# A Novel Algorithm for Human Activity Determination Feature Selection and Classification

**Win Win Myo**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Science (International Program)**
**Prince of Songkla University**
**2019**

# A Novel Algorithm for Human Activity Determination Feature Selection and Classification

**Win Win Myo**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Science (International Program)**
**Prince of Songkla University**
**2019**

| | |
|---|---|
| **Dissertation Title** | A Novel Algorithm for Human Activity Determination Feature Selection and Classification |
| **Author** | Mrs. Win Win Myo |
| **Major Program** | Computer Science (International Program) |

---

**Major Advisor**

……
(Asst. Prof. Dr. Pattara Aiyarak)

**Co-advisor**

……
(Asst. Prof. Dr. Wiphada Wettayaprasit)

**Examining Committee:**

……Chairperson
(Asst. Prof. Dr. Jittat Fakcharoenphol)

…… Committee
(Asst. Prof. Dr. Ladda Preechaveerakul)

…… Committee
(Asst. Prof. Dr. Jarunee Duangsuwan)

…… Committee
(Asst. Prof. Dr. Pattara Aiyarak)

…… Committee
(Asst. Prof. Dr. Wiphada Wettayaprasit)

The Graduate School, Prince of Songkla University, has approved this dissertation as partial fulfillment of the requirements for the Doctor of Philosophy Degree in Computer Science (International Program).

……
(Prof. Dr. Damrongsak Faroongsang)
Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

……Signature
(Asst. Prof. Dr.  Pattara Aiyarak)
Major Advisor

……Signature
(Asst. Prof. Dr. Wiphada Wettayaprasit)
Co-advisor

……Signature
(Mrs. Win Win Myo)
Candidate

I hereby certified that this work had not been accepted in substance for any degree, and is not being currently submitted in candidature for any degree.

……. Signature
(Mrs. Win Win Myo)
Candidate

| | |
|---|---|
| **Dissertation Title** | A Novel Algorithm for Human Activity Determination Feature Selection and Classification |
| **Author** | Mrs. Win Win Myo |
| **Major Program** | Computer Science (International Program) |
| **Academic Year** | 2019 |

## ABSTRACT

Human Activity Determination (HAD) used the integrated sensors in a mobile phone is a very active research field to predict the everyday activities of humans in Machine Learning (ML). However, there are numerous challenges to develop systematically in the HAD system. In ML, feature selection is a critical issue due to containing redundant or irrelevant features to represent the target activity. Because of the large size of the dataset and the complexity of the features concerned, HAD relies mainly on feature selection to improve robustness and precision.

The primary objective of this research is to discover the best ideal classifier model-based feature selection technique for HAD and Machine Learning (ML) problems. As the aim of this, an Artificial Neural Network (ANN) model using Multi-Layer Perceptron (MLP) is designed with formulating to discover the number of hidden nodes in neuron. In addition, two novel feature selection methods: 1) LDC (Linearly Dependent Concepts) using linearly dependent concepts, and 2) CAT (Cyclic Attribution Technique) using group theory and basic properties of cyclic group are identified. Three datasets (UCI-HAR, DATASET-UCI dataset, HAPT) and five different classifiers (SVM, BAG, KNN, CART, and BAYES) support to conduct the statistical and comparative analyses.

Based on all systematic experiments, the performances with running time compared with each other. Although the CAT feature selection method could reduce more 30% of features than the LDC feature selection method, the accuracy of model-based LDC is better than the CAT model. The study concluded that the MLP model-based LDC approach was the most comprehensively applicable and effective methodology for HAD systems development.

**ACKNOWLEDGEMENTS**

Firstly, I would like to express my sincere gratitude to my advisor Asst. Prof. Dr. Pattara Aiyarak for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time for research and writing of this dissertation.

To my co-advisor, Asst. Prof. Dr. Wiphada Wettayaprasit, I would like to express my appreciation for being an excellent mentor for me. I would like to be grateful to her for encouraging my study and giving the valuable guidelines.

