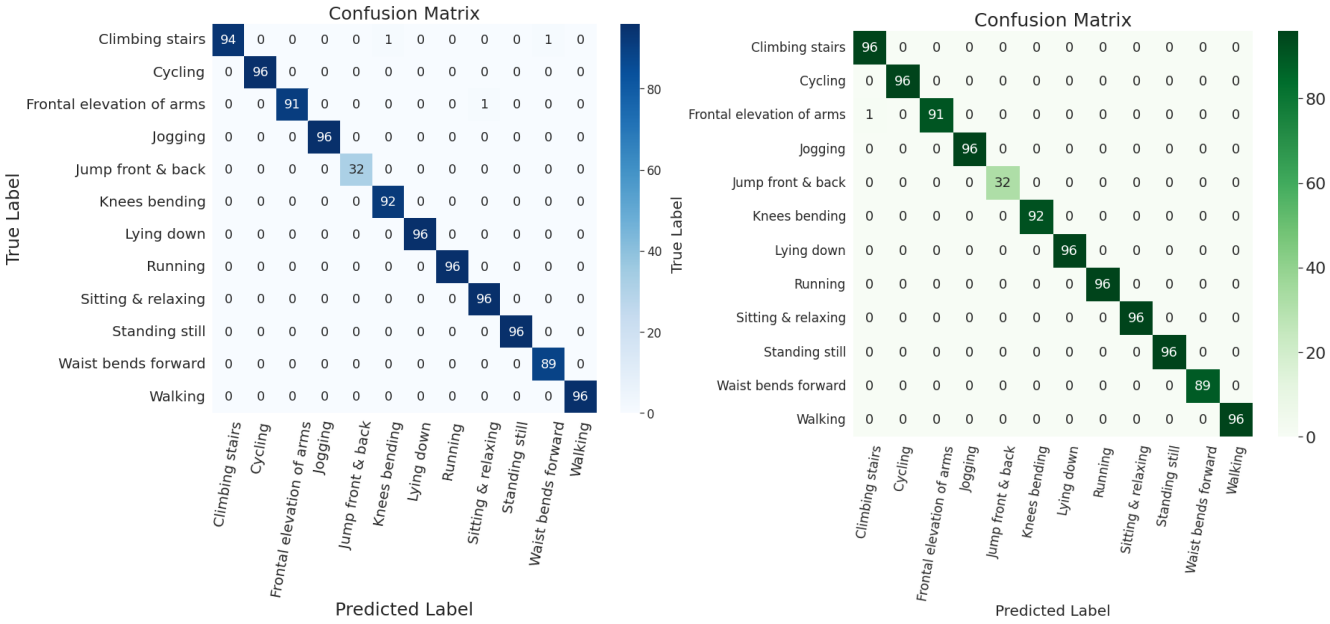


Figure 4.9: Confusion matrices for MHEALTH on the model (a) without FS, and (b) with FS

4.5 Impact of FS Hyper-parameters on Model Performance

The effectiveness of the FS algorithm is highly depends on various hyper-parameters such as population size, crossover probability, and number of iterations. As these hyper-parameter controls the quality of features selected by the FS framework, hence have direct influence on the classification model's performance. This section describes the effect of these FS hyper-parameter on the models overall accuracy.

4.5.1 Effect of Population Size

The population size is an important parameter that has a direct impact on the ability to find the best solution in the search space. Having a large population increases the likelihood of obtaining an optimal solution. In this work, different population sizes have been used, beginning with 5 and increasing to 30 with a fixed interval of 5. The population size vs accuracy graph for the three datasets is shown in Figure 4.10.

