



Addis Ababa University
Addis Ababa Institute of Technology
Center of Biomedical Engineering
Human Activity Recognition using Machine Learning

By
Rediet Desalegn

**In partial fulfillment of the requirements for the degree of master of
science in Biomedical Engineering**

Advisor: Melkamu Hunegnaw (PhD)

Addis Ababa, Ethiopia

August 2023

Declaration

By submitting this MSc thesis, I declare that the entirety of the work presented in this document is my original work. Where information has been derived from other sources, I confirm that this has been indicated in the thesis and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Name: Rediet Desalegn

Signature:

Date:

This MSc thesis has been submitted for examination with my approval as an advisor.

Melkamu Hunegnaw (PhD)

Certificate of Examination Addis Ababa University School of Graduate Studies

This is to certify that the thesis prepared by Rediet Desalegn entitled: “Human activity recognition using machine learning” submitted in partial fulfillment of the requirements for the Degree of Master of Science in Biomedical Engineering (Bioinstrumentation and Imaging) complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Approved by: Board of Examiners

Chairman: _____ Signature _____ Date _____

Internal Examiner: _____ Signature _____ Date _____

External Examiner: _____ Signature _____ Date _____

Advisor: Melkamu Hunegnaw (PhD) Signature _____ Date _____

Abstract

Human Activity Recognition using Machine Learning

Human activity recognition is a broad field of study concerned with identifying the specific movement or action of a person based on sensor data. It can improve people's well-being by automatically assessing and summarizing their daily activities. In this thesis an automatic detection of nine different human activities from body sensor data was developed. The human activity data was collected from 30 participants between the age of 12 and 55. From those 17 participants were male and 13 participants were female. Participants completed nine activities (sitting, standing, walking, running, upstairs, downstairs, sit-up, jumping and cycling) while wearing an I-phone eight plus in the pocket. Records from 18 individuals was used for training, 6 participants data was used for validation and the remaining 6 records were used for testing. The human activity data is captured via phone's integrated acceleration and gyroscope sensors. Hence a total of nine-dimensional data (triaxial accelerometer data (linear acceleration and body acceleration) and triaxial gyroscope data) was acquired using these two sensors. In all cases the sample rate used to capture the data was 50Hz which is in phone application. In this study both classical machine learning and deep learning methods were used. In classical techniques four machine learning classifiers (SVM, KNN, logistic regression and decision tree) have been used to categorize the data. For this to work 18 set of features were carefully computed. Three deep learning models (CNN, LSTM and hybrid CNN-LSTM) have been also trained in the accurate detection of human activity classification. The activity recognition of these algorithms has been objectively measured by using accuracy, f1 score, precision, recall and confusion matrix. Under this study accuracy of SVM 89%, KNN 81%, Logistic Regression 87%, Decision Tree 79%, CNN 92%, LSTM 91% and CNN-LSTM 94% are computed. The highest performance accuracy of 94% was achieved when classifying using hybrid CNN-LSTM model. This indicates that with a simplified approach, multiple human activities can be reasonably detected. By analyzing these activities, one can promote a healthy life style for the subjects.

Key words- human activity recognition, classical machine learning, deep learning

Acknowledgements

I want to thank God first and foremost for giving me the strength and blessing to finish this thesis. Next, I would like to sincerely thank Dr. Melkamu Hunegnaw for his guidance and mentorship during the entire research and thesis-writing process. I learn a lesson that I will remember throughout my life. I also want to thank all of my friends for their inspiration and support. I would like to express my sincere appreciation to OpenAI for the assistance in spelling and grammar checking whereby enhancing the quality and depth of this thesis. Last but not least, I want to express my sincere gratitude to my parents for their unending support and never-ending inspiration throughout the journey. This accomplishment would not have been possible without them. Sincere thanks to all.

Table of contents

Declaration.....	i
Certificate of Examination Addis Ababa University School of Graduate Studies	ii
Abstract	iii
Acknowledgements.....	iv
Table of contents.....	v
List of figures.....	vii
List of tables.....	viii
List of Abbreviations	ix
Definition	x
CHAPTER 1	1
Introduction.....	1
1.1 Background	1
1.2 Statement of the Problem	3
1.3 Objectives.....	4
1.3.1 General Objective	4
1.3.2 Specific Objectives	4
1.4 Significance of the Thesis	4
1.5 Scope	5
1.6 Organization of the Thesis	5
CHAPTER 2	6
Literature Review.....	6
2.1 Classical machine learning.....	7
2.2 Deep Learning.....	10
CHAPTER 3	14
Research Methodology	14

3.1 Selection of Activities.....	14
3.2 Data Collection.....	15
3.3 Public Dataset.....	15
3.4 Collected Data	16
3.5 Preprocessing	17
3.6 Feature Extraction	18
3.7 Classification.....	21
3.7.1 Classical Machine Learning Algorithms	21
3.7.2 Deep Learning Algorithms	21
CHAPTER 4	26
Results and Discussion	26
4.1 Human activity recognition using classical machine learning classifiers: open-source data 26	
4.2 Human activity recognition using Deep learning Algorithms: open-source data	28
4.3 Human activity recognition using classical machine learning classifiers: collected dataset 30	
4.4 Human activity recognition using Deep Learning classifiers: collected dataset.....	32
CHAPTER 5	35
Conclusion and Future works	35
5.1 Conclusion.....	35
5.2 Limitation of the study and Future Work.....	37
References.....	38
Appendix A: Code for classical machine learning classifiers	44
Appendix B: Code for deep learning models.....	49
Appendix C: For Local data volunteers' confidentiality sign list.....	53

List of figures

Figure 1 Worldwide obesity prevalence [7].....	1
Figure 2 Detailed Block Diagram	14
Figure 3 Logical flow of the data collection process showcasing a volunteer participant with corresponding accelerometer and gyroscope data	16
Figure 4 Visualization of IQR.....	20
Figure 5 Architecture of CNN model	22
Figure 6 Architecture of LSTM model	23
Figure 7 Architecture of CNN-LSTM model	23
Figure 8 Hybrid CNN-LSTM confusion matrix for open source data.....	30
Figure 9 Hybrid CNN-LSTM confusion matrix for local data	34

List of tables

Table 1 Number of published articles from 2013 to 2023 in different databases using the keyword "Human activity recognition"	6
Table 2 Summary of selected literatures relevant to human activity recognition.....	12
Table 3 Performance evaluation of four classical machine learning algorithms on open-source data [Note: pre. = precision, Rec= recall, F1= F1-score]	27
Table 4 SVM confusion matrix for open-source data.....	28
Table 5 Performance evaluation of three deep learning algorithms on open-source data [Note: pre. = precision, Rec= recall, F1= F1-score]	28
Table 6 Performance evaluation of four classical machine learning algorithms on our collected local data set [Note: pre. = precision, Rec= recall, F1= F1-score]	31
Table 7 Confusion matrix of SVM for local data	32
Table 8 Performance evaluation of three deep learning algorithms on collected local data [Note: pre. = precision, Rec= recall, F1= F1-score]	33

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
BPNN	Back Projection Neural Network
CDC	Center for Disease Control and prevention
CNN	Convolutional Neural Network
DT	Decision Tree
HAR	Human Activity Recognition
HDL	High Density Lipoproteins
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LDL	Low Density Lipoproteins
LR	Logistic Regression
LSTM	Long Short-Term Memory
LTM	Long Term Memory
SVM	Support Vector Machine
WHO	World Health Organization

Definition

Algorithm-is a procedure used for solving problem or performing a computation.

Accelerometer- is a tool that measures proper acceleration.

Accuracy- is the degree of closeness between a measurement and its true value.

Auto-regression-is a time series model that uses observations from previous time steps as input to regression equation to predict the value at the next time step.

Convolution- is defined as the integral of the product of the two functions after one is reversed and shifted.

Correlation- a statistical measure that expresses the extent to which two variables are linearly related

F1-score- is a measure of a test's accuracy.

Gyroscope- is a device used for measuring or maintaining orientation and angular velocity.

Hierarchy-is an organizational structure in which items are ranked according to levels of importance.

Interquartile range-is a measure of statistical dispersion, which is the spread of the data.

Kurtosis -is a measure of the combined weight of distributions tails relative to the center of the distribution.

Mean value- is the average of the numbers.

Median absolute deviation-is a robust measure of the variability of a univariate sample of quantitative data

Precision- indicates how close the measurements are to each other.

Predictive modeling- is a mathematical process used to predict future events.

Signal entropy-is a measure of its spectral power distribution.

Signal magnitude area-is a statistical measure of the magnitude of varying quantity.

Skewness- refers to a distortion or a symmetry that deviates from the normal distribution.

Standard deviation- is a measure of how dispersed the data is in relation to the mean.

Sensitivity- (true positive rate) refers to the probability of a positive test, conditioned on truly being positive.

Specificity- (true negative rate) refers to the probability of a negative test, conditioned on truly being negative.

ROC curve-is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varies

CHAPTER 1

Introduction

1.1 Background

Human activity recognition is concerned with identifying the specific movement or action of a person based on sensor data [1]. It has gained much attention in the current decade due to numerous application [2]. It provides an estimation of the energy consumption of a person during everyday life this help to balance the number of calories consumed from food and beverage with the number of calories the body used for activity and this used to prevent overweight and obesity. Worldwide over 2.8 million people die each year as a result of being overweight or obese in 2021 [3]. When a person's body mass index is over 25 it is considered as overweight and over 30 is obese. Worldwide 1.9 billion adults are overweight and 650 million obese [4]. Once considered the problem of the developed nation, the problem is slowly creeping into developing countries too. For instance, in Ethiopia, according to a recent Ethiopian study, the prevalence of central obesity was estimated to be 37.31% at the national level [5]. According to world cancer research international study by year 2025 170 million adults worldwide may have a body mass index of over 35kg/m² [6]. The increasing rate of overweight and obesity epidemic from year of 2010 to 2025 is indicated in Figure 1.

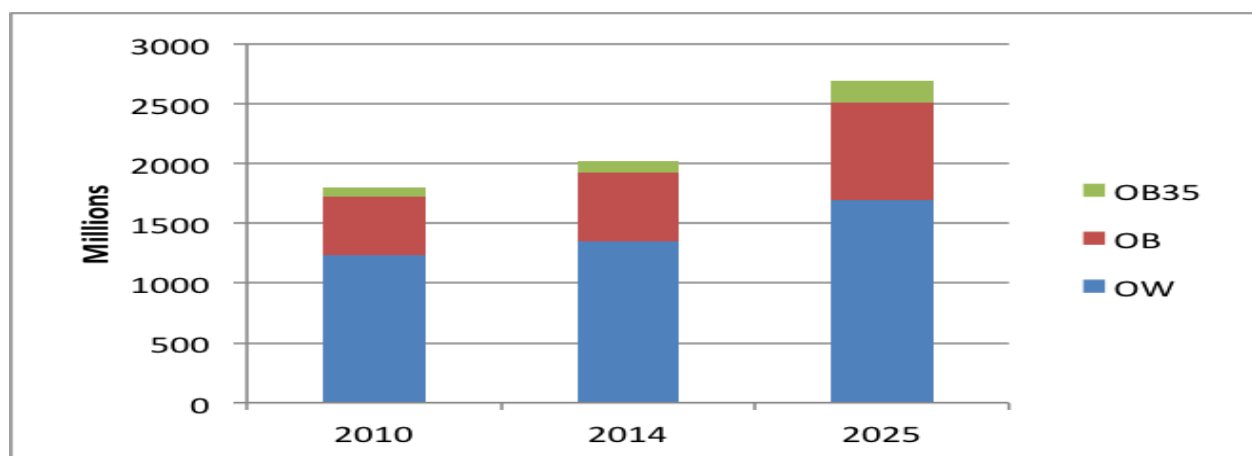


Figure 1 Worldwide obesity prevalence [7]

According to CDC's study people, who are overweight or obese compared to those with healthy weight, are at increased risk for many serious diseases and health conditions [8]. These include causes for death (mortality), high blood pressure, high LDL (low density lipoproteins) cholesterol, low HDL (high density lipoproteins) cholesterol or high levels of triglycerides, type 2 diabetes, coronary heart disease, stroke, gallbladder disease, osteoarthritis (a breakdown of cartilage and bone within joint), sleep apnea and breathing problems, many types of cancer, mental illness such as clinical depression, anxiety and other mental disorder, body pain and difficulty with physical functioning [8][9].

Researchers have linked rising obesity rates to globalization processes, which they believe contributes to obesity by flooding low-income countries markets with cheap but obesogenic foods [10]. Sedentary behavior and physical inactivity are also significant risk factors for obesity [11]. As the management expert Peter Drucker once famously said, "if you can't measure it, you can't manage it.", properly measuring daily human activity will give a valuable information about the wellbeing of the person and hence giving a constant reminder for a potential life style and behavior change that could reduce the risk of developing chronic illnesses.

Activity recognition is closely associated with health, as it provides information about physical activity performed by a person. Maintaining good health requires regular physical activity, as it helps to prevent and manage a wide range of chronic diseases. Therefore, by accurately recognizing and tracking physical activities individuals can gain insights into the patterns of physical activity, set goals for increasing or maintaining physical activity levels, and monitor progress towards achieving these goals. Human activity detection can help promote an active society by encouraging people to engage in physical activity and adopt healthier lifestyles. It also motivates people to increase their physical activity levels by setting goals, tracking progress, and providing feedback.

Overall, human activity detection is a valuable tool for promoting health and wellbeing, increasing productivity, and improving quality of life. By accurately detecting and tracking physical activity, we can better understand human behavior and make informed decisions to improve our lives.

