**Department of Computer Science**
**Memorial University of Newfoundland & Labrador**
240 Prince Philip Dr, St. John's, Newfoundland and Labrador, Canada

# Investigating Effects of Window Length on 1D-CNN-LSTM and Effectiveness of Heuristic Features in Solving Sensor Orientation and Placement Problems in Human Activity Recognition Using a Single Smartphone Accelerometer

Authored by
**Arnab Barua**

Supervised by
**Dr. Xianta Jiang**

Co-supervised by
**Dr. Daniel Fuller**

A thesis submitted to the School of Graduate Studies
to partially fulfil the requirements for the Master of
Science degree.

May 2023

# Abstract

Human Activity Recognition (HAR) using smartphone sensors can offer multiple applications in different spheres. Using deep learning classifiers such as Convolutional Neural Networks (CNN), Short-Term Long Memory (LSTM), or their hybrid showed promising improvement in HAR. However, using these deep learning networks requires segmenting the input data into multiple data windows of similar length. The length of the data windows can significantly affect HAR's performance. Therefore, the influence of the window lengths needs to be investigated to choose an optimal window length. Additionally, the orientation and placement of the smartphone sensor also present significant challenges to HAR. Many approaches have been proposed to solve the orientation and placement problems. In my study, I first evaluated the effects of window length on 1D-CNN-LSTM in HAR for six activities: Lying, Sitting, Walking, and Running at 3-METs (Metabolic Equivalent of Tasks), 5-METs and 7-METs. Subsequently, I evaluated the effectiveness of the heuristic features in HAR in solving sensor orientation and sensor placement problems for three smartphone locations: Pocket, Backpack and Hand. I performed this evaluation using 1D-CNN-LSTM by using the optimal window length found in the first part.

# General Summary

Perfecting Human Activity Recognition (HAR) using smartphone sensors can allow us to innovate multiple essential functions for our daily lives. The smartphone contains multiple sensors, such as an accelerometer, gyroscope, magnetometer, etc. The sensors can generate different signal values and patterns for the same activities if the smartphone is fixed at constant orientation and location. Consequently, researchers conducted many studies to remove the influence of sensor orientation and locations on smartphone sensors. One of these approaches proposed heuristic features to solve these problems. They evaluated the features' effectiveness for sensor orientation problems on synthetically modified datasets. In addition, the window length of the input data segment can affect the performance of deep learning classifiers used vastly in HAR. Therefore, my study explored the influence of window length and the heuristic features' effectiveness in HAR in solving sensor orientation and placement problems on a dataset replicating practical daily life scenarios.

# Co-Authorship Statement

The studies in this thesis are the outcome of the cumulative contribution of Arnab Barua, Dr Xianta Jiang and Dr Daniel Fuller. Arnab Barua conducted the primary investigation under continuous supervision and monitoring of senior researchers Dr Xianta Jiang and Dr Daniel Fuller.

The author's role and contributions in different areas are listed below:

- **Conceptualization:** Arnab Barua, Xianta Jiang and Daniel Fuller
- **Data curation:** Daniel Fuller
- **Formal analysis:** Arnab Barua
- **Funding acquisition:** Xianta Jiang and Daniel Fuller
- **Investigation:** Arnab Barua
- **Methodology:** Arnab Barua
- **Supervision:** Xianta Jiang and Daniel Fuller
- **Writing:** Arnab Barua
- **Review and editing**: Arnab Barua, Xianta Jiang and Daniel Fuller

Arnab Barua obtained approval from the senior researchers and co-authors to publish the studies and corresponding findings as part of his thesis.

# Acknowledgements

This thesis would not be possible without continuous supervision and guidelines from Dr. Xianta Jiang during my whole Master's program. I am also grateful to Dr. Daniel Fuller for his support and direction in advancing the research conducted for my thesis. I would like to thank my lab mates for their valuable feedback on my research.

The chapters in this thesis have been published or are under review in peer-reviewed journals. I acquired the necessary copyright permission from the publisher to include the publications in my thesis. Chapter 2 has been published as “Barua, A.; Fuller, D.; Musa, S.; Jiang, X. Exploring Orientation Invariant Heuristic Features with Variant Window Length of 1D-CNN-LSTM in Human Activity Recognition. Biosensors 2022, 12, 549. https://doi.org/10.3390/bios12070549”

Chapter 3 is under review in the peer-reviewed journal “Biomedical Engineering Online” with the title “The Effectiveness of Simple Heuristic Features in Sensor Orientation and Placement Problems in Human Activity Recognition Using A Single Smartphone Accelerometer”. The authors are Arnab Barua, Dr. Xianta Jiang and Dr. Daniel Fuller.

The Memorial University Interdisciplinary Committee on Ethics in Human Research granted us ethical permissions for secondary use of the dataset under application number 20230193-SC.

I thank the Lord Buddha for providing me with this opportunity and blessing me with my family, who has supported me my whole life with everything they have. I dedicate this work to them as a token of gratitude. Finally, I thank my friends and brothers, who were there for me whenever I needed them.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Overview

## 1.1. Background

Human Activity Recognition(HAR) is the process of enabling a machine to identify human activities by processing raw data accumulated in appropriate mediums. Researchers have been using various types of data in HAR, such as data from accelerometer sensors[1-3], gyroscope sensors[4-6], magnetometer sensors[7-9], images[10] and videos[11]. The advantages of using sensor units rather than images or videos are their flexibility, availability, privacy, and lower computational complexity[12]. The availability of the mentioned sensors in present-day smartphones has motivated researchers to accomplish HAR using smartphone sensors.

The two primary sensors in modern smartphones are the accelerometer and gyroscope sensors. An accelerometer sensor in a smartphone computes the smartphone's acceleration over the sensor's three axes. On the other hand, the gyroscope sensor measures the angular velocity and the smartphone's orientation. When a user keeps the smartphone attached to his body and performs different activities, the accelerometer and gyroscope sensors generate unique sensor values and patterns that can be used to recognize human activities using machine learning classifiers. However, when using smartphone sensors in HAR, researchers ran into some major problems related to the smartphone's orientation and placement[13]. Therefore, many studies have proposed different methods to eliminate these problems to perfect the HAR using smartphone sensors.

## 1.2. Problem Statement

Implementing Human Activity Recognition (HAR) using smartphone sensors requires us to consider two major scenarios. A user may keep the smartphone attached to the body in multiple places, such as the hand, pocket or backpack, and perform different human activities. In a particular location, the user can place the smartphone in multiple orientations, as depicted in Figure 1.1. The smartphone sensor may generate unique sensor values or patterns for a particular human activity depending on the location and orientation. However, for a particular activity, Researchers want sensor values to stay in a constant range and follow a uniform pattern despite the different smartphone locations or orientations of the sensors.

Figure 1.1. Possible orientations for a smartphone in a particular placement.

To address these challenges, different studies proposed distinct methods. These proposed methods include two major types of approaches. The first approach needs the sensor values to be sent into a universal reference frame from the sensor frame[14]. This approach requires a gyroscope and magnetometer sensor along with the accelerometer sensor. The other approach includes extracting meaningful features from the raw sensor signals, which can significantly reduce the influence of sensors' orientations and locations[15] [16] [17] [18-20]. In this approach, the studies used one or multiple types of sensors. Some studies followed this approach using a single smartphone accelerometer, which could reduce the time complexity as data volume was low. Researchers investigated different extracted features, including time-domain, frequency domain, statistical, and heuristic features. To achieve an orientation- and location-independent HAR system, studies had to extract numerous time-domain, frequency-domain, and statistical features. The number of extracted features multiplies with the number of sensors used.

Along with the proposed approaches for solving orientation and location problems, researchers had to carefully select proper machine learning classifiers with proper parameter tuning for an efficient HAR system. Generally, two types of machine learning classifiers are used in HAR: deep learning classifiers and simple machine learning classifiers. Deep learning classifiers include Convolutional Neural Networks (CNN) [21] [22] [23] [24-26], Recurrent Neural Networks(RNN) [27-29], and their different variants or hybrids. Among the different variants of RNN, Long Short-Term Memory (LSTM) has been utilized by researchers in recent studies. Simple or conventional machine learning refers to classifiers such as Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Random Forest, Decision Tree, etc. Between these two types of classifiers, deep classifiers gained comparatively more popularity with the advancement of computational resources and their high efficiency. Among the deep learning classifiers, CNN, LSTM and their hybrid were found to be sufficiently effective for an improved HAR system[30] [31] [32]. When using these deep classifiers, the input data needs to be segmented into fixed-length data windows. However, previous studies for HAR rarely investigated the influence of window lengths on the performance of these classifiers in HAR and chose the window length arbitrarily. Choosing a proper window length for the deep classifiers can improve the accuracy of the HAR system, and investigating the effects of window lengths can help other researchers choose the window length in future work more efficiently.

In addition, the dataset and validation procedure also play an important role in bringing out proper findings in HAR. Most publicly available datasets had very few samples[33] [34] [35] that do not offer sufficient variations to replicate the real-life scenario. The scenario is the same for the other non-public datasets used in HAR. Besides, the validation procedure also needs to be appropriately selected to evaluate the credibility of a HAR system in a practical scenario. Previous HAR studies generally used validation techniques such as the holdout method[36-40], K-Fold cross-validation[41, 42], and Leave-N-out cross-validation[43, 44]. Due to data leakage, these validation techniques, except for Leave-N-out cross-validation, can not offer many challenges to the classifiers. The classifiers must be tested using unseen participants' data for proper evaluation. To accomplish this, some studies performed Leave-One-Subject-Out cross-validation, but still, it should not be enough to assess the classifier and methods in a practical scenario.

Considering the above-mentioned issues, there should be studies regarding HAR with simple features with low sensor units and evaluating the effectiveness in both intra-location and inter-location scenarios. Besides, the influence of window lengths on deep classifiers in HAR needs to be investigated. Finally, we need to replicate the real-life scenario during the evaluation of the HAR system.

## 1.3. Literature Review

A lot of studies have proposed different methods to improve the HAR system using smartphone sensors to tackle the typical challenges. These studies included different types of sensors, numbers of sensor units, machine learning models, smartphone orientations, placements and activities. Studies acquired different results depending on the sensor types and the number of sensor units available in different public HAR datasets accumulated from smartphones.

One of the most popular public datasets is the WISDM [35]. For this dataset, data was collected from a single smartphone accelerometer while 29 participants carried the smartphone in their front pant leg pockets and performed walking, jogging, ascending stairs, descending stairs, sitting, and standing. The publisher of the dataset used decision trees (J48), logistic regression and multilayer neural networks with the help of extracted features to record the activity recognition accuracy for the dataset. They achieved the highest accuracy of 91.7% using neural networks. Another popular public dataset is the UCI-HAR [34] dataset. For this dataset, the publisher collected data from 30 volunteers who performed six daily activities in two trials. In the first trial, the smartphone was mounted on the waist, whereas in the second trial, the volunteers kept the phone as they preferred. The collected data was raw signals from the smartphone's accelerometer and gyroscope sensors. They primarily used an SVM classifier and achieved an accuracy of 96% using 561 extracted features. Another comparatively new dataset was published in [45]. They collected a huge dataset containing almost 29 million samples collected from 19 participants. The dataset contained signals from the smartphone's accelerometer, gyroscope, magnetometer and GPS for four generalized activities related to smartphone conditions. Besides, they filtered the accelerometer data to keep it safe from the influence of smartphone rotation. They used an SVM classifier and acquired an accuracy of 69.28%. There are some other public datasets for HAR, but the data was not collected from the smartphone. Many studies used these public datasets with different methods and reported different results. The studies mentioned above suggest that different datasets used different types and a different number of sensor units. It also seems that the number of extracted features also increased with an increase in sensor units.  However, the datasets rarely introduced smartphone orientation and placement issues. Still, many studies considered these issues and proposed different approaches to solve them.

Previous studies used two major approaches to eliminate the sensor orientation and placement problem. The first approach mainly deals with the problem by transforming raw signals from the smartphone to a reference frame from the sensor frame, which we can also refer to as coordinate transformation. Some of the studies which used these approaches are [46], [47], [48]. However, for the coordinate transformation approach, the studies always required gyroscope or magnetometer signals along with accelerometer signals. These studies will be further discussed in Chapter 3 with detailed descriptions.  The second approach is extracting orientation-invariant features. In this approach, studies extracted different types of feature sets, such as time-domain, frequency domain, statistical and heuristic features. [18] extracted 31 time and frequency domain features from the raw signal of a single smartphone accelerometer to implement a HAR system for recognizing five human activities: walking, limping, jogging, walking upstairs, and walking downstairs. They placed the smartphone near the waist in arbitrary orientations and accumulated data from only 3 participants. They used both passive and active learning classifiers to reach a classification rate of 84.40%. [19] extracted magnitude norm vector and rotation feature (pitch and roll angles) from raw signals of a tri-axial accelerometer of a smartphone for classifying 13 human activities. They used a deep stacked encoder to extract further feature representation and acquired a 97.13% accuracy using a combination of extracted features with raw signals. For their dataset, they collected about 1 million samples from 10 participants fixing the smartphone in two locations: wrist and pockets. For their dataset using only the raw signals, they achieved 92.92% accuracy. It seems that the raw signals could achieve higher accuracy in their study, and one possible reason could be their validation method. They used a holdout validation method, which is prone to data leakage. Another study extracted 21 features from the raw signals of a single smartphone accelerometer for a HAR system capable of classifying five human activities. They collected their dataset from 8 participants, and they had the freedom to keep the phone in their right or left trouser pocket. They performed online and offline activity recognition using KNN and Quadratic Discriminant Analysis (QDA) and achieved accuracies of about 93%-95% for all cases. Along with the previously mentioned studies, many other available studies [15] [16] used extracted features for implementing a HAR system. One of the important studies to follow for my thesis is [17]. This study introduced nine simple orientation-invariant heuristic features and evaluated their effectiveness on five publicly available datasets. The features they introduced were simple, but they did not evaluate their performance on datasets with practical orientations and placement problems. They also did not evaluate the performance of heuristic features in solving placement problems. I could have used the first approach, but it was not applicable as I used a single accelerometer sensor. I preferred this study to others for the second approach because other studies used many extracted features to solve the orientation and placement problem. However, choosing an effective approach alone cannot make a HAR system accurate. Researchers should also choose their classification algorithm and parameters wisely.

Initially, previous studies used simple machine learning algorithms like SVM [18, 49-51], KNN [52, 53], Decision tree [54, 55], Random forest [56-58], etc in HAR. They were used because of their simplicity, and those algorithms offered less computational complexity. However, advancements in computational resources have helped the studies use deep learning classifiers such as CNN, Recurrent Neural Network (RNN), LSTM, Gated Recurrent Unit (GRU), etc. Moreover, these deep classifiers had additional capabilities to enhance the performance of the classification system. For instance, CNN can extract compact feature representations, and LSTM or GRU can maintain a flow of old information hidden in the time series data. Due to these additional capabilities, new studies chose CNN [21] [22] [23] [24-26] or LSTM [59-61] for their HAR systems. Furthermore, studies started to combine the abilities of CNN and LSTM by hybridizing them into a single model and achieved better performance in HAR [30] [31] [32] [62-65]. However, in almost all studies, the researchers did not evaluate one significant parameter of CNN, LSTM, or their hybrid (CNN-LSMT), which was the window length. Deep classifiers such as CNN or LSTM need windows of data as input. A data window contains multiple data samples. Depending on the data window length, the models' performance can vary. Again, choosing a very high window length increases computational complexity, whereas a low window length can degrade the model's performance. So, choosing the correct window length can significantly affect a model’s performance. For HAR systems, previous studies rarely investigated the influence of the window length on the performance of these deep classifiers. Therefore, in my thesis, another major investigation was the influence of window length on performance of 1D-CNN-LSTM in HAR. I choose the hybrid CNN-LSTM by seeing its effectiveness in earlier studies. Besides choosing the correct classification algorithms, researchers also need to prepare a proper dataset that would have required variations and samples to make the HAR process more practical. In addition, studies should also choose a validation method that would help to assess the performance of the HAR process in real-life scenarios..

The size of a HAR dataset and variations in it can significantly impact the models' performance. Some datasets contain very few samples from a few participants and fewer variations to replicate the real-life scenario. For instance, this dataset [33] only had 9120 feature vectors from 8 participants, although it contained data for 19 activities. Another vastly used dataset had only 165,633 samples from 4 participants for five activities. The UCI-HAR [34] and WISDM [35] datasets mentioned before only had 10299 samples and 5424 samples, respectively. A relatively new dataset: KU-HAR [66] only had 1945 samples, although they accumulated data from 90 participants and 17 activities. I went through many other studies, where I noticed that the dataset contained very few samples which could not offer many variations. Considering the number of samples in previous studies, I used a dataset containing about 12 million smartphone accelerometer samples from 42 participants for six activities by keeping the smartphone in three different places. In addition, I considered the validation method adopted by previous studies. Many studies in the HAR domain used various validation procedures such as the Holdout method [36-40], K-Fold cross-validation method [41, 42], Leave-N-Subject-Out Cross-Validation Method [43, 44] etc. Among these methods, the Holdout and K-Fold cross-validation methods are unsuitable for evaluating the HAR system. When these methods are used, random samples are taken from the primary dataset for training and testing. As a result, the training and test data may contain samples from the same participant. But in real life, the test data remains unseen to the model. Therefore, these methods do not replicate the real-life scenario. A proper validation method is the Leave-N-Subject-Out Cross-Validation Method, in which the test dataset contains a dataset of some participants unseen to the trained model. Few studies adopted this procedure in their study, but they mostly did Leave-1-Subject-Out Cross-Validation probably because of a low number of participants, which means the test dataset would contain data from only one participant. If the test dataset contains more participants' data, I can evaluate the model's performance more practically. Considering the above discussion, I followed a Leave-10-subject-Out Cross-Validation method in which the test dataset contained data from 10 participants unseen to the trained model. Furthermore, I also performed intra-location and inter-location validation methods. Previous studies used the terms "location-dependent" and "location-independent" instead of "intra-location" and "inter-location", respectively. In the intra-location method, the model was trained and tested using data from the same smartphone location, whereas, in the inter-location, the model was trained using data from one smartphone location but tested using data from other smartphone locations. Therefore, I could evaluate the heuristic features' effectiveness in intra-location and inter-location scenarios.

## 1.4. Thesis Work

In this thesis, I performed two major analyses. Firstly, I investigated the effects of window length on the performance of 1D-CNN-LSTM in HAR. In my second analysis, I inspected the effectiveness of heuristic features in HAR for intra-location and inter-location scenarios, where the same 1D-CNN-LSTM model was used. My first analysis aimed to find a suitable range for window length, which would be appropriate for a better performance from the model and reduce the resource and time complexity. I used that suitable window length in my second analysis to train the 1D-CNN-LSTM model when inspecting the effectiveness of the heuristic features. In addition to this major investigation, I also emphasized some other issues. For instance, I used a Leave-10-Subject-Out Cross-Validation technique in my second analysis to make the evaluation process more realistic and complex for the HAR system. Furthermore, fusing my adapted cross-validation approach with inter-location analysis made the evaluation procedure more challenging.

In Chapter 2, I investigated the influence of the window length on 1D-CNN-LSTM in HAR for only one smartphone location: pocket. I extracted the heuristic features to solve the orientation problem. I trained and tested the 1D-CNN-LSTM model using the heuristic features for 20 different window lengths starting from 5 and ending at 195 with a gap of 10. I followed the Leave-1-Subject-Out Cross-Validation Procedure. I inspected the effects of window length on the models' overall performance. I also analysed the effects of window length on the model's performance for individual participants and activity classes. This investigation revealed that, up to a specific window length, the model's accuracy increases as the window length increases. After that certain window length, the model's performance stabilises. So, rather than choosing a long window length, we should choose one after which the model's performance becomes steady.

In Chapter 3, I focused on investigating the effectiveness of the heuristic features for different smartphone locations. The heuristic features were generally proposed for solving the orientation problem, but I also wanted to observe how well they were in solving the placement problem. By placement problem, I mean that if the HAR system is trained using data for one smartphone location, such as a pocket, what will be its performance if the test data comes from another smartphone location, such as a hand or backpack? This study has the important implication that the model would be location-invariant in real-life applications. Besides, in the study where the heuristic features were introduced, the researchers did not evaluate the heuristic features' effectiveness in solving the orientation using data with practically introduced orientation. In my study, I first evaluated the efficiency of the heuristic features in solving the sensor orientation problem for three smartphone locations: pocket, hand and backpack. I called it intra-location evaluation. In this part, a 1D-CNN-LSTM model was trained and tested using data from the same smartphone location. So, I could observe the performance of the heuristic features in solving orientation problems for three smartphone locations. I had 42 participants who performed six activities: lying, sitting, walking, and running at 3-METs, 5-METs and 7-METs by keeping three smartphones in three different locations/locations: pocket, backpack and hand. I collected data from the accelerometer signal of the smartphone. For intra-location evaluation, I followed the Leave-10-Subject-Out Cross-Validation technique. I depicted three types of results: overall performance, participant-specific performance and activity-specific performance. I had a similar overall performance for every smartphone location. However, the performance was better for the location “Backpack” as the smartphone stayed less shaky in the backpack than the location of the pocket and hand.

I conducted an inter-location evaluation to inspect the effectiveness of the heuristic features in solving the placement problem. In inter-location evaluation, I trained the model using heuristic features extracted from the raw signal of one smartphone location. I tested the model using the heuristic features extracted from the other two smartphone locations. I followed the same Leave-10-Subject-Out Cross-Validation technique with the 1D-CNN-LSTM model. I depicted the results as I depicted for the intra-location evaluation. The performance of the heuristic features was lower than the intra-location evaluation results. However, considering the difficulties offered, such as the use of only one single accelerometer’s data, Leave-10-Subject-Out Cross-Validation and simple heuristic features, the results seemed promising. It should be mentioned that, in Chapter 3, all the models were trained using a suitable window length, which I found in Chapter 2.

To summarize, I investigated the influence of window lengths on the 1D-CNN-LSTM model’s performance to select the optimal window length. Using the optimal window length, I trained all the models to inspect the effectiveness of heuristic features in intra-location and inter-location scenarios. I found that the heuristic features are considerably effective in solving orientation problems. With further analysis, I can also achieve a better result using the heuristic features for inter-location evaluation.

## 1.5. Contributions

I prepared the report for this thesis in "Manuscript Style". The work in this thesis is published and under review in peer-reviewed journals. I have the appropriate permission from the publisher to use the findings in my thesis. The published or under-review versions are included as Chapter 2 and Chapter 3 as follows:

- Chapter 2 has been published as "Barua, A.; Fuller, D.; Musa, S.; Jiang, X. Exploring Orientation Invariant Heuristic Features with Variant Window Length of 1D-CNN-LSTM in Human Activity Recognition. *Biosensors* **2022**, *12*, 549. https://doi.org/10.3390/bios12070549". In this study, we explored the influence of window lengths on the performance of the 1D-CNN-LSTM model in HAR.
- Chapter 3 is under review in the journal "Biomedical Engineering Online", titled "The Effectiveness of Simple Heuristic Features in Sensor Orientation and Placement Problems in Human Activity Recognition Using A Single Smartphone Accelerometer". The authors are Arnab Barua, Dr. Daniel Fuller and Dr. Xianta Jiang.

It is important to note that due to the manuscript style of this thesis, there may be instances of repeated text between chapters 2 and 3.

# Chapter 2

# Exploring Orientation Invariant Heuristic Features with VariantWindow Length of 1D-CNN-LSTM in Human Activity Recognition

## 2.1. Introduction

Human Activity Recognition (HAR) has allowed for the implementation of distinct applications such as user identification[67], health monitoring[68], identifying the early stage of depression [69], fall detection[70] and more. Improving these applications requires ongoing methodological development. Researchers have conducted many studies to improve HAR by introducing the recognition of various daily activities using distinct approaches that include non-identical machine learning algorithms. Improving HAR requires considering some inevitable challenges, which involve sensor orientation, sensor location, device independency, study sample length, and data volume [71-74]. Among the mentioned challenged the most signifiacnt issue to solve is problem of sensor orientation and location.

For solving the orientational and locational problem due to sensor placement, different studies introduced techniques such as transforming the sensor signals to a universal frame, extracting orientation-invariant features from raw signals, removing orientation and location-specific information by introducing statistical alterations, estimating the orientation of the sensor to the earth frame by using the tri-axial sensors (accelerometer, magnetometer and gyroscope) and then transforming the raw signals from sensor frame to the earth frame. Using the earth frame transformation approach, [14] achieved an average accuracy of 86.4% in recognizing 19 activities using Support Vector Machine (SVM). [17] introduced heuristic orientation invariant transformation and singular value decomposition-based transformation to tackle sensor orientation problems. They evaluated their approaches using 4 different classifiers on 5 distinct datasets. They found out that their proposed approaches can reduce the accuracy drop by a considerable margin compared to other state-of-the-art approaches. [15] decomposed the accelerometer signal into horizontal and vertical acceleration. They extracted 9 features from the tri-axial gyroscope sensor and horizontal and vertical acceleration signals to solve the location and orientation dependency problem. They acquired an accuracy of 91.27%, employing SVM to recognize 5 activities using the data from a smartphone in 4 different locations. For our study, we decided to utilize the features proposed by [17] since it requires extracting 9 simple features to eliminate the variation in data produced from sensor orientation and location. Along with the orientational and locational dependency obstacles, we should also consider the number of participants and activities appraised in former studies.

As the number of participants and activities varies, distinct variations in sensor signals appear due to the differences in the participants' body attributes and the uniqueness in movements of the body parts for different activities. The number of participants matters, especially for studies where the inter-participant evaluation technique is accepted as the validation method. There are many publicly available datasets to work with, and they have already been used in several studies that introduced different numbers of participants and activities [75-79]. However, inter-participant evaluation for a large number of participants in the field of HAR is yet to be explored.