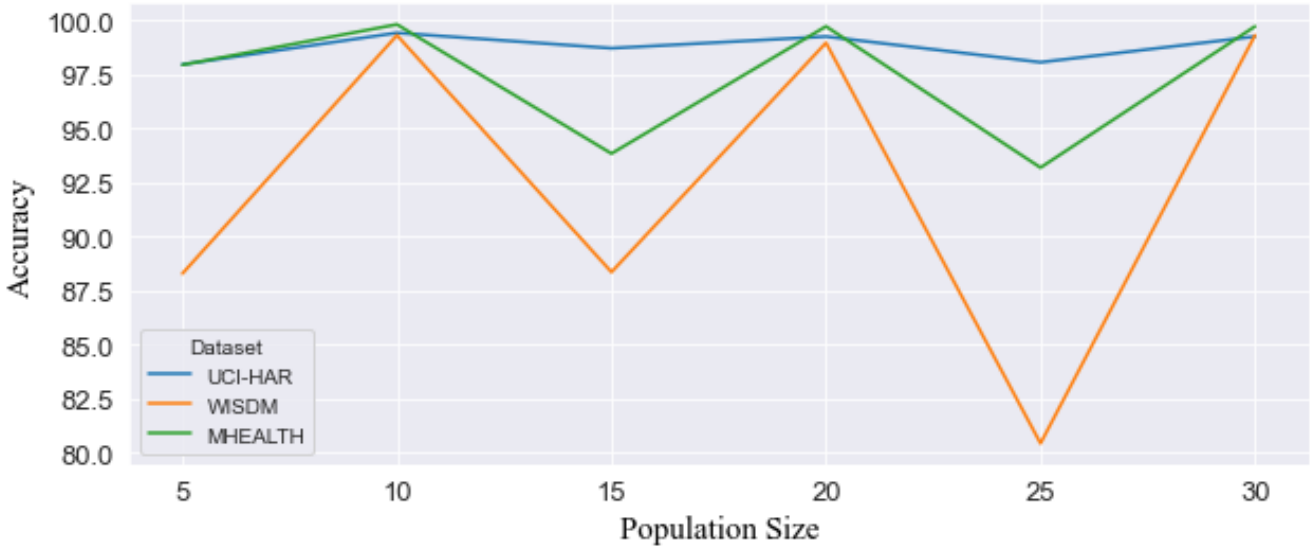


Figure 4.10: Population Size of GA vs Accuracy graph for all three HAR datasets

For all three datasets, the accuracy increases linearly and reaches the global maximum when the population size was 10. As the population size increases, the accuracy follows a zigzag pattern. For WISDM and MHEALTH, accuracy reaches the minimum when the population size is 25. For the proposed method, 10 is used as the default population size.

4.5.2 Effect of Crossover Probability

Crossover is used as a genetic operator for generating new solutions from an existing population stochastically. The crossover probability is the likelihood that a crossover will occur in specific mating. In this experiment, the crossover probability is varied as 0.1, 0.2 to 0.9, to find how accuracy changes.