I would also like to thank the members of my dissertation committee, Asst. Prof. Dr. Jittat Fakcharoenphol (Department of Computer Engineering, Kasetsart University), Asst. Prof. Dr. Ladda Preechaveerakul, and Dr. Jarunee Duangsuwan for their insightful comments, suggestions, and encouragement to finish my work.

My sincere thanks also go to the department head, all lecturers, all staffs, and all friends at the Computer Science department of Prince of Songkla University (PSU) for encouraging me and letting me through all the difficulties.

A very special gratitude goes to the officials of Prince of Songkla University for supporting the PSU President scholarship (Ph.D. degree) and Graduate School of PSU for supporting Financial Research Fund in Fiscal year 2018.

I am grateful to the researchers for their datasets (UCI-HAR dataset, DATASET-UCI-HAR dataset, and UCI-HAPT dataset) which are publicly available at the UCI repository in the field of human activity recognition (available from http://archive.ics.uci.edu/ml).

Sincere thanks are extended to Dr. Hla Htay, the rector (Former) of Dagon University, Yangon, Myanmar for allowing me to study here and encouraging me throughout my study. Finally, a special thanks to my family: my parents, my husband, my son, my daughter, my sisters, and my brothers for supporting me throughout my study life in general.

**Win Win Myo**

# TABLE OF CONTENTS


| | Page |
|---|---|
| ABSTRACT | V |
| ACKNOWLEDGEMENT | VI |
| LIST OF TABLES | X |
| LIST OF FIGURES | XIII |
| LIST OF ACRONYMS | XIV |
| LIST OF NOMENCLATURES | XVII |
| LIST OF GREEK SYMBOLS | XVIII |
| LIST OF SUBSCRIPTS | XVIII |
| **CHAPTER** | |
| **1 INTRODUCTION** | **1** |
| 1.1 Research Problems | 2 |
| 1.1.1 High Dimensionality Dataset | 3 |
| 1.1.2 Redundant Features | 3 |
| 1.1.3 Sensitivity of Classifiers | 3 |
| 1.2 Understanding HAD | 4 |
| 1.3 Research Contributions and Objectives | 5 |
| 1.4 Outline | 7 |
| **2 BACKGROUNDS AND REVIEWS** | **8** |
| 2.1 Review on human activity determination (had) | 8 |
| 2.1.1 Main had process | 8 |
| 2.1.2 Sensors for had | 9 |
| 2.1.3 Human activity | 12 |
| 2.2.1 Data collection | 14 |
| 2.2.2 Step counting | 16 |

# TABLE OF CONTENTS (Continued)


| | | Page |
|---|---|---|
| 2.3 | A review on feature selection and classification | 18 |
| 2.3.1 | Feature selection techniques | 18 |
| 2.3.2 | Classification techniques | 22 |
| 2.4 | A review on datasets and main ideas on feature selection | 25 |
| 2.4.1 | dataset | 25 |
| 2.4.2 | linearly dependent concepts | 31 |
| 2.4.3 | group theory | 38 |
| **3** | **METHODOLOGY** | **41** |
| 3.1 | Task_1 (understanding work in had) | 41 |
| 3.1.1 | Data collection | 41 |
| 3.1.2 | Step counting | 45 |
| 3.2 | Task_2 (implementing the system design) | 49 |
| 3.2.1 | Dataset | 50 |
| 3.2.2 | Data pre-processing | 52 |
| 3.2.3 | Feature selection | 55 |
| 3.2.4 | Designing ANN classifier [ii] | 63 |
| 3.2.5 | MLP model | 66 |
| 3.2.6 | Performance evaluation | 67 |
| **4** | **RESULTS AND DISCUSSIONS** | **70** |
| 4.1 | Step counting | 70 |
| 4.3 | LDC feature selection method | 77 |
| 4.3.1 | Results of feature selection by LDC | 77 |
| 4.3.2 | Performance evaluations to the whole dataset based on LDC | 79 |
| 4.3.2 | Performance evaluations by applying the LDC method to three groups (t, t', f) of the datasets | 87 |
| 4.4 | CAT feature selection method | 89 |