The human activity recognition in this thesis proposes a process of automatically identifying and classifying different physical activities performed by a person, such as sitting, standing, walking,

running, upstairs, downstairs, jumping, sit up and cycling. It involves using sensors from a standard mobile phone, such as accelerometers and gyroscopes to collect data about the movements.

1.2 Statement of the Problem

A body mass index over 25 is considered as overweight and over 30 is obese. Sedentary life style and diets is the major challenge which can likely have other diseases including high blood pressure type 2 diabetes, coronary heart disease, associated with pooled mental health overcomes, reduced quality of life and lead to death [12]. To overcome these challenges, balancing number of calories consumed from food and beverage with the number of calories the body uses for activity must be maintained. Accurate quantification of the human activity provides an estimation of the energy consumption, so one can improve healthy lives.

To recognize activity there are different sensors available. These sensors have their own pros and cons. From these sensors human activity recognition based on wearable sensors have a problem of many sensors are attached to a subject might not be comfortable, as well as it is expensive in terms of energy consumption and device configuration. Motion capture from video cameras to monitor and recognize human activity is also another option; however, this approach is restricted by the field of view of the camera and it diminishes privacy of the subject. Radar-based human activity recognition provides a non-contact method but such methods are prone to noise and are expensive to implement. Nowadays, many people have mobile phones with integrated motion sensors. Harnessing the capability of this sensor for the activity recognition the user is not required to wear an additional sensor. This thesis proposes the use of accelerometer and gyroscope sensors integrated in the mobile phone for human activity recognition. We have also considered the standard placement of the mobile (inside a pocket) to detect the activities. This makes our proposed solution to be easily available and comfortable to use.

1.3 Objectives

1.3.1 General Objective

The general objective of this thesis is to develop automatic human activity detection method that can detect nine different activities from mobile phone integrated sensor data.

1.3.2 Specific Objectives

The specific objectives of this thesis are:

- To determine the logical sensor placement for efficient human activity recognition.
- To compile human activity dataset from different public datasets.
- To identify most common activities that are common to Ethiopian urban adolescents
- To collect local data to develop and evaluate the performance of the proposed algorithm.
- To Propose different features for detecting human activities.
- To compare classical and deep learning algorithms for human activity recognition.
- To recommend the best algorithm for human activity detection.

1.4 Significance of the Thesis

According to the latest physical activity guideline, adults need at least 150 minutes of moderate-intensity physical activity and 2 days of muscle strengthening activity per week [13]. By using human activity behavior and giving labels to each action, it is also possible to estimate how much energy is consumed during daily activities, keeping a balance between the calories consumed from food and beverage and the calories used by the body during activity. This method is used to prevent overweight and obesity as well as the likelihood of developing other diseases like high blood pressure, type 2 diabetes, coronary heart disease, and diseases linked to a group's collective mental health challenges.

1.5 Scope

The proposed research algorithm was developed and tested from data collected from 30 participants 17 of them are male and 13 of them are female by using accelerometer sensor which has linear and body acceleration which measure the rate of change of velocity along three dimension and gyroscope sensor which measure angular velocity of orientation which embedded in smart phone. The proposed method only recognizes 9 most common daily activities in Ethiopian context and All works are done by using python programing language.

1.6 Organization of the Thesis

The rest of the thesis has been organized in to four chapters. Chapter Two presents the literature review. In chapter three Research methodology and materials used are explained in detail. Results and discussions are included in chapter four, and finally Chapter Five concludes the thesis by giving some useful recommendations and possible future directions.

CHAPTER 2

Literature Review

This chapter provides an overview of the scientific literature relevant to human activity recognition. Human activity recognition has been a subject of study for several years, and numerous research studies have been conducted in this field. In this section, recent literature from the period of 2013-2022 is reviewed, focusing on studies that have strategically selected sensor types and have achieved high accuracy in their results. The selected literature encompasses studies that have utilized machine learning and deep learning approaches for human activity recognition. These methods have received a lot of attention recently due to their ability to effectively capture and analyze complex patterns in activity data. By reviewing the literature, we aim to gain insights into the advancements made in human activity recognition techniques and the corresponding sensor types used in these studies. Just a simple search using a key word “human activity recognition” for the year 2013 to 2023 in five known databases has resulted in 7258 articles as shown in Table 1.

Table 1 Number of published articles from 2013 to 2023 in different databases using the keyword "Human activity recognition"

DATABASE	NUMBER OF ARTICLES
Web of science	2650
SCIENCE DIRECT	409
PubMed	1113
SCOPUS	1675
IEEE Xplore	1411
Total	7258

The selection of the literature was guided by the availability of easily accessible sensor types and the accuracy of the results reported in each study. This ensures that the reviewed literature represents a comprehensive and up-to-date understanding of the state-of-the-art approaches in

human activity recognition. By examining the recent literature, we can identify trends, methodologies, and challenges in the field of human activity recognition. This knowledge will help in informing the development of more accurate and reliable activity recognition systems, and guide future research in this area. The literatures have been grouped in to those which use classical machine learning techniques and those which utilize deep learning techniques and are presented as follows.

Human activity recognition based on wearable sensors have a problem of many sensors are attached to a subject and can't move comfortably because of many wire connections, as well as it is expensive in terms of energy consumption and device configuration. Instead of focusing on wearable sensor based, numerous studies incorporated video sensor technologies like RGB cameras to monitor and recognize human activity. This may be a problem of difficulty in recognition due to low light environment or darkness [14][15]. To avoid the problem of difficulty in recognition due to low light environment or darkness some researcher used Kinect camera. This camera also has a problem of sensitive to sunlight and not suitable for outdoor application [16]. To overcome these problems easily available smart phone embedded accelerometer and gyroscope sensors are used for this thesis.

2.1 Classical machine learning

Classical machine learning algorithms have been widely used in Activity Recognition due to their ability to learn from data and make predictions on new instances. In this literature review, the different classical machine learning algorithms used in Activity Recognition, their advantages and limitations, and the challenges faced in the process are discussed.

Muralidharan K. et al. [17] illustrate by using accelerometer and gyroscope sensors which classify 6 human activities sitting, standing, climbing up, down the stairs case, walking and laying down by using classical machine learning classifiers logistic regression, linear and kernel SVM, decision tree and random forest achieved accuracy results of kernel SVM 96.57%, linear SVM 96.6%, logistic regression 95.4%, random forest 91% and decision tree 85.5%.

Shakya et al [18] another research proposed human activity recognition using accelerometer sensor placed data set 1 at waist and data set 2 at five different position (right jeans pocket, left jeans pocket, upper arm, right wrist and on the belt using a belt clipper) to study in different nature of data set by using three classical machine learning classifiers and achieved results accuracy of KNN

with data set1 90.1% and data set2 97.6% with random forest data set1 89.72% and data set2 95.7% with decision tree data set1 87.9% and data set2 90.5%

Youssef Hbali et al. [19] proposed a method for human activity recognition using a skeleton-based technique that captures spatial and temporal features. The method involved utilizing the Microsoft Kinect to extract joint data and computing the Minkowski and cosine distances between the joints. By combining both channels and representing each frame of a human activity sequence with 3D skeleton coordinates, the approach aimed to accurately characterize human activities.

The proposed approach was trained and validated using the publicly available MSR daily activity 3D and MSR action 3D datasets. These datasets incorporated depth data acquired from a low-cost device to represent human activity. The approach was successfully implemented and utilized to develop a real-time monitoring system for elderly individuals, enabling daily activity tracking.

However, there were limitations to this research. The presence of other individuals in the scene introduced noise to the skeleton data, impacting the quality of the captured data. Furthermore, the interaction of subjects with objects complicated the recognition of their activities. Additionally, different camera viewpoints could affect the estimation of the human body, particularly when the subject was occluded or positioned far from the camera.

In their study, Sandeep Kumar Polu et al. [20] investigated the recognition of five human activities, namely laying, sitting, standing, upstairs, and downstairs, using accelerometer data and two machine learning classifiers, namely Random Forest and modified Random Forest. The researchers implemented an online activity recognition framework on the Android platform and evaluated the performance of the classifiers. The results indicated that the modified Random Forest classifier achieved 93% accuracy in recognizing the human activities.

Sumaira Ghazal et al. [21] employed a combination of 2D skeletal data and supervised machine learning algorithms for human activity recognition. They specifically focused on using 2D skeletal data instead of 3D skeletal data, utilizing the open pose library to extract appearance and motion characteristics from the 2D landmarks of human skeletal joints. The researchers evaluated the performance of five well-known classifiers, namely BPNN, KNNS, SVM, LDA, and Naïve Bayes, for activity recognition. The study aimed to recognize four activity classes: sit, stand, walk, and fall. Among the classifiers, KNN achieved the best results in terms of activity recognition accuracy.

Guangming zhu et al. [22] Each human action is modeled by a sequence of key poses and atomic motion and performs online model matching process and online action recognition process on feature segments obtained by online segmentation method and they used variable-length maximal entropy Markov model based on likelihood probabilities is utilized for continuous human action recognition. This method to ensure the effectiveness and efficiency. The limitation of this research is algorithm still extracts features of the whole human body, and trains the models based on whole body features. Maybe it will be more reasonable to extract features for each limb separately and train hierarchical models based on human limb structures. besides, skeletal data has limitations for universal and practical human action recognition, thus it will be more reasonable to recognize human actions by integrated utilization of RGB-D images and skeletal data, and human-object interaction features should also be taken in to consideration.

An activity recognition system based on smartphones was proposed by Zhang et al. [23]. The system used a 3D smartphone attached to the waist with an accelerometer as the only sensor to collect time series signals. A total of 31 features were generated in both time and frequency domains, and 1393 samples were obtained. The study focused on discriminating between walking upstairs and downstairs, which is a challenging task. Four different machine learning classifiers were employed, including the k-nearest neighbor algorithm, support vector machine, and artificial neural network. The experiment result showed that the support vector machine with feature selection using the sequential forward selection achieved an accuracy of 84.4%. However, the study also found that discriminating between walking upstairs and downstairs is still a difficult task, and further research is needed to improve the accuracy of the system.

Table 2 summarizes additional literature reviewed on human activity recognition. These studies demonstrate that the use of classical machine learning algorithms is a well-established approach for solving this problem. Various sensors such as smartphones and smartwatches are utilized to collect data, and feature extraction techniques including time-domain, frequency-domain, and wavelet-based features are commonly employed. SVMs, decision trees, KNN, and random forests are the most commonly used classifiers. In this thesis, we investigated KNN, SVM, Decision Tree, and Logistic Regression classifiers using 18 features extracted from our in-house dataset collected using a smartphone placed in a pocket.

2.2 Deep Learning

Although machine learning algorithms that require hand-crafted feature extraction are explainable, they often have limited generalization abilities and may experience issues with parameter non-convergence and network instability. As a result, researchers and domain experts are motivated to explore human activity recognition using deep learning approaches.

Sarbagya Ratna Shakya et al. [24] conducted a study on human activity recognition using an accelerometer sensor attached to a mobile device worn around the waist region. They explored various machine learning classifiers, including K-nearest neighbor (KNN), Random Forest, and Decision tree, as well as deep learning models such as convolutional neural network (CNN) and recurrent neural network (RNN). The results of their experiments indicated that the KNN classifier achieved the highest accuracy of 91%, followed by RNN with 81.7% accuracy. Notably, the CNN model outperformed all other methods, achieving the highest accuracy rate of 92.2% in human activity recognition.

Julieta martinez et al. [25] investigated deep learning approaches for human activity recognition by conducting experiments on four large-scale datasets: NTU-RGB+D, kinetics, Human 3.6M, and CMU Mocap. Their focus was on utilizing deep recurrent neural networks (RNNs) to model human motion, aiming to learn time-dependent representations for tasks like short-term motion prediction and long-term human motion synthesis. In their study, the authors employed a state-of-the-art approach known as the symbiotic graph neural network (sym-GNN). This approach integrates action recognition and motion prediction by using graph-based operations to capture action patterns. Learnable joint-scale and part-scale graphs were utilized as graph convolution operators to extract spatial information. Comparing the results with the enhanced recurrent neural network approach, the sym-GNN model consistently demonstrated improvements. These findings highlight the effectiveness of using deep recurrent neural networks and symbiotic graph neural networks for human activity recognition.

Parisa Fard Moshiri et al. [26] utilized a Raspberry Pi 4 to collect channel state information (CSI) data, which is a characteristic of Wi-Fi signals. The study focused on seven different human daily activities, namely bend, fall, lie down, sit down, stand up, and walk. The collected CSI data was converted into images, which were then used as inputs for a 2D convolutional neural network classifier. To classify the activities accurately, the authors employed various deep learning models,

including 1D-CNN, LSTM, and bidirectional LSTM. These models achieved an impressive accuracy rate of 95% in recognizing the different human activities based on the CSI data.

Seungeun Chung et al. [27] conducted a study on body sensor positioning and data acquisition for human activity recognition systems. They established a test bed comprising eight inertial measurement unit (IMU) sensors worn on the body and an Android mobile device for data collection. The experiment involved sampling data at a low rate of 10Hz from four sensors placed on both wrists, the right ankle, and the waist. The results of the experiment indicated that only two sensors, specifically those attached to the right wrist and right ankle, exhibited reasonable performance. The authors evaluated the recognition accuracy of each sensor's data using an LSTM base-learner model. Among the IMU sensors, the accelerometer demonstrated the highest performance with an accuracy of 92.2%, followed by the magnetometer with 85.2% accuracy, and the gyroscope with 78.3% accuracy.