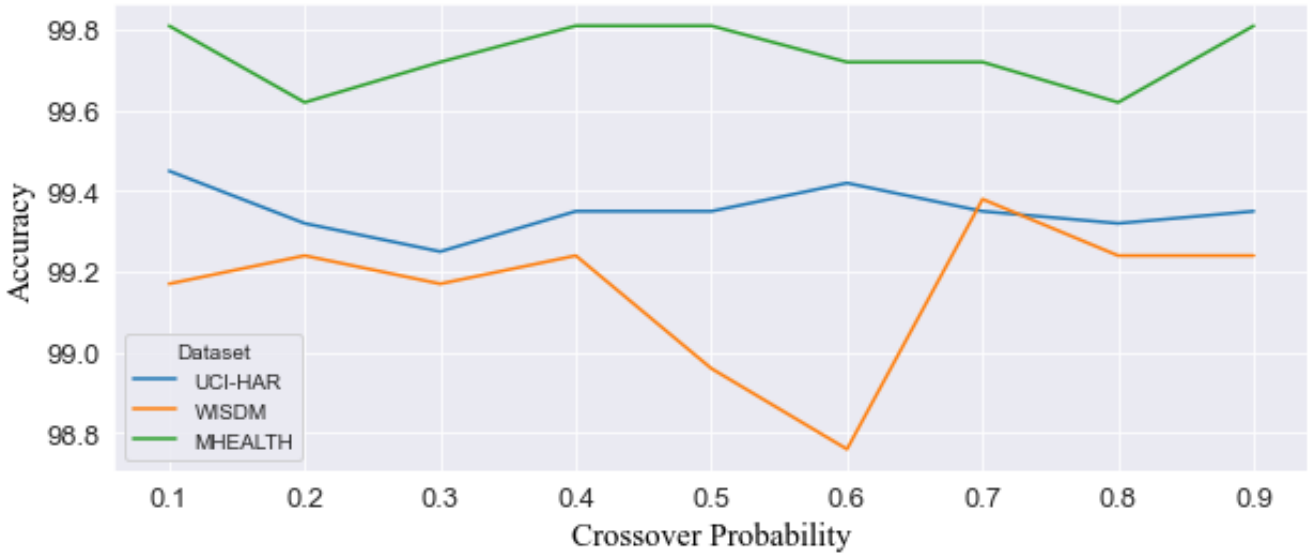


Figure 4.11: Crossover Probability of GA vs Accuracy graph for all three HAR datasets

Figure 4.11 depicts the relation of the crossover probabilities and the accuracy. As the crossover probability gets increased, the change in accuracy varies differently for different datasets. For UCI-HAR and MHEALTH datasets, initially, accuracy decreases and then starts to increase as the crossover probability increases. The accuracy reaches the minimum when the crossover probability is 0.3 for UCI-HAR and 0.2 for MHEALTH. For the WISDM dataset, the accuracy first follows a zigzag pattern followed by a sharp fall and reaches the minimum when the crossover probability is 0.6. Further increase in the crossover probability increases the accuracy.

4.5.3 Effect of Number of Iterations

Figure 4.12 depicts the change in accuracy as the number of iterations of GA increases. The accuracy of this hyper-parameter, like that of other hyper-parameters, varies depending on the dataset. As the number of iterations gets increased from 5 to 30 with a uniform interval of 5, the accuracy of the UCI-HAR dataset gradually increases and reaches a maximum when the number of iterations is 30. Whereas for the WISDM and MHEALTH datasets, the accuracy initially increases and then begins to decrease as the number of iterations exceeds 15. When the number of iterations exceeds 25, the accuracy begins to increase again. The accuracy reaches its peak when the number of iterations is set to 10 for the WISDM dataset and 15 for the MHEALTH dataset. In this experiment, 30 as the default number of iterations has been used.