# TABLE OF CONTENTS (Continued)

**Page**

4.4.1 Implemented results of features of the cat method 89
4.4.2 Performances evaluations on the whole data by applying the cat method 91
4.4.3 Performances evaluations by dividing the data into three groups (t, t', f), picking up 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50% of features, and evaluation the performances without cat 100
4.5 Comparative analysis of the LDC method and the cat method 104
4.5.1 Dimensionality reduction 104
4.5.2 Performance comparison on each activity with and without LDC & CAT 106

**5 CONCLUSIONS 110**

5.1 Summary 110
5.2 Future work 113

**APPENDIX 114**

**REFERENCES 133**

**VITAE 145**

# LIST OF TABLES


| Table | | Page |
|---|---|---|
| 2.1 | A brief description of some integrated mobile phone sensors | 10 |
| 2.2 | A brief review of HAD system using built-in sensors of mobile phone | 13 |
| 2.3 | Overview of some data retrieving techniques for HAD system | 15 |
| 2.4 | Overview of Some Step Detection using Mobile's Sensors | 17 |
| 2.5 | A brief overview on some feature selection methods | 21 |
| 2.6 | General details of three datasets used in this dissertation | 25 |
| 2.7 | The percentage of each activity per dataset | 26 |
| 2.8 | Description of variable types of features | 27 |
| 2.9 | Description of the types of raw signal in the time domain | 28 |
| 2.10 | Description of the types of raw signal in the frequency domain | 29 |
| 2.11 | Feature categorization and number of features from the specified datasets | 30 |
| 2.12 | Row vector and column vector | 31 |
| 2.13 | Description of R2 vector space and Rn vector space | 33 |
| 2.14 | Example of the total number of elements of cyclic subgroup G | 40 |
| 3.1 | Descriptions of Activities | 42 |
| 3.2 | Detail descriptions of TED data collection | 43 |
| 3.3 | A comparison of UCI-HAR dataset and TAD dataset | 44 |
| 3.4 | Three types of walking during step counting | 46 |
| 3.5 | Four usage modes for mobile location | 46 |
| 3.6 | General Information of three datasets | 50 |
| 3.7 | Activities of each dataset | 51 |
| 3.8 | Four datasets to use in system | 52 |
| 3.9 | Total numbers of features for each subgroup of data in each dataset | 54 |
| 3.10 | Grouping feature set and instance set of each dataset | 54 |
| 3.11 | Accuracy of MLP models with 1 hidden layer and 2 hidden layers used with 25%, 50%, and proposed numbers of neurons | 66 |
| 3.12 | Specific requirements to build an MLP model | 66 |
| 3.13 | Confusion matrix | 67 |

# LIST OF TABLES (Continued)


| Table | | Page |
|---|---|---|
| 3.14 | Activity classes of UCI-HAR, DATASET -UCI, and HAPT_1 | 68 |
| 3.15 | Activity Classes for HAPT_2 | 69 |
| 4.1 | Comparison results between algorithm counting and actual walking steps in four mobile locations | 72 |
| 4.2 | Accuracy rates and running times of six classification methods | 74 |
| 4.3 | Accuracy results of MLP model with different number of neurons | 75 |
| 4.4 | Accuracy results of MLP with different numbers of neurons | 76 |
| 4.5 | Constant values related with features of LDC | 77 |
| 4.6 | Selected and removed features from each dataset by the LDC | 78 |
| 4.7 | Dimensionality reductions on each dataset by the LDC | 78 |
| 4.8 | Confusion matrix of MLP with and without LDC to UCI-HAR dataset | 79 |
| 4.9 | Confusion matrix of MLP with and without LDC to the DATASET-UCI | 81 |
| 4.10 | Confusion matrix of MLP on the HAPT_1 dataset with and without LDC | 83 |
| 4.11 | Confusion matrix of MLP on HAPT_2 dataset with LDC | 85 |
| 4.12 | Confusion matrix of MLP on HAPT_2 dataset without LDC | 86 |
| 4.13 | Evaluation comparisons for UCI-HAR dataset: Performance results by individual features selected with LDC to form three data groups (t, t', f) and performance results based on proposed LDC method | 88 |
| 4.14 | Evaluation comparison for DATASET-UCI dataset: Performance results by individual features selected with LDC to form three data groups (t, t', f) and performance result based on proposed LDC method | 88 |
| 4.15 | Evaluation comparisons for HAPT-1 dataset: Performance results by individual features selected with LDC to form three data groups (t, t', f) and performance results based on proposed LDC method. | 89 |
| 4.16 | Selected new features group and the total numbers of features | 90 |
| 4.17 | New feature groups, total numbers of features, and CAT features | 90 |
| 4.18 | Dimensionality reductions on four datasets by CAT | 91 |