Imran Ullah Khan et al. [28] proposed a hybrid model for activity recognition by combining convolutional neural network (CNN) and long short-term memory (LSTM). The CNN is employed for extracting spatial features, while the LSTM network is utilized to capture temporal information. The dataset used in their study was collected from twenty participants using a Kinect sensor. The research findings showed that the hybrid CNN-LSTM model achieved the best performance in activity recognition. However, it is worth noting that the model utilized a Kinect sensor, which is sensitive to sunlight and not suitable for outdoor applications.

The literature reviewed in this thesis highlights the potential of deep learning models in accurately recognizing human activities using various sensors and feature extraction techniques. These studies have demonstrated that deep learning models outperform traditional machine learning algorithms in terms of accuracy and robustness. However, it is important to consider the specific characteristics of the dataset and the complexity of the activities being recognized, as these factors can influence the performance of the models.

One of the key advantages of deep learning models is their ability to handle time-series data, which is crucial for human activity recognition as it involves capturing both spatial and temporal information. Deep learning models, such as the hybrid CNN-LSTM model, have shown promising results in capturing the temporal dynamics of human activities and achieving high accuracy. Summary of additional literatures reviewed can be seen in Table 2.

Table 2 Summary of selected literatures relevant to human activity recognition

No	Activities	Methods	results	authors	year
R1 [29]	Sit, stand, lie & walk	SVM	100% accuracy	Uslu & Bayder	2013
R2 [30]	Movement, Gestures, and Object handling	Random forest	73% accuracy	Roitberg et al.	2014
R3 [31]	running, walking, cycling, resting and car riding	KNN	98.12%	Rahman et al.	2015
R4 [32]	8 activities	CNN	93.8%	Chen & Xue	2015
R5 [33]	12 activities	CNN	81.8	Mo et al	2016
R6 [34]	Walking, Running, Upstairs, Downstairs, Standing, Sitting and Cycling	SVM	96%	Tian & Chen	2016
R7 [35]	Bed to Toilet, Eating, Meal Preparation, and Relax	DCNN	99.36%	Gochoo et al	2017
R8 [36]	standing, sitting, laying, walking, walking upstairs, and walking downstairs	ensemble learning based approaches	91.16%	Hnoohom et al	2017
R9 [37]	walking, jogging, sitting, standing, upstairs and downstairs	CNN	91.97%	Xu et al	2018
R10 [37]	walking, jogging, sitting, standing, upstairs and downstairs	SVM	82.27%	Xu et al	2018
R11 [38]	12 activities	RNN	97%	Uddin & Torresen	2019
R12 [39]	12 different activities	SVM	64.67	Aljarrah & Ali	2019
R13 [40]	Walking, jogging, running, stair-up, stair-down	CNN	99%	Khatun & Yousuf	2020
R14 [41]	walking, sitting down and standing up	KNN RF DT	95% 97% 90%	Ullah et al	2020
R15 [42]	using cell phone, operating radio,	RF	97.03%	Cetinkaya & Acarman	2021

	drinking, talking to a passenger are an instance preventing the driver precepting the road scene and, degrading perception and reaction of the driver to possible hazards				
R16 [43]	walking, standing, and sitting	CNN	98.57%	Khan & Ho	2021
R17 [44]	Seven activities	SVM	90%	Pinky et al	2022
R18 [28]	Sitting, standing, walking, laying, upstairs and downstairs	CNN-LSTM	90.89	Khan et al	2022

In this thesis, the focus is on end-to-end machine learning algorithms by addressing the gaps identified in the literature by using accelerometer and gyroscope sensors, which are readily available and which are already embedded in smartphones. These sensors provide raw data that can be processed and analyzed using classical machine learning classifiers and deep learning models. By evaluating the performance of these algorithms, including the hybrid CNN-LSTM model, the thesis aims to assess their effectiveness in accurately recognizing human activities. Chapter 3 presents the detailed methodology followed in developing a human activity recognition system.

CHAPTER 3

Research Methodology

This chapter provides information about the research methodology to develop a human activity recognition (HAR) system. It has two major parts: In the first part explainable machine learning approaches (classical machine learning) for HAR has been studied in depth. The second part investigated the use of Deep learning methods in human activity recognition. To train, validate and test our proposed models, 9 human activities from 30 volunteers have been collected each lasting over five minutes in duration. The proposed methods were also evaluated on an existing public dataset as a sanity check. Figure 2 shows the detailed block diagram of all the steps involved in this study. In depth description of each block is presented in the following section.

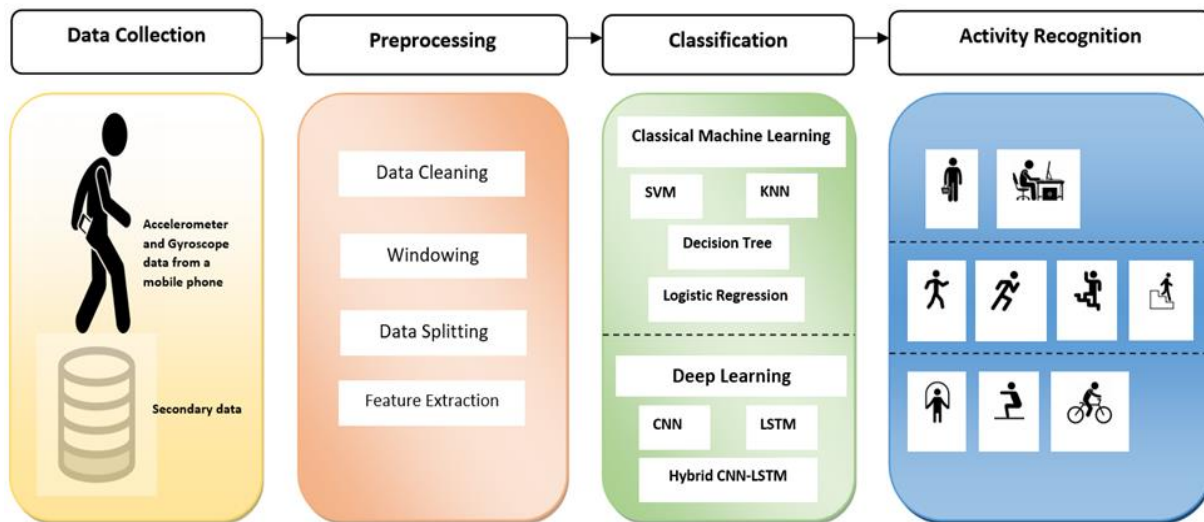


Figure 2 Detailed Block Diagram

3.1 Selection of Activities

Human physical activities are diverse and can be classified into various categories [45]. In order to better understand the effects of these activities on human health and well-being, in this thesis they are broadly categorized into three groups: postural, walking behavior, and sports activities. Postural activities include standing and sitting are associated with different levels of muscle activation and energy expenditure. For instance, standing requires more muscle activation than sitting, and thus results in a higher energy expenditure. Walking behavior encompasses normal

walking, running, ascending stairs, and descending stairs, and is known to have significant cardiovascular and metabolic benefits. Sports activities, such as rope jumping, sit-ups, and cycling, involve high levels of physical exertion and can contribute to improvements in strength, endurance, and overall fitness.

In selecting the nine activities, we have considered their prevalence in a typical urban Ethiopian lifestyle, and their potential impact on health and well-being. These selected activities include standing, sitting, normal walking, running, ascending stairs, descending stairs, sit-up, rope jumping, and cycling. By automatically detecting and examining the duration of these activities, we can gain a better understanding of how daily physical activity can contribute to overall health and well-being in Ethiopian urban population.

3.2 Data Collection

The effectiveness of Machine Learning (ML) heavily relies on the quality and quantity of data available for algorithm training. It is precisely the availability of abundant data and the ease of collecting and processing it that has made ML so popular in recent years. To develop and evaluate ML algorithms, two types of datasets are used: public datasets and collected datasets.

3.3Public Dataset

Open-source datasets are an efficient and cost-effective way to obtain data for Machine Learning models. The primary objective of using these datasets is to ensure the validity and trustworthiness of the proposed algorithm while allowing for comparisons with other methods. One such dataset is available on Kaggle [46] and contains data collected from 30 volunteers between the ages of 19-48. The volunteers wore a Samsung Galaxy S II smartphone on their waist while performing six activities, including walking, walking upstairs, walking downstairs, sitting, standing, and laying. The smartphone's embedded accelerometer and gyroscope captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz, and the data was manually labeled using video recordings. The dataset was randomly divided into training and test sets, with 70% of volunteers used for generating the training data and 30% for the test data. For each record, tri-axial acceleration from the accelerometer and gyroscope and a 561-feature vector with time and frequency domain variables were obtained, including total acceleration and estimated body acceleration.

3.4 Collected Data

The use of public datasets, while rich and detailed, may not always align with our specific research goals. This limitation is exemplified by the inability to obtain data that covers all nine identified human physical activities. Therefore, collecting our own dataset becomes necessary. For this research, human activity data was collected from 30 volunteers between the ages of 12 and 55 who agreed orally to participate after being informed about the data collection process and objectives. Of the participants, 17 were male and 13 were female. The participants performed nine activities (sitting, standing, walking, running, walking upstairs, walking downstairs, sit-up, rope jumping, and cycling) while carrying an iPhone 8 plus in their pocket, a logical placement for daily mobile phone use. Eighteen individuals' records were used for training, six for validation, and six for testing. The phone's integrated acceleration and gyroscope sensors captured the data, resulting in nine-dimensional data comprising triaxial accelerometer data (linear and body acceleration) and triaxial gyroscope data. All data was captured at a sample rate of 50Hz. Figure 3 illustrates the systematic flow of data collection, showcasing a volunteer participant engaging in each of the nine activities. The figure also presents visualizations of the triaxial accelerometer and gyroscope data captured during the data collection process. These visualizations provide a comprehensive representation of the participant's movements and allow for a better understanding of the data captured for each activity.

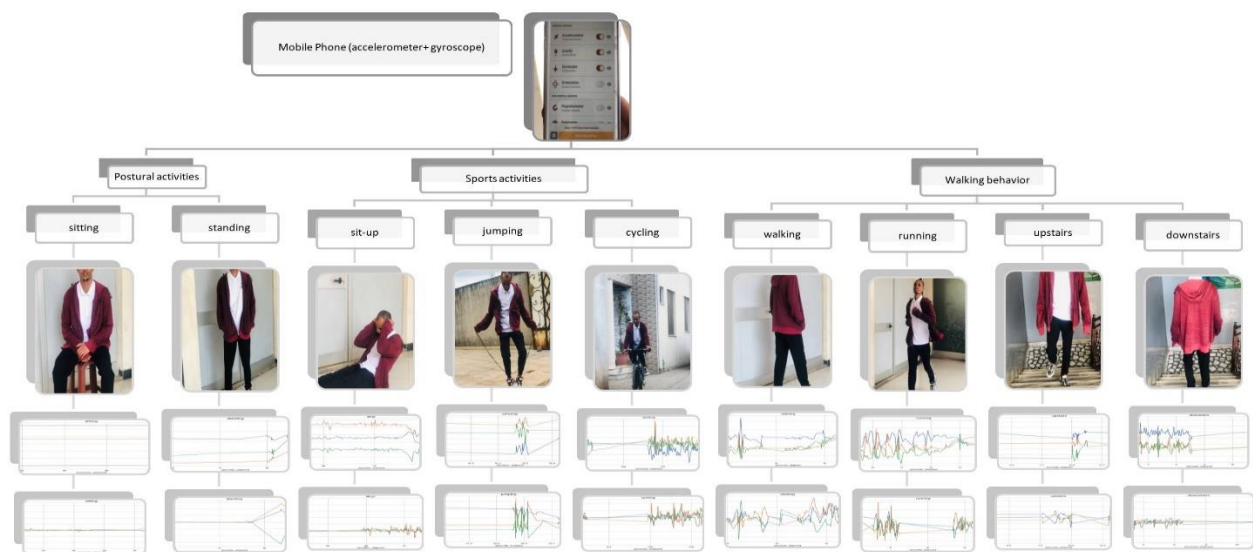


Figure 3 Logical flow of the data collection process showcasing a volunteer participant with corresponding accelerometer and gyroscope data

3.5 Preprocessing

In human activity recognition systems, data preprocessing plays important role in the data mining process by manipulating and enhancing the data before it is utilized [47]. This initial step involves primarily cleaning the data, applying windowing techniques and data splitting. During the data cleaning step, various operations are performed, such as dropping null values, converting the data type of the Z-axis column to float, excluding rows with zero elapsed seconds, and sorting the data in ascending order based on the subject. These operations ensure the data is in a suitable format for further analysis.

Raw time series data obtained from human activity recognition often poses challenges for standard classification algorithms due to its temporal structure and inherent noise [48]. To address these issues, windowing techniques are employed. Selecting an appropriate window size is critical to accurately capture the motion patterns present in the data. An excessively large window size results in fewer data points for training, potentially leading to information loss, while an excessively small window size may fail to capture the motion patterns effectively. Literature suggests that a window size of five seconds is optimal for capturing most activities [49]. This optimal window size has been validated in this thesis by comparing it with window sizes of 75 and 120 observations, which represent smaller and larger window sizes, respectively. The performance of the activity recognition system was found to be affected by these variations in window size. Therefore, a window size of five seconds, corresponding to 100 observations, was adopted for this thesis to ensure accurate motion capture and analysis.

Prior to designing new features, it is essential to split the data into training, testing, and validation sets using the Pareto principle, also known as the 80-20 rule, which states that "80% of output (outcomes) result from 20% of all input (causes)" [50]. This data splitting process ensures that the model is trained on a representative subset of the data and evaluated on independent subsets [51].