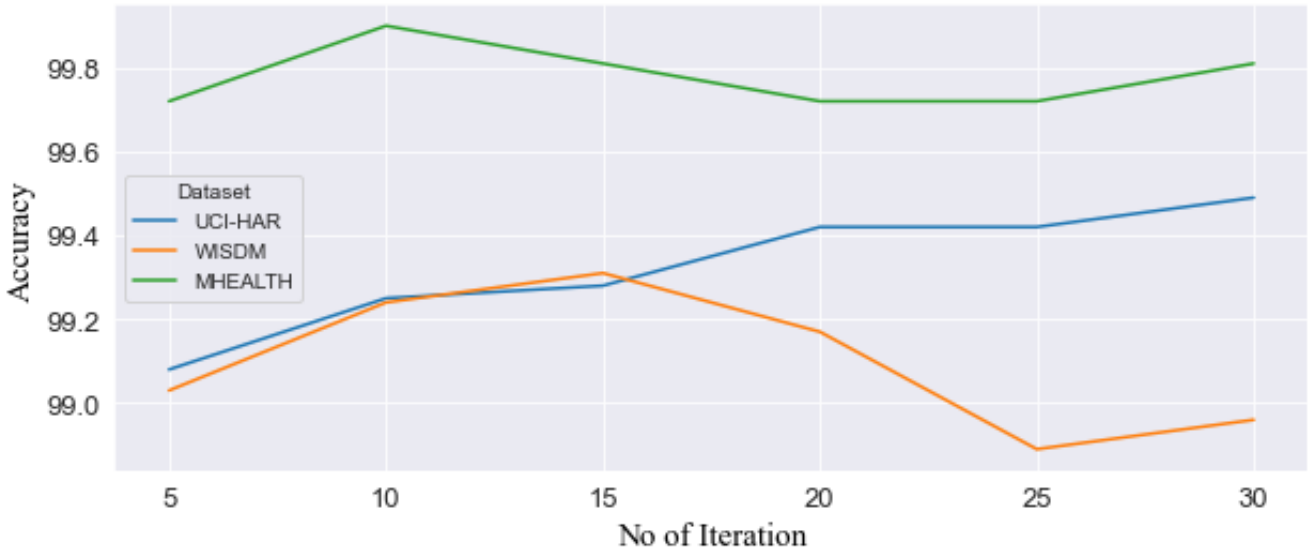


Figure 4.12: No of iteration vs accuracy graph for all three HAR datasets

4.6 Comparison with State-of-the-art Methods

This section shows the comparative analysis in term of the overall accuracy of the proposed model with other state-of-the-art HAR model. This comparison also helps to assess the efficacy and generalizability of the proposed model.

The comparative results for the UCI-HAR, WISDM, and MHEALTH datasets are shown in Tables4.5,4.6 and 4.7 respectively. The comparison is done based on the classification accuracy. The results show that the proposed model without FS has achieved higher recognition accuracy compared to other HAR models. The use of FS technique has improved recognition accuracy even more. For all three datasets, the proposed method with FS outperforms the state-of-the-art algorithms considered here for comparison.

Table 4.5: Performance comparison of the proposed model with past methods for the UCI-HAR dataset.

Model	**Accuracy (in %)**
Wang et al [69]	91.65
Nair et al. [14]	94.60
Xia et al. [34]	95.78
Ronao et al. [18]	95.75
Dua et al. [35]	96.20
Challa et al. [70]	96.37
Ignatov et al [71]	97.63
Proposed model without FS	98.74
Proposed model with FS	**99.45**

Table 4.6: Performance comparison of the proposed model with past methods for the WISDM dataset.

Model	**Accuracy (in %)**
Sena et al. [72]	89.01
Ignatov et al. [71]	93.32
Lu et al. [73]	93.50
Xia et al. [34]	95.85
Challa et al. [70]	96.05
Mukherjee et al. [74]	97.20
Dua et al. [35]	97.21
Proposed model without FS	98.34
Proposed model with FS	**99.38**

Table 4.7: Performance comparison of the proposed model with past methods for the MHEALTH dataset.

Model	**Accuracy (in %)**
Chen et al. [75]	94.05
Nguyen et al. [76]	94.72
Lu et al. [73]	96.10
Sena et al. [72]	96.27
Qin et al. [46]	98.50
Uddin et al. [77]	99.00
Abdel et al. [78]	99.68
Proposed model without FS	99.72
Proposed model with FS	**99.90**

Chapter 5

Conclusion

Despite numerous studies on human activity recognition, few solutions are practical enough to be used in mission-critical real-world applications such as health, fitness, social work, sociology, and gaming. Especially in the health domain, these solutions either require excessive number of wearable devices (multi-device approach) or demand complex classification models and large training datasets (deep classification approach). This research work attempts to investigate the very nature of daily activities in order to propose effective and efficient feature selection based approaches to address the fundamental challenges in HAR. The quality of features extracted by the CNN based feature extractor is improved by the proposed CWT based time-frequency representation of sensor signal. Furthermore, by selecting the most relevant features, the suggested FS framework reduces the dimension of feature vector obtained from the CNN model. As a result, the suggested method greatly boosts overall recognition ability of the proposed HAR model.

The proposed strategy, on the other hand, necessitates a significant amount of memory and additional time to train the model. The CWT of a single sample activity necessitates extra storage space for the produced wavelet coefficient while also necessitating a lengthy computation time. It takes a long time and a lot of resources to effectively train the data-hungry CNN features extractor.

Apart from that the results of this study's experiments reveal that the proposed strategy significantly improves recognition accuracy and outperforms the state-of-the-art.

As a next step, there is a plan to incorporate the feature selection approaches into real-world IoT-based applications, specifically health and sports, to investigate their applicability in more complex real-time scenarios. At the same time, there will be endeavours to keep the execution time of the model and computational resources to a minimum.

Bibliography

[1] J. H. Mosquera, H. Loaiza, S. E. Nope, and A. D. Restrepo, "Identifying facial gestures to emulate a mouse: navigation application on facebook.," *IEEE Latin America Transactions*, vol. 15, no. 1, pp. 121–128, 2017.