# LIST OF TABLES (Continued)


| Table | | Page |
|---|---|---|
| 4.19 | Confusion matrix of MLP to UCI-HAR dataset with and without CAT | 92 |
| 4.20 | Confusion matrix of MLP to DATASET-UCI with and without CAT | 94 |
| 4.21 | Classification results to HAPT_1 with and without CAT | 96 |
| 4.22 | Classification results of MLP to HAPT_2 with CAT features selection | 98 |
| 4.23 | Classification results of MLP to HAPT_2 without CAT feature selection | 99 |
| 4.24 | Performance results of data divided into three groups (t, t', f) with picked-up data of 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50% without feature selection by the CAT method and performance results based on feature selection by CAT method for the UCI-HAR dataset | 101 |
| 4.25 | Performance results of data divided into three groups (t, t', f) with picked-up data of 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50% without feature selection by the CAT method and performance results based on feature selection by CAT method for the DATASET-UCI dataset | 102 |
| 4.26 | Performance results of data divided into three groups (t, t', f) with picked-up data of 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50% without feature selection by the CAT method and performance results based on feature selection by CAT method for the HAPT_1 dataset | 103 |
| 4.27 | Comparison of original features, selected features using LDC and CAT | 104 |
| 4.28 | Dimensionality data sizes of original, the CAT and LDC data | 106 |
| 5.1 | Performance of MLP on three original datasets before and after feature selection methods (LDC vs. CAT) | 111 |

## LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 2.1 | Four main processes of HAD system | 9 |
| 2.2 | Some examples of physical activities of humans | 12 |
| 2.3 | A single artificial neuron model | 22 |
| 2.4 | Multi-layer perceptron model with two hidden layers | 23 |
| 3.1 | Twelve categorizations of walking activity for each user | 47 |
| 3.2 | Step detection algorithm | 48 |
| 3.3 | System design of HAD based on LDC and CAT feature selections | 49 |
| 3.4 | Dividing data into three feature subgroups | 53 |
| 3.5 | Algorithm design for LDC feature selection method | 59 |
| 3.6 | Algorithm design for CAT feature selection method | 61 |
| 3.7 | Comparison of learning rates and MSE on UCI-HAR dataset | 65 |
| 3.8 | A new MLP model design | 67 |
| 4.1 | Comparison of step counting results for 3 walking styles in each mode | 71 |
| 4.2 | Overall results of step counting in four different modes | 73 |
| 4.3 | Performances of 6 classifiers to UCI-HAR dataset after the LDC | 80 |
| 4.4 | Performance results of six classifiers on DATASET-UCI dataset | 82 |
| 4.5 | Performances of 6 classifiers to HAPT_1 dataset with the LDC | 84 |
| 4.6 | Performances of 6 classifiers to HAPT_2 dataset with the LDC | 87 |
| 4.7 | Performance values of six classifiers on UCI-HAR data with CAT | 93 |
| 4.8 | Performances of six classifiers on DATASET-UCI with the CAT | 95 |
| 4.9 | Performances of six classifiers on HAPT_1 with the CAT (78 features) | 97 |
| 4.10 | Comparison of performance values with the MLP model and other classifiers using HAPT_2 dataset with CAT | 100 |
| 4.11 | Comparison of data percentage size of original, LDC and CAT features | 105 |
| 4.12 | Performance comparison of each activity to UCI-HAR dataset with original data, LDC data, and CAT data | 107 |
| 4.13 | Performance comparison of each activity to DATASET-UCI dataset with original data, LDC data, and CAT data | 108 |
| 4.14 | Performances of HAPT_1, LDC data, CAT data in each activity | 109 |