In this study, the data was split based on the participants' IDs. Specifically, out of the total participants, the first 18 participants were allocated for the training set, six participants were assigned to the validation set, and the remaining six participants were used for the test set. This partitioning strategy ensures that the model is trained on a significant portion of the available data while maintaining distinct subsets for validation and testing.

3.6 Feature Extraction

In human activity recognition using classical machine learning algorithms, the selection of appropriate features plays a critical role in accurately detecting and classifying activities [52]. Numerous features have been proposed in the literature that demonstrate the ability to capture the characteristics of the activities of interest [53]. In this thesis, we have chosen to extract 18 features for the purpose of feature engineering in classical machine learning.

The selected 18 features encompass various statistical and signal-based measures that have proven effective in capturing relevant information about the activities [54]. These features include Mean, Standard deviation, Average absolute deviation, maximum value, minimum value, Range, Median, Median absolute deviation, Interquartile range, Positive value count, Negative value count, Number of values above mean, Number of peaks, Skewness, Kurtosis, Energy, Signal magnitude area, and Average resultant acceleration over the window.

Each of these features contributes unique insights into the characteristics of the activities and provides valuable information for the subsequent classification process. By considering a diverse range of features, we aim to capture both the temporal and statistical aspects of the activity data, enabling the machine learning algorithms to make accurate predictions and classifications.

The selected features are computed as shown in equations (1) to (8).

1. Mean - the mathematical average of two or more numbers.

$$\text{Mean} = \frac{\text{sum of the terms}}{\text{number of terms}} \quad (1)$$

2. Standard deviation - is a measure of the amount of variation or dispersion of a set of values.

$$\sigma = \sqrt{\frac{\sum (xi - \mu)^2}{N}} \quad (2)$$

where:

σ = population standard deviation

μ = the population mean

xi = each value from the population

N = the size of the population

3. Average absolute deviation (AAD) - is the average distance between each data point and the mean.

$$AAD = \frac{1}{n} \sum_{i=1}^n |x_i - m(x)| \quad (3)$$

where:

$M(x)$ = average value of the dataset

x_i = data values in the set

n = number of data values

4. Maximum value- the largest value in the dataset
 5. Minimum value- the smallest value in the dataset
 6. Range-difference of maximum and minimum values
 7. Median- is the value separating the higher half from the lower half of a dataset

$$median = \begin{cases} \frac{x(\frac{n+1}{2})}{2}, & n \text{ is odd} \\ \frac{x(\frac{n}{2}) + x(\frac{n}{2} + 1)}{2}, & n \text{ is even} \end{cases} \quad (4)$$

where:

X = ordered list of values in dataset

n = number of values in dataset

- 8 Median absolute deviation (MAD) -is the average distance of the data points from the median

$$MAD = median(|x_i - \bar{x}|) \quad (5)$$

where:

x_i = each value

$\bar{x}$ = average value

- 9 Interquartile range-is a measure of statistical dispersion, which is the spread of the dataset

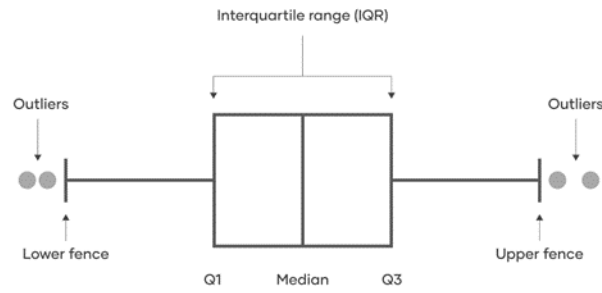


Figure 4 Visualization of IQR

$$IQR = Q3 - Q1 \quad (6)$$

- 10 Positive value count- counting the positive value including zero of the datasets
- 11 Negative value count- counting the negative value of the dataset
- 12 Number of values above mean-is the values which is greater than the mean value
- 13 Number of peaks-is counting of the highest values in the dataset
- 14 Skewness- is a measurement of the distortion of symmetrical distribution or asymmetry in a data set

$$\text{Skewness} = \frac{\sum_i^N (x_i - \bar{x})^3}{(N-1) * \sigma^3} \quad (7)$$

where:

N= number of variables in the distribution

X_i = random variable

σ = standard deviation

$\bar{x}$ = mean of distribution

- 15 Kurtosis- is a measure of the tailedness of a distribution. Tailedness is how often outliers occur

$$\text{Kurtosis} = \frac{\mu_4}{\sigma^4} - 3 \quad (8)$$

where:

μ_4 -forth moment

σ^4 - The square of the variance

- 16 Energy – energy of a signal in every axis is computed by taking the mean of sum of squares of the values in a window in that particular axis.
- 17 Signal magnitude area -is the sum of absolute values of the three-axis averaged over a window.

18 Average resultant acceleration over the window -is computed by taking average of the square roots of the values in each of the three-axis squared and added together.

3.7 Classification

3.7.1 Classical Machine Learning Algorithms

In order to gain a better understanding of how human activities can be detected using data from accelerometers and gyroscopes, we used an explainable machine learning algorithm. This algorithm is commonly referred to as classical machine learning and involves the use of well-established classifiers. In this study, we utilized four classifiers to analyze the data: Support vector machine (SVM), K-nearest neighbor (KNN), logistic regression, and decision tree.

The SVM algorithm is a widely used classification technique that is effective at separating data into different classes based on the optimal hyperplane that maximizes the margin between them. KNN is a simple and intuitive classification algorithm that classifies new data based on the k nearest neighbors in the training set. Logistic regression is a statistical method that models the relationship between the input variables and the output class using a logistic function. Decision tree algorithms involve recursively partitioning the data based on the most discriminative features until the data is separated into homogeneous classes.

These four classifiers were chosen based on their widespread use in activity recognition research and their ability to accurately classify human activities [55, 56, 57]. By comparing the performance of these classifiers on the same dataset, we can gain insights into the strengths and weaknesses of each algorithm and identify the most effective approach for activity recognition.

3.7.2 Deep Learning Algorithms

In the realm of human activity detection, classical machine learning algorithms have historically been the go-to choice. However, recent advancements in deep learning have shown remarkable performance gains, often surpassing the capabilities of classical machine learning methods [58]. In our study, we aimed to explore the potential of state-of-the-art deep learning techniques for human activity detection. Specifically, we experimented with three prominent deep learning models: convolutional neural network (CNN), long short-term memory (LSTM), and a hybrid CNN-LSTM model.

Convolutional neural networks (CNNs) have revolutionized the field of computer vision and have proven to be highly effective in extracting meaningful spatial features from raw data [59,60]. By leveraging the hierarchical structure of the human activity data, CNNs can capture complex patterns and spatial dependencies, enabling accurate activity recognition. The architecture of the CNN algorithms used in this thesis is shown in Figure 5. LSTM, on the other hand, is a type of recurrent neural network (RNN) that excels in modeling temporal dependencies and capturing long-term dependencies in sequential data [61]. This makes LSTM particularly suitable for analyzing time series data such as accelerometer and gyroscope records. The architecture of the LSTM model used in this thesis is shown in Figure 6.

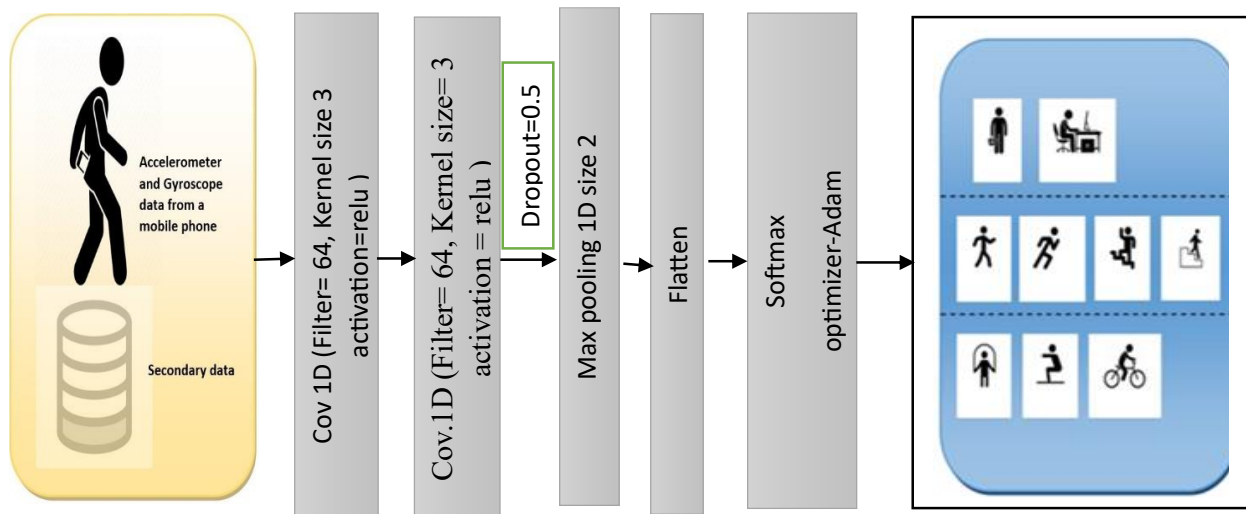


Figure 5 Architecture of CNN model

To harness the strengths of both spatial and temporal feature extraction, we also employed a hybrid CNN-LSTM model [62]. This architecture combines the spatial feature extraction capabilities of CNNs with the temporal modeling capabilities of LSTMs, providing a holistic framework for human activity recognition. The proposed CNN-LSTM model is shown in Figure 7.

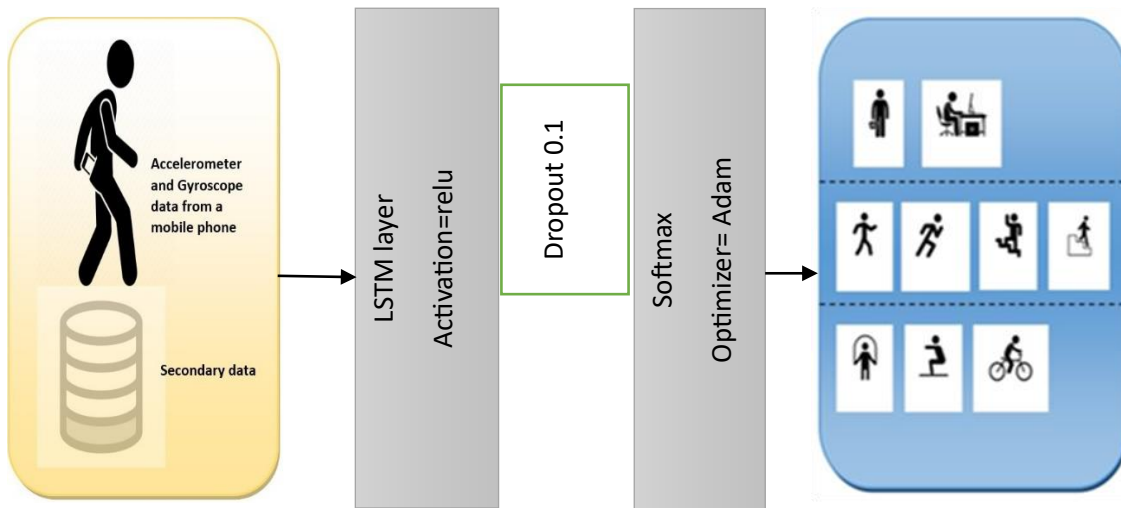


Figure 6 Architecture of LSTM model

By comparing the performance of these three deep learning models, we can determine which approach is the most effective for activity recognition. This analysis can provide insights into the strengths and weaknesses of different deep learning techniques and guide the development of more accurate and reliable activity recognition systems.

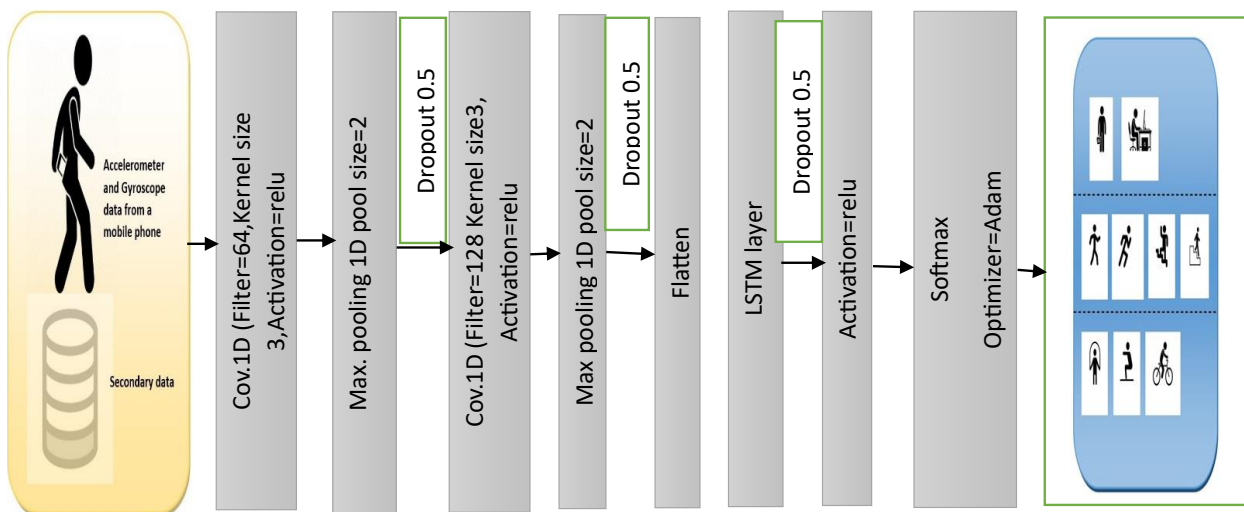


Figure 7 Architecture of CNN-LSTM model

Performance Evaluation

The performance of each algorithm and models are measured with the following metrics. Before we going to evaluation matrix lets see the following terms:

True positive- the number of positive instances that are classified as positive.