[2] K. K. Roudposhti, J. Dias, P. Peixoto, V. Metsis, and U. Nunes, "A multilevel body motion-based human activity analysis methodology," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 1, pp. 16–29, 2016.

[3] G. Ogbuabor and R. La, "Human activity recognition for healthcare using smartphones," in *Proceedings of the 2018 10th international conference on machine learning and computing*, pp. 41–46, 2018.

[4] A. B. Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Expert Systems with Applications*, vol. 91, pp. 480–491, 2018.

[5] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[6] H. Xu, Z. Huang, J. Wang, and Z. Kang, "Study on fast human activity recognition based on optimized feature selection," in *2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES)*, pp. 109–112, IEEE, 2017.

[7] K. Nurhanim, I. Elamvazuthi, L. Izhar, and T. Ganesan, "Classification of human activity based on smartphone inertial sensor using support vector machine," in *2017 ieee 3rd international symposium in robotics and manufacturing automation (roma)*, pp. 1–5, IEEE, 2017.

[8] P. Paul and T. George, "An effective approach for human activity recognition on smartphone," in *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, pp. 1–3, IEEE, 2015.

[9] S. Sani, N. Wiratunga, and S. Massie, "Learning deep features for knn-based human activity recognition.," in *In Proceedings of the ICCBR 2017 Workshops*, CEUR Workshop Proceedings, 2017.

[10] Z. Liu, S. Li, J. Hao, J. Hu, and M. Pan, "An efficient and fast model reduced kernel knn for human activity recognition," *Journal of Advanced Transportation*, vol. 2021, 2021.

[11] L. Fan, Z. Wang, and H. Wang, "Human activity recognition model based on decision tree," in *2013 International Conference on Advanced Cloud and Big Data*, pp. 64–68, IEEE, 2013.

[12] S. Brajesh and I. Ray, "Ensemble approach for sensor-based human activity recognition," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pp. 296–300, 2020.

[13] N. Hnoohom, S. Mekruksavanich, and A. Jitpattanakul, "Human activity recognition using triaxial acceleration data from smartphone and ensemble learning," in *2017 13th international conference on signal-image Technology & Internet-Based Systems (SITIS)*, pp. 408–412, IEEE, 2017.

[14] N. Nair, C. Thomas, and D. B. Jayagopi, "Human activity recognition using temporal convolutional network," in *Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction*, pp. 1–8, 2018.

[15] S. Münzner, P. Schmidt, A. Reiss, M. Hanselmann, R. Stiefelhagen, and R. Dürichen, "Cnn-based sensor fusion techniques for multimodal human activity recognition," in *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pp. 158–165, 2017.

[16] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *2017 ieee international conference on big data and smart computing (bigcomp)*, pp. 131–134, IEEE, 2017.

[17] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[18] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[19] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "Tse-cnn: A two-stage end-to-end cnn for human activity recognition," *IEEE journal of biomedical and health informatics*, vol. 24, no. 1, pp. 292–299, 2019.

[20] Q. Teng, K. Wang, L. Zhang, and J. He, "The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 7265–7274, 2020.

[21] R. Zhu, Z. Xiao, Y. Li, M. Yang, Y. Tan, L. Zhou, S. Lin, and H. Wen, "Efficient human activity recognition solving the confusing activities via deep ensemble learning," *Ieee Access*, vol. 7, pp. 75490–75499, 2019.

[22] N. Zehra, S. H. Azeem, and M. Farhan, "Human activity recognition through ensemble learning of multiple convolutional neural networks," in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–5, IEEE, 2021.

[23] P. Agarwal and M. Alam, "A lightweight deep learning model for human activity recognition on edge devices," *Procedia Computer Science*, vol. 167, pp. 2364–2373, 2020.

[24] T. Zebin, M. Sperrin, N. Peek, and A. J. Casson, "Human activity recognition from inertial sensor time-series using batch normalized deep lstm recurrent networks," in *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pp. 1–4, IEEE, 2018.