## LIST OF ACRONYMS

| Notation | Description |
|---|---|
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| BAG | Bagging |
| BAYES | Bayasian |
| CART | Classification and Regression Tree |
| CAT | Cyclic Attribution Technique |
| CFS | Correlation-based Feature Selection |
| DATASET-UCI | DATASET for Machine Learning Repository |
| DBN | Deep Belief Network |
| DDR | Data Dimensionality Reduction |
| DSF | Disease-Specific Feature |
| DT | Decision Tree |
| DTW | Dynamic Time Warping |
| EAC | Euler Angle Conversion |
| EM | Expectation Maximization |
| ET | Extra Tree Classifier |
| FFT | Fast Fourier Transform |
| FN | False Negative |
| FP | False Positive |

**LIST OF ACRONYMS (Continued)**

| Notation | Description |
|---|---|
| GB | Gradient Boosting |
| GDA | Generalized Discriminant Analysis |
| HAD | Human Activity Determination |
| HAPT | Human Activities and Postural Transitions |
| HAR | Human Activity Recognition |
| HMM | Hidden Markov Model |
| HPF | High Pass Filter |
| IFS-CoCo | Co-operative Co-evolutionary algorithm for Instance and Feature Selection |
| J48 | C.45 Decision Tree Algorithm |
| KNN | K Nearest Neighbour |
| KPCA | Kernel Principal Component Analysis |
| LASSO | Absolute Shrinkage and Selection Operator |
| LDA | Linear Discriminant Analysis |
| LDC | Linear Dependent Concept |
| LPF | Low Pass Filter |
| ML | Machine Learning |
| MLP | Multi-layer Perceptron |
| MSE | Means square error |
| MC-SVM | multi-class SVM |

**LIST OF ACRONYMS (Continued)**

| Notation | Description |
|---|---|
| NN | Neural Networks |
| PCA | Principal Component Analysis |
| PLS | Partial Least Square |
| RF | Random Forest |
| RS | R-Square |
| RVM | Relevance Vector Machines |
| SM | Sammon Mapping |
| SVD | Singular Value Decomposition |
| SVM | Support Vector Machine |
| TAD | Testing Activity Data |
| TMDP | Tensor Manifold Discriminant Projections |
| TN | True Negative |
| TP | True Positive |
| UCI | University of California, Irvine |
| UCI-HAR | University of California, Irvine, Human Activity Recognition |
| UCI-HAPT | UCI Machine Learning Repository-Human Activities and Postural Transitions |
| UCI-ML | UCI Machine Learning Repository |
| USC-HAD | University of Southern California-Human Activity Dataset |

## LIST OF NOMENCLATURES

| Notation | Description |
|---|---|
| AC | Accelerometer |
| GY | Gyroscope |
| MG | Magnetometer |
| LN | Linear |
| R | Actual walking steps |
| P | Predicted walking steps |
| N-w | Normal walking |
| S-w | Slowing walking |
| F-w | Fast walking |
| WK | Walking |
| WKU | Walking-Upstairs |
| WKD | Walking-Downstairs |
| SIT | Sitting |
| STD | Standing |
| LY | Laying |
| STD-SIT | Stand-to-Sit |
| SIT-STD | Sit-to-Stand |
| SIT-LY | Sit-to-Lie |
| LY-SIT | Lie-to-Sit |
| STD-LY | Stand-to-Lie |
| LY-STD | Lie-to-Stand |