Regarding the employed classifiers for HAR, researchers evaluated the performance of both machine learning and deep learning algorithms. Research primarily assessed the performance of conventional machine learning algorithms such as Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbor, and Random Forest [47, 80-82]. However, with the emergence of advanced computational power, deep learning algorithms became more common in HAR. [75] evaluated the performance of Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional-LSTM, and Multilayer Perceptron (MLP) using two public datasets named UCI[83] and Pamap2[84]. They found that CNN outperformed other classifiers with 92.71% and 91% accuracy on UCI and Pamap2, respectively. [85] compared CNN with state-of-the-art classifiers for classifying six activities and showed that CNN performed better than all other classifiers using features extracted by Fast Fourier Transform (FFT) with an accuracy of 95.75%. CNN remains favoured for executing HAR because of its powerful ability to automatically extract features from raw signals using multiple filters[21]. Studies then tried to combine the feature extraction power of CNN with the LSTM's power of persisting old information about time-series data. LSTM is an upgraded version of the Recurrent Neural Network (RNN) that can preserve older information than RNN [86]. The hybrid of CNN and LSTM, also called CNN-LSTM, has been used in different HAR studies. [30] evaluated the performance of CNN-LSTM on HAR using three public datasets named UCI[83], WISDM[87], and OPPORTUNITY[88]. They achieved accuracies of 95.78%, 95.85% and 92.63% on UCI, WISDM, and OPPORTUNITY datasets, respectively, using a CNN-LSTM architecture. [89] explored distinct deep learning architectures, including CNN-LSTM with their proposed margin-based loss function on OPPORTUNITY, UniMiB-SHAR[78], and PAMAP2 datasets. [90] ensembled three models, namely CNN-Net, Encoded-Net, CNN-LSTM and found the performance of the ensembled model superior over 6 benchmark datasets.

There are a number of different implementations of CNN for sensor data. Typically, 1-Dimensional CNN (1D-CNN) is used for accelerometer, gyroscope and magnetometer signals. An important consideration with 1D-CNN, LSTM, or their hybrid is that these methods require data windows as inputs. Each window resembles a data matrix with a fixed number of samples as rows and the features as columns. Each consecutive window may or may not overlap. 1D-CNN uses filters on each window to extract features automatically. 1D-CNN maps these internally extracted features to different activities in HAR research. However, when 1D-CNN is combined with LSTM, the internally extracted features from 1D-CNN work as inputs to the LSTM layers. These LSTM layers further process these automatically extracted features. The advantage of using a 1D-CNN-LSTM hybrid rather than using a single CNN or single LSTM is that 1D-CNN-LSTM can use the ability of CNN to extract spatial features present in the input data as well as preserve the temporal information present in the extracted spatial features using the ability of LSTM. Although a 1D-CNN-LSTM system takes more time in training than a single CNN, it should not impose any problem in the deployment of real-life applications since, in real life, pre-trained models are deployed. A detailed explanation of the working mechanism of 1D-CNN-LSTM will be given in a later section. Now that 1D-CNN works with windows of data, the length of windows can affect the performance of 1D-CNN. With a large window length, the model will have a bigger picture of the signals' nature at a particular time. In contrast, a smaller window length portrays comparatively less information regarding the signal nature at any specific time. Again, bigger windows increase the computational complexity and time complexity, whereas smaller windows keep the computational burden and processing time considerably lower. Previous studies selected the window length arbitrarily in HAR execution while using CNN, LSTM or their hybridization, or they did not provide any discussion regarding window length selection [29, 85, 91-94]. It is yet to be explored how different window length s may affect the performance of CNN, LSTM, or their hybrid models in HAR research.

Considering the shortcomings mentioned above, the orientation problem, study sample lengths, and window length considerations, this paper systematically examines these limitations using feature extraction methods and window length experiments with 1D-CNN-LSTM models. A pictorial view of the overall procedure is portrayed in Figure 2.1. Data from a study including 42

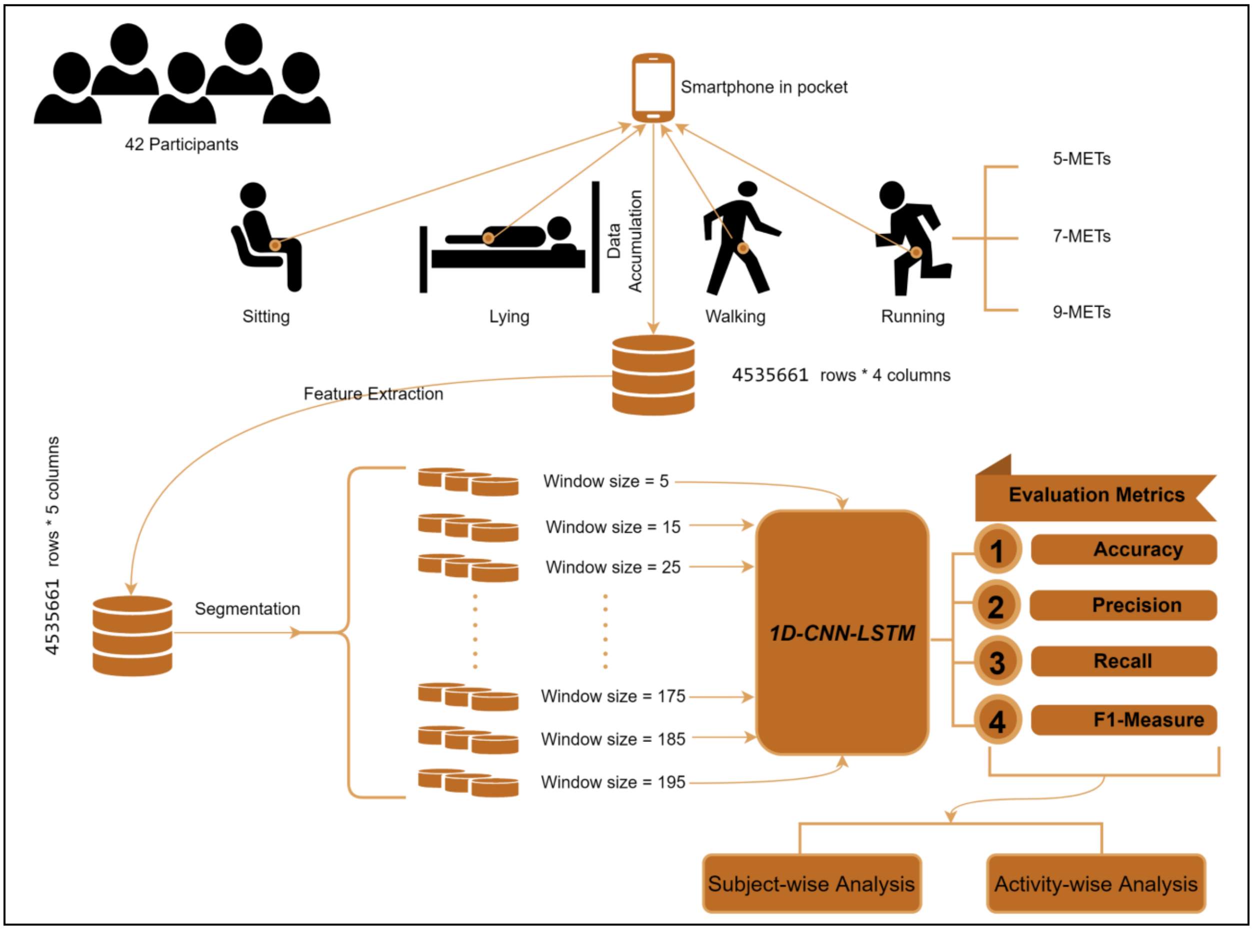


Figure 2.1. Structural diagram of our overall study.

participants performing 6 activities, namely, sitting, lying, walking, and running at 3 Metabolic Equivalent Tasks (METs), 5 METs, and 7 METs pace, were used. Data were collected using an accelerometer sensor of a smartphone carried by participants in their pockets. Specific research questions were:

- How can sensor orientation be solved?

To solve the sensor orientation problem due to the flippable locations of the smartphone in the pocket, we selected 4 orientation-invariant heuristic features from the proposed 9 heuristic features in [17].

- What is the impact of window length on model accuracy?

Results show that after a particular window length, the performance of 1D-CNN-LSTM is not influenced by the window length. Further examination explores how different window lengths influence the recognition metrics for high and low-intensity activities.

- What is the impact of the inter-participant validation method in the case of a vast number of participants?

We found that the model did not produce the same performance when evaluated using data from different participants. Still, the effects of window length on the performance of the different participants were the same.

## 2.2. Materials and Methods

In this section, we will start by discussing the data accumulation process. Then we will explain the data preprocessing step and the features we extracted to solve the smartphone orientation problem. Following, we will present the structure of the 1D-CNN-LSTM we employed in our study to discuss its effectiveness briefly. Finally, we will elaborate on the validation procedure, evaluation metrics and data reshaping process. We presented the discussion regarding the window lengths and their impact in the next section.

### 2.3.1. Data Accumulation

We acquired verbal and written consent from 42 healthy participants above 18 years to collect the required data. Before participating in the protocol, the participants completed the Physical Activity Readiness Questionnaire (PAR-Q). We acquired the necessary ethical approvals from the Memorial University Interdisciplinary Committee on Ethics in Human Research (ICEHR #20180188-EX). A summary of the demographics (gender, age, height and weight) is presented in Table 2.1. We did not include the participants' demographics as attributes in our dataset because, in another study[95], we found that these attributes did not significantly influence the performance of the machine learning models.

Table 2.1. Demographics of the participants.

| Number of participants | | Age (years) | | | Height (centimetres) | | | Weight (kilograms) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Male | Female | Average | Maximum | Minimum | Average | Maximum | Minimum | Average | Maximum | Minimum |
| 18 | 24 | 29 | 56 | 18 | 169.17 | 185 | 143 | 68.19 | 95.2 | 43 |

Each participant carried a Samsung Galaxy S7(SM-G930W8) smartphone in their pocket with a pre-installed Ethica application[96] that recorded the X, Y and Z-axis values of the Accelerometer sensor embedded in the mentioned smartphone while completing the protocol. The lab-based protocol required 65 minutes for each participant to be completed entirely. During the 65 minutes, each participant conducted the activities according to the order and for the time presented in Table 2.2. The Rank column in Table 2.2 resembles the order of the activity trial. Rank 1 means each participant started with the corresponding activity trial, and Rank 9 points to the activity trial that each participant completed at the end. Participants had the freedom to keep the smartphone in their pocket in any arbitrary orientation.

Table 2.2. Duration and order of activities performed by each participant.

| Rank | Activity | Duration (minutes) |
|---|---|---|

| | | |
|---|---|---|
| 1 | Lying down | 5 |
| 2 | Sitting | 5 |
| 3 | Walking | 10 |
| 4 | Lying down | 5 |
| 5 | Running at 3-METs | 10 |
| 6 | Lying down | 5 |
| 7 | Running at 5-METs | 10 |
| 8 | Sitting | 5 |
| 9 | Running at 7-METs | 10 |

The participants walked and ran on a treadmill set up in the lab. To measure the intensities of Running, we used the Metabolic Equivalent of Task (MET)[97], a relative measure of energy expenditure related to the participant's weight and volume of oxygen consumed per minute. We used METs rather than walking speed, cadence or stride length because we wanted to quantify the intensity of activities using energy expenditure. For the same walking speed, cadence or stride length, we may experience different energy expenditures from different participants. Furthermore, MET has been highly recommended by other studies to measure energy expenditure. We ensured that the participants ran at a particular intensity (MET) by setting up the intensity level on the treadmill. In the treadmill, there were options to select a particular intensity, such as 3-METs, 5-METs, or 7-METs. Equation 1 defines the calculation process of METs.

$$METs = \frac{Oxygen\ Cosumption\ rate\ (\frac{ml}{\text{minute}})}{3.5\ \times weight\ (kg)}$$

Here $\boldsymbol{ml}$ is a unit of volume of oxygen that stands for millilitre, and kg stands for Kilogram, which is a unit for measuring weight.

The reason for choosing the aforementioned activity types and activity intensities was that these were reported to be the most common type of activities included in former studies[98]. Besides, walking or running with different intensities was overlooked in most of the former studies.

Therefore we focused on studying the effects of window length for 1D-CNN-LSTM on HAR for common types of activities and intensities.

### 2.3.2. Data Preprocessing and Feature Extraction

We used programming languages R 3.6.1 and Python 3.9.7 to execute the required preprocessing of the data and extract heuristic features, respectively. We used Python packages named Pandas 1.3.4 and Numpy 1.20.3 for performing feature extraction.

### 2.3.3. Data Resampling and Data Imputation

The Ethica App could not accumulate the sensor data in a constant frequency; rather, the frequency varied from 5Hz to 19Hz. The reason behind this varying frequency was the application's forced optimization technique to keep the app running by constraining the amount of data uploaded to the server. Because of this varying frequency, each activity class had a very different number of observations, although they should have an almost similar amount of observations. We upsampled the data to a constant frequency of 30Hz by using the resampling method introduced in [33] to eliminate this data imbalance. In addition, we also experienced missing values in our accumulated data. This problem happened due to the momentary connection loss between the Ethica App and the server. To get rid of the problem regarding these missing values, we conducted data imputation. For performing linear imputation, we used the R package named ImputeTS. After data resampling and imputation execution, we had the following amount of data points for each activity presented in Table 2.3.

Table 2.3. Number and ratio of samples for each type of activity class.

| Activity | Number of data points | Ratio to total dataset |
|---|---|---|
| Running 7 METs | 926606 | 21.43% |
| Running 5 METs | 812135 | 18.78% |

| | | |
|---|---|---|
| Running 3 METs | 815498 | 18.86% |
| Self Pace walk | 609406 | 14.09% |
| lying | 696329 | 16.10% |
| sitting | 464559 | 10.74% |

### 2.3.4. Feature Extraction and Selection

We extracted some suitable heuristic features to resolve the sensor orientation problem. As mentioned earlier, during the data collection phase, the participants had the freedom to keep the mobile phone in their pocket in any arbitrary orientation. Therefore, different participants might perform the same activity trial while keeping the smartphone in a non-identical orientation.

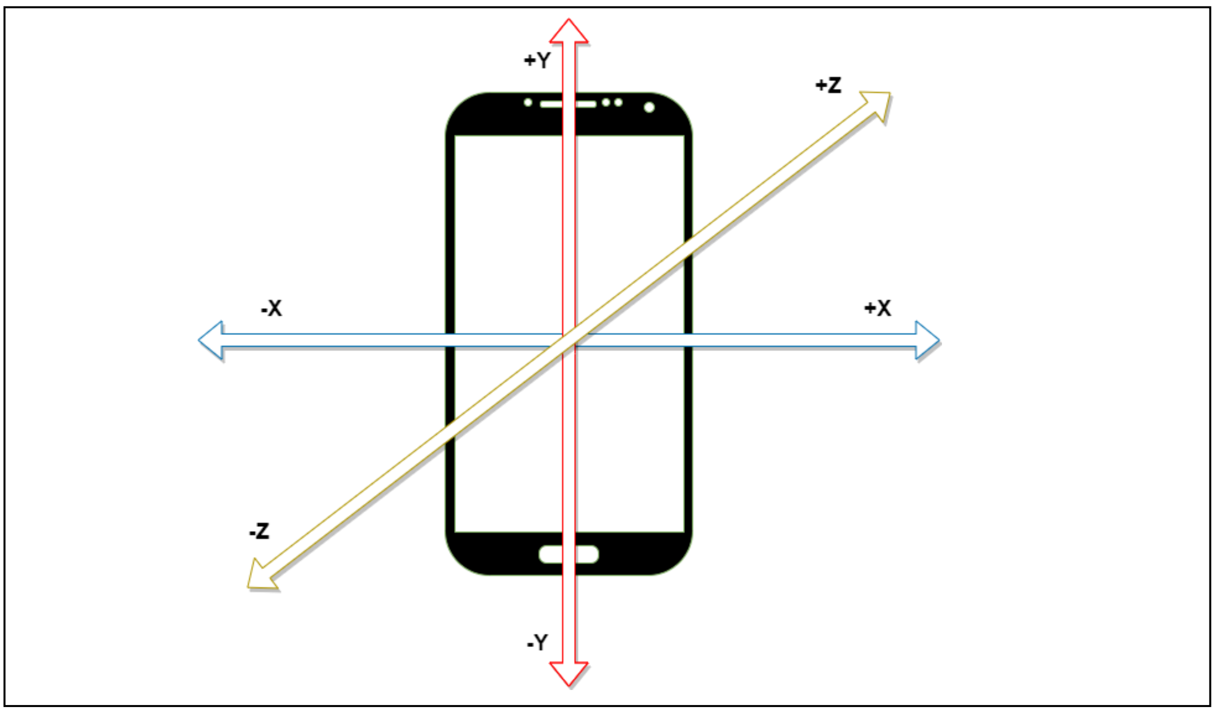


Figure 2.2. The direction of three axes of an accelerometer sensor in a smartphone.

If we observe Figure 2.2, we can see the direction of the axes of the accelerometer with respect to the smartphone. The X-axis goes from left to right of the smartphone screen, Y-axis goes from top to bottom, and Z-axis goes perpendicularly through the screen. As we orient the smartphone, the axes also get oriented accordingly. As a result, we observed different Accelerometer X, Y and Z-axis values for the same activity if the users kept the smartphone in different orientations while performing the same activity.

If we observe Figure 2.3, we can see how the values of the axes of the accelerometer differ in values while different participants ran at a speed of 9 METs. To solve this problem, we extracted 4 simple heuristic features which were proposed in [17]. This study originally proposed 9 sensor invariant heuristic features. They defined each data vector of the accelerometer as $\overrightarrow{v_n} = (v_x[n], v_y[n], v_z[n])$ where $v_x[n]$, $v_y[n]$, $v_z[n]$, were values of the accelerometer x-axis, y-axis and z-axis, respectively, at any time-sample n. They also defined first-order and second-order time differences as $\Delta\overrightarrow{v_n} = v_{n+1} - v_n$ and $\Delta^2\overrightarrow{v_n} = v_{n+1} - v_n$ respectively. The equations for computing the 9 heuristic features are given below,

$$w_1[n] = \|\overrightarrow{v_n}\| \quad (1)$$

$$w_2[n] = \|\Delta\overrightarrow{v_n}\| \quad (2)$$

$$w_3[n] = \|\Delta^2\overrightarrow{v_n}\| \quad (3)$$

$$w_4[n] = \angle(\overrightarrow{v_n}, \overrightarrow{v_{n+1}}) \quad (4)$$

$$w_4[n] = \angle(\Delta\overrightarrow{v_n}, \Delta\overrightarrow{v_{n+1}}) \quad (5)$$

$$w_4[n] = \angle(\Delta^2\overrightarrow{v_n}, \Delta^2\overrightarrow{v_{n+1}}) \quad (6)$$

$$w_7[n] = \angle(\overrightarrow{p_n}, \overrightarrow{p_{n+1}}) \text{ where } \overrightarrow{p_n} = \overrightarrow{v_n} \times \overrightarrow{v_{n+1}} \quad (7)$$

$$w_8[n] = \angle(\overrightarrow{q_n}, \overrightarrow{q_{n+1}}) \text{ where } \overrightarrow{q_n} = \Delta\overrightarrow{v_n} \times \Delta\overrightarrow{v_{n+1}} \quad (8)$$

$$w_9[n] = \angle(\overrightarrow{r_n}, \overrightarrow{r_{n+1}}) \text{ where } \overrightarrow{r_n} = \Delta^2\overrightarrow{v_n} \times \Delta^2\overrightarrow{v_{n+1}} \quad (9)$$

Here,

$$w_t = extracted\ heuristic\ features\ for\ t = 1\ to\ 9$$

$$\|\vec{m}\| = Euclidean\ norm\ of\ vector\ m$$

$$\angle(\vec{a},\vec{b}) = \ cos^{-1}\left(\frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\ \|\vec{b}\|}\right) = angle\ between\ vector\ a\ and\ vector\ b\ where\ \vec{a}\ \cdot \vec{b}\ denotes\ their\ dot\ product$$

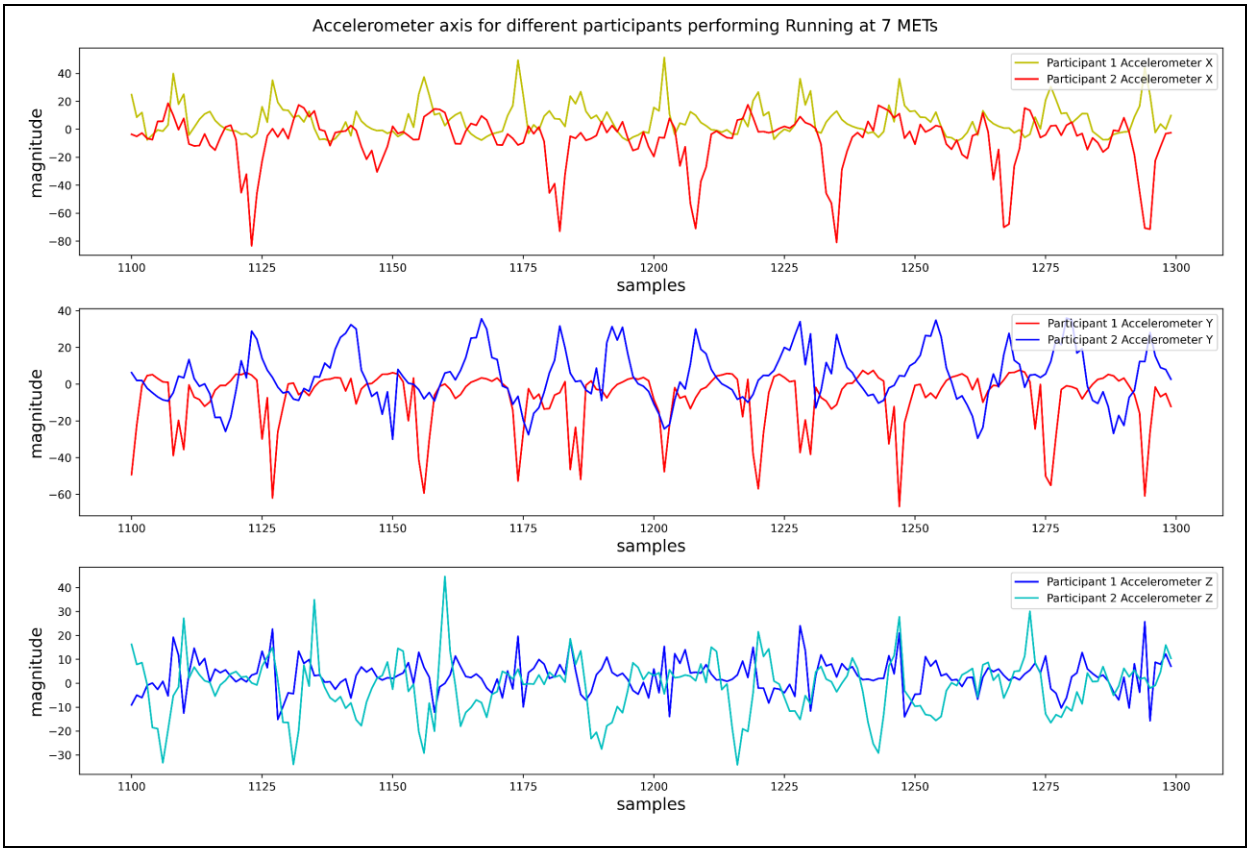


Figure 2.3. Values of three accelerometer axes for the same activity performed by two different participants.

They claimed these 9 heuristic features to be irresponsive to the orientation of the sensor and mathematical elaborated on the reason behind being invariant to the orientation. Further analysis can be found in [17]. Although they examined the performance of all 9 features in HAR execution, we conducted a more detailed analysis of these features. We found that the first 4 features $w_1, w_2, w_3$ and $w_4$ are most important and distinguishable for different activities. In Figure 2.4, we can observe the pattern and magnitude range of the first 4 heuristic features for two participants performing the same running activity at a speed of 7 METs. We can observe some similarities in the pattern and magnitude range of 4 heuristic features.


Figure 2.4. Values of the first four heuristic features for activity running at 9 METs from two different participants.

In Figure 2.5, we plotted the values of the first 4 heuristic features for different participants performing two different activities named lying and running at a speed of 7 METs. We can observe a considerable difference in the patterns and magnitude of the first 4 features for two different activities performed by two different participants. That means the first 4 heuristic features showed similarities in their values for the same activity and dissimilarities for different activities, which is essential for any classifier to distinguish them.

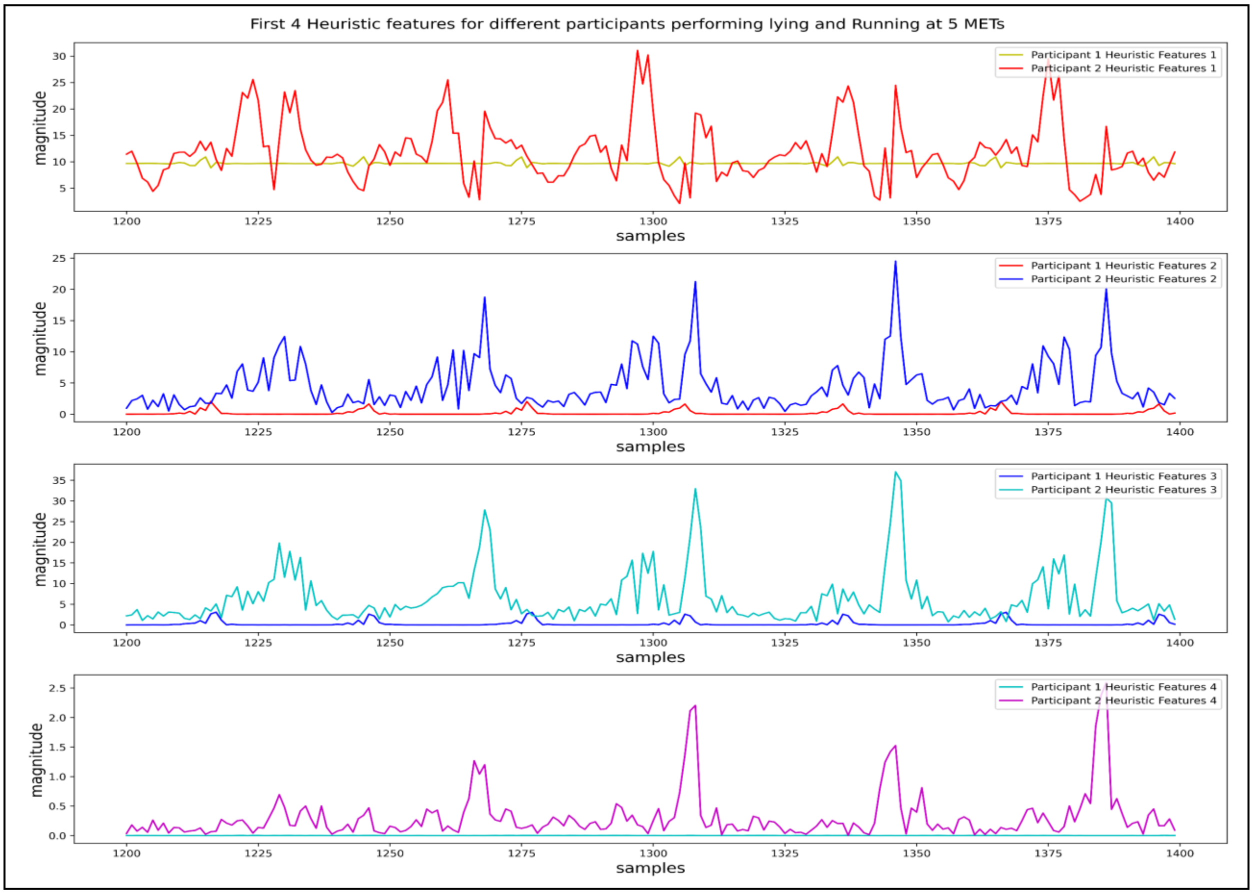


Figure 2.5. Values of the first four heuristic features for two different activities from two different participants.