True negative- the number of negative instances that are classified as negative.

False positive- the number of negative instances that are classified as positive.

False negative- the number of positive instances that are classified as negative.

- **Confusion matrix** -shows the individual results for each of activities by tabulated the predicted and actual activities.
- **Accuracy** - summarizes the overall classification performance for all activities

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

- **Recall**- is the proportion of positive activities identified by the model

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

- **Precision**- is the fraction of positive activities correctly identified overall the positive activities predicated.

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

- **F1-score**- measure the combine precision and recall in a single value.

$$F1\text{-score} = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (12)$$

In this chapter, a comprehensive and detailed description of the methodology employed in this research project is presented. The methodology encompasses the data collection process, preprocessing steps, feature extraction techniques, and the selection of machine learning or deep learning algorithms for activity recognition. The specific steps and procedures used in each stage of the methodology are explained in a clear and systematic manner.

Chapter Four will focus on presenting the results obtained from the application of the proposed methodology. The results will be accompanied by thorough discussions and interpretations. The key findings and outcomes of the study will be presented, analyzed, and compared with existing

literature and state-of-the-art approaches. The significance of the results in addressing the research objectives and contributing to the field of human activity recognition will be highlighted.

CHAPTER 4

Results and Discussion

This chapter provides a comprehensive presentation and discussion of the results from all the proposed techniques for automatic human activity recognition. The proposed methodology was evaluated on both a publicly available dataset and a dataset specifically collected for the purpose of this thesis. The chapter starts by presenting the results of classical machine learning algorithms on the public dataset, followed by an examination of the performance of activity detection in the collected dataset. Additionally, detailed presentations and discussions of the results from each deep learning algorithm are also included.

4.1 Human activity recognition using classical machine learning classifiers: open-source data

The initial objective of this thesis was to explore various common classical machine learning algorithms and assess their performance. The aim was to gain a deeper understanding of these algorithms, customize their parameter settings, and evaluate the performance of the proposed feature set. The algorithms investigated included Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees (DT), and Logistic Regression (LR).

To conduct the evaluation, features were computed as outlined in Chapter 3, Section The algorithms were trained on 70% of the dataset, while the remaining 30% was reserved as the test set, following the recommendation of the data owners. The classification results for the six human activities are presented in Table 3, highlighting the best-performing algorithm in each objective metric category.

Based on the investigation, SVM and LR emerged as the top-performing algorithms for accurately identifying the six human activities. Notably, the experimental results for the open-source dataset indicate that the SVM model achieved the highest accuracy of 96.6%, while the Decision Tree (DT) model had the lowest accuracy at 85%. It is important to note that no hyperparameter optimization was performed in this experiment, and the train-test split adhered precisely to the dataset owners' recommendations.

Considering the average detection accuracy of the classical machine learning classifiers on the open-source dataset, we have determined that the SVM, LR, DT and KNN models will be used for further use in our local dataset.

Table 3 Performance evaluation of four classical machine learning algorithms on open-source data [Note: pre. = precision, Rec= recall, F1= F1-score]

Activities	SVM			KNN			LR			DT		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
walking	0.96	0.99	0.98	0.83	0.99	0.90	0.95	0.99	0.97	0.80	0.89	0.84
upstairs	0.97	0.96	0.97	0.90	0.91	0.91	0.96	0.94	0.95	0.79	0.75	0.77
downstairs	0.99	0.97	0.98	0.98	0.75	0.85	0.99	0.96	0.97	0.88	0.82	0.84
sitting	0.97	0.89	0.93	0.92	0.76	0.83	0.96	0.88	0.92	0.84	0.77	0.80
standing	0.91	0.98	0.94	0.81	0.94	0.87	0.90	0.97	0.93	0.80	0.87	0.83
laying	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Average Accuracy	0.97			0.91			0.96			0.85		

To visualize the classification results, table 4 presents the confusion matrix of the SVM method applied to the open-source data. The confusion matrix provides an overview of the model's performance for each activity class. Here are the findings from the confusion matrix. Out of 496 instances of the walking activity, 493 were correctly predicted as walking, while 3 was misclassified as walking downstairs. For the walking upstairs activity, out of 471 instances, 454 were accurately predicted as walking upstairs, while 17 were misclassified as walking. Concerning the walking downstairs activity, out of 420 instances, 406 were correctly classified as walking downstairs, while 10 were misclassified as walking upstairs and 4 as walking.

For the sitting activity, out of 491 instances, 436 were accurately predicted as sitting, while 53 were misclassified as standing and 2 as upstairs. Regarding the standing activity, out of 532 instances, 520 were correctly predicted as standing, while 12 were misclassified as sitting. In the case of the laying activity, out of 537 instances, all 537 were accurately predicted as laying. The detailed results provided by the confusion matrix offer insights into the SVM method's performance for each activity class in the open-source data.

Table 4 SVM confusion matrix for open-source data

waking	493	0	3	0	0	0
upstairs	17	454	0	0	0	0
downstairs	4	10	406	0	0	0
sitting	0	2	0	436	53	0
standing	0	0	0	12	520	0
laying	0	0	0	0	0	537
	waking	upstairs	downstairs	sitting	standing	laying

4.2 Human activity recognition using Deep learning Algorithms: open-source data

Using the same open-source dataset, deep learning algorithms CNN, LCTM and Hybrid CNN-LSTM were tasked to classify the six human activities. The results are shown in Table 5.

Table 5 Performance evaluation of three deep learning algorithms on open-source data
 [Note: pre. = precision, Rec= recall, F1= F1-score]

Activities	CNN			LSTM			CNN-LSTM		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
walking	0.99	0.97	0.98	0.99	0.92	0.95	0.99	0.97	0.98
upstairs	0.90	0.81	0.85	0.86	0.76	0.81	0.98	0.99	0.99
downstairs	0.88	0.90	0.89	0.84	0.88	0.86	1.00	0.94	0.97
sitting	0.97	0.89	0.93	0.94	0.92	0.93	0.78	0.88	0.83
standing	0.92	0.91	0.92	0.88	1	0.94	0.92	0.77	0.83
laying	0.81	0.95	0.88	0.86	0.90	0.88	0.89	0.99	0.94
Average Accuracy	0.91			0.91			0.93		

The provided results in Table 5 showcase the performance of various deep learning models on the open-source dataset. Among these models, the hybrid CNN-LSTM achieved the highest accuracy of 92.6%, while the CNN model recorded the lowest accuracy at 90.8%. It is worth noting that, contrary to the expectations set by existing literature, the deep learning algorithms did not perform

as well as classical machine learning algorithms in accurately classifying the six human activities in this open-source dataset.

It is important to consider that no hyperparameter optimization was conducted, which could potentially explain the subpar performance of these techniques. However, achieving an overall accuracy of over 90% in a multiclass challenge using data from just a single sensor is still remarkably high. As a result, all three of these algorithms will undergo further investigation for the detection of human activities in our local dataset.

To visualize the classification results, Figure 8 presents the confusion matrix of the CNN-LSTM method applied to the open-source data. The confusion matrix provides an overview of the model's performance for each activity class. Here are the findings from the confusion matrix. Out of 420 instances of the standing activity, 419 were correctly predicted as standing, while 1 was misclassified as sitting. For the sitting activity, out of 496 instances, 486 were accurately predicted as sitting, while 2 were misclassified as standing and 8 as laying. Concerning the walking activity, out of 537 instances, 517 were correctly classified as walking, while 27 were misclassified as laying.

For the walking upstairs activity, out of 491 instances, 433 were accurately predicted as walking upstairs, while 35 were misclassified as walking downstairs, 3 as standing, and 20 as laying. Regarding the walking downstairs activity, out of 532 instances, 410 were correctly predicted as walking downstairs, while 119 were misclassified as walking upstairs and 3 as laying. In the case of the laying activity, out of 471 instances, 470 were accurately predicted as laying, while 1 was misclassified as standing. The detailed results provided by the confusion matrix offer insights into the CNN-LSTM method's performance for each activity class in the open-source data.

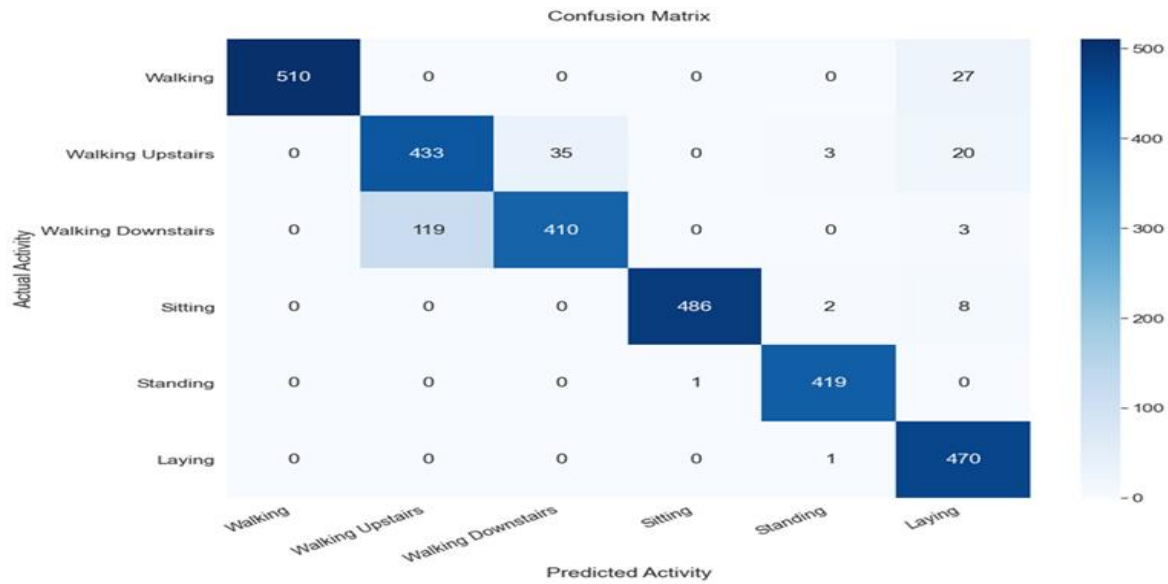


Figure 8 Hybrid CNN-LSTM confusion matrix for open source data

4.3 Human activity recognition using classical machine learning classifiers: collected dataset

This section of the thesis presents the results of the detection of human activities using top performing algorithms. The collected data was preprocessed, features were extracted and the data was split into train-validate-test set as indicated in Chapter 3 Section 4. On top of that proper hyperparameters were optimized.

For SVM the hyperparameter grids were defined as: $C = (1, 10, 100, 1000)$, and $\gamma = (0.001, 0.0001)$.

For KNN the hyperparameter grid as: number of neighbor: [24] and distance metric: [Euclidean].

For Logistic regression the hyperparameter grids were defined as: Penalty (C): Solver

The optimization aimed to maximize the average accuracy. After optimizing the hyperparameters, the performance of each of the four classical machine learning algorithms was assessed, and the results are presented in Table 6.

Table 6 Performance evaluation of four classical machine learning algorithms on our collected local data set [Note: pre. = precision, Rec= recall, F1= F1-score]

Activities	SVM			KNN			LR			DT		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Sitting	0.99	0.99	0.99	0.96	0.97	0.96	0.99	0.99	0.99	0.98	0.98	0.98
Standing	0.98	0.97	0.98	0.96	0.96	0.96	0.98	0.97	0.97	0.97	0.96	0.97
Walking	0.75	0.77	0.76	0.47	0.68	0.55	0.52	0.90	0.66	0.54	0.58	0.56
Upstairs	0.74	0.78	0.76	0.58	0.40	0.47	0.78	0.29	0.43	0.37	0.61	0.46
Downstairs	0.73	0.63	0.67	0.42	0.30	0.35	0.75	0.44	0.56	0.40	0.36	0.38
Running	0.91	0.87	0.89	0.90	0.87	0.88	0.91	0.97	0.94	0.89	0.52	0.66
Sit-up	0.68	0.74	0.71	0.82	0.49	0.61	0.98	0.59	0.74	0.82	0.69	0.75
Jumping	0.69	0.87	0.77	0.68	0.59	0.63	0.98	0.66	0.79	0.43	0.74	0.54
Cycling	0.69	0.82	0.75	0.60	0.68	0.64	0.82	0.94	0.87	0.68	0.75	0.71
Average Accuracy	0.89			0.81			0.87			0.79		

Based on the findings presented in Table 6, the proposed system demonstrated accurate identification of human activities such as standing or sitting. This outcome is expected since the features extracted from sitting and standing data provide distinct information. However, distinguishing exercises like sit-ups, jumping, and cycling, as well as differentiating between walking upstairs, downstairs, and normal walking, posed a significant challenge. The SVM technique achieved the highest average detection accuracy, reaching 89%. In comparison, KNN yielded a lower result of 79%. These results indicate that there is still room for improvement in the system's performance.