## GREEK SYMBOLS

| Notation | Description |
|---|---|
| $\varphi$ | Euler Phi function |

## SUBSCRIPTS

| Notation | Description |
|---|---|
| $a_1, a_2$ | Input of single ANN model |
| $w_1, w_2$ | Weights for the layers of single ANN model |
| $y$ | Output of single ANN model |
| $a$ | Magnitude of acceleration value |
| $m$ | Total number of timestamps |
| $w$ | A peak point |
| $B$ | Step occurrences |
| $A$ | $(k \times n)$ matrix |
| w | Weights for activation function |
| b | Bias for activation function |
| $\boldsymbol{x}$ | A vector |

# CHAPTER 1

# INTRODUCTION

Human Activity Determination (HAD) or Human Activity Recognition (HAR) is a process that uses Machine Learning (ML) to monitor the behaviour and characteristics of daily human activity [1]. It is becoming a critical and challenging research area due to its strong contributions and demand from areas such as healthcare, eldercare, security, entertainment, and fitness centres [2], [3]. It is a vital component of healthcare that enhances maintenance of human health by providing information about movement and behaviour [1], [4]–[6]. The detection of normal and abnormal behavior and monitoring the rehabilitation of patients are good examples of how HAD-supports healthcare systems.

HAD is a sensor-based method of determining human physical activity using ML [7]. By using the sensor technology and associated information in mobile phones, HAD can infer a subject's daily activity [8]. Due to the powerful built-in sensors of smartphones, HAD is becoming a significant focus of research [9]–[11]. Present mobile phones come together with embedded sensors including accelerometers, gyroscopes, and magnetometers that implement users to implement HAD software [12]. These mobile phone sensors enable the recording of complicated activities performed by users such as lying, walking, ascending and descending stairs, waving, running, jogging, sitting, standing, and cycling [13].

A typical HAD task included four procedures: data retrieval, preprocessing, feature extraction/selection, and classification [14]. Data retrieval involves the acquisition of data derived from mobile phone or wearable sensors; preprocessing is the method in which raw data are cleaned and appropriate data transformed for further processing; feature extraction makes accurate information easier to obtain from the preprocessed data and feature selection is a way of selecting the most significant features for predictive model construction; finally, classification in HAD predicts the activities of the user and evaluates their performances using an algorithm.

In feature extraction, three types of features are recognized: relevant features that have an influence on the output, irrelevant features that have no influence on the output, and redundant features that occur whenever one feature can take the role of another. Feature selection can delete redundant or unnecessary features and a well-selected feature set not only improved recognition efficiency but also reduced computational complexity [15] and prevented the output of classifiers with no useful information [16]. Feature selection is a key component of many pattern recognition applications in ML and good feature selection improves the performance of classification. Useful features were dependent features that supported critical information and improved the capability of the performance model [16]. A model with irrelevant features generated negative predictions but good feature selection with a well-defined classification theory improved performance [17]. Classifiers are utilized for recognition tasks ranging from simple categorization to complicated applications in various areas of Artificial Intelligence (AI). It is essential to select the correct classification algorithm to determine human activities. There are many classifiers used in ML, but a prediction tool conventionally used in HAD is Artificial Neural Network (ANN).

ANN is a computational paradigm based on mathematical models with a mammalian brain framework. It automatically makes choices for multiple pattern recognition issues. ANN learns from the perceived data, justifies with undefined function, and transmits the results of the predicted models. The strategy for learning of this classifier is based on algorithms of supervised learning. To transmit the outcomes needed, simple models are constructed in which a vector may compose a pattern consisting of interrelationships attributed to meaningful categories.

### 1.1 Research Problems

Although many researchers have improved the general efficiency, stability and precision of HAD, there is still room for further development. Several researchers have discussed and analyzed the determination of human activities by investigating preprocessing techniques, feature selection methods, and classification models. However, the evaluation of HAD has not yet been sufficiently developed. The

following are the most important study issues that come within the scope of this dissertation.