In addition, we computed the feature importance of all the 9 features using the two classifiers named Decision Tree (DT) and the Random Forest (RF). Here feature importances define how impactful a feature is to a classifier to make a prediction. We chose these two classifiers for computing feature importance because they are proven to be very effective. There are different versions of DT. We used the version named Classification and Regression Tree (CART). DT and

RF have two metrics to decide the importance of features, named Gini Impurity and Information Gain.

Our study used both as metrics to calculate the feature importance. Although RF uses multiple decision trees to compute its result and is comparatively better than DT, there are still differences between DT and RF regarding how they use the whole dataset. Therefore, we wanted to compute feature importance using both classifiers because of their different nature. For functioning these classifiers in Python 3.8.10, we used their implementation provided in the package named scikit-learn 0.22.1. We feed the whole dataset to both classifiers to calculate the feature importance. Feature importance for the 9 features is depicted graphically in Figure 2.6.

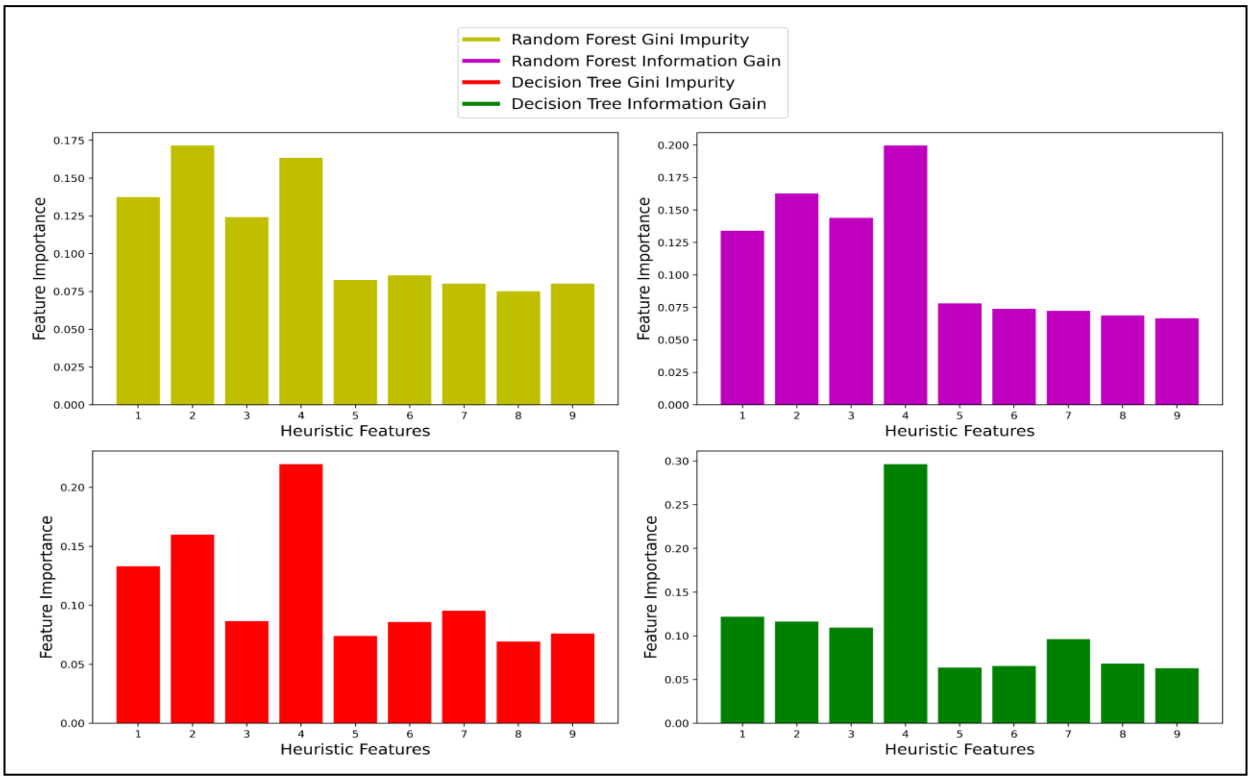


Figure 2.6. Feature importance of the 9 heuristic features.

From Figure 2.6, we can see that the first four features had greater feature importance than the last 5 heuristic features when we used RF using Gini impurity and information gain to compute the feature importance. In the case of the DT, it assigned greater feature importance for the first 4 heuristic features using Gini impurity and information gain; it also assigned considerable importance for the 7th feature. However, considering all four scenarios of feature importance, we elected to use the first four features as they were assigned greater feature importance in all cases. We also wanted to find if scaling the data had an impact on feature selection and which is why we computed the feature importance after scaling the features between 0 and 1. The result did not change. Even after scaling, the first 4 heuristic features had superior feature importance. Since we used the raw heuristic features for our study, we did not present the feature importance found after scaling the data in this paper.

### 2.3.5. The architecture of 1D-CNN-LSTM

A conventional CNN consists of an input, convolution, pooling, fully connected, and output layer. The input layer takes the data matrix as input. A data matrix encapsulates a portion of the data. The convolution layer consists of multiple filters, where each filter is also a matrix with lower dimensions than the feed data matrix. Each filter can move on the input data matrix in two directions or one direction for 2D-CNN and 1D-CNN, respectively. Each filter performs a convolution operation and constructs a feature map when moving on the input data matrix. A convolution layer with n number of filters constructs n number of feature maps for a single data matrix. We can define a feature map as a representation of the original data matrix with equal or lower dimensions but concentrate on prioritizing a particular feature of the data matrix. The pooling layer reduces the length of feature maps by moving averaging filter or max filter on them. We can feed these feature maps into an LSTM model. An LSTM model[86] can have one or more

LSTM layers. Each LSTM layer consists of multiple LSTM cells. Each cell encapsulates three gates named Forget gate, Input gate and Output gate. The forget gate is responsible for removing unnecessary information from the previous time step, where the time step resembles a row in a data matrix. The input gate processes the current data fed into the cell, and the output gate generates the output to be combined with the next input to the LSTM cell. In our hybrid 1D-CNN-LSTM[99], we had Convolution and pooling layers followed by an LSTM and fully connected layers, as depicted in Figure 2.7. The final pooling layer generates final feature maps fed into LSTM cells. The LSTM cells then consider each feature map as a timestep and try to learn from it and propagate the information to use it in processing the next feature map (time step). The LSTM layer's output then goes to the fully connected layer, composed of conventional neurons of Artificial Neural Network (ANN). The predictive result comes out of the final fully connected layer.

Our proposed 1D-CNN-LSTM architecture started with a CNN encompassing 6 convolutional layers, 3 pooling layers and 4 dropout layers. The convolution layers used different kernel lengths and Rectified Linear Units (relu) as their activation function. We introduced dropout layers in the model to reduce the risk of overfitting. For LSTM's portion, we used an LSTM layer with 512 hidden units and tanh as its activation function. 4 fully connected dense layers followed the LSTM layer. The first 3 layers had a different number of neurons and relu as their activation function.

The last dense layer had 6 neurons with softmax as its activation function to render the probability regarding 6 types of activity. Although many former studies used the concept of the CNN-LSTM hybrid model, the elements, including the number of layers, filters, LSTM cells, neurons and dropout rate in the structure we proposed here, were determined by us.

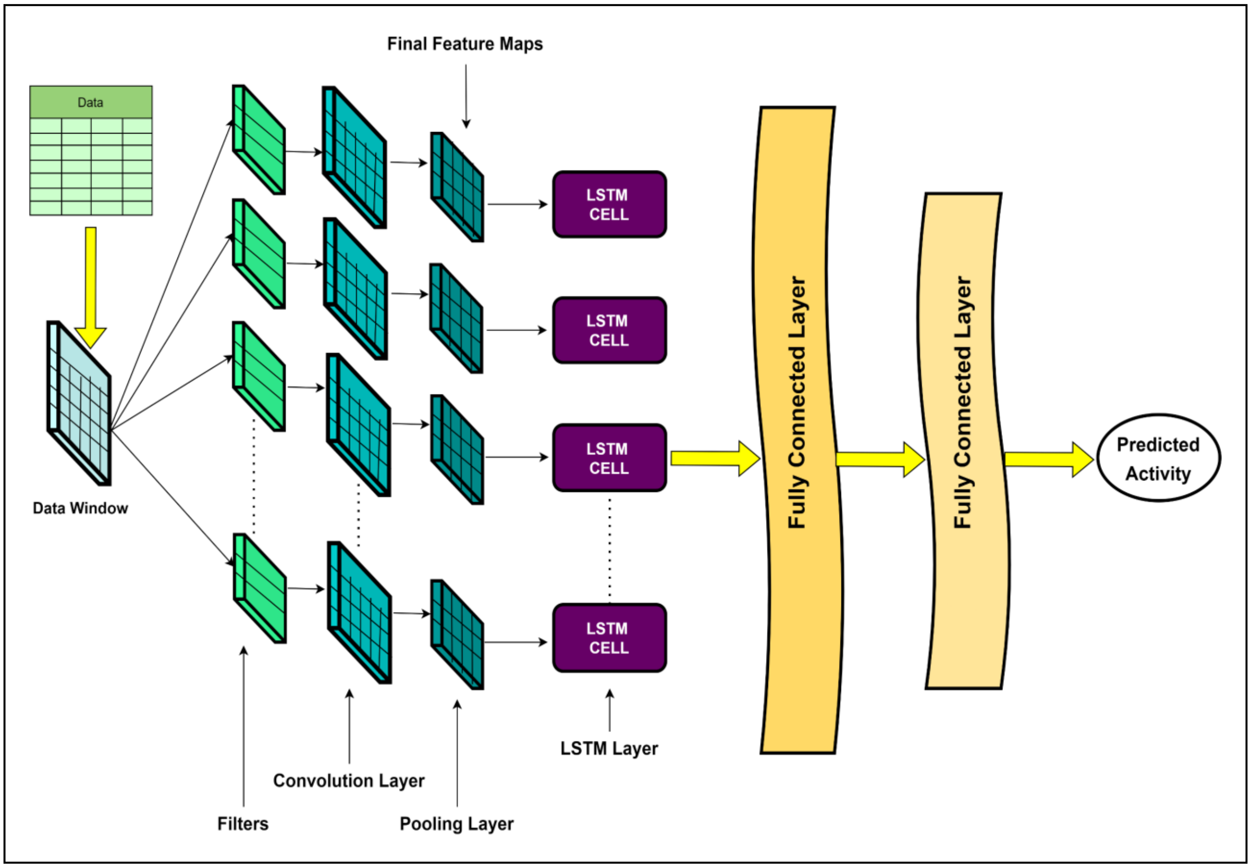


Figure 2.7. Proposed structure of the 1D-CNN-LSTM model.

We implemented the 1D-CNN-LSTM model in the programming language Python 3.8.10 using a package called Tensorflow 2.5.0. We used the 'Adam' optimizer for training our model with a learning rate of 0.001. We trained each model for 500 epochs using a batch size of 2000. After each epoch, we evaluated the performance of our model using the test data and saved the best model that showed the best accuracy on test data to calculate other evaluation metrics. The model can be reimplemented easily, and the results are completely reproducible. It should be mentioned that the parameters of our model were decided by trial and error method. We know that deciding the parameters of the deep learning model is challenging and time-consuming if we use a parameter optimization technique. Therefore, using a comparatively small sample size, I tried several layers, filters and LSTM units with variable learning rates and activation functions. Finally, I used the model for which I found the best result. A detailed summary of the overall architecture is presented in Table 2.4,

Table 2.4. A detailed description of our proposed 1D-CNN-LSTM model.

| Parts of Architecture | Components of each part (blank cell = not available for this layer) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| CNN | **Layer's Name** | **Number of Filters** | **Kernel Size** | **Activation Function** | **Dropout Ratio** | **Pooling Type** | **Pool Size** | **Padding type** |
| | Convolution | 512 | 5 | relu | | | | same |
| | Dropout | | | | 0.3 | | | |
| | Pooling | | | | | Average | 3 | same |
| | Convolution | 256 | 3 | relu | | | | same |
| | Dropout | | | | 0.3 | | | |
| | Convolution | 64 | 3 | relu | | | | same |
| | Pooling | | | | | Average | 3 | same |
| | Convolution | 128 | 3 | relu | | | | same |
| | Convolution | 256 | 5 | relu | | | | same |
| | Dropout | | | N/A | 0.3 | | | |
| | Convolution | 512 | 7 | relu | | | | same |
| | Dropout | | | | 0.3 | | | |
| | Pooling | | | | | Average | 3 | same |
| LSTM | **Layer's Name** | | | **Number of Units** | | **Activation Function** | | |
| | LSTM | | | 512 | | tanh | | |
| Fully Connected Network | **Layer's Name** | | | **Number of Neurons** | | **Activation Function** | | |
| | Dense | | | 100 | | relu | | |
| | Dense | | | 28 | | relu | | |
| | Dense | | | 64 | | relu | | |
| | Dense | | | 6 | | softmax | | |

## 2.3.6. Validation Procedure

We accumulated data from 42 participants. We used the leave-one-out cross-validation method for our study, which we can also refer to as inter-participant evaluation. We had to train and test our model 42 times to execute this procedure. Each time we had data from 41 participants in the training data and data from the other participant in the test data. Every time we trained and tested our model, we had data from a different participant in the test set. We were able to investigate the overall impact of our study on each participant.

### 2.3.7. Evaluation Metrics

To evaluate the performance of our findings, we used 4 evaluation metrics; accuracy, precision, recall and f1-measure. The definition and purpose of using each metric are given below:

#### 2.2.7.1. Accuracy

Accuracy[100] is defined by the ratio of the correct number of predictions to the total number of predictions. It is well suited for the classification task where each class has an almost similar number of samples. It can be calculated using the following formula,

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

#### 2.2.7.2. Precision

This metric is used to identify if the model is equally capable of identifying each class. This metric is helpful in evaluating the model's performance for each class separately. High precision for a class refers to the model's efficient performance in identifying that class. Low precision for a class means that the model is not considered capable of recognizing that class. We can calculate precision[100] for any class A using the following formula,

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

Here,

True positives = number of correctly predicted samples of class A

False positives = number of samples predicted as class A but not belonging to the class

#### 2.2.7.3. Recall

This metric is also suitable for evaluating models' performance for each class separately. It helps determine if the model is pruned to misclassification for a particular class. High recall for a class means that the model is not pruned to misclassify that class as another class. Low recall for a class refers to models' proneness in misclassifying that class as another class. We use the following formula to calculate recall[100] for any class A,

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

Here,

True positives = number of correctly predicted samples of class A

False negatives = number of samples not predicted as class A but belonging to the class

#### 2.2.7.4. F1-measure:

From the definition of Precision and Recall, precision emphasizes keeping the predictions accurate, whereas recall prioritizes increasing the number of correct predictions. For any model, we need to maintain a precision-recall trade-off, where we want to increase the number of correct predictions while keeping the predictions as accurate as possible. The F1 measure[101] represents a model's ability to maintain proper precision and recall. It does so by computing a harmonic mean of precision and recall using the following formula,

$$F1 - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

A model with a high F1 measure represents the model's ability in maintaining both high recall and precision, whereas a low F1 measure represents the opposite.

### 2.3.8. Data Reshaping

CNN was the first part of our proposed architecture of 1D-CNN-LSTM. So we initially reshaped the data to feed into CNN's first Convolution layer. As mentioned earlier, CNN works with a data matrix or data window. Each window may consist of a particular portion of data. This window helps CNN grasp knowledge about a current data point by providing some past data points or future data points. We can also refer to this data matrix as an image form where the CNN will extract features to learn more efficiently. Since CNN requires a data window as its input, we had to construct windows of data from our whole dataset. The number of samples each window contains is called window length. The training and test sets need to be segmented into windows of equal length. Each consecutive window may have common training samples. The number of common training samples between each window depends on the overlapping ratio[102]. We can use the following formula to calculate the overlapping ratio,

$$Overlapping\ Ratio = \frac{Number\ of\ common\ samples\ between\ two\ consecutive\ windows}{Window\ length}$$

In our study, we wanted to observe how different window length influences the performance of the 1D-CNN-LSTM model in the case of HAR. To conduct this study, we recorded the performance of our model for different window lengths. We started by segmenting the training set and test set into windows with a window length of 5 and then evaluated the model's performance. We then raised the window length by 10 and re-recorded the model's performance. We continued the process until we reached a window length of 195. We did not increase the window length further because it increased the training time considerably. The number of common samples between each consecutive window for a particular window length was $window\ length - 1$. So the overlapping ratio for a particular window length was,

$$Overlapping\ Ratio\ for\ a\ particular\ window\ lentgh = \frac{Window\ length - 1}{Window\ length}$$

It should be mentioned that although each window had multiple samples, there was only one label (activity) that corresponded to each window. The label for each window was the activity corresponding to the last sample of that window. Detailed information about the segmented datasets is illustrated in Table 2.5 below:

Table 2.5. The number of windows in the training set and the test set for each window length.

| **Window length** | **Overlapping ratio (%)** | **No. of windows in the training set ± standard deviation** | **No. of windows in the test set ± standard deviation** |
|---|---|---|---|
| 5 | 80.00 | 4221561 ± 31399 | 102959 ± 31399 |
| 15 | 93.33 | 4221551 ± 31399 | 102949 ± 31399 |
| 25 | 96.00 | 4221541 ± 31399 | 102939 ± 31399 |
| 35 | 97.14 | 4221531 ± 31399 | 102929 ± 31399 |
| 45 | 97.77 | 4221521 ± 31399 | 102919 ± 31399 |
| 55 | 98.18 | 4221511 ± 31399 | 102909 ± 31399 |
| 65 | 98.46 | 4221501 ± 31399 | 102899 ± 31399 |
| 75 | 98.66 | 4221491 ± 31399 | 102889 ± 31399 |
| 85 | 98.82 | 4221481 ± 31399 | 102879 ± 31399 |
| 95 | 98.94 | 4221471 ± 31399 | 102869 ± 31399 |
| 105 | 99.04 | 4221461 ± 31399 | 102859 ± 31399 |
| 115 | 99.13 | 4221451 ± 31399 | 102849 ± 31399 |
| 125 | 99.20 | 4221441 ± 31399 | 102839 ± 31399 |
| 135 | 99.25 | 4221431 ± 31399 | 102829 ± 31399 |
| 145 | 99.31 | 4221421 ± 31399 | 102819 ± 31399 |
| 155 | 99.35 | 4221411 ± 31399 | 102809 ± 31399 |
| 165 | 99.39 | 4221401 ± 31399 | 102799 ± 31399 |
| 175 | 99.42 | 4221391 ± 31399 | 102789 ± 31399 |
| 185 | 99.46 | 4221381 ± 31399 | 102779 ± 31399 |
| 195 | 99.49 | 4221371 ± 31399 | 102769 ± 31399 |

## 2.3. Results

This section will discuss the effect of window length on the overall result using different evaluation metrics and the effect of window length on each activity.

### 2.3.1. Effects of window length on the overall result:

We averaged the accuracy for all the participants at each window length, and the average accuracy gradually increased until we reached the window length of 55. We can observe the effect of window length on average accuracy in Figure 2.8. At the lowest window length, which is 5, the

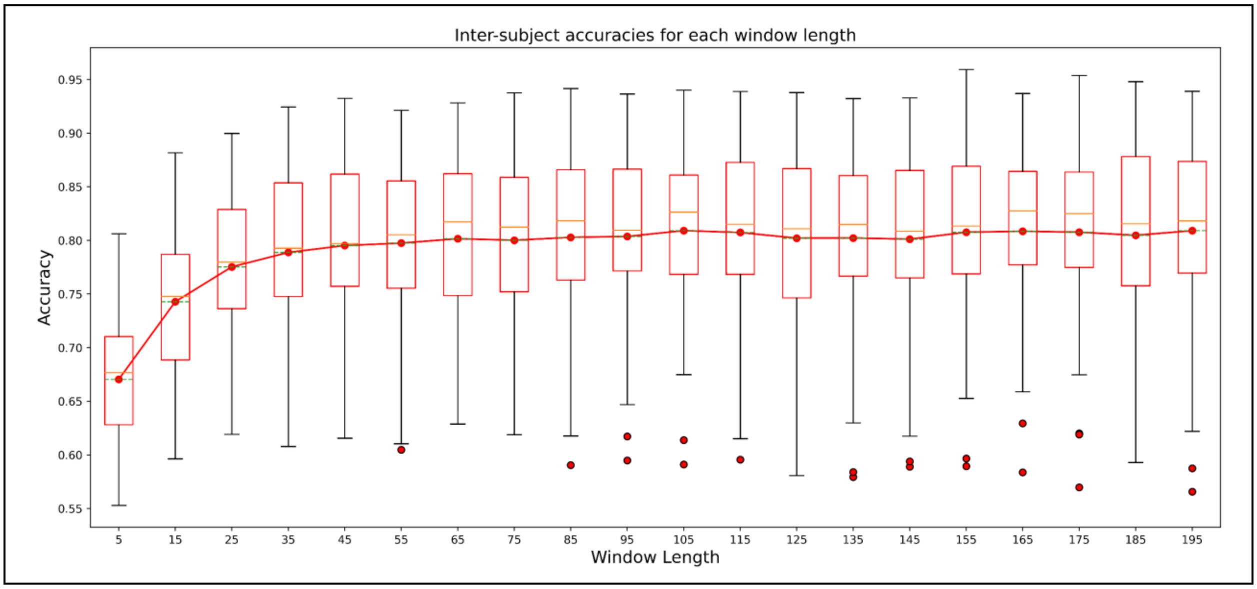


Figure 2.8. Box plot showing spreads of results for each participant for all window lengths. A red line connects the points, indicating average accuracy at each window length. The red horizontal line in the bar denotes the median values, whereas the green dashed horizontal line denotes the average values.

average accuracy was 67.04%. At a window length of 55, the average accuracy was 79.74%, and as the window length became greater than 55, we could not see considerable change. For the window length from 65 to 195, the average accuracy was around 80%. We recorded the highest average accuracy for the window length of 105, which was 80.91%. It is clear that the window length influenced the model's performance, but after a certain length, there was hardly any influence on the model's performance.

Furthermore, we can also observe the spread of accuracy for all the participants. The highest accuracy was about 81% for a particular participant at the lowest window length, and the lowest recorded accuracy was about 56%. However, as the window length increased, the highest recorded accuracy for any window length also increased. We recorded the highest accuracy, around 97% for a particular participant, for the window length of 155. Although the highest recorded accuracy for any window length increased with the increment of window length, the lowest recorded accuracy for any window length did not improve considerably. It seemed that the accuracy remained poor for some participants, even for bigger window lengths. This scenario can be explained better with participant-wise analysis.

### 2.3.2. Effect of window length on models performance for individual participants:

From the previous section, we found out that the average accuracy for all the participants became steady with an increment of window length. We will now observe if the scenario was the same for the individual participant. We can determine the effect of window length for each participant from Figure 2.9.

From Figure 2.9, we can see that the accuracy improved for the first 14 participants as the window length increased and remained steady after the window length of 55. Most of the participants showed a slight increment in the accuracy for the window length of 105. Among participants 1 to participant 14, the model performed best when we tested the model using the data from participant 10. We recorded the highest accuracy, around 97%, for participant 11 at a window length of 155;

however, the performance was not consistent. For participants 15 to 28, the scenario was almost the same as we had for the first 14 participants, but for participant 16, the accuracy did not improve with window length; rather, we found a downward trend. We recorded the highest accuracy for participant 16 at a window length of 65, which was about 65%. Accuracy for participant 19 was found to be as poor as we saw for participant 16. Although accuracy for participant 10 had an increased accuracy of around 70% at a window length of 105. From participants 15 to 28, we recorded the best performance for participant 20 almost at every window length. Regarding the model's performance for participants 29 to 42, we observed poor outcomes from the model for participant 37, which resembled the model performance for participant 16. The best performance from the model was recorded for participant 33, almost for every window length.

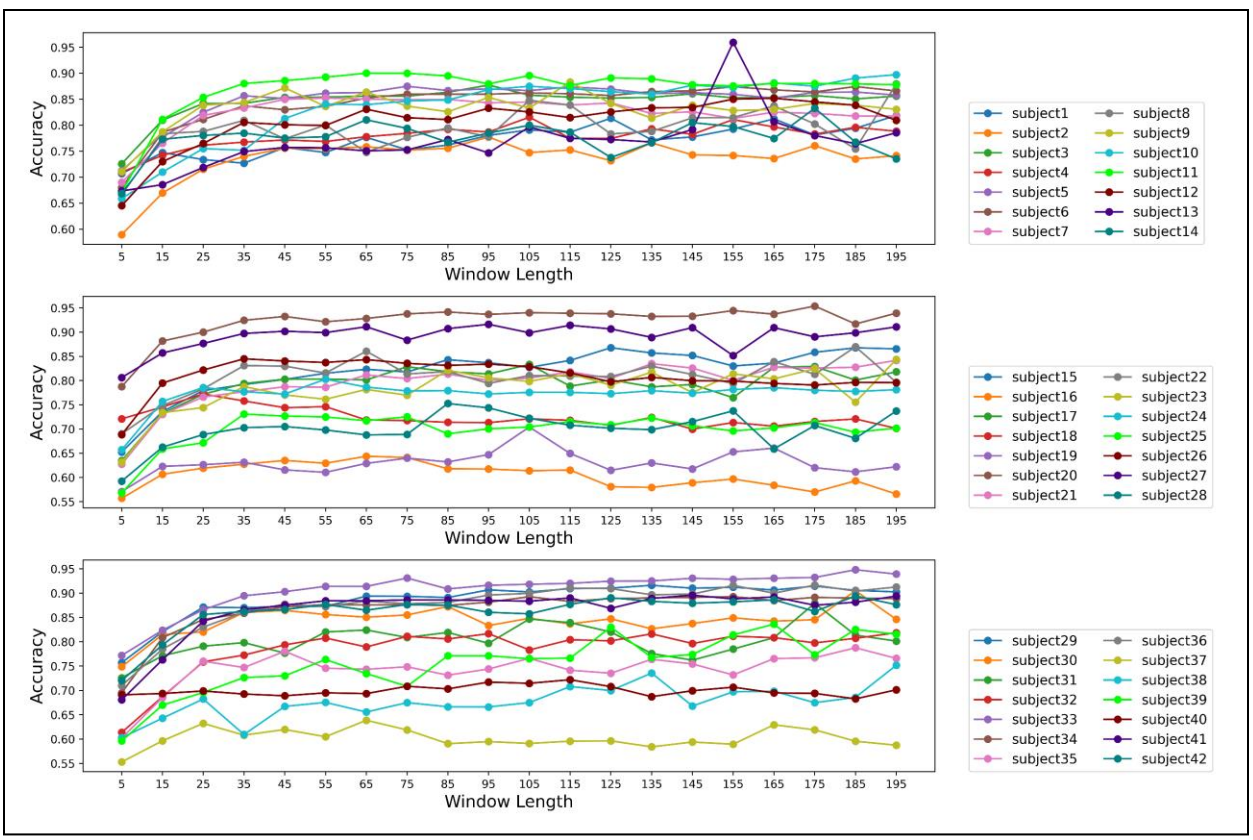


Figure 2.9. Accuracies for each participant for different window lengths.