[25] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni, "A sequential deep learning application for recognising human activities in smart homes," *Neurocomputing*, vol. 396, pp. 501–513, 2020.

[26] A. Malshika Welhenge and A. Taparugssanagorn, "Human activity classification using long short-term memory network," *Signal, Image and Video Processing*, vol. 13, no. 4, pp. 651–656, 2019.

[27] S. Yu and L. Qin, "Human activity recognition with smartphone inertial sensors using bidir-lstm networks," in *2018 3rd international conference on mechanical, control and computer engineering (icmcce)*, pp. 219–224, IEEE, 2018.

[28] M. Lv, W. Xu, and T. Chen, "A hybrid deep convolutional and recurrent neural network for complex activity recognition using multimodal sensors," *Neurocomputing*, vol. 362, pp. 33–40, 2019.

[29] S. P. Singh, M. K. Sharma, A. Lay-Ekuakille, D. Gangwar, and S. Gupta, "Deep convlstm with self-attention for human activity decoding using wearable sensors," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8575–8582, 2020.

[30] J. V. Jeyakumar, E. S. Lee, Z. Xia, S. S. Sandha, N. Tausik, and M. Srivastava, "Deep convolutional bidirectional lstm based transportation mode recognition," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1606–1615, 2018.

[31] S. Perez-Gamboa, Q. Sun, and Y. Zhang, "Improved sensor based human activity recognition via hybrid convolutional and recurrent neural networks," in *2021 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, pp. 1–4, IEEE, 2021.

[32] S. Mekruksavanich and A. Jitpattanakul, "Smartwatch-based human activity recognition using hybrid lstm network," in *2020 IEEE SENSORS*, pp. 1–4, IEEE, 2020.

[33] R. Mutegeki and D. S. Han, "A cnn-lstm approach to human activity recognition," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 362–366, IEEE, 2020.

[34] K. Xia, J. Huang, and H. Wang, "Lstm-cnn architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.

[35] N. Dua, S. N. Singh, and V. B. Semwal, "Multi-input cnn-gru based human activity recognition using wearable sensors," *Computing*, vol. 103, no. 7, pp. 1461–1478, 2021.

[36] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International conference on pervasive computing*, pp. 1–17, Springer, 2004.

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

[38] V. M. Souza, D. F. Silva, and G. E. Batista, "Extracting texture features for time series classification," in *2014 22nd International Conference on Pattern Recognition*, pp. 1425–1430, IEEE, 2014.

[39] E. Garcia-Ceja, M. Z. Uddin, and J. Torresen, "Classification of recurrence plots distance matrices with a convolutional neural network for activity recognition," *Procedia computer science*, vol. 130, pp. 157–163, 2018.

[40] N. Hatami, Y. Gavet, and J. Debayle, "Classification of time-series images using deep convolutional neural networks," in *Tenth international conference on machine vision (ICMV 2017)*, vol. 10696, p. 106960Y, International Society for Optics and Photonics, 2018.

[41] Y. Zhang, Y. Hou, S. Zhou, and K. Ouyang, "Encoding time series as multi-scale signed recurrence plots for classification using fully convolutional networks," *Sensors*, vol. 20, no. 14, p. 3818, 2020.

[42] C. Ito, X. Cao, M. Shuzo, and E. Maeda, "Application of cnn for human activity recognition with fft spectrogram of acceleration and gyro sensors," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1503–1510, 2018.

[43] I. A. Lawal and S. Bano, "Deep human activity recognition with localisation of wearable sensors," *IEEE Access*, vol. 8, pp. 155060–155070, 2020.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[45] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[46] Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K. R. Choo, "Imaging and fusing time series for wearable sensor-based human activity recognition," *Information Fusion*, vol. 53, pp. 80–87, 2020.

[47] Z. Ahmad and N. Khan, "Inertial sensor data to image encoding for human action recognition," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 10978–10988, 2021.

[48] T. Hur, J. Bang, J. Lee, J.-I. Kim, S. Lee, *et al.*, "Iss2image: A novel signal-encoding technique for cnn-based human activity recognition," *Sensors*, vol. 18, no. 11, p. 3910, 2018.