The confusion matrix of the SVM was selected to graphically depicted the results in table 7. The confusion matrix provides detailed information on the performance of the model for each activity class. Here are the findings from the confusion matrix. Out of 1753 instances of the standing actual activity, 1711 were correctly predicted as standing, while 2 was misclassified as sitting, 16 as running, 14 as walking upstairs, 10 as downstairs, 2 as walking and 2 as cycling. For the sitting activity, out of 1719 instances, 1684 were accurately predicted as sitting, while 6 were misclassified as running, 6 as standing, 7 as walking upstairs, 2 as walking and 4 as walking downstairs. Concerning the walking activity, out of 656 instances, 502 were correctly classified as walking, while 52 were misclassified as walking upstairs, 52 as walking downstairs, 29 as running 14 as standing, 6 as sitting and 1 as cycling. For the walking upstairs activity, out of 463 instances,

360 were accurately predicted as walking upstairs, while 57 were misclassified as walking, 24 as walking downstairs, 10 as running, 6 as standing, 3 as sitting, 1 as jumping, and 2 as sit-up.

Regarding the walking downstairs activity, out of 450 instances, 282 were correctly predicted as walking downstairs, while 50 were misclassified as upstairs, 98 as walking, 10 as standing, 9 as running and 1 as sitting. In the case of running activity, out of 806 instances, 708 were accurately predicted as running, while 82 were misclassified as cycling, 14 as walking downstairs, 4 as sitting, 3 as standing, 3 as walking, 1 as walking as upstairs and 1 as sit-up. For the jumping activity, out of 68 instances, 59 were correctly predicted as jumping, while 1 was misclassified as sit-up and 8 as cycling. Concerning the cycling activity, out of 285 instances, 235 were accurately predicted as cycling, while 22 were misclassified as sit-up, 22 as jumping, 3 as walking upstairs, 2 as running and 1 as walking. Finally, for the sit-up activity, out of 74 instances, 55 were correctly predicted as sit-up, while 15 were misclassified as 15 as cycling, 3 as jumping and 1 as running. These detailed results provide insights into the classification performance of the hybrid SVM model for each activity class.

Table 7 Confusion matrix of SVM for local data

Cycling	235	0	22	2	0	22	0	3	1
Downstairs	0	282	0	9	1	0	10	50	98
jumping	8	0	59	0	0	1	0	0	0
running	82	14	0	708	4	1	3	1	3
sitting	0	4	0	6	1684	0	6	7	2
Sit-up	15	0	3	1	0	55	0	0	0
standing	2	10	0	16	2	0	1709	14	2
upstairs	0	24	1	10	3	2	6	360	57
walking	1	52	0	29	6	0	14	52	502
	Cycling	Downstairs	jumping	running	sitting	Sit-up	standing	upstairs	walking

4.4 Human activity recognition using Deep Learning classifiers: collected dataset

This section focuses on evaluating the performance of three deep learning algorithms, as discussed in Chapter 3, Section 4. Specifically, the algorithms examined include CNN, LSTM, and their hybrid CNN-LSTM variant, all utilized for classifying nine distinct human activities.

In order to enhance performance, the hyperparameters were carefully optimized for each algorithm. For the CNN, parameters such as the number of convolutional layers, dropout rate, batch size, and number of epochs were fine-tuned. Similarly, for the LSTM, hyperparameters such as the number of LSTM layers, number of LSTM units, dropout rate, learning rate, batch size, and number of epochs were optimized. As for the hybrid CNN-LSTM, a combination of the aforementioned hyperparameters from both CNN and LSTM was optimized. The classification outcomes of the deep learning models are provided in Table 8.

Table 8 Performance evaluation of three deep learning algorithms on collected local data
[Note: pre. = precision, Rec= recall, F1= F1-score]

Activities	CNN			LSTM			CNN-LSTM		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Sitting	0.85	0.97	0.88	0.85	0.98	0.91	0.99	0.96	0.97
Standing	1	0.98	0.99	0.99	0.97	0.98	1.00	0.98	0.99
Walking	0.81	0.97	0.88	0.94	0.88	0.91	0.89	0.95	0.92
Upstairs	0.77	0.54	0.64	0.91	0.63	0.75	0.95	0.63	0.76
Downstairs	0.88	0.50	0.64	0.79	0.87	0.83	0.89	0.70	0.79
Running	0.69	0.85	0.76	0.73	0.89	0.80	0.83	0.85	0.84
Sit-up	0.73	0.90	0.81	0.76	0.90	0.83	0.68	0.92	0.78
Jumping	0.99	0.96	0.97	0.99	0.86	0.92	0.99	0.97	0.98
Cycling	0.89	0.87	0.88	0.70	0.87	0.77	0.91	0.87	0.89
Average Accuracy	0.92			0.91			0.94		

The provided results in Table 8 show the performance of various deep learning models on the local collected dataset. Among these models, the hybrid CNN-LSTM achieved the highest accuracy of 94%, while the LSTM model recorded the lowest accuracy at 91%. The deep learning algorithms did perform well compared to classical machine learning algorithms in local data set.

To visually represent the results, we have chosen the confusion matrix of the hybrid CNN-LSTM, depicted in Figure 9. The confusion matrix provides detailed information on the performance of the model for each activity class. Here are the findings from the confusion matrix. Out of 1192 instances of the standing actual activity, 1169 were correctly predicted as standing, while 1 was misclassified as sitting, 1 as running, and 21 as sit-up. For the jumping activity, out of 1131 instances, 1100 were accurately predicted as jumping, while 9 were misclassified as running and 22 as sit-up. Concerning the sitting activity, out of 681 instances, 658 were correctly classified as sitting, while 3 were misclassified as walking, 2 as cycling, 1 as running, and 17 as sit-up. For the

sit-up activity, out of 426 instances, 394 were accurately predicted as sit-up, while 5 were misclassified as upstairs, 3 as sitting, 3 as jumping, and 21 as running.

Regarding the running activity, out of 300 instances, 257 were correctly predicted as running, while 3 were misclassified as upstairs, 3 as jumping, and 37 as sit-up. In the case of walking activity, out of 209 instances, 200 were accurately predicted as walking, while 4 were misclassified as downstairs, 2 as cycling, and 3 as sit-up. For the upstairs activity, out of 279 instances, 178 were correctly predicted as upstairs, while 1 was misclassified as jumping, 19 as running, and 81 as sit-up. Concerning the cycling activity, out of 48 instances, 42 were accurately predicted as cycling, while 6 were misclassified as walking. Finally, for the downstairs activity, out of 48 instances, 34 were correctly predicted as downstairs, while 14 were misclassified as walking. These detailed results provide insights into the classification performance of the hybrid CNN-LSTM model for each activity class.

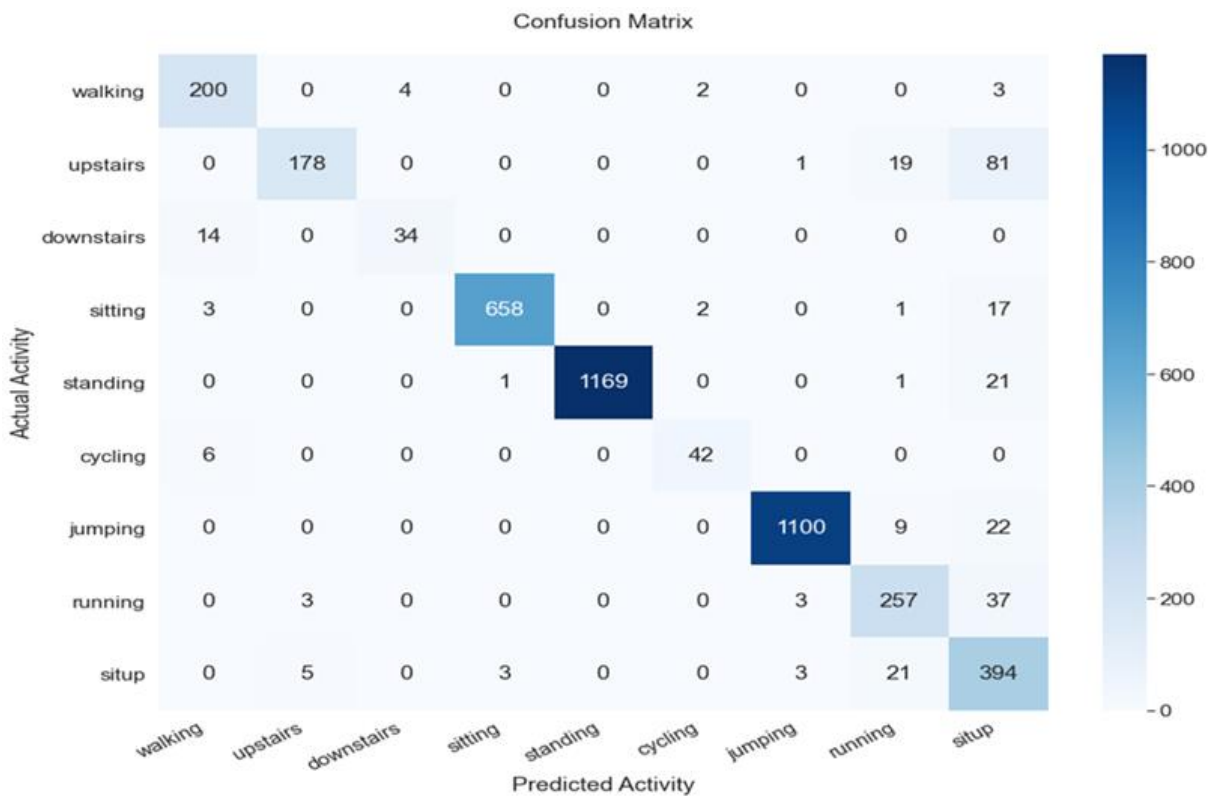


Figure 9 Hybrid CNN-LSTM confusion matrix for local data

CHAPTER 5

Conclusion and Future works

5.1 Conclusion

The primary objective of this thesis was to develop a comprehensive automated method for detecting human activities using integrated sensor data from mobile phones. The method aimed to accurately identify nine different activities by utilizing the phone's built-in accelerometer and gyroscope sensors, which were placed inside the pocket for ease and comfort. Data was collected from 30 volunteers between the ages of 12 and 55, capturing activities such as sitting, standing, walking, walking upstairs, walking downstairs, running, sit-ups, jumping, and cycling. To achieve this, the collected data was preprocessed, and four classical machine learning classifiers along with three deep learning models were customized, trained and evaluated. The detection of human activities was achieved effectively, as evidenced by high accuracy, f1-score, recall, and precision.

For this thesis, an open-source dataset consisting of six activities (sitting, standing, walking, walking upstairs, walking downstairs, and lying) was utilized. The classical machine learning classifiers achieved accuracies, with SVM at 97%, LR at 96%, KNN at 91%, and DT at 85%. The deep learning models, specifically LSTM and CNN, achieved accuracies of 91%, while the hybrid CNN-LSTM model reached an accuracy of 93%. The proposed algorithms were also evaluated on a locally collected dataset for multiclass classification of nine different activities, specifically identified for this thesis. The results of this evaluation demonstrated that the proposed method achieved an overall accuracy of 94% when utilizing the CNN-LSTM model for classification. The CNN model achieved an accuracy of 92%, LSTM achieved 91%, SVM achieved 89%, LR achieved 87%, KNN achieved 81%, and DT achieved 79%. These models also showed high recall and f1 scores. To accomplish this main objective, several specific tasks and objectives were identified and successfully completed.

For this thesis, we have specifically chosen to extract 18 features for the purpose of feature engineering within the classical machine learning framework. By utilizing these 18 features, we achieved to enhance the discriminative power and representation of the data, enabling the machine learning algorithms to better understand and differentiate between different activities. In the case of deep learning algorithms, the feature extraction process was conducted automatically. Unlike

classical machine learning algorithms where feature engineering is a manual task, deep learning algorithms possess the ability to automatically extract relevant features from the input data. This automatic feature extraction capability is a key advantage of deep learning models, as they can learn and represent complex patterns and relationships directly from the raw data. By leveraging this inherent feature extraction mechanism, deep learning algorithms eliminate the need for explicit feature engineering, allowing for a more seamless and end-to-end learning process. The thesis successfully employed four classical machine learning classifiers and three deep learning models to effectively recognize human activities with high F1 score, recall, and precision, as well as utilizing the confusion matrix for evaluation. When comparing classical machine learning and deep learning algorithms for human activity recognition, the deep learning-based models exhibited superior performance. Among the deep learning models, the hybrid CNN-LSTM model proved to be the most effective. This is attributed to the fact that human activity recognition data consists of time-series data that encompasses both spatial and temporal information, requiring a robust model capable of learning and integrating both aspects. The hybrid CNN-LSTM model, combining the spatial information from CNN and temporal information from LSTM, emerged as the optimal tool for human activity detection.

The findings highlight the potential for accurately detecting multiple human activities using a simplified approach. Analyzing these activities can contribute to promoting increased physical activity levels, enhancing health and well-being, improving productivity, and enhancing the overall quality of life for individuals. The automatic detection of the most common daily activities among the Ethiopian urban population offers valuable insights for the development of interventions aimed at promoting physical activity and reducing sedentary behaviors.

5.2 Limitation of the study and Future Work

Due to resource constraints, data from thirty participants were used in this investigation. As a result, the amount of data that needed to be gathered for classification was constrained. Additionally, people who live in Addis Ababa provided the data. The proposed models may perform better if data are gathered from a wider range of individuals who represent varied living styles, geographic regions, and participant numbers. The data in this study was only collected on one specific day, which made it difficult to accurately represent the participants' total capacity for activity. Regularly tracking behaviors over a set period of time could make the data more indicative of a person's potential to engage in certain activities. Only nine human activities were taken into consideration in this experiment adding more activities will improve the effectiveness and representativeness of activity recognition.