### 2.3.3. Effect of window length on models performance for each activity

As mentioned earlier, we considered the metrics precision, recall, and F1 measure to evaluate the impacts of window length on each activity. We calculated the precision for each activity class at each window length for all participants and averaged the precision depicted in Figure 2.10.

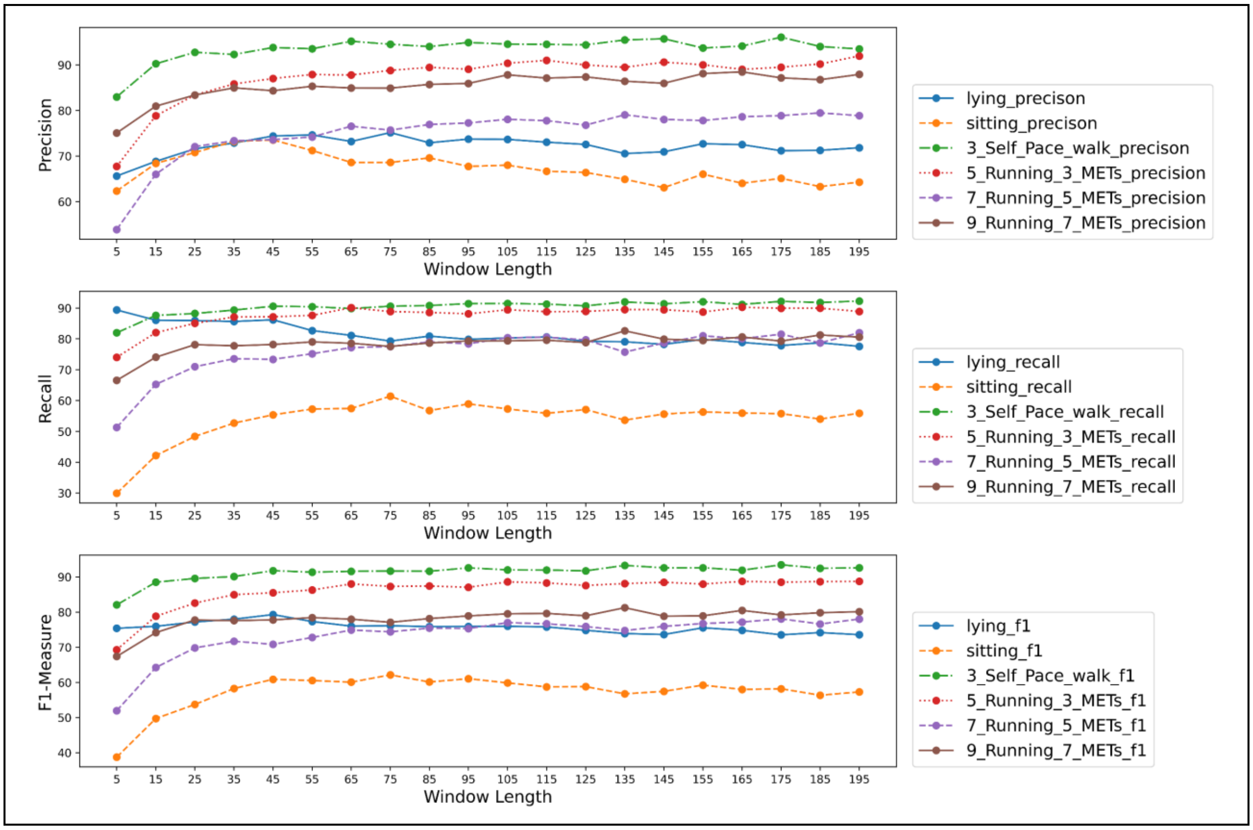


Figure 2.10. Precision, recall and f1-measure for each activity class at different window lengths.

From Figure 2.10, we can see that precision increased until a certain window length for all six different activities. After a window length of 45, precision either remained steady or improved for all activities except for sitting. In addition, we experienced high precision for high-intensity activities such as walking and running at 3 METs, 5 METs and 7 METs. Still, the precision was comparatively poor for low-intensity activities such as lying and sitting. We recorded higher precision for activity walking than all other activities for every window length, which means that the models were highly accurate in predicting activity walking. A more detailed scenario of precision is shown in Table 2.6. We can see the models' highest average precision for each activity and the window length at which the highest precision we recorded in Table 2.6.

Regarding recall, we observed very poor recall for the activity sitting. Experiencing poor precision and recall for the activity sitting means that the models experienced difficulties correctly identifying it. It also means that the models misclassified many samples belonging to other activities as sitting and many samples from the activity sitting as other activities. We recorded impressive recall for the activities walking and running at 3 METs. Recalls recorded for the other 3 activities were considerably decent. An interesting trend we observed for the activity lying is that the recall was about 90% at the lowest window length, and the recall reduced as the window length increased. In contrast, recalls for all other activities increased until a certain window length and then became steady. A detailed numerical description of recall is given in Table 2.6.

The F1 measure depicted similar trends in precision and recall. We recorded a high F1 measure for walking and running at 3 METs. We had a decent F1 measure for all other activities except for sitting. The F1 measure was very poor to consider for activity sitting. F1 measures increased until a certain window length for all the activities and remained steady as window length increased. We also provided a numerical description of the F1 measure in Table 2.6.

Table 2.6. Highest and lowest averaged precision, recall and f1-measure for each activity and respective window length

| Activities | Properties (Highest Precision) | | Properties (Lowest Precision) | | Properties (Highest Recall) | | Properties (Lowest Recall) | | Properties for the (Highest F1 Measure) | | Properties (Lowest F1 Measure) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Highest Value** | **Window Length** | **Lowest Value** | **Window Length** | **Highest Value** | **Window Length** | **Lowest Value** | **Window Length** | **Highest Value** | **Window Length** | **Lowest Value** | **Window Length** |
| Lying | 76.16 | 75 | 65.63 | 5 | 89.38 | 5 | 77.58 | 195 | 79.30 | 45 | 73.55 | 175 |
| Sitting | 73.53 | 45 | 62.34 | 5 | 61.41 | 175 | 29.91 | 5 | 62.15 | 75 | 38.74 | 5 |
| Walking | 96.10 | 175 | 82.99 | 5 | 92.29 | 195 | 82.00 | 5 | 93.46 | 175 | 82.07 | 5 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Running 3 METS | 91.98 | 195 | 67.73 | 5 | 90.26 | 165 | 74.04 | 5 | 88.76 | 195 | 69.30 | 5 |
| Running 5 METS | 79.48 | 185 | 53.87 | 5 | 81.99 | 195 | 51.30 | 5 | 78.08 | 175 | 51.95 | 5 |
| Running 7 METS | 88.49 | 165 | 75.08 | 5 | 82.62 | 135 | 66.57 | 5 | 81.26 | 135 | 67.39 | 5 |

## 2.4. Discussion

In [17], they studied the performance of heuristic features for 5 publicly available datasets, which they labelled as A[33], B[103], C[83], D[104], and E[105]. Besides using the 9 features altogether, they also used only the first 3 or the first 6 heuristic features and recorded the performance of 4 classifiers Bayesian Decision Making (BDM), K- Nearest Neighbours (KNN), Support Vector Machine (SVM) and Artificial Neural Network (ANN). They used a 10-fold cross-validation technique where each fold contained data for a particular participant. We can call our used inter-participant validation technique a 42-fold cross-validation technique where each fold contains data for a particular participant. They recorded accuracy from each classifier using first 3 heuristic features, first 6 heuristic features and all 9 heuristic features. They achieved the highest accuracy for datasets B, C, and D using the first 3 heuristic features. To acquire the highest accuracy for datasets A and E, they used the first 6 and all 9 heuristic features, respectively. For all the datasets, they found the best performance using SVM. From their results, it was clear that all 9 heuristic features were not critically important to model performance since, for 3 of their datasets, they recorded the best performance using only the first 3 heuristic features. However, they did not try all the combinations of features, and there was no analysis to select the most significant features. We performed that analysis and found the first 4 features to be the most important. We plotted the highest accuracy they achieved using the heuristic features for each dataset and also indicated the number of features for which they found the best performance in **Error! Reference source not found.**. We also included the highest average accuracy we achieved, using the 4 most important features we found in the plot to provide a comparative perspective.

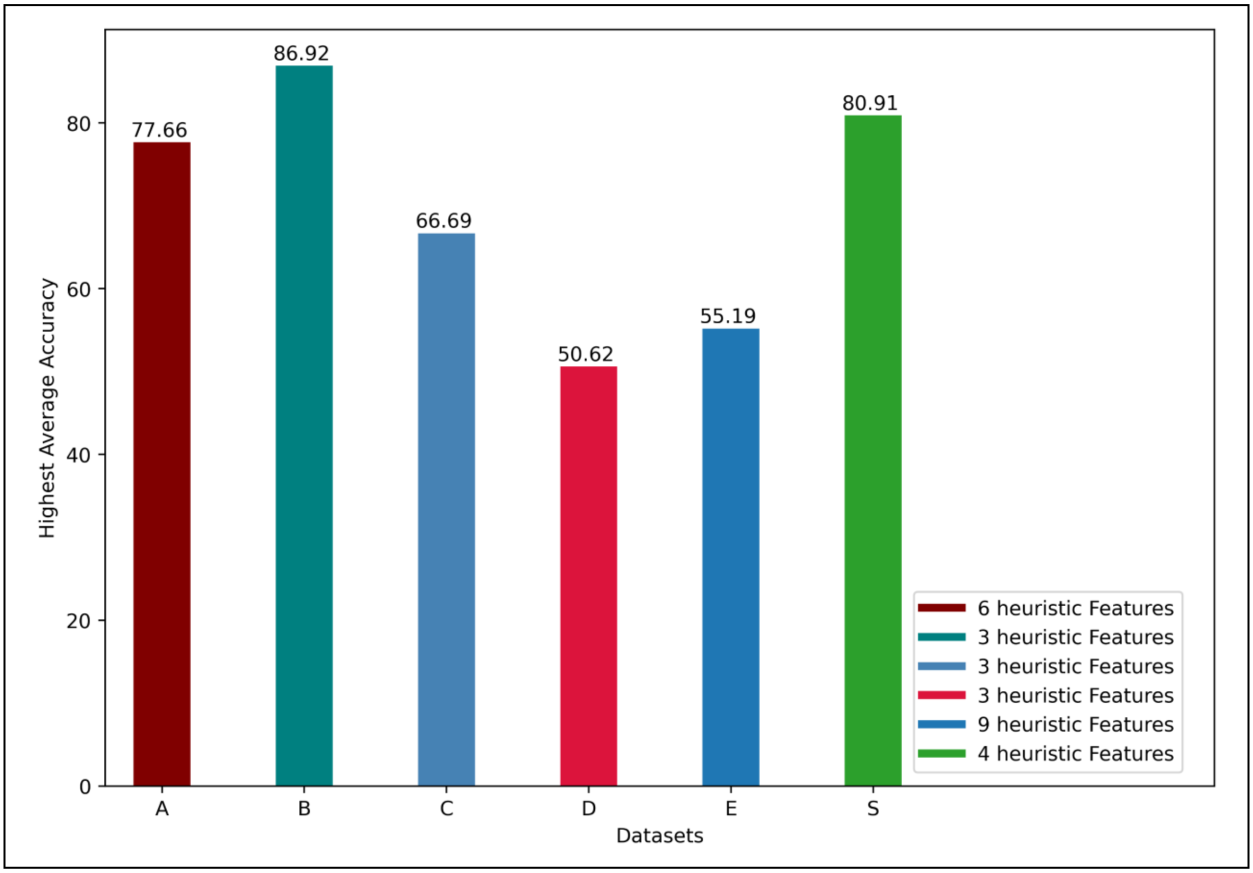


Figure 2.11. The highest accuracies achieved for datasets A, B, C, D and E by the different number of heuristic features by [17], along with the accuracy we found for our dataset denoted by S using 4 most significant heuristic features.

Although it is not feasible to compare our results with the results found in [17]since they used different classifiers and datasets, we acquired results comparable to their performance, even with more participants than they had for an inter-participant validation method. However, our main objective was to explore the effects of window length in HAR execution for 1D-CNN-LSTM.

Many studies have explored HAR using deep neural networks like CNN, LSTM or hybrids. But few studies in HAR reported the effects of window length or time steps. Most of the studies chose the time steps or window length, claiming that they achieved the best performance using that particular window length. For instance, [106] used a CNN and Gated Recurrent Unit (GRU) hybrid on three datasets named UCI-HAR, WISDM, PAMAP2 and acquired 96.20%, 97.21%, 95.27%, respectively. Still, they did not mention how they chose the window length of 128 for their model

. In another study [65], which used the same dataset as in [106], they also did not mention the reason behind choosing 128 as their window lengths rather, they emphasized their proposed CNN, Bidirectional LSTM hybrid model architecture and acquired accuracies. Many other studies[76, 107-109] explored non-identical forms of deep learning architectures using the popular UCI-HAR dataset and used the same window length of 128 samples. Another study[110] using a dataset called "Complex Human Activities using smartphones and smartwatch sensors" explored the performance of different deep neural networks, including LSTM, Bidirectional LSTM, GRU, Bidirectional GRU, CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU for 5 differents window lengths(in seconds) of 5, 10, 20, 30 and 40s. They achieved the highest accuracy of 98.78% using CNN-BiGRU when they used the window length of 40. However, exploring only 5 different window lengths was insufficient to depict the influence of window lengths. Therefore in our study, we explored the performance of 1D-CNN-LSTM for 19 different window lengths. We can observe from Figure 2.8 that the recorded results showed that window length had a significant impact on the performance of the models in HAR. However, the impact was noticeable until we reached a window length between 45 to 75 in the study. After that, the performance was not influenced substantially by incremental increases in window length. We can call the window length range of 45 to 75 the saturation range for the models' performance. The reason behind such a trend could be that, after a certain length, even if we increase the window length, the model could not extract significant knowledge to enhance its performance. Although Figure 2.8 displayed the averaged effect of window length, we observed the influence on individual participants in **Error! Reference source not found.**. We experienced a similar trend for almost all participants. We had a contradictory trend in performance for participants 16 and 37. After the saturation range with increments in window length models, performance for those participants was reduced. Although the decrement was not considered important, it was not usual if we observed the performance trend for other participants. This could happen due to noisy samples in the datasets belonging to those two participants, which need to be further analyzed. Observing the effect of window length on each activity class, we found that precision, recall, and f1-measure had very poor values for the sitting activity. In addition, the values of evaluation metrics were reduced with an increase in window length for lower-intensity activities like sitting and lying. For instance, recall for lying was about 90% when the window length was the smallest, but as the window length increased, the recall decreased. The effect of window length on lower-intensity activities was an interesting observation which was not evaluated in previous studies. We can assume that window length had different effects for activities with different intensities considering our outcome. When choosing a window length, we should also consider a window length that will help generate a better outcome for lower and higher-intensity activities. Another reason behind such poor performance could be the lower number of samples for activity sitting, which we can see in Table 2.3. Balancing the classes could have helped to improve the situation. Still, we did not do it in our study as our main objective was to observe the influence of window length rather than increasing the models' performance.

From Table 2.6, we can see that for most of the activity classes highest metric values we found were when the window length was above 150, but the highest values were not considerably greater than the values we found at the saturation point. That means we need not choose a very high window length to achieve the best performance from the model rather, we need to select a window length around the saturation range that will be considerably smaller than the window lengths where we found the highest metric values. If we manage to keep the window length smaller, it would reduce the time complexity for the models and increase the computational efficiency.

Although we have conducted analysis participant-wise and activity-wise, some analysis is yet to be done. For instance, we achieved very high performance for some participants, such as participants 26, 27 and 33 and very poor performance for participants 16, 19 and 37. Still, we did not try to determine why the model performed differently, especially for these participants; as we mentioned earlier, our objective was to study the effect of window length in 1D-CNN-LSTM in HAR.

In brief, we found that window length in 1D-CNN-LSTM had a significant effect on HAR. We found that the training time was affected by the window length. As the window length increased, the training time also increased. The approximate training time for the model using the lowest window length of 5 was about 40 minutes and almost 20 hours for the highest window length of 195. For our suggested saturation range of 55 to 85, the training time was about 4 hours. Here we approximated the mentioned training time for each iteration of inter-subject validation, which means there were data from 41 subjects in training data, and test data included data from 1 subject, which we did not include in the training data. So, window length should not be arbitrarily long; rather, it should be chosen wisely by correctly identifying the saturation range so that the model offers less time complexity while training and more efficiency. In addition, for the 1D-CNN-LSTM model, other studies may choose a window length from our suggested saturation range of 55 to 85 for HAR. We resampled the whole dataset to 30Hz, so our proposed saturation range should be 1.83s to 2.83s in seconds.

There are a number of limitations to our current study. We only used one accelerometer for our study to keep the computation complexity low, as we conducted inter-participant validation for 42 participants. But we may experience improvements in our study if the gyroscope sensor was also used with the accelerometer sensor since the gyroscope sensor can provide substantial information regarding the rotational nature. Moreover, the data was collected from only one smartphone location, a phone in the pocket, and we did not study how much the analysis would get affected if we used data from different body parts. In addition, we studied the effect of window length on one type of model, but other models also take windows of data as input. We do not know if the effect would be the same for those models. However, we initiated this type of analysis using many participants, one accelerometer sensor, data from one smartphone location and only one type of model. In future, we will try to conduct the same study using different models and settings.

## 2.5. Conclusions

Our study wanted to depict the influence of window length in 1D-CNN-LSTM on HAR. We used a large dataset accumulated from 42 participants for six different activities. The samples were from the accelerometer sensor in a smartphone kept in the pocket. We used 4 heuristic features to eliminate variations produced due to the rotation of the smartphone. We found a saturation range for window length, after which the model does not get considerably influenced by the window length.

**Author Contributions:** Conceptualization, Arnab Barua and Xianta Jiang; Data curation, Daniel Fuller and Musa Sumayyah Bamidele; Formal analysis, Daniel Fuller; Funding acquisition, Daniel Fuller and Xianta Jiang; Investigation, Daniel Fuller and Xianta Jiang; Methodology, Arnab Barua; Project administration, Daniel Fuller and Xianta Jiang; Resources, Daniel Fuller; Software, Arnab Barua; Supervision, Daniel Fuller and Xianta Jiang; Validation, Arnab Barua, Daniel Fuller and

Xianta Jiang; Visualization, Arnab Barua; Writing – original draft, Arnab Barua; Writing – review & editing, Daniel Fuller and Musa Sumayyah Bamidele.

**Funding:** This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant number RGPIN-2020-05525.

**Institutional Review Board Statement:** The study was by the Memorial University Interdisciplinary Committee on Ethics in Human Research (ICEHR #20180188-EX).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to ethical constraints.

**Conflicts of Interest:** The authors declare no conflict of interest.

# Chapter 3

# The Effectiveness of Simple Heuristic Features in Sensor Orientation and Placement Problems in Human Activity Recognition Using A Single Smartphone Accelerometer

## 3.1. Introduction

Human Activity Recognition (HAR) is a critical research area because of its multiple applications that are beneficial for humanity. HAR is the process of enabling computers to recognize human activities by analyzing patterns in different data types, including sensor data, images, and videos. Research on HAR is important as it is the principal method for accomplishing applications such as identifying risk factors regarding depression[69], diabetes[111], health condition surveillance[112], eldercare[113], sports performance analysis[114], and abnormal activity identification[115]. Since HAR is the primary foundation for the successful implementation of many applications, researchers are trying to overcome the challenges which cause inaccuracy in HAR. Sensor data is one of the most reliable and popular types of data used in HAR. Sensor data includes data from accelerometers, gyroscopes, and magnetometers [4, 82, 116]. Studies have used these sensors in distinct ways to accumulate data for HAR. Some researchers attached the sensors separately to different body parts[80, 117, 118], and some used sensors embedded in smartphones[85, 119-122] or smartwatches[123-125]. Among these different types of sensory devices and placements, smartphones are efficient, feasible and beneficial to for HAR research because they address a number of advantages including applicability to a large population. Almost every smartphone contains an accelerometer and gyroscope sensor. Data from both of these sensors are capable of distinguishing different human activities, which means they are feasible for HAR applications. Smartphones are also an inseparable part of daily human life. As a result, researchers emphasized perfecting the HAR using smartphone sensors by adapting various techniques to diminish the difficulties posed by smartphones in HAR.

Perfecting the HAR process using smartphone sensors requires the researchers to overcome some significant challenges related to sensor orientation, sensor placement, and algorithm choice. The sensor orientation problem is one of the most concerning issues faced when using smartphones in HAR, as a smartphone can be kept in different orientations. A user can keep the smartphone in any orientation and perform different activities. When two different users perform the same activity while keeping the smartphone in different orientations, the sensor data is different, and it becomes hard for HAR methods to identify the sensor data as the same activity. Many studies have proposed different methods to deal with the sensor orientation problem of the smartphone in HAR. Researchers also had to propose various approaches to diminish the sensor placement problem along with the sensor orientation problem. The sensor placement problem happens as smartphone users tend to keep their smartphones in different body locations, including backpacks, hands, or pockets. The smartphone sensors, particularly the accelerometer and gyroscope sensors, generate non-identical patterns for similar activity if the smartphone is kept in non-identical locations. For dealing with the sensor orientation and placement problem, researchers generally try to extract features with no orientation or placement effect that could generate substantially different sensor patterns for different activities. For example, [15] used extracted features in their proposed activity recognition process where they included data from four different body locations (coat pocket, hand, trouser pocket and bag) and for five human activities (going upstairs, going downstairs, walking, standing and running). They started by extracting horizontal and vertical acceleration data from a raw accelerometer to diminish the influence of device orientation. Later they extracted eight features from the raw gyroscope signal and separated horizontal and vertical accelerations to develop a location identification system. Finally, they performed feature selection, and using this location recognition system, they conducted some data adjustments to the selected features, which were later used in their activity recognition process. They achieved an accuracy of 91.27% using Support Vector Machine (SVM) with a four folds cross-validation technique. [16] extracted 89 time and frequency domain features from the accelerometer and gyroscope sensor of smartphones to make the activity recognition process orientation invariant and location independent. They then performed feature selection and feature normalization on the extracted features. Using these features, they evaluated the performance of three classifiers, K-Nearest Neighbours (KNN), Random Forest (RF), and SVM in recognition of five human activities (descending stairs, ascending stairs, walking, jogging and jumping) for five non-identical smartphone locations (trousers pocket, jacket pocket, hand and upper arm). They considered different validation procedures named one-to-one, all-to-one and rest-to-one and compared the performance of the classifiers for different validation procedures. [17] extracted 9 heuristic features from the data of different sensors available in five public datasets, which they claimed to be free of the influence of sensor orientation. They evaluated the performance of these 9 features in HAR using four machine learning algorithms and found the features to be effective enough to diminish the orientational effects. Along with these studies, many other studies extracted features to solve the sensor orientation and location dependency problem[18-20]. However, along with feature extraction process, coordinate transformation is a promising method to address the sensor orientation and location dependency problem.

In the coordinate transformation approach, first, a global reference coordinate system is discovered, and then all the signals are projected to that reference system. [46] used the gyroscope signal from a smartphone to develop the global reference coordinate system and then transformed the acceleration signal into that uniform reference coordinate system. Following this, they used a motif discovery algorithm to find activity patterns and then developed a Vector Space Model for classification purposes. They used their approach on a dataset containing smartphone signals from four different body locations (the left upper arm, the shirt pocket, the trousers front pocket, and the behind trouser pocket) and four different orientations and performed cross-orientation and cross-placement validation. [47] also performed coordination transformation by calculating quaternion to transform the linear acceleration signal from the device-coordinate system to the earth-coordinate system. Followed they extracted the first two principal components from the transformed acceleration signal to eliminate the direction effect for different activities. In addition, they extracted time and frequency domain features to make their approach more reliable and accurate. To validate their method, they collected data from a smartphone placed in three different locations pants' pocket, shirt's pocket and backpack) and three different orientations. They performed leave one orientation out cross-validation technique using an Online SVM algorithm and compared results for different orientations, placements and participants. [48] also performed coordinate transformation and feature extraction using an accelerometer, gyroscope and magnetic sensor to eliminate the orientational effect of the smartphone sensor and evaluated their method for two smartphone orientations (vertical and horizontal) placed in trouser pockets. They achieved 97% accuracy in recognizing five human activities using a KNN classifier. In brief, a number of studies and methods have been used to address the sensor orientation and placement problem. However, the classification algorithm also plays an important role in HAR classification accuracy; in particular, deep learning algorithms are potentially promising given a sufficiently large dataset. Studies have evaluated the performance of different machine learning algorithms in HAR and provided comparisons to decide the most suitable classifiers to use. Early classification studies used simple machine learning classifiers such as SVM[18, 49-51], RF[56-58], KNN[52, 53], and Decision Trees[54, 55]. These were employed because of their low complexity and resource-efficient nature. However, the advancement of computational resources enabled the usage of deep learning algorithms such as Artificial Neural Networks (ANN), Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), Long Short-Term Memory(LSTM), and Gaited Recurrent units(GRU). These deep learning algorithms offer additional advantages in HAR, especially CNN and LSTM because of CNN's automated feature extraction capability and LSTM's ability to persist older information from time series data. [21] used CNN's ability of automatic feature learning and found it to outperform four conventional machine learning algorithms in recognizing 18 human activities and 12 hand gestures. They also concluded that CNN was suitable for online HAR. [22] also exploited the feature extraction ability of CNN for HAR on three public datasets (Opportunity, Skoda and Actitracker) and acquired an accuracy of 88.19%, 76.83%, and 96.88% on Skoda, Opp, and Antitracker, respectively, using the CNN-based model. [23] evaluated the feature extraction capability of CNN as well in recognizing six daily human activities (sitting, standing, walking, jogging, upstairs, and downstairs) using accelerometer signals. They achieved an accuracy of 94.2% which outperformed traditional machine learning algorithms such as Decision Trees (J48) and SVM. There are also other studies that used CNN as their final classification model, along with their early data pre-processing layer, to enhance the recognition rate of human activities[24-26]. Along with CNN, another deep neural network variation called RNN is being widely used in HAR[27-29]. RNN has few variations of itself and among them, LSTM is useful in HAR, especially when studies combined the information persisting ability of LSTM with the feature extraction capability of CNN. [30] utilized the combination of CNN and LSTM, also called CNN-LSTM, to evaluate its performance in HAR for two datasets (iSPL and UCI HAR). They acquired accuracies of 99.06% and 92.13% on iSPL and UCI HAR datasets, respectively. [31] also employed the CNN-LSTM model for HAR using data from smartwatch sensors from 44 subjects performing 18 activities. They achieved an accuracy of 96.20% using CNN-LSTM, which was found to be better than the performance of CNN and LSTM when the models were used separately. [32] proposed a 4-layered CNN-LSTM model and evaluated its performance using the UCI HAR dataset. They found that the CNN-LSTM hybrid model can outperform Vanilla LSTM network, 2-Stacked LSTM network, 3-Stacked LSTM network with an accuracy of 99.39% using a 10-fold cross-validation technique. Many other researchers have used CNN-LSTM in HAR to utilize its capabilities of feature extraction and preserving temporal dependencies[62-65]. There has been considerable research for HAR which proposed non-identical techniques to solve the major challenges including sensor orientation, sensor placement, and algorithm choice.