[49] N. Daniel and I. Klein, "Inim: Inertial images construction with applications to activity recognition," *Sensors*, vol. 21, no. 14, p. 4787, 2021.

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[51] C. V. San Buenaventura and N. M. C. Tiglao, "Basic human activity recognition based on sensor fusion in smartphones," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 1182–1185, 2017.

[52] C. Fan and F. Gao, "Enhanced human activity recognition using wearable sensors via a hybrid feature selection method," *Sensors (Basel)*, vol. 21, p. 6434, 2021.

[53] C. Dewi and R.-C. Chen, "Human activity recognition based on evolution of features selection and random forest," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 2496–2501, 2019.

[54] N. D. Nguyen, D. T. Bui, P. H. Truong, and G.-M. Jeong, "Position-based feature selection for body sensors regarding daily living activity recognition," *Journal of Sensors*, 2018.

[55] K. Aminian, P. Robert, E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon, "Physical activity monitoring based on accelerometry: validation and comparison with video observation," *Medical & biological engineering & computing*, vol. 37, no. 3, pp. 304–308, 1999.

[56] R. W. Selles, M. A. Formanoy, J. B. Bussmann, P. J. Janssens, and H. J. Stam, "Automated estimation of initial and terminal contact timing using accelerometers; development and validation in transtibial amputees and controls," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 1, pp. 81–88, 2005.

[57] J. M. Jasiewicz, J. H. Allum, J. W. Middleton, A. Barriskill, P. Condie, B. Purcell, and R. C. T. Li, "Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals," *Gait & posture*, vol. 24, no. 4, pp. 502–509, 2006.

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

[59] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[60] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[61] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig, "The mutual information: Detecting and evaluating dependencies between variables," *Bioinformatics*, vol. 18, pp. S231–S240, 2002.

[62] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992* (D. Sleeman and P. Edwards, eds.), pp. 249–256, Morgan Kaufmann, 1992.

[63] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[64] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, pp. 437–442, 2013.

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

[66] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *International workshop on ambient assisted living*, pp. 91–98, Springer, 2014.

[67] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas, "Design, implementation and validation of a novel open framework for agile development of mobile health applications," *Biomedical engineering online*, vol. 14, no. 2, pp. 1–20, 2015.

[68] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, "Pywavelets: A python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019.

[69] L. Wang and R. Liu, "Human activity recognition based on wearable sensor using hierarchical deep lstm networks," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 837–856, 2020.

[70] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch cnn-bilstm model for human activity recognition using wearable sensor data," *The Visual Computer*, pp. 1–15, 2021.

[71] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[72] J. Sena, J. Barreto, C. Caetano, G. Cramer, and W. R. Schwartz, "Human activity recognition based on smartphone and wearable sensors using multiscale dcnn ensemble," *Neurocomputing*, vol. 444, pp. 226–243, 2021.

[73] W. Lu, F. Fan, J. Chu, P. Jing, and S. Yuting, "Wearable computing for internet of things: A discriminant approach for human activity recognition," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2749–2759, 2018.

[74] D. Mukherjee, R. Mondal, P. K. Singh, R. Sarkar, and D. Bhattacharjee, "Ensemconvnet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications," *Multimedia Tools and Applications*, vol. 79, no. 41, pp. 31663–31690, 2020.

[75] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 5, pp. 1747–1756, 2019.

[76] H. Nguyen, K. P. Tran, X. Zeng, L. Koehl, and G. Tartare, "Wearable sensor data based human activity recognition using machine learning: a new approach," *arXiv preprint arXiv:1905.03809*, 2019.

[77] M. Z. Uddin, M. M. Hassan, A. Alsanad, and C. Savaglio, "A body sensor data fusion and deep recurrent neural network-based behavior recognition approach for robust healthcare," *Information Fusion*, vol. 55, pp. 105–115, 2020.

[78] M. Abdel-Basset, H. Hawash, V. Chang, R. K. Chakrabortty, and M. Ryan, "Deep learning for heterogeneous human activity recognition in complex iot applications," *IEEE Internet of Things Journal*, 2020.