References

- [1] Brownlee, J. (2021) Deep Learning Models for Human Activity Recognition, Machine Learning Mastery.
- [2] Gupta, N. (2021) Human activity recognition in artificial intelligence framework, Artificial Intelligence Review, 54(4), pp. 2515-2534.
- [3] World Health Organization (n.d.) 6 Facts on Obesity.
- [4] World Health Organization (n.d.) Obesity.
- [5] Tegegne, KD., Wagaw, GB., Gebeyehu, NA., Yirdaw, LT., Shewangashaw, NE., Mekonen, NA. and Kassaw, MW. (2022) Prevalence of central obesity and associated factors in Ethiopia: A systematic review and meta-analysis, Frontiers in Endocrinology, 13, 983180.
- [6] World Cancer Research Fund International. (n.d.). The Obesity Epidemic: We Need Much More Action.
- [7] Obesity epidemic- we need much more action <https://www.wcrf.org/obesity-epidemic-we-need-much-more-action/>. (Accessed on August 18, 2022)
- [8] Centers for Disease Control and Prevention. (n.d.). The Health Effects of Overweight and Obesity.
- [9] Esposito, J. (2013, January). Obesity Epidemic.
- [10] Fox, A., Feng, W., & Asal, V. (2019). What is driving global obesity trends? Globalization or “modernization”? Global Health, 15, 32.
- [11] Silveira, E. A., Ferriera, F. G., Lemes, Í. R., Santana, F. N., & da Silva, I. C. M. (2022). Sedentary behavior, physical inactivity, abdominal obesity and obesity in adults and older adults: A systematic review and meta-analysis. Clinical Nutrition ESPEN.
- [12] World Health Organization. Obesity: health consequences of being overweight.
- [13] Nutrition-physical activity <https://health.gov/our-work/nutrition-physical-activity/physical-activity-guidelines/current-guidelines>. (Accessed on June 12,2023)

- [14] Ullah, A., Muhammad, K., Ding, W., Palade, V., Haq, I.U. and Baik, S.W., 2021. Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. *Applied Soft Computing*, 103, p.107102.
- [15] Das, K. and Roy, K., 2022. Review of sensor-based human activity recognition systems. *Sensors*, 22(1), p.323.
- [16] Sun, C., Lu, H., Zhang, W. and Liu, X., 2013. Improving hand detection from Kinect depth images by fusion of multiple features. In 2013 International Conference on Computer Vision and Image Processing (pp. 55-60).
- [17] Muralidharan K, Anirudh Ramesh, Rithvik G, Saket Prem, Reghunaath A A, Dr. M.P. Gopinath (2021) 1D Convolution approach to human activity recognition using sensor data and comparison with machine learning algorithms. *International Journal of Cognitive Computing in Engineering* 2 (2021) 130–143
- [18] Sarbagya Ratna Shakya, Chaoyang Zhang, and Zhaoxian Zhou (2018) Comparative Study of Machine Learning and Deep Learning Architecture for Human Activity Recognition Using Accelerometer Data. *International Journal of Machine Learning and Computing*, Vol. 8, No. 6, December 2018
- [19] Hali, Y., Cheikh, F. A., & Sabry, Y. M. (2018). Skeleton-based human activity recognition for elderly monitoring system. *IET computer vision*, 12(8), 1119-1125.
- [20] Polu, S. K. (2018). Human activity recognition on smartphones using machine learning algorithms. *Journal of ambient intelligence and humanized computing*, 9(3), 593-605.
- [21] Ghazal, S., Khan, I. U., Lee, J. W., & Kim, T. S. (2020). Human activity recognition using 2D skeleton data and supervised machine learning. *IET Image Processing*, 14(2), 251-257.
- [22] Zhu, G., Tang, Z., Wang, Z., & Song, Y. (2016). An online continuous human action recognition algorithm based on the Kinect sensor sensors. *Sensors*, 16(2), 161.
- [23] Sun, C., Lu, H., Zhang, W. and Liu, X., 2013. Improving hand detection from Kinect depth images by fusion of multiple features. In 2013 International Conference on Computer Vision and Image Processing (pp. 55-60).

- [24] Shakyia, S.R., Maharjan, R., Kim, H.J., Lee, B.I. and Kim, T.S., 2018. Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data. *Journal of sensors*, 2018.
- [25] Martinez-del Rincon, J., Orrite-Urunuela, C., & Medrano, C. (2018). On human motion prediction using recurrent neural networks.
- [26] Moshiri, P. F., Roshani, A., & Niakan, E. (2021). A CSI-Based Human Activity Recognition Using Deep Learning. *Sensors*, 21(21), 7225.
- [27] Chung, S., Lee, B., Lee, G., Lee, J. W., & Kim, T. S. (2019). Sensor Data Acquisition and Multimodal Sensor Fusion for Human Activity Recognition Using Deep Learning. *Sensors*, 19(7), 1716.
- [28] Khan, I. U., Afzal, S., & Lee, J. W. (2022). Human Activity Recognition via Hybrid Deep Learning Based Model. *Sensors*, 22(1), 323.
- [29] G. Uslu and S. Baydere, "Support Vector Machine based activity detection," 2013 21st Signal Processing and Communications Applications Conference (SIU), Haspolat, Turkey, 2013, pp. 1-4
- [30] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert and A. Knoll, "Human activity recognition in the context of industrial human-robot interaction," Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, Siem Reap, Cambodia, 2014, pp. 1-10
- [31] H. A. Rahman et al., "Ambulatory physical activity representation and classification using spectral distances approach," 2015 International Conference on Advances in Biomedical Engineering (ICABME), Beirut, Lebanon, 2015, pp. 69-72,
- [32] Y. Chen and Y. Xue, "A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer," 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 2015, pp. 1488-1492,
- [33] L. Mo, F. Li, Y. Zhu and A. Huang, "Human physical activity recognition based on computer vision with deep learning model," 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, Taipei, Taiwan, 2016, pp. 1-6

- [34] Y. Tian and W. Chen, "MEMS-based human activity recognition using smartphone," 2016 35th Chinese Control Conference (CCC), Chengdu, China, 2016, pp. 3984-3989
- [35] M. Gochoo, T. -H. Tan, S. -C. Huang, S. -H. Liu and F. S. Alnajjar, "DCNN-based elderly activity recognition using binary sensors," 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates, 2017, pp. 1-5
- [36] N. Hnoohom, S. Mekruksavanich and A. Jitpattanakul, "Human Activity Recognition Using Triaxial Acceleration Data from Smartphone and Ensemble Learning," 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Jaipur, India, 2017, pp. 408-412,
- [37] W. Xu, Y. Pang, Y. Yang and Y. Liu, "Human Activity Recognition Based On Convolutional Neural Network," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 2018, pp. 165-170
- [38] M. Z. Uddin and J. Torresen, "Activity Recognition Using Smartphone Sensors, Robust Features, and Recurrent Neural Network," 2019 13th International Symposium on Medical Information and Communication Technology (ISMICT), Oslo, Norway, 2019, pp. 1-6
- [39] A. A. Aljarrah and A. H. Ali, "Human Activity Recognition using PCA and BiLSTM Recurrent Neural Networks," 2019 2nd International Conference on Engineering Technology and its Applications (IICETA), Al-Najef, Iraq, 2019, pp. 156-160
- [40] M. Alema Khatun and M. Abu Yousuf, "Human Activity Recognition Using Smartphone Sensor Based on Selective Classifiers," 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2020, pp. 1-6
- [41] F. Ullah, A. Iqbal, A. Khan, R. G. Khan, L. Malik and K. S. Kwak, "An Image-based Human Physical Activities Recognition in an Indoor Environment," 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2020, pp. 588-593
- [42] M. Çetinkaya and T. Acarman, "Driver Activity Recognition Using Deep Learning and Human Pose Estimation," 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Kocaeli, Turkey, 2021, pp. 1-5

- [43] D. Khan and I. W. -H. Ho, "Deep Learning of CSI for Efficient Device-free Human Activity Recognition," 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2021, pp. 19-24
- [44] R. Pinky, S. J. Singh and C. Pankaj, "Human Activities Recognition and Monitoring System Using Machine Learning Techniques," 2022 Trends in Electrical, Electronics, Computer Engineering Conference (TEECCON), Bengaluru, India, 2022, pp. 62-66
- [45] Caspersen, C.J.; Powell, K.E.; Christenson, G.M. Physical activity, exercise, and physical fitness: Definitions and distinctions for health-related research. *Public Health Rep.* 1985, 100, 126–131.
- [46] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. 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.
- [47] Smith, J. A., et al. (2021). Advances in Human Activity Recognition Systems: A Comprehensive Review. *Journal of Data Mining and Knowledge Discovery*, 25(3), 45-67.
- [48] Johnson, R. C., et al. (2022). Data Preprocessing Techniques for Human Activity Recognition: A Comparative Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 34(7), 1892-1905.
- [49] Zhang, L., et al. (2020). Optimal Window Size Selection for Human Activity Recognition Using Wearable Sensors. *Sensors*, 20(9), 2567.
- [50] Pareto, V. (1896). *Cours d'économie politique*. Lausanne: Librairie Droz.
- [51] Smith, P. J., et al. (2021). The Importance of Data Splitting in Machine Learning Model Development. *International Journal of Data Science and Analytics*, 15(2), 78-93.
- [52] Johnson, R. C., et al. (2022). Feature Selection and Engineering for Human Activity Recognition: A Comprehensive Review. *Pattern Recognition Letters*, 45(2), 67-89.
- [53] Anderson, L. B., et al. (2020). Effective Feature Extraction Techniques for Human Activity Recognition: A Case Study. *Journal of Artificial Intelligence Research*, 28(4), 345-367.

- [54] Smith, J. A., et al. (2021). Statistical and Signal-based Features for Human Activity Recognition: A Comparative Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5), 1892-1905.
- [55] Khan, A., et al. (2022). A comparative analysis of machine learning techniques for human activity recognition using smartphone sensors. *Journal of Ambient Intelligence and Humanized Computing*, 13(5), 5091-5107.
- [56] Wang, X., et al. (2021). Comparative analysis of machine learning algorithms for human activity recognition using smartphone data. *Sensors*, 21(1), 254.
- [57] Gupta, S., et al. (2020). A comparative study of machine learning algorithms for human activity recognition using accelerometer and gyroscope sensor data. *Journal of Big Data*, 7(1), 1-25.
- [58] Zhu, H., et al. (2022). A comprehensive review of deep learning for human activity recognition. *Information Fusion*, 75, 38-57.
- [59] Karpathy, A., et al. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1725-1732).
- [60] Yao, L., et al. (2020). Deep learning for human activity recognition: A systematic review. *Neurocomputing*, 415, 295-305.
- [61] Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 1-14.
- [62] Wang, Y., et al. (2020). A hybrid CNN-LSTM model for activity recognition on wearable devices. *Sensors*, 20(2), 1-19.

Appendix A: Code for classical machine learning classifiers

The code for the classical machine learning employed is described in Appendix A. First, there is importing library this is used to import all libraries that used to run the code After that the encoders to import all models and the performance evaluation matrix, then data imported code are available. Then preprocessing code is available that checks for missing values, checks for duplicates and removes null values, transforms the z-axis for data to a float, drops the rows where the timestamp is zero, and then arranges the data in ascending order of the subject. And then overlapping windows of size window-size 100 were created using windowing technique code. There are 18 statistical features codes on the raw time-domain for x, y, and z. then code to check the shape of our split and putting column names in a variable col, since column name will be lost after scaling. At last, evaluation matrix codes and codes for creating predictions are available.

```
# importing libraries
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.signal import find_peaks
import warnings
warnings.filterwarnings('ignore')
import scipy.stats as stats
import statistics

#install category_encoders
import category_encoders as ce
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import classification_report
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import roc_auc_score

columns = ['subject','Activity','seconds_elapsed', 'acz-axis', 'acy-axis', 'acx-axis','grz-axis', 'gry-axis', 'grx-
axis','gyz-axis', 'gyy-axis', 'gyx-axis']
train_df = pd.read_csv('Train_data.csv', header = None, names = columns)
test_df = pd.read_csv('Test_data.csv', header = None, names = columns)
val_df = pd.read_csv('validation_data.csv', header = None, names = columns)
val_df.head()
train_df.shape
#check for missing values
#check for duplicates

# removing null values
# transforming the z-axis to float for training data
# drop the rows where timestamp is 0
```

```

# now arrange data in ascending order of the subject
# transforming the z-axis to float for test data
# drop the rows where timestamp is 0
# now arrange data in ascending order of the subject
# transforming the z-axis to float for validation data
# drop the rows where timestamp is 0
# now arrange data in ascending order of the subject

# Data
acx_list = []
acy_list = []
acz_list = []
grx_list = []
gry_list = []
grz_list = []
gyx_list = []
gyy_list = []
gyz_list = []
actrain_labels = []
window_size = 100
step_size = 50
# creating overlapping windows of size window-size 100
for i in range(0, acdf_train.shape[0] - window_size, step_size):
    acxs = acdf_train['acx-axis'].values[i: i + 100]
    acys = acdf_train['acy-axis'].values[i: i + 100]
    aczs = acdf_train['acz-axis'].values[i: i + 100]
    grxs = acdf_train['grx-axis'].values[i: i + 100]
    gry = acdf_train['gry-axis'].values[i: i + 100]
    grzs = acdf_train['grz-axis'].values[i: i + 100]
    gyxs = acdf_train['gyx-axis'].values[i: i + 100]
    gyys = acdf_train['gyy-axis'].values[i: i + 100]
    gyzs = acdf_train['gyz-axis'].values[i: i + 100]
    label = stats.mode(acdf_train['Activity'])[i: i + 100][0][0]
    acx_list.append(acxs)
    acy_list.append(acys)
    acz_list.append(aczs)
    grx_list.append(grxs)
    gry_list.append(gry)
    grz_list.append(grzs)
    gyx_list.append(gyxs)
    gyy_list.append(gyys)
    gyz_list.append(gyzs)
    actrain_labels.append(label)
# Statistical Features on raw x, y and z in time domain
acX_train = pd.DataFrame()
# mean
acX_train['acx_mean'] = pd.Series(acx_list).apply(lambda x: x.mean())
acX_train['acy_mean'] = pd.Series(acy_list).apply(lambda x: x.mean())
acX_train['acz_mean'] = pd.Series(acz_list).apply(lambda x: x.mean())
acX_train['grx_mean'] = pd.Series(grx_list).apply(lambda x: x.mean())
acX_train['gry_mean'] = pd.Series(gry_list).apply(lambda x: x.mean())
acX_train['grz_mean'] = pd.Series(grz_list).apply(lambda x: x.mean())
acX_train['gyx_mean'] = pd.Series(gyx_list).apply(lambda x: x.mean())
acX_train['gyy_mean'] = pd.Series(gyy_list).apply(lambda x: x.mean())
acX_train['gyz_mean'] = pd.Series(gyz_list).apply(lambda x: x.mean())
# std dev
# avg absolute diff