In this study, we contributed to this field by evaluating the performance of previously introduced heuristic features[17] using our dataset in both intra-location (i.e. senor orientation) and inter-location. (ie. sensor placement) scenarios. In the original study[17], the researchers introduced heuristic features to tackle the sensor orientation problem. However, they evaluated those heuristic features' performance by introducing orientation in the dataset synthetically. In our study, we evaluated the performance of these features in solving the orientation problem for three different smartphone locations where the sensor orientations were ensured during the data accumulation process. Moreover, we assessed the performance of those heuristic features in solving the sensor placement problem. By doing this, we wanted to inspect if the heuristic features alone can solve the sensor placement issue. In addition, only a few studies adopted the leave N-Subject-out cross-validation approach and did it for a considerably small-scale dataset. In our study, we adopted the Leave-N-Subject-out cross-validation approach for a dataset accumulated from 42 subjects and consisted over 12 million samples. To be precise, we worked on the following points in this study,

- We evaluated the effectiveness of previously proposed sensor invariant features[17] performance in the case of sensor orientation problem in HAR for a large-scale dataset where the sensor orientations were introduced practically. (Intra-location Evaluation)
- We assessed the performance of heuristic features in tackling the sensor placement issue in HAR. (Inter-location Evaluation)
- We analyzed the performance of the proposed approach in HAR using a leave N-out cross-validation technique for a huge dataset containing enormous variations.
- We analyzed the performance of the proposed architecture for six activities with varying intensities (Lying, Sitting, Walking, Running at 3-METs, Running at 5-METs, and Running at 7-METs).

The rest of the paper is arranged as follows. Section 2 introduces the materials and methods where we discuss the data accumulation procedures, data pre-processing and feature extraction approach, the architecture of the models and its workflow. Section 3 describes activity-specific and participant-specific results for both intra-location and inter-location scenarios. Section 4 discusses our findings, and we conclude our study in Section 5. The entire study procedure is depicted in

Figure 3.1.

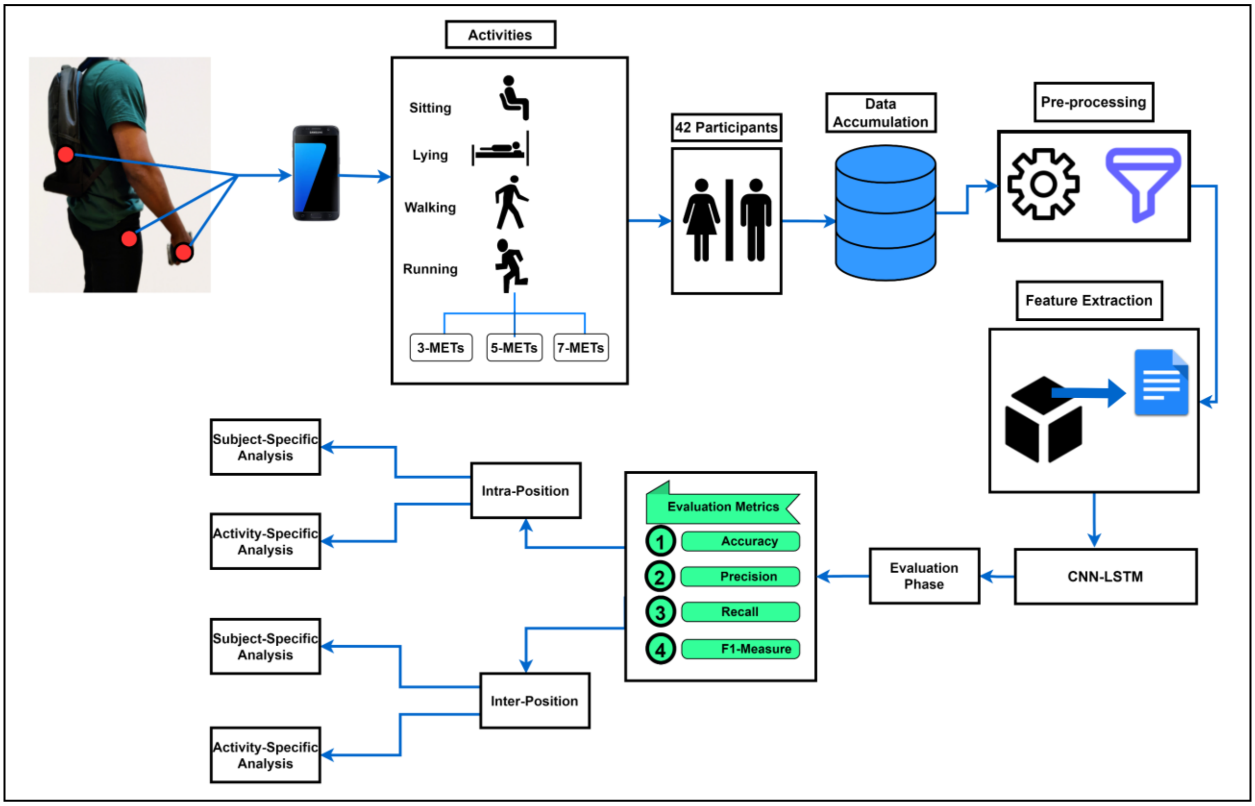


Figure 3.1. Overall workflow diagram of our study.

## 3.2. Materials and Methods

In this section, we will first discuss the data accumulation process. Following, we will explain the data pre-processing and feature extraction procedure. Then, we will briefly discuss the feature selection approach and describe the CNN-LSTM architecture we used.

### 3.2.1. Data Accumulation

This section has segments similar to section 2.2.1. The sole difference was introducing data collection from the smartphone location backpack and right hand. For our study, we collected data from 42 healthy participants for six different activities with varying intensities: Lying, Sitting, Walking, Running at 3-METs, Running at 5-METs, and Running at 7-METs. We acquired ethical approval from the Memorial University Interdisciplinary Committee on Ethics in Human Research (ICEHR #20180188-EX). Before commencing the data accumulation procedure, each participant had to complete the Physical Activity Readiness Questionnaire (PAR-Q). There were 18 male and 24 female participants. The average age, height and weight were 29 (range = 18-56 years) years, 169.17cm (range = 143-185cm) and 68.19 kg (range = 43-95.2kg), respectively. Each participant performed nine trials to complete the data collection protocol. While performing the trials, the participant carried three Samsung Galaxy S7 smartphones (SM-G930W8) in three locations. The locations were the participant's right pocket, backpack and right hand. The data accumulation process was 65 minutes long. The order of the trials with duration is given in Table 3.1. Trial 1 is the trial with which the participants started the data collection protocol, and Trial 9 refers to the last trial to be completed.

An android application called Ethica Data[96] was used to collect sensor data. The application recorded the data from the accelerometer sensor's X, Y and Z axes embedded into the smartphone. The application continuously recorded the sensor's value and uploaded the value to the server. During the data collection procedure, the participants were free to keep the smartphone in any arbitrary orientation. We collected the data in an indoor environment. We used a treadmill to accumulate data for walking and running at three speeds. We used the Metabolic Equivalent of Task (MET) to quantify the running intensities or speeds. METs are a ratio of the oxygen consumption rate of a person to the corresponding person's weight. We preferred MET to walk speed, cadence or stride length to measure the intensity because those units are prone to generate different expenditures for different persons. We wanted to ensure that the participants performed the activity with the same intensity. The mathematical equation to define the MET is given in equation 1,

$$MET = \frac{Oxygen\ Consumption\ Rate\ (\frac{mili leter}{minute})}{3.5 \times weight\ (kg)}$$

We selected these particular activities in our study to ensure the presence of the most common daily activities. As well, few HAR studies combined activity types and activity intensity recognition in their work.

Table 3.1. Trial order and duration of data collection protocol.

| **Trial Order** | **Activity** | **Duration (Minutes)** |
|---|---|---|
| 1 | Lying down | 5 |
| 2 | Sitting | 5 |
| 3 | Walking | 10 |
| 4 | Lying down | 5 |
| 5 | Running at 3-METs | 10 |
| 6 | Lying down | 5 |
| 7 | Running at 5-METs | 10 |
| 8 | Sitting | 5 |
| 9 | Running at 7-METs | 10 |

### 3.2.2. Data Pre-processing

This section has repeated segments from section 2.2.3. The new addition in this section was the sample counts for the smartphone location backpack and right hand. We performed data resampling and data imputation on the dataset. The optimization technique of the Ethica App did not let the app maintain the same data uploading frequency. As a result, the frequency ranged from 5 Hz to 19 Hz. We upsampled the dataset to a constant frequency of 30 Hz to eliminate this data imbalance using a published method [33]. Another challenge with the data was missing values. Missing data occurred because of the temporal connection loss between the Ethica App and the server. We used linear data imputation to impute missing values using the ImputeTS package in

R. The number of samples for each activity at each smartphone location after pre-processing is shown in Table 3.2**,**

Table 3.2. Number of samples for each activity class at each location after pre-processing.

| **Location: Pocket** | | **Location: Backpack** | | **Location: Hand** | |
|---|---|---|---|---|---|
| **Activity Name** | **Sample Count** | **Activity Name** | **Sample Count** | **Activity Name** | **Sample Count** |
| Sitting | 879325 | Sitting | 879612 | Sitting | 879855 |
| Lying | 1266292 | Lying | 1268078 | Lying | 1269428 |
| Walking | 765246 | Walking | 765375 | Walking | 765266 |
| Running at 3 METS | 984997 | Running at 3 METS | 986512 | Running at 3 METS | 986310 |
| Running at 5 METS | 986446 | Running at 5 METS | 986064 | Running at 5 METS | 985187 |
| Running at 7 METS | 992654 | Running at 7 METS | 995726 | Running at 7 METS | 992154 |

### 3.2.3. Feature Extraction

The feature extraction process and formulas have already been discussed in section 2.24. This section has been added to this chapter for the reader's convenience. Some new segments and figures have been added in this section to show the effect of the extracted features for different smartphone locations. We extracted 9 orientation-invariant heuristic features using the formulas in [17] to address the orientational dependency problem. Since the participants had the freedom to place the smartphones in the pre-determined smartphone location in any orientation, we experienced different ranges and patterns in sensor values for different participants even though they were performing the same activity. Figure 3.2 shows the differences in patterns and ranges of accelerometer axes due to the sensor orientation while different participants performed the same activity (Running at 7 METs), keeping the smartphone in their backpacks. We extracted the previously introduced 9 orientation invariant

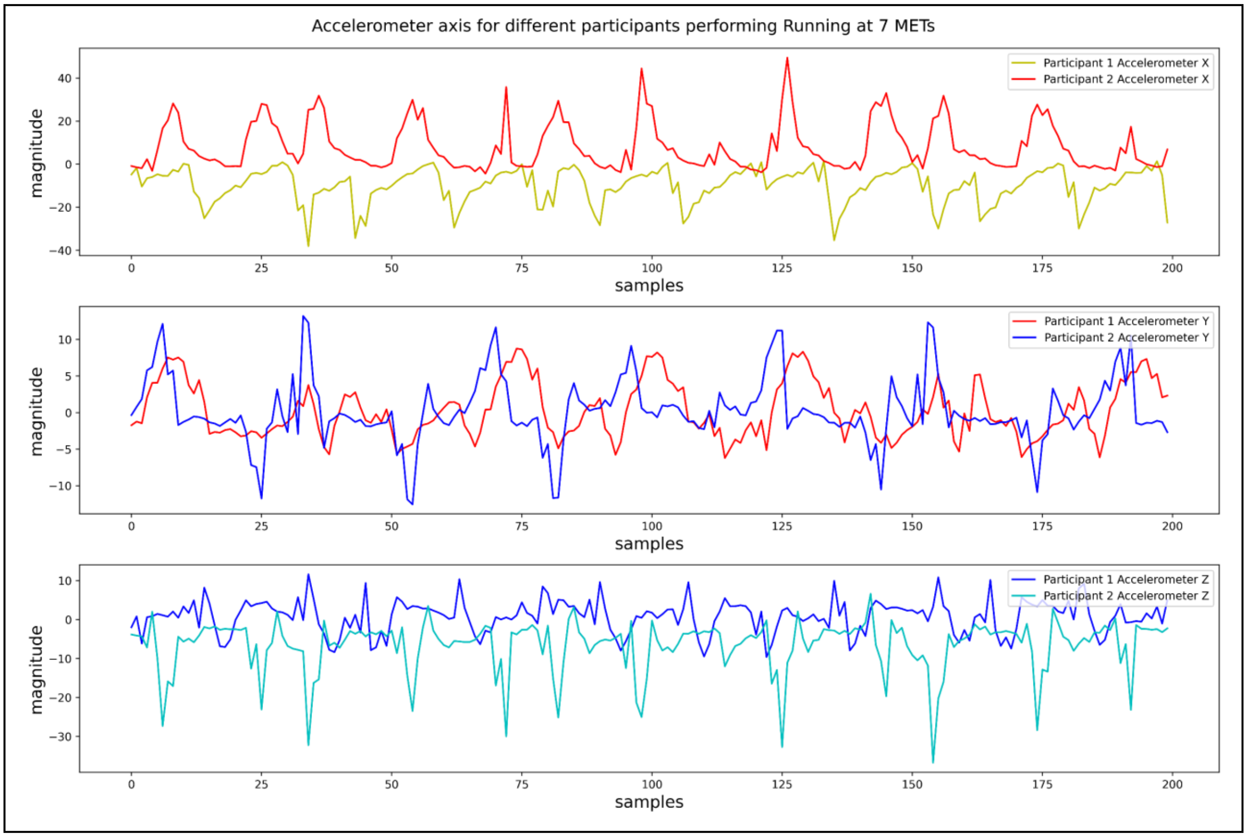


Figure 3.2. The difference in range and patterns of accelerometer axes due to orientation (Backpack)

heuristic features to eliminate this problem. The formulas to extract the 9 orientation-invariant heuristic features are given below,

$$w_1[n] = \|\overrightarrow{v_n}\| \quad (1)$$

$$w_2[n] = \|\Delta\overrightarrow{v_n}\| \quad (2)$$

$$w_3[n] = \|\Delta^2\overrightarrow{v_n}\| \quad (3)$$

$$w_4[n] = \angle(\overrightarrow{v_n}, \overrightarrow{v_{n+1}}) \quad (4)$$

$$w_4[n] = \angle(\Delta\overrightarrow{v_n}, \Delta\overrightarrow{v_{n+1}}) \tag{5}$$

$$w_4[n] = \angle(\Delta^2\overrightarrow{v_n}, \Delta^2\overrightarrow{v_{n+1}}) \tag{6}$$

$$w_7[n] = \angle(\overrightarrow{p_n}, \overrightarrow{p_{n+1}}) \text{ where } \overrightarrow{p_n} = \overrightarrow{v_n} \times \overrightarrow{v_{n+1}} \tag{7}$$

$$w_8[n] = \angle(\overrightarrow{q_n}, \overrightarrow{q_{n+1}}) \text{ where } \overrightarrow{q_n} = \Delta\overrightarrow{v_n} \times \Delta\overrightarrow{v_{n+1}} \tag{8}$$

$$w_9[n] = \angle(\overrightarrow{r_n}, \overrightarrow{r_{n+1}}) \text{ where } \overrightarrow{r_n} = \Delta^2\overrightarrow{v_n} \times \Delta^2\overrightarrow{v_{n+1}} \tag{9}$$

Here,

$\overrightarrow{v_n} = \left(v_x[n], v_y[n], v_z[n]\right)$ defines a vector where $v_x[n]$, $v_y[n]$, $v_z[n]$, were values of the accelerometer x-axis, y-axis, and z-axis, respectively, at any time sample n. $\Delta\overrightarrow{v_n} = v_{n+1} - v_n$ and $\Delta^2\overrightarrow{v_n} = v_{n+1} - v_n$, defined first-order and second-order time differences, respectively.

$$w_t = extracted\ heuristic\ features\ for\ t = 1\ to\ 9$$

$$\|\overrightarrow{m}\| = Euclidean\ norm\ of\ vector\ m$$

$$\angle(\vec{a}, \vec{b}) = cos^{-1}\left(\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\ \|\vec{b}\|}\right)$$

$$= angle\ between\ vector\ a\ and\ vector\ b\ where\ \vec{a} \cdot \vec{b}\ denotes\ their\ dot\ product$$

A more detailed explanation of the features can be found in [17]. Although they introduced the nine features mentioned and used them to eliminate the orientational effect, in a previous study[126], we found the first four features $w_1, w_2, w_{3,}$ and $w_{4,}$ to be the most significant and effective in reducing the orientational effect. Therefore, for our study, we only used the first four features.


Figure 3.3. The similarity in pattern and ranges of heuristic features while two participants performed the same activity (Backpack)

From Figure 3.3**,** we can observe that the 4 heuristic features were able to introduce enough similarity for the feature values while two different participants placed the smartphone in a backpack and performed the same activity (Running at 7 METs). Visual inspection of Figure 3.4 shows that features were able to maintain dissimilarity for the accelerometer values while two different participants placed the smartphone in a backpack and performed different activities (Sitting and Running at 5 METs). The heuristic features reduced the sensor orientation effect from the sensor values. The four features are also simple to extract, which can reduce computational complexity compared to the other feature-extracting methods with numerous features that need to be extracted to eliminate the orientational problem.

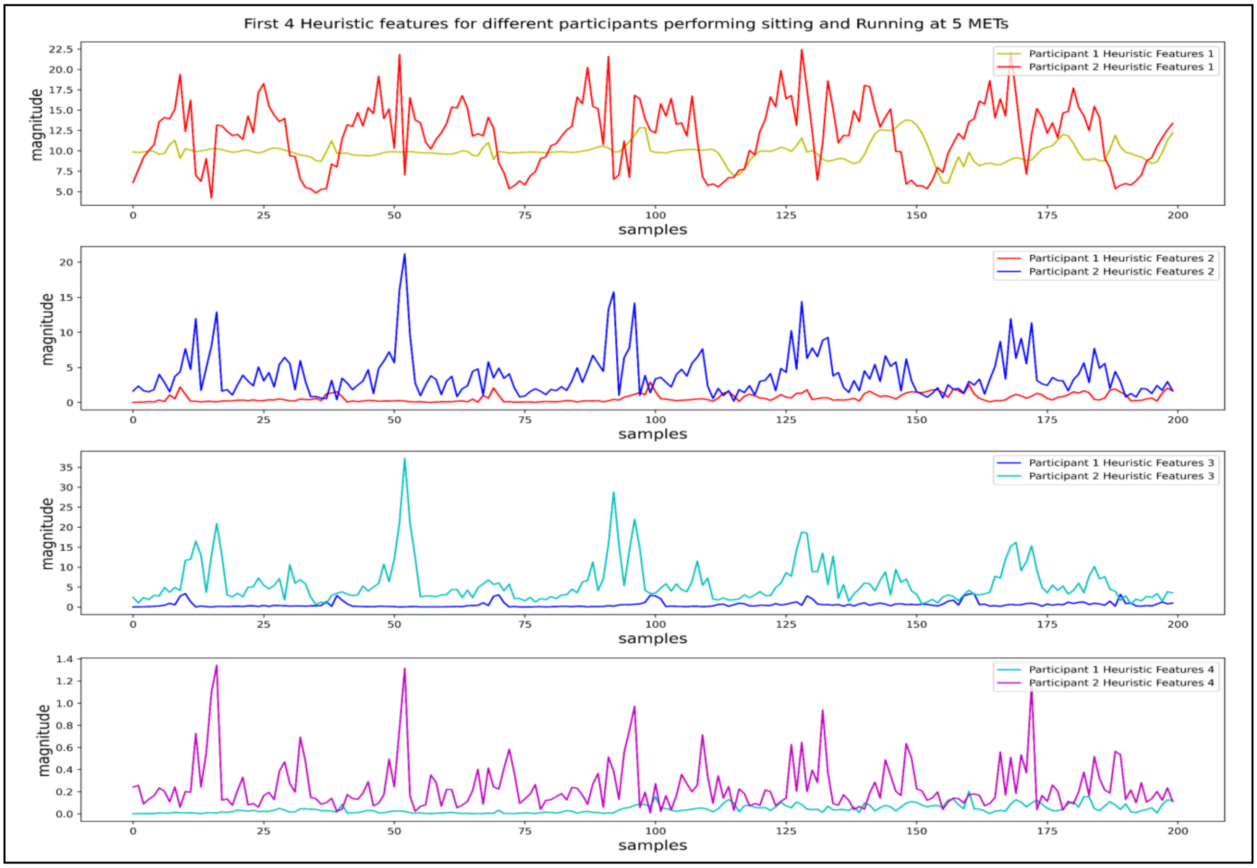


Figure 3.4. Dissimilarity in pattern and ranges of heuristic features while two participants performed different activities. (Backpack)

We investigated the patterns and ranges of the raw accelerometer values and heuristic features in different smartphone locations. The raw accelerometer values should differ in ranges and patterns for the same activity performed by different participants when keeping the smartphone in different locations. From Figure 3.5, we can observe the dissimilarity in patterns and ranges of raw accelerometer values while two participants performed running at 5-METs, keeping the smartphone in two different locations (Backpack and Pocket).

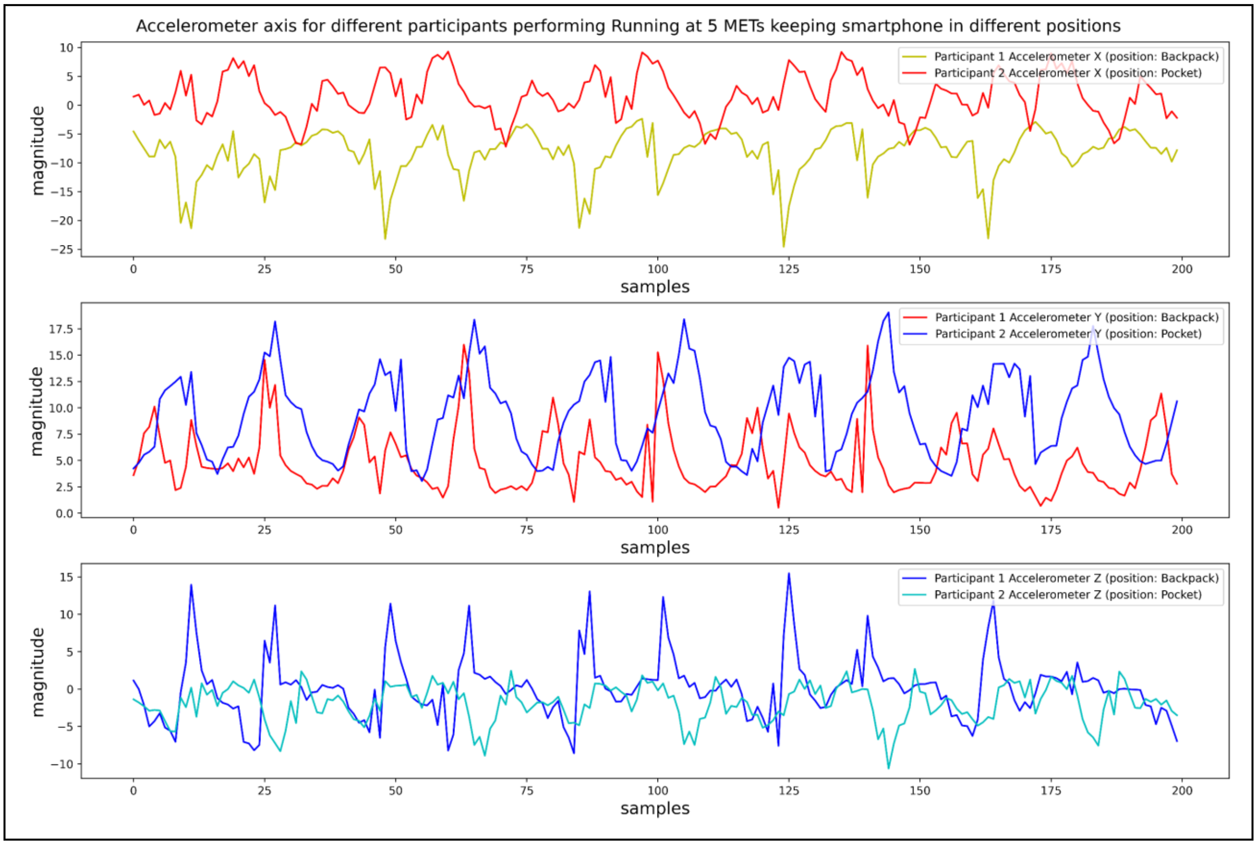


Figure 3.5. The difference in range and patterns of accelerometer axes due to different sensor placements

From Figure 3.6, we can observe that the first heuristic feature showed similarities in the values and patterns while the two participants performed the same activity, running at 5-METs by keeping the smartphones in different locations (Backpack and Pocket). For the remaining 3 heuristic features, the similarities in ranges looked promising, but the patterns differed substantially. Besides, the heuristic features could maintain the dissimilarity in the feature values and range for different activities (Sitting and Running at 5 METs) performed by different participants, keeping the smartphone in different locations (Backpack and Pocket), as depicted in Figure 3.7.

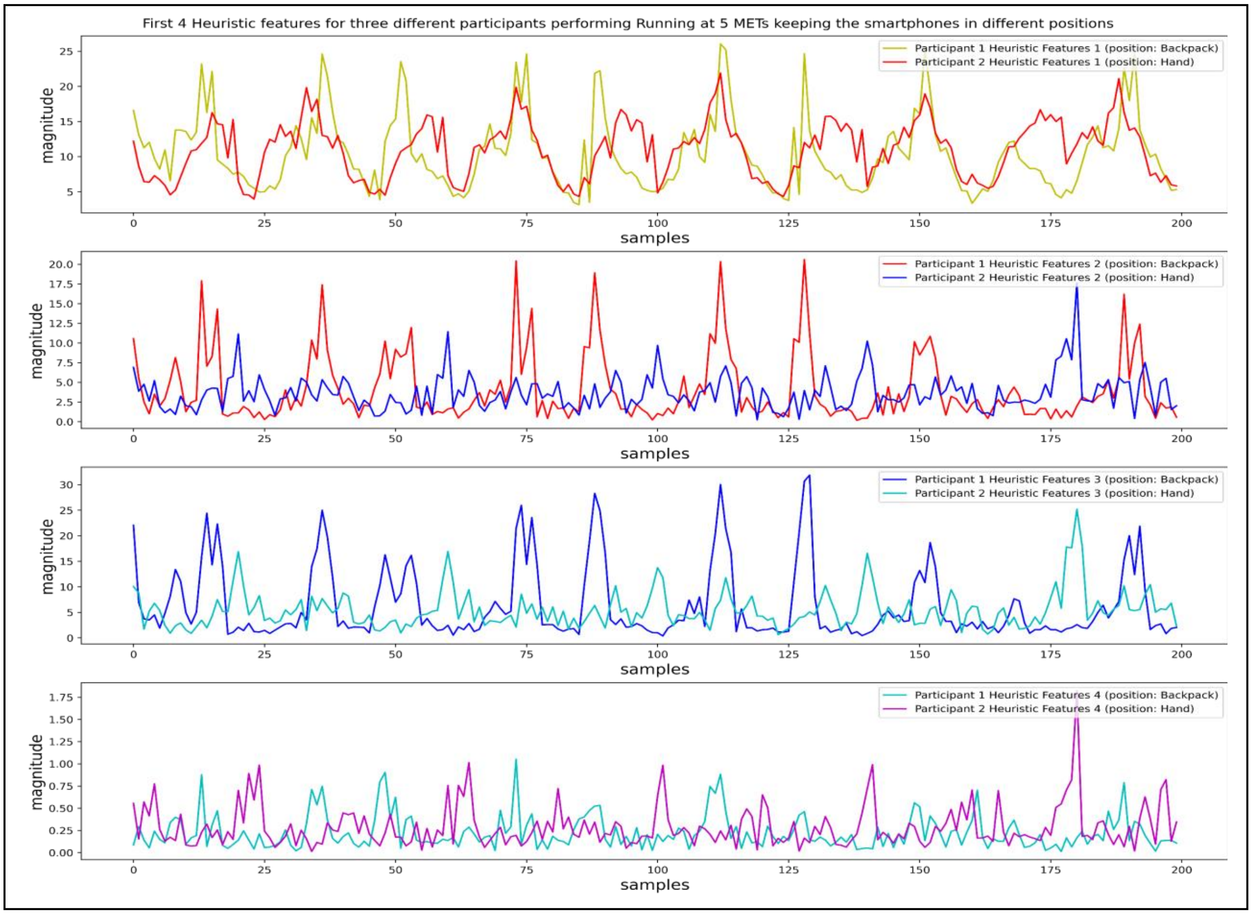


Figure 3.6. The similarity in pattern and ranges of heuristic features while two participants performed Running at 5-METs, keeping the smartphone in different locations (Backpack and Pocket)

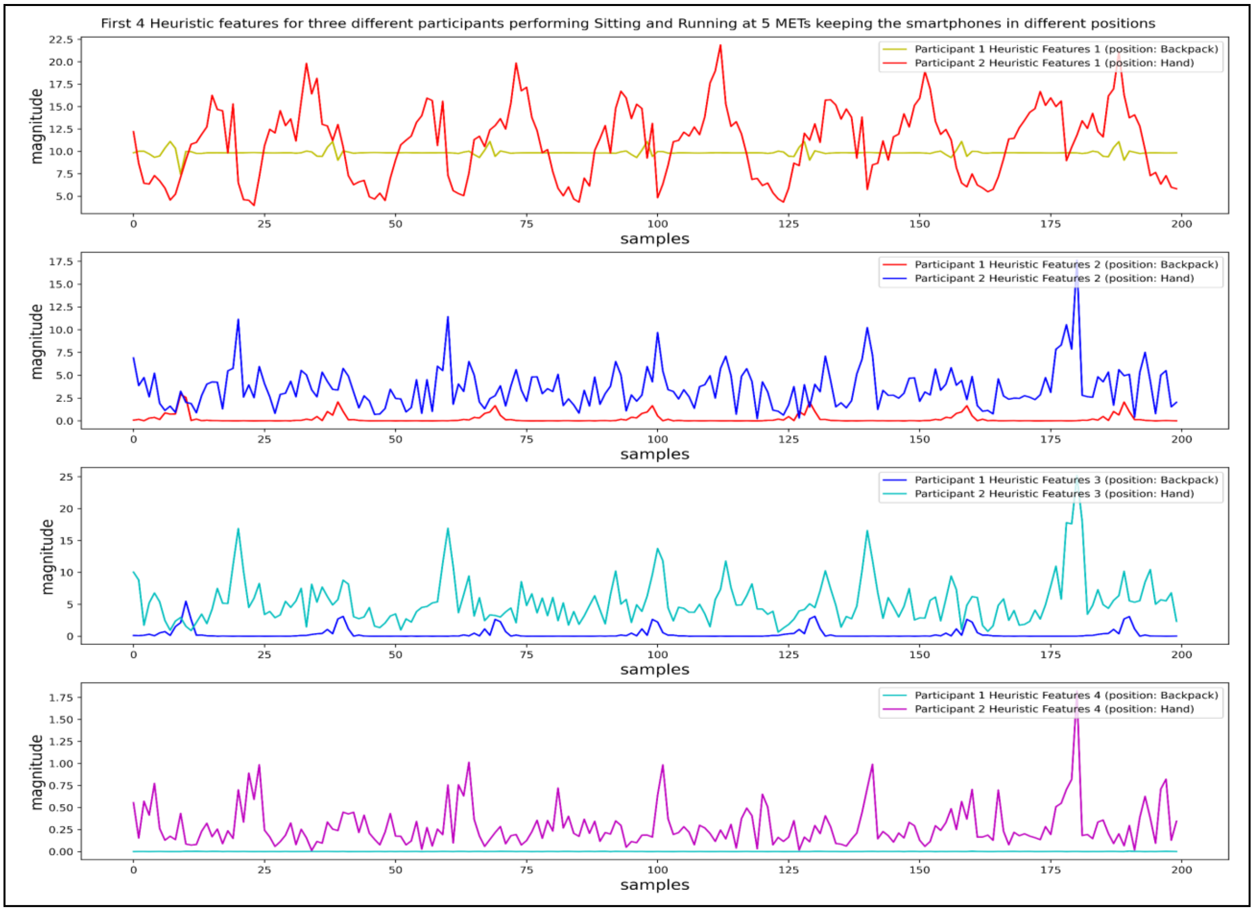


Figure 3.7. Dissimilarity in pattern and ranges of heuristic features while two participants performed Sitting and Running at 5 METs, keeping the smartphone in different locations (Backpack and Pocket)

### 3.2.4. 1D-CNN-LSTM Architecture

The structure of the 1D-CNN-LSTM model has already been discussed in section 2.2.5. This section has been repeated for the reader's convenience. In this section, we will discuss our deep-learning approach. Although the heuristic features tried to reduce the gap between the sensor values for the same activity in different placements (i.e., hand, pocket or backpack), there were still substantial differences among the sensor values for different placements.

We decided to use a hybrid 1D-CNN-LSTM Architecture to address the sensor placement problem. In our proposed model architecture, there were two major parts. The first part contained the CNN model, and the second part included the LSTM and fully connected layers. The reason behind using CNN was its automatic feature extraction capability. Generally, a CNN model takes images or data matrices as input. The convolution layer of CNN applies multiple filters or kernels on the feed images or data matrices and extracts meaningful feature maps. The number of feature maps depends on the number of filters applied. If $n$ number of filters are applied on a single data matrix, then we will get $n$ number of feature maps where each feature map will try to extract a distinctive feature for that data matrix. After extracting the data matrix, CNN uses the pooling layer to reduce the size of the filter maps.

The average or max pooling layer is used to reduce the feature map's size. The feature maps can be regarded as the automatically extracted features for the input data matrices. In the convolution layer, we propagate the kernels or filters on the data matrices in two different ways. If we propagate the filters in two directions at a time, we call the model 2D-CNN or conventional CNN. If we propagate the filters in only one direction, we call it 1D-CNN. Generally, we use 2D-CNN for images and 1D-CNN for data matrices. As mentioned earlier, CNN can be combined with LSTM to maintain temporal and spatial dependency. In an LSTM model, there can be one or more LSTM layers. Each LSTM contains multiple LSTM cells, and each LSTM cell contains three gates named Forget gate, Input gate and Output gate. We need to feed data matrices as input to the LSTM model. If a data matrix has n samples, then we can denote the samples as $t_i$ where $i = 1,2,3 \ldots \ldots n$. When the Input gate processes any particular sample $t_i$, the forget gate decides the information to preserve from the previous sample $t_{i-1}$. The output gate then combines the information from the Input gate and Forget gate to make a prediction for the current sequence or data matrix. If an LSTM model follows a CNN model, then the feature maps generated by the CNN model act as input to the LSTM model. Using the extracted feature maps of the CNN model, the LSTM model can find better temporal dependency for a sequence or data matrix. The output from the LSTM model goes to the fully connected layers made of conventional neurons to make the final prediction. We assumed that the CNN portion of the proposed architecture would be capable of bringing meaningful feature maps, which will help reduce the similarity gap in heuristic feature values observed in the case of non-identical placements of smartphones.

Our proposed 1D-CNN-LSTM architecture was designed as a classification model for classifying human activities. The 1D-CNN-LSTM model contained six convolution layers with 512, 256, 64, 128, 256, and 512 filters, followed by an LSTM layer with 512 LSTM cells. Then, we added four fully connected layers with 100, 28, 64 and 6 neurons. We had average pooling layers after the first, third and final convolution layers with a pool size of 3. We also introduced some dropout layers to reduce the overfitting issue in our model. A more detailed description of the model is depicted in Table 3.3. We used an "Adam" optimizer with a learning rate of 0.0001. We implemented the model using the programming language Python with the "Tensorflow" and "Keras" packages.

Table 3.3. The architecture of the 1D-CNN-LSTM Model

| **Parts of Architecture** | **Components of Each Part** | | | | | | |
|---|---|---|---|---|---|---|---|
| CNN | **Layer's Name** | **Number of Filters** | **Kernel Size** | **Pool Size** | **Activation Function** | **Padding Type** | **Dropout Ratio** |
| | Convolution | 512 | 5 | | relu | same | |
| | Dropout | | | | | | 0.3 |
| | Average Pooling | | | 3 | | same | |
| | Convolution | 256 | 3 | | relu | same | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Dropout | | | | | | 0.3 |
| | Convolution | 64 | 3 | | relu | same | |
| | Average Pooling | | | 3 | | same | |
| | Convolution | 128 | 3 | | relu | same | |
| | Convolution | 256 | 5 | | relu | same | |
| | Dropout | | | | | | 0.3 |
| | Convolution | 512 | 7 | | relu | same | |
| | Dropout | | | | | | 0.3 |
| | Average Pooling | | | 3 | | same | |
| LSTM | **Layer's Name** | | | **Number of Units** | | **Activation Function** | |
| | LSTM | | | 256 | | tanh | |
| Fully Connected Network | **Layer's Name** | | | **Number of Neurons** | | **Activation Function** | |
| | Dense | | | 100 | | relu | |
| | Dense | | | 28 | | relu | |
| | Dense | | | 64 | | relu | |
| | Dense | | | 6 | | softmax | |

### 3.2.5. Validation Procedure

There were data from 42 participants. We used 30 participants' data in the training phase, 10 for testing and 2 for validation. The participant's data in the validation set were constant, but the training data and test data changed as we used Leave-N-Subject-Out Cross-Validation. In our case, the value of N was 10, which made our procedure a Leave-10-Subject-Out Cross-Validation technique. As we had 40 participants' data for the training and testing phase, 4 iterations were required for the whole validation procedure. We had 10 different participants' data in the test set at each iteration. As mentioned before, we decided to inspect two separate kinds of scenarios: intra-location (i.e., sensor orientation) and inter-location (i.e., sensor placement) evaluations. In the intra-location evaluation, the 1D-CNN-LSTM model was trained and tested using the data from the same smartphone location, and in the inter-location scenario, the model was trained using data from one smartphone location but tested using the data from the other two smartphone locations. To accomplish the intra-location and inter-location evaluation, we trained the 1D-CNN-LSTM model for a particular smartphone location using the heuristic features for the 30 participants in the training set. Then, we computed the evaluation metrics using the data of 10 participants in the test set for all three smartphone locations. For instance, if the model was trained using the heuristic features of 30 participants in the training set for the pocket location, then we computed the evaluation metrics using the heuristic features of 10 participants in the test set for all three smartphone locations: pocket, hand and backpack. In this manner, both intra-location and inter-location results were accumulated for all three smartphone locations. According to our validation approach, we had to train the model 4 times to follow the Leave-10-Out Cross-Validation technique for each smartphone location. Since we were conducting our study for 3 different smartphone locations, we needed to train the model 12 times in total. We used an early-stopping technique in the training of the 1D-CNN-LSTM model. The early stopping technique was designed so that the model would stop training if the accuracy of the model for the validation data did not improve within the successive 20 epochs. The 1D-CNN-LSTM model needed the training and test data to be segmented into data matrices or data windows because 1D-CNN-LSTM works with data windows or data matrices. We segmented the training and test data in each iteration using a window length of 65 samples with an overlapping ratio of 98.46%. That means each window had 65 samples, and two consecutive windows had 64 samples in common. We used the window length of 65 because we found it to be both computational and time-efficient in our previous study [126]. The information for the validation approach is organized in Table 3.4.

Table 3.4. A detailed description of the validation procedure for each model in each iteration of the Leave-10-Out Cross-Validation technique.

| Locations of smartphone | | | Number of participants | | Parameter Values | |
|---|---|---|---|---|---|---|
| Training data | Test data for intra-location validation | Test data for inter-location validation | Training set | Test set | Window size | Batch Size |
| Pocket | Pocket | Backpack and Hand | 30 | 10 | 65 | 2000 |
| Backpack | Backpack | Pocket and Hand | 30 | 10 | 65 | 2000 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Hand | Hand | Pocket and Backpack | 30 | 10 | 65 | 2000 |

## 3.3. Results

We used data from three smartphone locations, and the users had the freedom to keep the smartphone in each smartphone location at any orientation. We used the four most common evaluation metrics for multi-class classification studies: Accuracy[100], Precision[100], Recall[100], and F1-Score[101]. Accuracy is the most suitable metric to present a classification model's overall performance. The other three metrics are well-suited to describe the model's performance for the class-specific scenario.

### 3.3.1. Results for Intra-Location Scenario

In Intra-location evaluation, we first analysed the model's overall performance for each smartphone location. Following, we performed participant-specific and activity-specific analyses.

#### 3.3.1.1. Overall Result

For the intra-location case, the model was trained and tested using the heuristic features corresponding to the same smartphone location. We computed results using the Leave-10-Out Cross-Validation procedure and averaged the test results. The average accuracy, recall, precision and F1-Score for each smartphone location are depicted in **Error! Reference source not found.**.

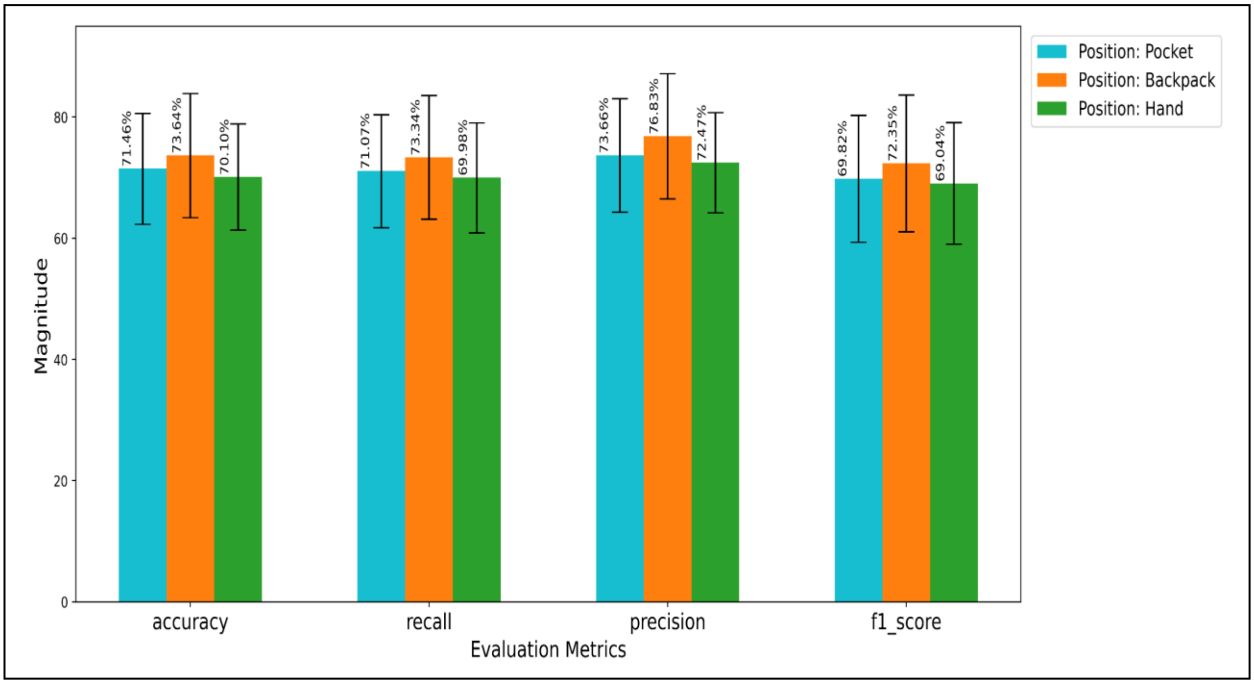


Figure 3.8. Bar plots with error bars showing averaged evaluation metrics for the intra-position scenario.

In the intra-location scenario, we achieved the highest result for the location backpack for every evaluation metric. We recorded 73.65% accuracy, 73.34% recall, 76.83% precision and 72.36% F1-score for the location backpack. We recorded the second-best results for the location pocket. For the location pocket, the accuracy, recall, precision and F1-score were 71.46%, 71.07%, 73.66% and 69.82%, respectively. Although the results were lower for the location hand among all the smartphone locations, it was not much lower if compared with the results for the location pocket. We recorded 70.10% accuracy, 69.68% recall, 72.47% precision and 69.04% F1-score for the location hand. We hypothesize better results for the backpack location because the smartphone was more stable in the backpack than in the other smartphone locations. For activities with high intensities, such as walking or running, the hand frequently moved with the body, which allowed additional variations for the values from the accelerator sensor of the smartphone. Consequently, the values for the heuristic features were affected for the location hand, and the results became lower. If we observe the overall results, the evaluation metrics ranged between 69%-74% for all the smartphone locations. We cannot consider it the best result compared to the previously conducted studies. Still, considering the number of participants, the volume of the dataset and the number of sensors, the results seem promising.

The accuracies were over 70% for all the smartphone locations, which means that the 1D-CNN-LSTM model performed decently as a classification model. The average precision and recall were good, indicating that our model tried to keep the number of false predictions lower and true predictions higher for each activity class. However, these two metrics will be more meaningful when we observe their value for the activity-specific scenario. The satisfactory F1-Scores meant that the 1D-CNN-LSTM model tried to maintain a balanced trade-off between precision and recall.

#### 3.3.1.2. Participant-Specific Scenario

We only considered accuracy as a summary metric of model performance for the participant-specific result analysis in the intra-location scenario. We wanted to observe how consistent the model's performance was for each subject. The accuracy of each participant for each smartphone location is depicted using a line plot in **Error! Reference source not found.**. For the location pocket, we achieved the highest accuracy of 88.49% for participant 20. For most participants, the accuracy ranged from 60% to 80%. However, we recorded inferior accuracy in the case of some participants, such as participants 16, 35, 37 and 38. For the location backpack, we recorded the highest accuracy of 90.29% for participant 27. The accuracy range for most participants was the same as we observed in the pocket case. We also observed very poor performance from the model for some participants, such as participants 4, 16, 22 and 37. For the location hand, the highest accuracy was 84.25% for participant 33. The overall accuracy range was the same as we observed for other smartphone locations. Again, for participants such as 19, 25 and 37, the model rendered insufficient accuracy. Considering the overall pictures, for the intra-location scenario, the performance of heuristic features can be considered sufficient and propitious. Some participants, including 16 and 37, consistently had low accuracy across all intra-location scenarios. It is somewhat unclear why this is the case, but the result is likely due to noise in the raw data.

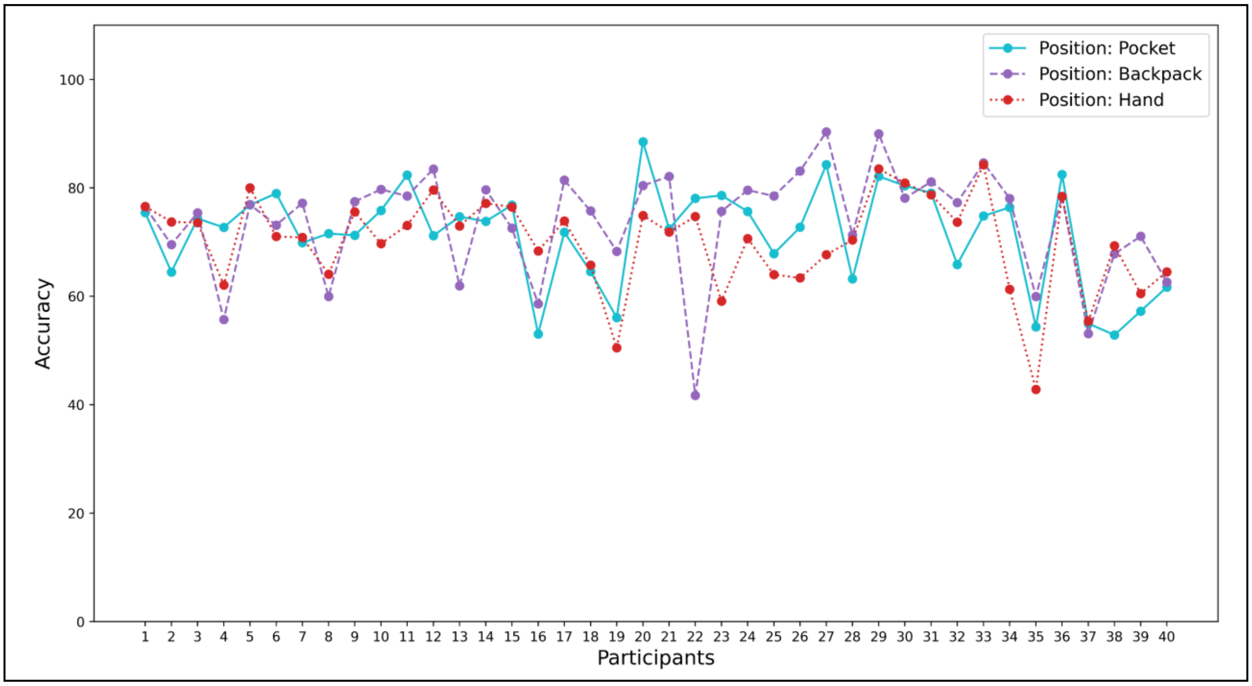


Figure 3.9. Line plot showing accuracies for all participants at every position in the intra-position scenario.

#### 3.3.1.3. Activity-Specific Scenario

We also analysed the result of the intra-location scenario for the activity-specific case. For this analysis, we considered the evaluation metrics such as recall, precision and F1-Score to demonstrate how the heuristic features performed with the help of the 1D-CNN-LSTM model for each activity class. The results for the activity-specific case are depicted in **Error! Reference**

**source not found.**. First, we will discuss the values of evaluation metrics for the pocket location. We recorded the highest precision of 86.58% for the activity "Walking". We generally expect a model to generate high precision for all the classes. Our 1D-CNN-LSTM model generated high precision for high-intensity activities such as Walking and running at 3, 5 and 7 METs for the data of the location pocket. However, the precision for low-intensity activities such as Sitting (61.61%) and Lying (63.70%) was low. This is a well-known result because sensor signals tend to be small, and models misclassify these activities.