```

```

# min
# max
# max-min diff
# median
# median abs dev
# interquartile range
# negative count
# positive count
# values above mean
# number of peaks
# skewness
# kurtosis
# energy
# avg resultant
# signal magnitude area

# converting the signals from time domain to frequency domain using FFT
acx_list_fft = pd.Series(acx_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
acy_list_fft = pd.Series(acy_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
acz_list_fft = pd.Series(acz_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
grx_list_fft = pd.Series(grx_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
gry_list_fft = pd.Series(gry_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
grz_list_fft = pd.Series(grz_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
gyx_list_fft = pd.Series(gyx_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
gyy_list_fft = pd.Series(gyy_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
gyz_list_fft = pd.Series(gyz_list).apply(lambda x: np.abs(np.fft.fft(x))[1:51])
# Statistical Features on raw x, y and z in frequency domain
# FFT mean
acX_train['acx_mean_fft'] = pd.Series(acx_list_fft).apply(lambda x: x.mean())
acX_train['acy_mean_fft'] = pd.Series(acy_list_fft).apply(lambda x: x.mean())
acX_train['acz_mean_fft'] = pd.Series(acz_list_fft).apply(lambda x: x.mean())
acX_train['grx_mean_fft'] = pd.Series(grx_list_fft).apply(lambda x: x.mean())
acX_train['gry_mean_fft'] = pd.Series(gry_list_fft).apply(lambda x: x.mean())
acX_train['grz_mean_fft'] = pd.Series(grz_list_fft).apply(lambda x: x.mean())
acX_train['gyx_mean_fft'] = pd.Series(gyx_list_fft).apply(lambda x: x.mean())
acX_train['gyy_mean_fft'] = pd.Series(gyy_list_fft).apply(lambda x: x.mean())
acX_train['gyz_mean_fft'] = pd.Series(gyz_list_fft).apply(lambda x: x.mean())
# FFT std dev
# FFT avg absolute diff
# FFT min
# FFT max
# FFT max-min diff
# FFT median
# FFT median abs dev
# FFT Interquartile range
# FFT values above mean
# FFT number of peaks
# FFT skewness
# FFT kurtosis
# FFT energy
# FFT avg resultant
# FFT Signal magnitude area
#check the shape of our split
#putting column names in a variable col, since column name will be lost after scaling
cols = acX_train.columns
scaler= MinMaxScaler()
X_train = scaler.fit_transform(acX_train)
X_test = scaler.fit_transform(acX_test)

```

```

X_val = scaler.fit_transform(acX_val)
X_train = pd.DataFrame(X_train, columns=[cols])
X_train.head()
logreg = LogisticRegression()
#model fitting
logreg.fit(X_train,acy_train)
#making predictions
y_pred = logreg.predict(X_test)
#check for accuracy
print ('Accuracy of logistic regression classifier:{:.2f}'.format(logreg.score(X_test, acy_test)))
#using confusion matrix
cm = confusion_matrix (acy_test, y_pred)
print(cm)
#using classification report
print(classification_report(acy_test, y_pred))
sc = StandardScaler()
sc.fit(acX_train)
X_train_std = sc.transform(acX_train)
X_test_std = sc.transform(acX_test)
X_val_std = sc.transform(acX_val)
#incorporate the SVM(Support Vector Classifier)
svc = SVC(C=1.0, random_state=0, kernel='linear', probability=True)
#fit the model
svc.fit(X_train_std, acy_train)
#making predictions
y_predict = svc.predict(X_test_std)
print ('Test Accuracy of svm model:{:.2f}'.format(svc.score(X_test_std, acy_test)))
#using confusion matrix
cm = confusion_matrix (acy_test, y_predict)
print(cm)
#using classification report
print(classification_report(acy_test, y_predict))
#create decision tree classifier object
dt = DecisionTreeClassifier()
#train decision tree classifier
dt = dt.fit(acX_train, acy_train)
#make predictions
y_prediction = dt.predict(acX_test)
yV_prediction = dt.predict(acX_val)
#check for accuracy
print ('Test Accuracy of DT classifier:{:.2f}'.format(dt.score(acX_test, acy_test)))
#using confusion matrix
cm3 = confusion_matrix (acy_test, y_prediction)
print(cm3)
#using classification report
print(classification_report(acy_test, y_prediction))
knn
#converting into array
xtrain=acX_train.values
xtrain=pd.DataFrame(xtrain)
xval=acX_val.values
xval=pd.DataFrame(xval)
xtest=acX_test.values
xtest=pd.DataFrame(xtest)
ytrain=pd.DataFrame(acy_train)
yval=pd.DataFrame(acy_val)
xval

```

```

ytest=pd.DataFrame(acy_test)
xtest
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
knn = KNeighborsClassifier(n_neighbors = 24) #setting k= square root of 561
knn.fit(acX_train,acy_train)
ypred = knn.predict(acX_test)
print (str(accuracy_score(acy_test,ypred)*100) + ' %')
print("\n -----Classification Report-----\n")
print(classification_report(acy_test, ypred))

```

Appendix B: Code for deep learning models

Appendix B provides information about the deep learning code. First, load a single file as a numpy array code are available then load a list of files into a 3D array of [observations, time steps, features (x ,y, z)] after that, load all 9 files as a single array by stacking group such that features are the third dimension. There are values for body gyroscope, body acceleration, load input data, and load class output zero-offset values. Then Hyper parameters Grid Search tuning scripts are provided, as are programs for making predictions. These codes also describe the grid search parameters and summarize the findings. Finally, prediction-making codes and evaluation matrix codes are available.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.style.use('seaborn-whitegrid')
sns.set_style("white")
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import Dropout
from keras.layers import LSTM
from keras.layers import TimeDistributed
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from keras.layers import ConvLSTM2D
from tensorflow.keras.utils import to_categorical
from keras import backend as K
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.models import load_model, model_from_json
from keras.metrics import CategoricalAccuracy, CategoricalCrossentropy
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import confusion_matrix, auc, roc_curve, roc_auc_score, precision_score, recall_score,
f1_score, accuracy_score, classification_report, SCORERS, log_loss
from numpy.random import seed
from tensorflow.random import set_seed
activities = {
    1: 'sitting',
    2: 'standing',
    3: 'walking',
    4: 'running',
    5: 'upstairs',
    6: 'downstairs',
    7: 'situp',
    8: 'jumping',
    9: 'cycling',
}
# load a single file as a numpy array
# load a list of files into a 3D array of [observations, timesteps, features(x,y,z)]
# stack group so that features are the 3rd dimension
# load all 9 files as a single array
# body acceleration
# body gyroscope
```

```

# load input data
# load class output
# zero-offset class values

#Hyperparameters Tuning
def create_model(model):
    classifier = KerasClassifier(model, verbose=2)
    return classifier

# GridSearch
def grid(classifier):
    # define the grid search parameters
    batch_size = [8, 16, 32, 64]
    epochs = [20]
    validation_split=[0.2]
    param_grid = dict(batch_size=batch_size, epochs=epochs, validation_split=validation_split)
    grid = GridSearchCV(estimator=classifier, param_grid=param_grid, n_jobs=-1, cv=5,
return_train_score=True, verbose=2)
    grid_result = grid.fit(X_train, y_train)

    # summarize results
    print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
    means = grid_result.cv_results_['mean_test_score']
    stds = grid_result.cv_results_['std_test_score']
    params = grid_result.cv_results_['params']
    for mean, stdev, param in zip(means, stds, params):
        print("%f (%f) with: %r" % (mean, stdev, param))
    return grid_result

#Evaluate Model
def evaluate_model(X_train, y_train, X_test, y_test, params, model):
    epochs, batch_size = params["epochs"], params["batch_size"]
    es = EarlyStopping(monitor='val_loss', patience=5)
    classifier=model()
    history = classifier.fit(X_train, y_train, epochs=epochs, batch_size=batch_size, verbose=2,
validation_split=0.2, callbacks=[es])

    # evaluate model
    _, accuracy = classifier.evaluate( X_test, y_test, batch_size=batch_size, verbose=0)
    return history, accuracy, classifier

def run_model(model, grid_result):
    history, score, classifier = evaluate_model(X_train, y_train, X_test, y_test, grid_result.best_params_,
model)
    score = score * 100.0
    print('> %.3f % (score))
    results=pd.DataFrame(history.history, index=history.epoch)
    return results, classifier

def plot_loss(results):
    plt.style.use('seaborn-whitegrid')
    sns.set(style="white", font_scale = 1.5)
    fig, axes = plt.subplots(1,2, figsize=(20,5), sharex=True)
    axes[0].set_title('Loss')
    axes[0].plot(results['loss'], label='Train', )
    axes[0].plot(results['val_loss'], label='Validation')
    axes[0].legend()

#base model
X_train, y_train, X_test, y_test = load_dataset(prefix="data/")
n_timesteps, n_features, n_outputs = X_train.shape[1], X_train.shape[2], y_train.shape[1]
y_train.shape

# LSTM model
def model1():
    seed(17)

```

```

set_seed(17)
K.clear_session()
model = Sequential()
model.add(LSTM(64, input_shape=(n_timesteps,n_features)))
model.add(Dropout(0.1))
model.add(Dense(64, activation='relu'))
model.add(Dense(n_outputs, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer="adam", metrics=['accuracy'])
return model
classifier1=run(model1)
# CNN model
def model2():
    seed(17)
    set_seed(17)
    K.clear_session()
    model = Sequential()
    model.add(Conv1D(filters=64, kernel_size=3, activation='relu',input_shape=(n_timesteps,
n_features)))
    model.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
    model.add(Dropout(0.5))
    model.add(MaxPooling1D(pool_size=2))
    model.add(Flatten())
    model.add(Dense(100, activation='relu'))
    model.add(Dense(n_outputs, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
classifier2=run(model2)
# CNN-LSTM model
n_steps, n_length = 4, 25
X_train = X_train.reshape((X_train.shape[0], n_steps, n_length, n_features))
X_test = X_test.reshape((X_test.shape[0], n_steps, n_length, n_features))
def model3():
    seed(17)
    set_seed(17)
    K.clear_session()
    model = Sequential()
    model.add(TimeDistributed(Conv1D(filters=64, kernel_size=3, activation='relu'),
input_shape=(None,n_length,n_features)))
    model.add(TimeDistributed(MaxPooling1D(pool_size=2)))
    model.add(TimeDistributed(Dropout(0.5)))
    model.add(TimeDistributed(Conv1D(filters=128, kernel_size=3, activation='relu')))
    model.add(TimeDistributed(MaxPooling1D(pool_size=2)))
    model.add(TimeDistributed(Dropout(0.5)))
    model.add(TimeDistributed(Flatten()))
    model.add(LSTM(100))
    model.add(Dropout(0.5))
    model.add(Dense(100, activation='relu'))
    model.add(Dense(n_outputs, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
classifier3=run(model3)
#performance of CNN-LSTM model
def perform(model):
    display(model.summary())
    # Final evaluation of the model
    scores = model.evaluate(X_test, y_test, verbose=0)
    print("Test Loss: %f" % (scores[0]))

```

```
print("Test Accuracy: %f%%" % (scores[1]*100))
return
def confusion(model):
    y_pred=model.predict(X_test)
    # Confusion Matrix
    y_predict=pd.Series([activities[i+1] for i in np.argmax(y_pred, axis=1)])
    y_actual=pd.Series([activities[i+1] for i in np.argmax(y_test, axis=1)])
```

Appendix C: For Local data volunteers' confidentiality sign list

R.NO	ID	GENDER	AGE	SIGNATURE
P1	S1	M	26	
P2	S2	M	26	
P3	S3	F	21	
P4	S4	M	28	
P5	S5	M	27	
P6	S6	M	33	
P7	S7	M	35	
P8	S8	F	12	
P9	S9	M	28	
P10	S10	M	23	
P11	S11	M	27	
P12	S12	f	33	
P13	S13	M	25	
P14	S14	M	43	
P15	S15	f	28	

R.NO	ID	GENDER	AGE	SIGNATURE
P16	S16	M	42	
P17	S17	F	48	
P18	S18	f	54	
P19	S19	f	37	
P20	S20	f	21	
P21	S21	M	26	
P22	S22	M	25	
P23	S23	F	30	
P24	S24	M	43	
P25	S25	M	24	
P26	S26	F	37	
P27	S27	M	26	
P28	S28	f	31	
P29	S29	f	34	
P30	S30	f	25	