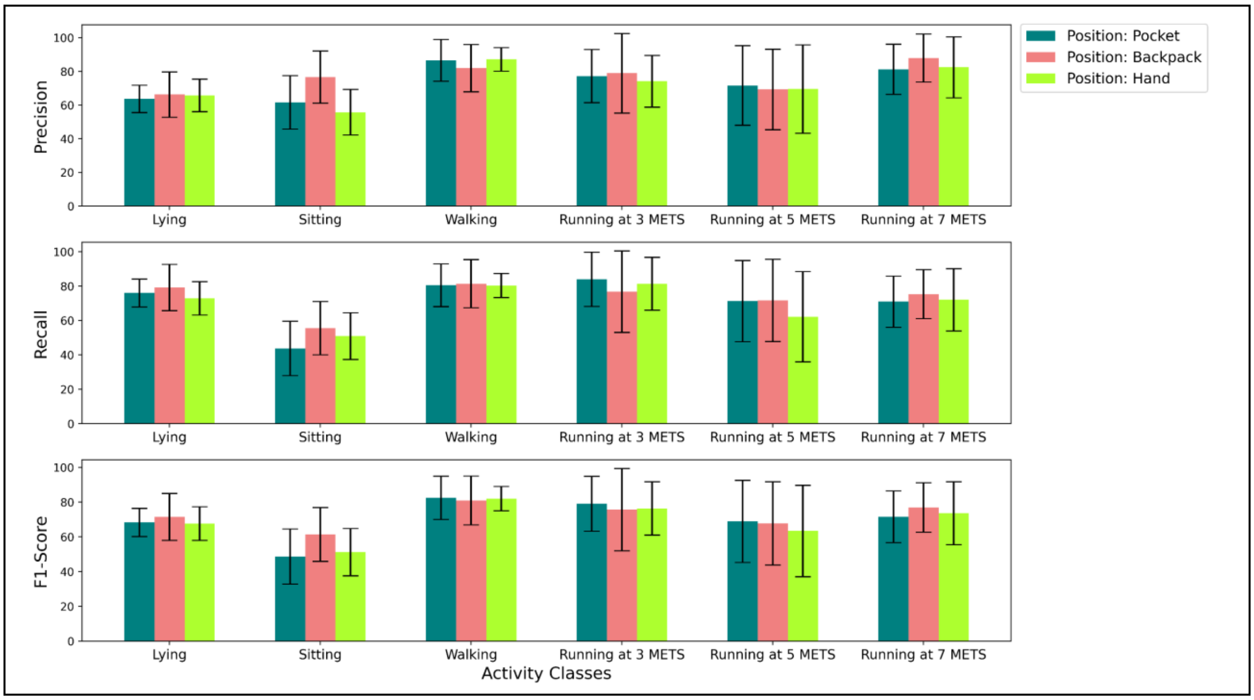


Figure 3.10. Bar plot showing activity-specific results with error bars for the intra-position scenario.

For the pocket location, all activity classes, except Sitting, had a model-generated recall greater than 70%. We recorded the highest recall of 83.96% for the activity Running at 5 METs. We also expect high recall from a classification model along with high precision. However, our model for the pocket location generated very poor recall (43.76%) for the low-intensity activity of Sitting. Now, the F1-Score in our classification model is a very important metric to consider because it explains how well-balanced our model is for precision and recall. For the pocket location, the F1-Score was promising for all the activities except for Sitting. For the activity class Sitting, the F1-Score was only 48.65%. The F1-Score was expected to be low for the activity Sitting as we experienced low precision and recall for that same activity. We acquired the highest F1-Score for the activity of walking (82.46%).

For the backpack location, we recorded the highest precision of 87.98% for activity Running at 7 METs. The precision was lower for activities such as lying (66.26%) and running at 5 METs (69.31%). The precision for the activity Sitting was poor for the pocket location; however, for the location backpack, it was improved (76.60%). The recall for the backpack location was similar to the pocket location. We recorded the lowest recall of 55.55% for the activity Sitting. The highest recall was found for the activity of Walking (81.42%). The recall ranged between 70% to 80% for all other activities. The scenario for F1-Score for the location backpack was similar to the location pocket. The lowest F1-Score was recorded for the activity of Sitting (61.41%), and the highest F1-Score was recorded for the activity of Walking (80.95%). Considering the evaluation metrics for the location backpack, our model seemed to struggle to identify the activity of Sitting correctly.

For the hand location, we expected the evaluation metrics to be poorer than the metrics for the other smartphone locations. This is because, as we mentioned before, the continuous movement of the hand during high-intensity activities causes extensive variations in the data collected from the accelerometer sensor. The precision values for the hand location had a pattern similar to that observed for the pocket location. We recorded the highest and lowest precision for the walking

(87.17%) and sitting (55.76%) activities, respectively. In the case of recall for the hand location, the scenario was the same as we observed for the pocket location.

The recall was highest for the activity Running at 3-METs (81.41%) and lowest for the activity Sitting (50.95%). The recall was poor for the activity Running at 5 METs. The lowest F1-Score for the hand location was recorded for the activity Sitting (51.25%), and the highest F1-Score was for the activity Walking (81.99%). Considering the precision, recall and F1-Score for all the smartphone locations, we found that the model struggled to recognize the activity Sitting for all three smartphone locations. For other activities, the model performed well using the heuristic features, especially for the activity of Walking.

### 3.3.2. Results for Inter-Location Scenario

In the case of the inter-location scenario, we performed the same analysis. We will start by discussing the overall results. Following, we will describe the participant-specific and activity-specific results.

#### 3.3.2.1. Overall Results

We trained our model using heuristic features extracted from the raw accelerometer data from one smartphone location and tested the model's performance using the heuristic features extracted from the raw accelerometer data from a different smartphone location. We averaged the evaluation metrics over all the iterations of the validation procedure to calculate the final overall results. The results are shown in Table 3.5. The highest accuracy was for the backpack location when the model was trained using the data from the hand location. We recorded 68.66% accuracy, 69.95% precision, 67.07% recall and 64.77% F1-Score in this case. The lowest accuracy result was recorded for the data from the hand location when the model was trained using data from the backpack location. The accuracy and F1-Score were below 60% in this case. When the model was trained using data from the pocket location and tested using data from the backpack location, we acquired results that were almost similar to the case where the model was trained using the data from hand and tested using the data from the backpack. For other cases, the metrics ranged between 62% to 66%.

Table 3.5. Averaged values of evaluation metrics for each inter-location case.

| Smartphone location for the training set | Smartphone location for test set | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Pocket | Backpack | 67.80% | 69.56% | 66.93% | 65.00% |
| | Hand | 62.39% | 63.27% | 64.20% | 60.86% |
| Backpack | Pocket | 62.62% | 66.68% | 63.59% | 61.45% |
| | Hand | 59.03% | 61.64% | 60.60% | 58.23% |
| Hand | Pocket | 64.10% | 66.09% | 62.92% | 60.38% |
| | Backpack | 68.66% | 69.95% | 67.07% | 64.77% |

We assumed to have poorer results in the case of inter-location evaluation since the training data and test data were from different smartphone locations and different participants. The values for the evaluation metrics were below 70%. However, the result seems acceptable considering the simple heuristic features and only data from a single accelerometer. The model seemed to perform the best when trained using the data from hand.

#### 3.3.2.2. Participant-Specific Result

We only considered accuracy as an evaluation metric for participant-specific evaluation. As mentioned before, the principal purpose of this analysis was to observe the number of participants for whom the model's performance was poor. The analysis is depicted graphically in Figure 3.11. When the model was trained using data from the pocket location and tested using the data from other smartphone locations, the accuracies were above 60% for most of the participants. In addition, the accuracies were consistent for each participant for every smartphone location, i.e., for a particular participant, if the model performed well for the data from the hand location, the model performed well for the data from the backpack location. However, there were some exceptions; for instance, for participant 22, the accuracy of the data from the backpack was the lowest (40.71%), but the accuracy of the data from hand was 76.81%. For participant 23, the accuracy was 42.05% when the model was tested using data from the hand location, but for the same participant, the accuracy was 75.97% when tested using data from the backpack location.

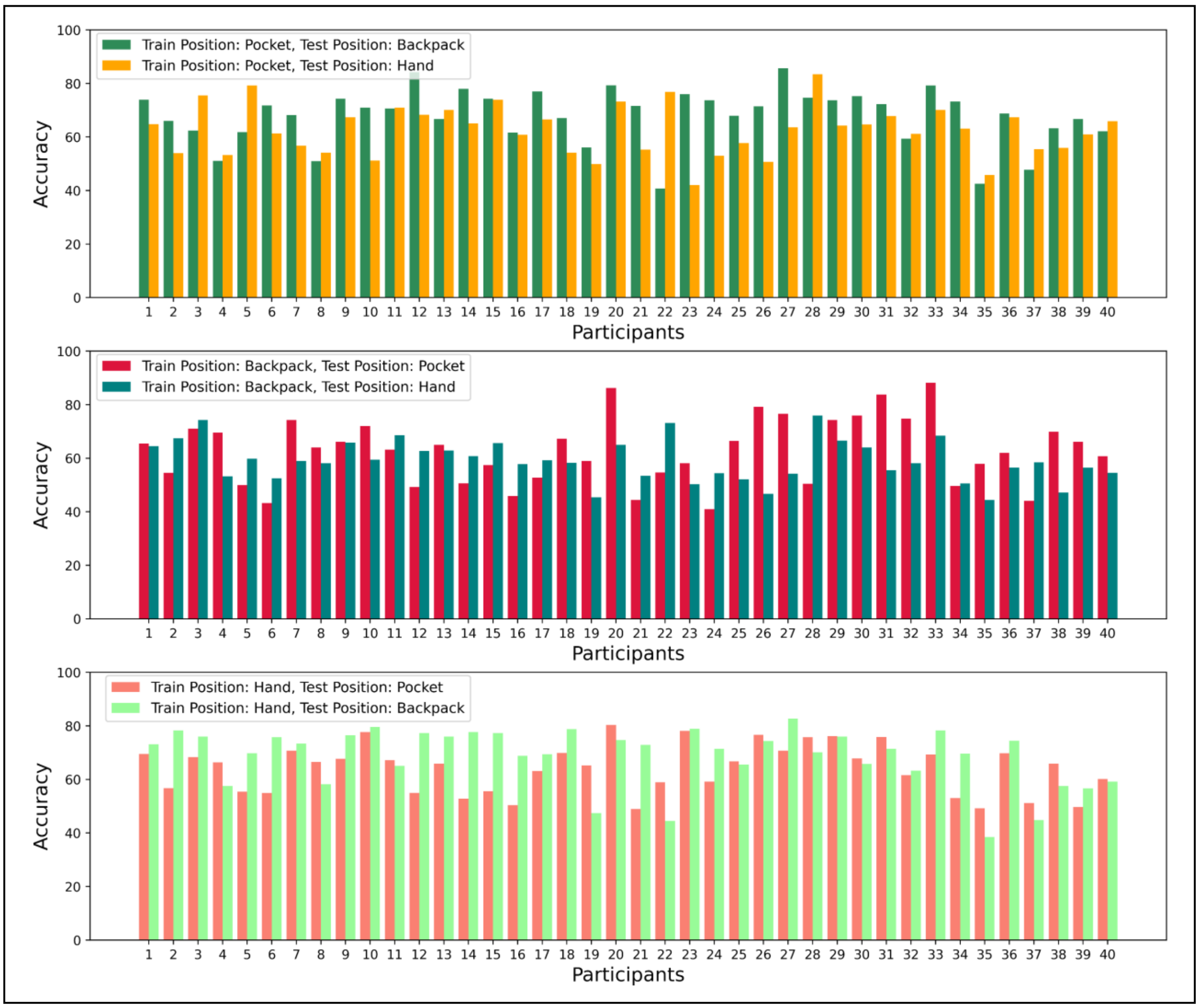


Figure 3.11. Participant-specific results for the inter-location scenario.

When the model was trained using the data from the backpack, for some participants, the model performed very well when the test data was from the pocket. For example, for participants 20, 31 and 33, the accuracies were 86.22%, 83.69% and 88.13%, respectively. However, the scenario was not the same when the test data was from the hand location. The highest accuracy recorded when the test data was from the hand location was 75.91% for Participant 28. When the test data was from the hand location, the lowest accuracy was below 47% for some participants, such as 19, 26 and 35. Similar accuracies were recorded for participants 6, 16, 21, 24 and 37 when the test data was from pocket.

For the final case, where the model was trained using data from hand and tested for two other smartphone locations, the results were better for most participants when the test data was from the backpack location. For most of the participants, the accuracies were about 70%. We recorded the highest accuracy of 82.67% for participant 27 when the training data was from the hand location, and test data was from the backpack location. Still, there were some participants, such as 19, 22, 35, and 37, for whom the accuracy was very low. When the test data was from the pocket location, we recorded the best accuracy (78.15%) for participant 21. For most of the participants, accuracies were around 60%, except for participants 21, 35 and 39, when the accuracies were below 50%.

#### 3.3.2.3. Activity-Specific Result

For the activity-specific results, we will discuss each evaluation metric for all the cases individually. The evaluation metrics for each inter-location scenario and every activity class are depicted in Figure 3.12. For precision, we can see that the lowest values were found for the activity Sitting. For all the inter-location cases, the precision for the activity Sitting was around 50%. We encountered similar results for intra-location cases. The model found it difficult to identify the activity Sitting correctly in both inter-location and intra-location situations. The precision for the activity Walking was satisfactory for all the inter-location cases and was around 80%. For the activity Lying, we experienced precision ranging between 50% to 60%. This means the model mislabelled many samples from other high-intensity activities to Sitting and Lying. For the activity Running at 3 METs, the precision was lower than 60% for two cases, and in both cases, the test data was from the hand location. For other cases, the precision was around 70%. For the activity Running at 5 METs, the precision was approximately 70% when test data was from the backpack location. For other cases, the precisions were around 60%. Finally, for the activity Running at 7 METs, the precision was approximately 80% except for two cases. The test data were from the pocket in both cases, and the precision was around 65%. Observing the precision for all activity classes, it is clear that the model had lower accuracy from low-intensity activities, which decreased overall precision.

Regarding the recall for all the inter-location cases, we recorded the poorest performance for two activity classes, Sitting and Running, at 5 METs. For the activity Sitting, we recorded poor precision (between 20% and 30%) in three cases. In two of those three cases, the test data was from the backpack location, and the other had data from the pocket location as test data and training data from the hand location. For other cases, the precision was between 55% to 70%. Poor precision and recall for the activity Sitting means that the model mislabelled other activities as the activity Sitting and mislabelled many samples from Sitting activities to other activities.

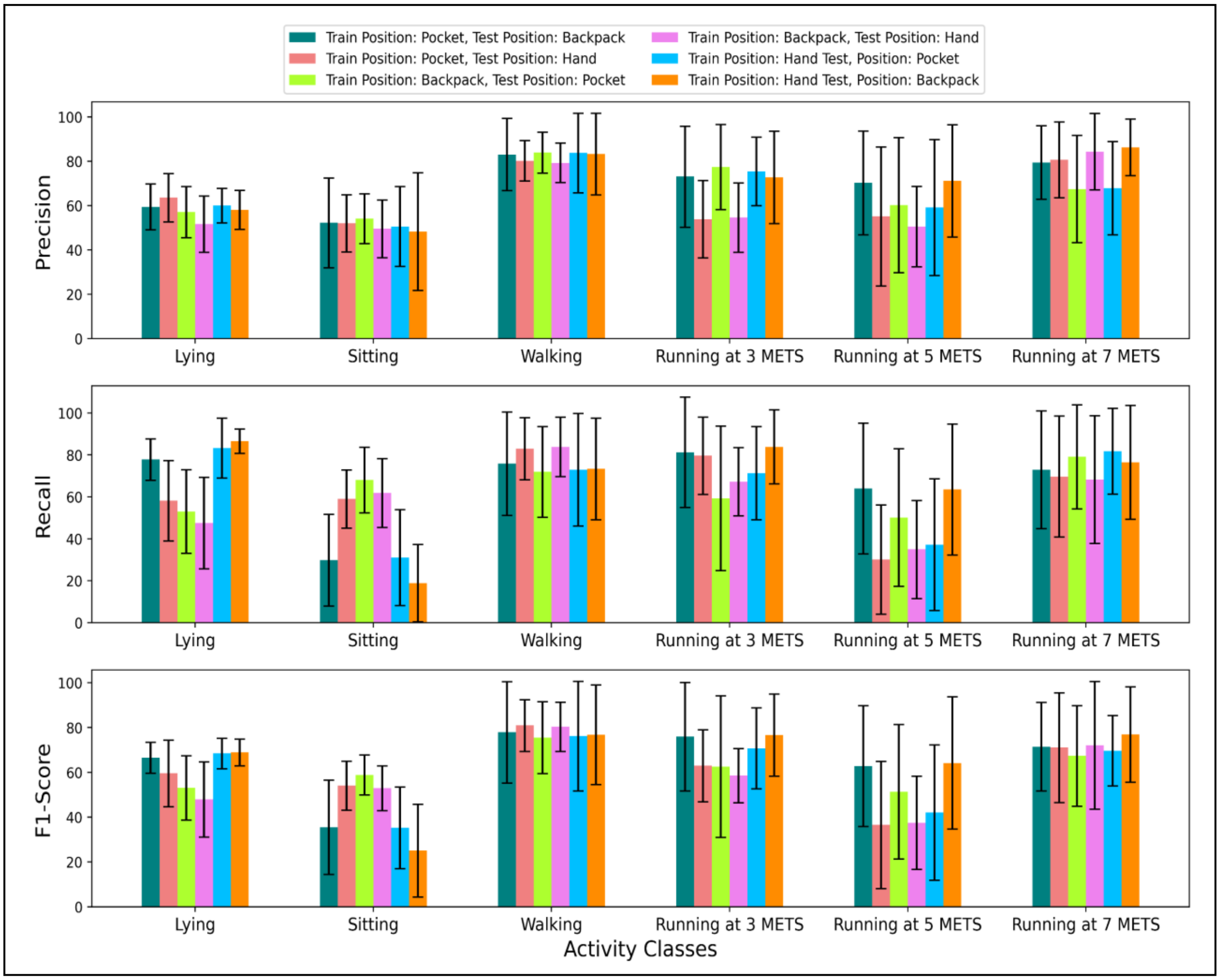


Figure 3.12. Activity-specific results for the inter-location scenario.

Although the precision was around 60% for all the cases of activity Lying, the recall was comparatively better and around 80% for three cases. In two of those three cases, the train data was from hand, and the other case had training data from the pocket location and test data from the backpack location. For Walking and Running at 3-METs activities, the recall was satisfactory and ranged between 70% and 80% for most inter-location cases. As mentioned before, the recall was poor for the activity Running at 5 METs. The recall (above 60%) was comparatively better for the activity running at 5 METs only when the test data was from the backpack. For the activity

Running at 7 METs, the recall was satisfactory. After observing the precision and recall, one significant finding was that the metrics were always better when the test data was from the backpack location.

In the case of F1-Score, the result was similar to the previous two metrics as it is a harmonic mean of recall and precision. The F1-Score was promising for the activity Walking, at approximately 80%, and Running at 7-METs, at approximately 70%, for all the cases. The F1-Score was low for the

activities Sitting and Running at 5-METs. The low value of F1-Score for the activities Sitting and Running was expected as we experienced low precision and recall for those activities. For the activities Lying and Running at 3-METS, the F1-Score was better for three cases. Among those three cases, two cases had the data from the hand as train data, and in the other case, the test data was from the backpack, and the train data was from the pocket.

Considering all evaluation metrics for all the inter-location cases, we can conclude that the heuristic features with the 1D-CNN-LSTM model struggled with low-intensity activities in both intra-location and inter-location cases. Along with the low-intensity activities, we also found poor performance for the activity Running at 5-METs.

## 3.4. Discussion

In our study, we wanted to explore the effectiveness of heuristic features in reducing the effect of sensor orientation and sensor placement on sensor data with the help of a 1D-CNN-LSTM model. We collected data from only a smartphone accelerometer sensor. We had 42 participants, and we followed the Leave-10-Subject-Out Cross-Validation approach. Our study had two types of analysis: intra-location (i.e., sensor orientation) and inter-location (i.e., sensor placement) analyses. In the intra-location scenario, we checked the effectiveness of the heuristic features on a dataset where the orientations were introduced during data collection. In the study[17], where the heuristic features were introduced, they evaluated the performance of the heuristic features on five public datasets, which they named A [33], B [103], C [83], D [104], and E [105]. These datasets were accumulated using sensors fixed at a constant orientation. So, the datasets had orientation information for only one orientation. Their study claimed that the heuristic features removed the orientational information from the dataset. Since the orientation information was removed, the dataset simulated the scenario where the orientation of the sensor did not matter anymore. However, their dataset did not represent a practical scenario where multiple orientations can be present. Since, in our case, different orientations of smartphone accelerometer were ensured, we were able to evaluate the performance of the heuristic features in a practical scenario. In addition, the dataset on which [17] performed the evaluation had a low volume of data. They had the highest number of data windows for dataset C (30 subjects); the number of data windows was 10,299, with a 50% overlapping ratio. In our case, we had around 4 million data windows for each smartphone location. For 3 smartphone locations, the number of data windows was about 12 million. Moreover, we had 42 participants' data, which offered more diversity for our dataset. Their study followed the Leave-1-Subject-Out Cross Validation approach, whereas Our Leave-10-Subject-Out Cross Validation approach ensured a more practical test case where 10 test participants offered completely unseen data to the model. In their study, three of their five datasets had more than one type of sensor. Datasets B and E only used a single accelerometer as we did. To be brief, our study protocol was more practical and simulated a real-life scenario, ensuring a more reliable evaluation of the heuristic features.

The study [23] followed two validation approaches: P-Fold Cross Validation and Leave-1-Out Cross-Validation. The Leave-1-Out Cross Validation approach and our Leave-10-Out Cross Validation approach tested the model using data from participants unseen by the model. As mentioned earlier, they introduced 9 heuristic features. They used these features in dividing them into 3 sets. The first set had the first 3 heuristic features, the second set had the first 6 features, and the third one had all 9 heuristic features. They evaluated the effectiveness of these features for 4 different classifiers: Bayesian Decision Making (BDM), K-Nearest Neighbour (KNN), Support Vector Machine (SVM) and Artificial Neural Network. In Table 3.6, we have tabulated the best result found for each dataset using the heuristic features, number of features used, name of the classifier, types of sensors, number of sensor units and the same information in our intra-location cases. We only included intra-location cases because their study was conducted to solve sensor orientation for a fixed smartphone location. We found better accuracies for all the intra-location cases when compared with the accuracies they found for datasets C, D and E. Moreover, from their results, we can see that they found the best result for three of their datasets using only the first 3 heuristic features. So, their study's findings supported selecting the first four heuristic features based on feature importance. In addition, the satisfactory results we found for intra-location cases depict that the heuristic features are effective in solving the orientation problem even for practical scenarios.

Table 3.6. Accuracy comparison between the result obtained in [23] and our study.

| Dataset | Types of Sensors | Number of sensor units | Classifier | Number of Features | Best Accuracy |
|---|---|---|---|---|---|
| A | Accelerometer, Gyroscope, Magnetometer | 5 | SVM | 6 | 77.66% |
| B | Accelerometer | 4 | SVM | 3 | 86.92% |
| C | Accelerometer, Gyroscope | 1 | SVM | 3 | 66.69% |

| | | | | | |
|---|---|---|---|---|---|
| D | Accelerometer, Gyroscope | 1 | SVM | 3 | 50.62% |
| E | Accelerometer | 1 | SVM | 9 | 55.19% |
| Our Dataset (Pocket) | Accelerometer | 1 | 1D-CNN-LSTM | 4 | 71.46% |
| Our Dataset (Backpack) | Accelerometer | 1 | 1D-CNN-LSTM | 4 | 73.65% |
| Our Dataset (Hand) | Accelerometer | 1 | 1D-CNN-LSTM | 4 | 70.10% |

In our study, we also presented the performance of the heuristic features in participant-specific and activity-specific scenarios. In participant-specific cases, we found that, for most of the participants at each smartphone location, the accuracies were around 70%. There were some participants for which the models' performance was reduced drastically. Moreover, the participants for whom we found poor performance for the model changed according to the sensor placement. For activity-specific scenarios, in intra-location cases, we found that the heuristic features work better for high-intensity activities. We recorded poor precision, recall, and F1-Score for low-intensity activities such as lying and sitting.

Regarding the inter-location cases, we found comparatively better results when we trained the model using the data from the hand location. The worst result we found was when we trained the model using the data from the backpack location. One interesting finding is the model trained using the data from the hand location, which encompasses data of comparatively more variations because of the frequent movement of the hand, showed the highest accuracy when using backpack or pocket location test data. On the contrary, the model performed poorly when it was trained using the data from the backpack, which encompasses fewer data variations because of the less frequent movements of the backpack. The difference between the values of the evaluation metrics for intra-location and inter-location cases was approximately 10%. The accuracies for the inter-location cases were around 65%. The model performance was informative, considering we used data from one single accelerometer and simple heuristic features. The heuristics features were particularly proposed for solving the orientation problem, and we wanted to find out the heuristic features' effectiveness in the placement problem. The performance of the heuristic features in inter-location cases indicates that if we fuse the heuristic features with other proposed approaches to solve the sensor-placement issue, then there is a high chance that the performance will increase. In addition, for both intra-location and inter-location cases, if we use other types of sensors, such as a Gyroscope and Magnetometer along with an accelerometer, we may find better results using the heuristic features.

We had some interesting findings regarding the participant-specific and activity-specific scenarios for inter-location cases. We observed that, in some cases, when we had good accuracy for a particular subject's data from a particular smartphone location, we had low accuracy for that same subject's data from a different smartphone location. Such a case was for Participant 22 when the model was trained using the data from the backpack location and tested using the data from the other two smartphone locations. Similar to the intra-location scenario, in inter-location scenarios, we observed that the models performed poorly for some participants in every inter-location case, reducing the average accuracy for all cases. In the activity-specific scenario, the findings' pattern was similar to those for intra-location cases. The heuristic features could not perform well for low-intensity activities, but the result was good for high-intensity activities, especially for Walking. However, among the high-intensity activities, the performance for Running at 5-METs was unsatisfactory.

In summary, the performance of the heuristic features with 1D-CNN-LSTM was promising in both intra-location and inter-location cases. We used data from only one accelerometer and performed a Leave-10-Subject-Out Cross Validation approach. We tried to replicate a practical scenario for a machine learning model and evaluate the performance of the heuristic feature in such cases. For inter-location cases, using other types of sensors might help. An interesting future study would be to observe how the heuristic features perform if fused with existing or newly collected data designed to solve the sensor placement problem.

Our study had research gaps that we would like to explore in future work. We explore the effectiveness of the heuristic features using one type of model. There are several other classifiers which could be promising. For our dataset, we used signals from a single smartphone accelerometer. We need to explore the results if we include signals from a gyroscope and other smartphone sensors. We can fuse the heuristic features with other proposed techniques for solving the sensor placement problem and evaluate its effectiveness for inter-location cases. We only conducted our study for six activities. In future, we intend to conduct the same study with more activities and variations. However, we think that exploring the effectiveness of the heuristic features for different smartphone locations in a practical manner would help other studies have a proper idea about the potency of the heuristic features and develop accuracy by using other techniques with these heuristic features.

## 3.5. Conclusion

This study aimed to examine if simple heuristic features could help solve the sensor orientation and sensor placement problems when conducting HAR using a single smartphone accelerometer. Our study concludes that the heuristic features adequately solve the sensor orientation problem despite a very simple study protocol.

**Author Contributions:** Conceptualization, Arnab Barua and Xianta Jiang; Data curation, Daniel Fuller; Formal analysis, Arnab Barua; Funding acquisition, Xianta Jiang and Daniel Fuller;

Investigation, Arnab Barua, Daniel Fuller and Xianta Jiang; Methodology, Arnab Barua; Project administration, Xianta Jiang and Daniel Fuller; Resources, Daniel Fuller; Software, Arnab Barua; Supervision, Xianta Jiang and Daniel Fuller; Validation, Arnab Barua, Xianta Jiang and Daniel Fuller; Visualization, Arnab Barua; Writing – original draft, Arnab Barua; Writing – review & editing, Xianta Jiang and Daniel Fuller.

**Funding:** This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant number RGPIN-2020-05525.

**Institutional Review Board Statement:** The study was by the Memorial University Interdisciplinary Committee on Ethics in Human Research (ICEHR #20180188-EX).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to ethical constraints.

**Conflicts of Interest:** The authors declare no conflict of interest.

# Chapter 4

# Discussion and Conclusion

In this thesis, I represented two major findings. I depicted how window lengths affected the performance of 1D-CNN-LSTM in recognizing six human activities: lying, sitting, walking, and running at 3-METS, 5-METs and 7-METS. I also inspected how simple heuristic features effectively solved sensor orientation and placement problems. For the study, I only used data from a single smartphone accelerometer sensor from 42 participants for three smartphone locations: pocket, hand and backpack.

To observe the effect of window length, I only used data from the smartphone location of the pocket. To solve the orientation problem, I used the heuristic features. I observed that the performance of 1D-CNN-LSTM increased at first as the window length increased. After a window length of 55, the performance did not improve considerably with an increment in window lengths. I recorded the highest average accuracy of 80.91% at a window length of 105. The average accuracy at a window length of 55 was 79.74%. I can see that the highest average accuracy and the accuracy at the window length of 55 had a shallow difference. This suggests that window length impacts the performance of 1D-CN-LSTM until a certain window length. After that certain window length, the performance of the model gets saturated. Therefore, when using deep classifiers like 1D-CNN-LSTM, I need not select a very high window length for better performance. I should investigate the effects and choose a window length where the performance becomes almost constant. In this way, I can reduce the resource and time complexity I encounter in case of long window length. Another interesting finding from this part was that the values of evaluation metrics such as precision, recall, and F1-Score were reduced for low-intensity activities such as lying and sitting when I increased the window length. The increment in the window length worked oppositely for high and low-intensity activities. Therefore, researchers should also select a window length at which the model performs well for high and low-intensity activities. It should be mentioned that for this part, I followed the Leave-1-Subject-Out Cross-Validation technique, but for my second study, I followed the Leave-10-Subject-Out Cross-Validation technique, so the evaluation scenario in my second part became harder for the model.

In the second part of my study, I investigated the effectiveness of the heuristic features in solving sensor orientation and sensor placement problems for different smartphone locations: pocket, hand and backpack. In my first part, along with investigating the effects of the window length, I also observed how effective the heuristic features were in solving the orientation problem for one smartphone location: pocket. I introduced more challenges for the heuristic features in the second part of the study. Instead of one smartphone location, I wanted to observe how effective the heuristic features were in solving the sensor orientation problem in three different smartphone locations. I called this investigation intra-location evaluation. Furthermore, instead of the Leave-1-Subject-Out Cross-Validation technique, I followed the Leave-10-Subject-Out Cross-Validation technique, making the evaluation process more challenging for both the 1D-CNN-LSTM model and the heuristic features. My intra-location evaluation found that the average accuracy for all three smartphone locations was close, around 70%-73%. The values of all the evaluation metrics, accuracy, precision, recall, and F1-Score, were higher for the location backpack. I assumed the reason behind achieving a better result for the location of the backpack was the smartphone remained steadier in the backpack compared to the location pocket and hand. The second-best result I achieved was for the location pocket. The poorest result was found for the location hand as there was a possibility of the smartphone being the shakiest when carried in hand. I also performed participant-specific analysis for intra-location evaluation. I found that the accuracy was very low for some participants, which reduced the overall average accuracy for all the participants. However, for different smartphone locations, I had a poor performance for different participants. I also performed activity-specific analysis for intra-location evaluation. I found that the heuristic features with the 1D-CNN-LSTM model performed better for high-intensity activities, especially for walking and running at 7-METs. For low-intensity activities, the values of evaluation metrics were very poor, especially for an activity like sitting. I observed that the heuristic features were more effective for high-intensity activities. After evaluating the intra-location scenario, I opted to check the effectiveness of the heuristic features for the inter-location scenarios, i.e., solving the sensor placement problem.

In inter-location evaluation, I found the best values of evaluation metrics from my model when I trained it using the data from hand and tested it using the data from the other two smartphone locations. The lowest performance I recorded was when I trained the model using the data from the backpack. The average results for all inter-location scenarios were between 59%-69%. After inspecting the overall performance, I conducted a participant-specific analysis. I found that for all inter-location scenarios, the models' accuracies were low for some participants, which lowered the overall accuracies. Another interesting finding was that the accuracy differed considerably for some participants if the test data were from different smartphone locations. I finally performed activity-specific analysis for all inter-location scenarios. From the activity-specific analysis, I found similar results for the activity classes I found in the intra-location evaluation. The values of the evaluation metrics were higher for high-intensity activities and lower for low-intensity activities. I recorded the worst performance for the activity of sitting and the best for the activity of walking. However, I found that the heuristic features in solving the placement problem were not considered effective. However, the effectiveness of heuristic features can be considered acceptable when accounting for the high number of variations in the dataset, simple study protocol, and Leave-10-Subject-Out Cross-Validation technique. It should be mentioned that for the second study, when I trained the 1D-CNN-LSTM model for intra-location and inter-location scenarios, I used a window length found suitable according to the investigation of my first study.

My study evaluated the effectiveness of heuristic features for the raw data from a single smartphone accelerometer signal. I am assuming that the result would have been better if I considered other sensors, such as a gyroscope and magnetometer. So, in future, I can explore the effectiveness of the heuristic features using more sensor types. In addition, many studies proposed different approaches to solve the sensor placement problem. Researchers can fuse one of those approaches with the heuristic features to see how much improvement they may get in solving the sensor placement problem. I also performed my investigation using only one type of classifier. I need to perform the same investigation on other types of classifiers to see the performance difference for different classifiers. Furthermore, I conducted very little data pre-processing, but I may get better results with further pre-processing. In my study, I considered just six activities. In the future, I plan to do the same investigation for other activities.

In brief, my study depicted the influence of window length on the performance of the 1D-CNN-LSTM, which will help other studies choose the proper window length for their studies with HAR. I also presented the effectiveness of the heuristic features for different smartphone locations. Other studies may fuse these heuristic features with other proposed methods and improve the performance of HAR. Finally, my findings will help other researchers understand the influence of different smartphone locations in HAR when using a smartphone.

# References


[1] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, “Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring,” *IEEE transactions on information technology in biomedicine,* vol. 10, no. 1, pp. 156-167, 2006.

[2] M. J. Mathie, B. G. Celler, N. H. Lovell, and A. C. Coster, “Classification of basic daily movements using a triaxial accelerometer,” *Medical and Biological Engineering and Computing,* vol. 42, pp. 679-687, 2004.

[3] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, “Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers,” *Pattern recognition letters,* vol. 29, no. 16, pp. 2213-2220, 2008.

[4] S. Ha, and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors." pp. 381-388.

[5] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "Human activities recognition using accelerometer and gyroscope." pp. 357-362.

[6] M. Webber, and R. F. Rojas, “Human activity recognition with accelerometer and gyroscope: A data fusion approach,” *IEEE Sensors Journal,* vol. 21, no. 15, pp. 16979-16989, 2021.

[7] A. K. M. Masum, E. H. Bahadur, A. Shan-A-Alahi, M. A. U. Z. Chowdhury, M. R. Uddin, and A. Al Noman, "Human activity recognition using accelerometer, gyroscope and magnetometer sensors: Deep neural network approaches." pp. 1-6.

[8] K. Altun, and B. Barshan, "Human activity recognition using inertial/magnetic sensor units." pp. 38-51.

[9] V. Bijalwan, V. B. Semwal, and V. Gupta, “Wearable sensor-based pattern mining for human activity recognition: Deep learning approach,” *Industrial Robot: the international journal of robotics research and application,* vol. 49, no. 1, pp. 21-33, 2022.

[10] T. Dobhal, V. Shitole, G. Thomas, and G. Navada, “Human activity recognition using binary motion image and deep learning,” *Procedia computer science,* vol. 58, pp. 178-185, 2015.

[11] F. Niu, and M. Abdel-Mottaleb, "View-invariant human activity recognition based on shape and motion features." pp. 546-556.

[12] O. D. Lara, and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE communications surveys & tutorials,* vol. 15, no. 3, pp. 1192-1209, 2012.

[13] J. Morales, and D. Akopian, “Physical activity recognition by smartphones, a survey,” *Biocybernetics and Biomedical Engineering,* vol. 37, no. 3, pp. 388-400, 2017.

[14] A. Yurtman, B. Barshan, and B. Fidan, “Activity recognition invariant to wearable sensor unit orientation using differential rotational transformations represented by quaternions,” *Sensors,* vol. 18, no. 8, pp. 2725, 2018.

[15] R. Yang, and B. Wang, “PACP: A location-independent activity recognition method using smartphone sensors,” *Information,* vol. 7, no. 4, pp. 72, 2016.

[16] Y. Chen, and C. Shen, “Performance analysis of smartphone-sensor behavior for human activity recognition,” *Ieee Access,* vol. 5, pp. 3095-3110, 2017.

[17] A. Yurtman, and B. Barshan, “Activity recognition invariant to sensor orientation with wearable motion sensors,” *Sensors,* vol. 17, no. 8, pp. 1838, 2017.

[18] A. Rasekh, C.-A. Chen, and Y. Lu, “Human activity recognition using smartphone,” *arXiv preprint arXiv:1401.8212*, 2014.

[19] U. R. Alo, H. F. Nweke, Y. W. Teh, and G. Murtaza, “Smartphone motion sensor-based complex human activity identification using deep stacked autoencoder algorithm for enhanced smart healthcare system,” *Sensors,* vol. 20, no. 21, pp. 6300, 2020.

[20] P. Siirtola, and J. Röning, "User-independent human activity recognition using a mobile phone: Offline recognition vs. real-time on device recognition." pp. 617-627.

[21] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition."

[22] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors." pp. 197-205.

[23] Y. Xu, and T. T. Qiu, "Human activity recognition and embedded application based on convolutional neural network," *Journal of Artificial Intelligence and Technology,* vol. 1, no. 1, pp. 51-60, 2021.

[24] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications,* vol. 25, pp. 743-755, 2020.

[25] N. Varshney, B. Bakariya, A. K. S. Kushwaha, and M. Khare, "Human activity recognition by combining external features with accelerometer sensor data using deep learning network model," *Multimedia Tools and Applications,* vol. 81, no. 24, pp. 34633-34652, 2022.

[26] B. Almaslukh, J. Al Muhtadi, and A. M. Artoli, "A robust convolutional neural network for online smartphone-based human activity recognition," *Journal of Intelligent & Fuzzy Systems,* vol. 35, no. 2, pp. 1609-1620, 2018.

[27] M. Inoue, S. Inoue, and T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *Artificial Life and Robotics,* vol. 23, pp. 173-185, 2018.

[28] M. Alessandrini, G. Biagetti, P. Crippa, L. Falaschetti, and C. Turchetti, "Recurrent neural network for human activity recognition in embedded systems using PPG and accelerometer data," *Electronics,* vol. 10, no. 14, pp. 1715, 2021.

[29] A. Murad, and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors,* vol. 17, no. 11, pp. 2556, 2017.

[30] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access,* vol. 8, pp. 56855-56866, 2020.

[31] S. Mekruksavanich, and A. Jitpattanakul, "Smartwatch-based human activity recognition using hybrid lstm network." pp. 1-4.

[32] S. Mekruksavanich, and A. Jitpattanakul, "Lstm networks using smartphone data for sensor-based human activity recognition in smart homes," *Sensors,* vol. 21, no. 5, pp. 1636, 2021.

[33] B. Barshan, and M. C. Yüksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Computer Journal,* vol. 57, no. 11, pp. 1649-1667, 2014.

[34] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." p. 3.

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

[36] Z. Chen, C. Jiang, and L. Xie, "A novel ensemble ELM for human activity recognition using smartphone sensors," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 5, pp. 2691-2699, 2018.

[37] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, and A. Aliverti, "A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone," *Sensors,* vol. 19, no. 17, pp. 3731, 2019.

[38] R. Mondal, D. Mukherjee, P. K. Singh, V. Bhateja, and R. Sarkar, "A new framework for smartphone sensor-based human activity recognition using graph neural network," *IEEE Sensors Journal,* vol. 21, no. 10, pp. 11461-11468, 2020.

[39] A. Subasi, A. Fllatah, K. Alzobidi, T. Brahimi, and A. Sarirete, "Smartphone-based human activity recognition using bagging and boosting," *Procedia Computer Science,* vol. 163, pp. 54-61, 2019.

[40] F. Hernández, L. F. Suárez, J. Villamizar, and M. Altuve, "Human activity recognition on smartphones using a bidirectional LSTM network." pp. 1-5.

[41] M. O. Gani, T. Fayezeen, R. J. Povinelli, R. O. Smith, M. Arif, A. J. Kattan, and S. I. Ahamed, "A light weight smartphone based human activity recognition system with high accuracy," *Journal of Network and Computer Applications,* vol. 141, pp. 59-72, 2019.

[42] G. Chetty, M. White, and F. Akther, "Smart phone based data mining for human activity recognition," *Procedia Computer Science,* vol. 46, pp. 1181-1187, 2015.

[43] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "On the personalization of classification models for human activity recognition," *IEEE Access,* vol. 8, pp. 32066-32079, 2020.

[44] H. Nematallah, S. Rajan, and A.-M. Cretu, "Logistic model tree for human activity recognition using smartphone-based inertial sensors." pp. 1-4.

[45] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco, and M. R. Luaces, "A public domain dataset for real-life human activity recognition using smartphone sensors," *Sensors,* vol. 20, no. 8, pp. 2200, 2020.

[46] H. Guo, L. Chen, G. Chen, and M. Lv, "Smartphone-based activity recognition independent of device orientation and placement," *International Journal of Communication Systems,* vol. 29, no. 16, pp. 2403-2415, 2016.

[47] Z. Chen, Q. Zhu, Y. C. Soh, and L. Zhang, "Robust human activity recognition using smartphone sensors via CT-PCA and online SVM," *IEEE Transactions on Industrial Informatics,* vol. 13, no. 6, pp. 3070-3080, 2017.

[48] Y. E. Ustev, O. Durmaz Incel, and C. Ersoy, "User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal." pp. 1427-1436.

[49] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine." pp. 216-223.

[50] D. N. Tran, and D. D. Phan, "Human activities recognition in android smartphone using support vector machine." pp. 64-68.

[51] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Training computationally efficient smartphone–based human activity recognition models." pp. 426-433.

[52] P. Paul, and T. George, "An effective approach for human activity recognition on smartphone." pp. 1-3.

[53] S. Kaghyan, and H. Sarukhanyan, "Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer," *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria,* vol. 1, pp. 146-156, 2012.

[54] L. Fan, Z. Wang, and H. Wang, "Human activity recognition model based on decision tree." pp. 64-68.

[55] O. D. Lara, and M. A. Labrador, "A mobile platform for real-time human activity recognition." pp. 667-671.

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

[57] M. T. Uddin, and M. A. Uddiny, "Human activity recognition from wearable sensors using extremely randomized trees." pp. 1-6.

[58] S. Gupta, and A. Kumar, "Human activity recognition through smartphone's tri-axial accelerometer using time domain wave analysis and machine learning," *International Journal of Computer Applications,* vol. 975, pp. 8887, 2015.

[59] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "LSTM networks for mobile human activity recognition." pp. 50-53.

[60] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, “Deep residual bidir-LSTM for human activity recognition using wearable sensors,” *Mathematical Problems in Engineering,* vol. 2018, pp. 1-13, 2018.

[61] M. Milenkoski, K. Trivodaliev, S. Kalajdziski, M. Jovanov, and B. R. Stojkoska, "Real time human activity recognition on smartphones using LSTM networks." pp. 1126-1131.

[62] K. Muhammad, A. Ullah, A. S. Imran, M. Sajjad, M. S. Kiran, G. Sannino, and V. H. C. de Albuquerque, “Human action recognition using attention based LSTM network with dilated CNN features,” *Future Generation Computer Systems,* vol. 125, pp. 820-830, 2021.

[63] S. Deep, and X. Zheng, "Hybrid model featuring CNN and LSTM architecture for human activity recognition on smartphone sensor data." pp. 259-264.

[64] M. A. Khatun, M. A. Yousuf, S. Ahmed, M. Z. Uddin, S. A. Alyami, S. Al-Ashhab, H. F. Akhdar, A. Khan, A. Azad, and M. A. Moni, “Deep CNN-LSTM with self-attention model for human activity recognition using wearable sensor,” *IEEE Journal of Translational Engineering in Health and Medicine,* vol. 10, pp. 1-16, 2022.

[65] S. K. Challa, A. Kumar, and V. B. Semwal, “A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data,” *The Visual Computer*, pp. 1-15, 2021.

[66] N. Sikder, and A.-A. Nahid, “KU-HAR: An open dataset for heterogeneous human activity recognition,” *Pattern Recognition Letters,* vol. 146, pp. 46-54, 2021.

[67] S. Mekruksavanich, and A. Jitpattanakul, “Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models,” *Electronics,* vol. 10, no. 3, pp. 308, 2021.

[68] X. Zhou, W. Liang, I. Kevin, K. Wang, H. Wang, L. T. Yang, and Q. Jin, “Deep-learning-enhanced human activity recognition for Internet of healthcare things,” *IEEE Internet of Things Journal,* vol. 7, no. 7, pp. 6429-6438, 2020.

[69] A. Barua, A. K. M. Masum, E. H. Bahadur, M. R. Alam, M. Chowdhury, and M. S. Alam, “Human activity recognition in prognosis of depression using long short-term memory approach,” *International Journal of Advanced Science and Technology,* vol. 29, pp. 4998-5017, 2020.

[70] A. Chelli, and M. Pätzold, “A machine learning approach for fall detection and daily living activity recognition,” *IEEE Access,* vol. 7, pp. 38670-38687, 2019.

[71] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges,” *Expert Systems with Applications,* vol. 105, pp. 233-261, 2018.

[72] A. D. Antar, M. Ahmed, and M. A. R. Ahad, "Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review." pp. 134-139.

[73] C. Jobanputra, J. Bavishi, and N. Doshi, “Human activity recognition: A survey,” *Procedia Computer Science,* vol. 155, pp. 698-703, 2019.

[74] S. Ramasamy Ramamurthy, and N. Roy, “Recent trends in machine learning for human activity recognition—A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 8, no. 4, pp. e1254, 2018.

[75] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, “Deep learning models for real-time human activity recognition with smartphones,” *Mobile Networks and Applications,* vol. 25, no. 2, pp. 743-755, 2020.

[76] R. Mutegeki, and D. S. Han, "A CNN-LSTM approach to human activity recognition." pp. 362-366.

[77] A. Ignatov, “Real-time human activity recognition from accelerometer data using Convolutional Neural Networks,” *Applied Soft Computing,* vol. 62, pp. 915-922, 2018.

[78] D. Micucci, M. Mobilio, and P. Napoletano, “Unimib shar: A dataset for human activity recognition using acceleration data from smartphones,” *Applied Sciences,* vol. 7, no. 10, pp. 1101, 2017.

[79] A. K. M. Masum, M. E. Hossain, A. Humayra, S. Islam, A. Barua, and G. R. Alam, "A statistical and deep learning approach for human activity recognition." pp. 1332-1337.

[80] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, “Physical human activity recognition using wearable sensors,” *Sensors,* vol. 15, no. 12, pp. 31314-31338, 2015.

[81] N. Ahmed, J. I. Rafiq, and M. R. Islam, “Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model,” *Sensors,* vol. 20, no. 1, pp. 317, 2020.

[82] A. Barna, A. K. M. Masum, M. E. Hossain, E. H. Bahadur, and M. S. Alam, "A study on human activity recognition using gyroscope, accelerometer, temperature and humidity data." pp. 1-6.

[83] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones." pp. 437-442.

[84] A. Reiss, and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring." pp. 108-109.

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

[86] S. Hochreiter, and J. Schmidhuber, “Long short-term memory,” *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[87] G. M. Weiss, K. Yoneda, and T. Hayajneh, “Smartphone and smartwatch-based biometrics using activities of daily living,” *IEEE Access,* vol. 7, pp. 133190-133202, 2019.

[88] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, and A. Ferscha, "Collecting complex activity datasets in highly rich networked sensor environments." pp. 233-240.

[89] T. Lv, X. Wang, L. Jin, Y. Xiao, and M. Song, “Margin-based deep learning networks for human activity recognition,” *Sensors,* vol. 20, no. 7, pp. 1871, 2020.

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

[91] Y. Chen, and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer." pp. 1488-1492.

[92] N. Y. Hammerla, S. Halloran, and T. Plötz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” *arXiv preprint arXiv:1604.08880*, 2016.

[93] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, “InnoHAR: A deep neural network for complex human activity recognition,” *Ieee Access,* vol. 7, pp. 9893-9902, 2019.

[94] B. Almaslukh, A. M. Artoli, and J. Al-Muhtadi, “A robust deep learning approach for location-independent smartphone-based human activity recognition,” *Sensors,* vol. 18, no. 11, pp. 3726, 2018.

[95] S. J. Khataeipour, J. R. Anaraki, A. Bozorgi, M. Rayner, F. A. Basset, and D. Fuller, “Predicting lying, sitting and walking at different intensities using smartphone accelerometers at three different wear locations: hands, pant pockets, backpack,” *BMJ Open Sport & Exercise Medicine,* vol. 8, no. 2, pp. e001242, 2022.

[96] E. Data, "Ethica [Mobile app]," 2020.

[97] M. Jetté, K. Sidney, and G. Blümchen, “Metabolic equivalents (METS) in exercise testing, exercise prescription, and evaluation of functional capacity,” *Clinical cardiology,* vol. 13, no. 8, pp. 555-565, 1990.

[98] A. Narayanan, F. Desai, T. Stewart, S. Duncan, and L. Mackay, “Application of raw accelerometer data and machine-learning techniques to characterize human movement behavior: a systematic scoping review,” *Journal of Physical Activity and Health,* vol. 17, no. 3, pp. 360-383, 2020.

[99] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks." pp. 4580-4584.

[100] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2020.

[101] Y. Sasaki, "The truth of the F-measure," *Teach tutor mater,* vol. 1, no. 5, pp. 1-5, 2007.

[102] A. Dehghani, O. Sarbishei, T. Glatard, and E. Shihab, "A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors," *Sensors,* vol. 19, no. 22, pp. 5026, 2019.

[103] W. Ugulino, D. Cardador, K. Vega, E. Velloso, R. Milidiú, and H. Fuks, "Wearable computing: Accelerometers' data classification of body postures and movements." pp. 52-61.

[104] M. Zhang, and A. A. Sawchuk, "USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors." pp. 1036-1043.

[105] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device." pp. 289-296.

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

[107] M. Ullah, H. Ullah, S. D. Khan, and F. A. Cheikh, "Stacked lstm network for human activity recognition using smartphone data." pp. 175-180.

[108] T. Yu, J. Chen, N. Yan, and X. Liu, "A multi-layer parallel lstm network for human activity recognition with smartphone sensors." pp. 1-6.

[109] N. Tufek, M. Yalcin, M. Altintas, F. Kalaoglu, Y. Li, and S. K. Bahadir, "Human action recognition using deep learning methods on limited sensory data," *IEEE Sensors Journal,* vol. 20, no. 6, pp. 3101-3112, 2019.

[110] S. Mekruksavanich, and A. Jitpattanakul, "Deep convolutional neural network with rnns for complex activity recognition using wrist-worn wearable sensor data," *Electronics,* vol. 10, no. 14, pp. 1685, 2021.

[111] E. H. Bahadur, A. K. M. Masum, A. Barua, and M. Z. Uddin, "Active Sense: Early Staging of Non-Insulin Dependent Diabetes Mellitus (NIDDM) Hinges upon Recognizing Daily Activity Pattern," *Electronics,* vol. 10, no. 18, pp. 2194, 2021.

[112] A. Subasi, M. Radhwan, R. Kurdi, and K. Khateeb, "IoT based mobile healthcare system for human activity recognition." pp. 29-34.

[113] A. Jalal, S. Kamal, and D. Kim, "A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments," *Sensors,* vol. 14, no. 7, pp. 11735-11759, 2014.

[114] Z. Zhuang, and Y. Xue, "Sport-related human activity detection and recognition using a smartwatch," *Sensors,* vol. 19, no. 22, pp. 5001, 2019.

[115] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE transactions on knowledge and data engineering,* vol. 20, no. 8, pp. 1082-1090, 2008.

[116] A. Wang, G. Chen, J. Yang, S. Zhao, and C.-Y. Chang, "A comparative study on human activity recognition using inertial sensors in a smartphone," *IEEE Sensors Journal,* vol. 16, no. 11, pp. 4566-4578, 2016.

[117] D. O. Olguın, and A. S. Pentland, "Human activity recognition: Accuracy across common locations for wearable sensors." pp. 11-14.

[118] M. Zhang, and A. Sawchuk, "A feature selection-based framework for human activity recognition using wearable multimodal sensors."

[119] A. K. M. Masum, A. Barua, E. H. Bahadur, M. R. Alam, M. A. U. Z. Chowdhury, and M. S. Alam, "Human activity recognition using multiple smartphone sensors." pp. 468-473.

[120] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, “A robust human activity recognition system using smartphone sensors and deep learning,” *Future Generation Computer Systems,* vol. 81, pp. 307-313, 2018.
[121] C. A. Ronao, and S.-B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors." pp. 46-53.
[122] A. M. Khan, Y.-K. Lee, S.-Y. Lee, and T.-S. Kim, "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis." pp. 1-6.
[123] A. Filippoupolitis, W. Oliff, B. Takand, and G. Loukas, “Location-enhanced activity recognition in indoor environments using off the shelf smart watch technology and BLE beacons,” *Sensors,* vol. 17, no. 6, pp. 1230, 2017.
[124] M. Shoaib, O. D. Incel, H. Scholten, and P. Havinga, "Smokesense: Online activity recognition framework on smartwatches." pp. 106-124.
[125] J. Knighten, S. McMillan, T. Chambers, and J. Payton, "Recognizing social gestures with a wrist-worn smartband." pp. 544-549.
[126] A. Barua, D. Fuller, S. Musa, and X. Jiang, “Exploring Orientation Invariant Heuristic Features with Variant Window Length of 1D-CNN-LSTM in Human Activity Recognition,” *Biosensors,* vol. 12, no. 7, pp. 549, 2022.