### 1.1.1 High Dimensionality Dataset

One of the major issues is that HAD produces high-dimensional datasets. Human behavior was not always consistent because various tasks were performed simultaneously during the cycle of human activity [4], [18], [19]. Moreover, numerous kinds of physical activities and specific characteristics may be included in the developer's inquiry and the changing position of the sensor during movement makes the information provided by the sensor very complicated. Therefore, the nature of sensor data and the user's motion create a dataset of high dimensionality when sensor data is constantly retrieved.

### 1.1.2 Redundant Features

Extracting features that infer precise information from the context of user activities is a necessary approach to a high dimensional dataset with redundant or irrelevant features. Since datasets in HAR have high dimensionality, feature selection is an essential component [20]. In samples for training and testing, the fact that a dataset contains many unrelated or unneeded features complicates the achievement of robust performance [21]. More redundant features require more training and the testing process makes classification more complex. The complicated data makes the model overfit and impacts generalization of the model. Also, more time is needed for storage and processing.

### 1.1.3 Sensitivity of Classifiers

To significantly improve performance evaluation, HAD requires proper classifier design. A HAD system mainly depends on a classifier receiving features input as attributed values to be output as classified activities. Usually, a classifier needs to be trained on labeled data where the activity is known. The classifier can then be used to predict unknown data. In ML, different classifier models give different performances. The sensitivity of the classification performance of different classification models depends on various parameters. In a standard strategy, better decision making has a

significant impact on performance precision. This implies that a particular classifier needs an appropriate design to correctly determine human activities.

### 1.2 Understanding HAD

The ML method of determining human activity operates on three levels. At the lowest level, sensor data is generated for retrieval using mathematical ideas and language. At the intermediate level, hypotheses are tested and estimates derived by preprocessing, feature extraction, or feature selection. At the highest level, classifiers determine the overall goal for activity sequences.

A preliminary study is necessary to understand the nature and type of sensor information to be processed later on. This study aims to establish the capabilities of the mobile phone sensors and to retrieve the data generated by complex user-performed activities. Smartphones were previously equipped with a single sensor (an accelerometer) but most Android devices nowadays come with a range of built-in multi-sensors [22]. These sensors can retrieve the context of substantial parameters such as movement, positioning, and different environmental conditions. They can be adapted to provide the user valuable information about numerous day-to-day physical activities.

There are three sensor categories: sensors for motion, sensors for the environment, and sensors for location. Sensors of motion such as accelerometers, gyroscopes, gravity sensors, and sensors of rotational vectors measure the rotations and acceleration along three axes. Environmental sensors such as barometers, photometers, thermometers, Wi-Fi and Bluetooth sensors are used to calculate various environmental parameters. Location sensors such as orientation sensors and magnetometers measure the physical location of a device. Among built-in mobile phone sensors, the accelerometer, gyroscope, and magnetometer are the most powerful at determining human activity [23].

Two preliminary tasks were carried out to establish the capabilities of the HAD system: sensor data retrieval and step counting; and there had to be a trade-off between processing power and the demands of the HAD system itself. Sensor data retrieval involves recognizing physical activities with the sensors in mobile phones.

The Samsung Galaxy Note 4 using the Android 6.0.1 operating system was selected for this task. Thirty volunteers performed eight activities (sitting, standing, walking on the spot, jogging on the spot, walking normally, walking-upstairs, walking-downstairs, and lying-down). Every 100 milliseconds, the phone's accelerometer, gyroscope, and magnetometer detected every motion from the activities of the 30 users. After retrieving the data, the nature of the sensor can be understood and the sensor data can be managed to take into account the required processing power.

The second task in the evaluation of HAD involves counting the walking steps of the user to get sufficient processing authority into the sensor data. Step counting is the automatic detection of paces walked. The use of smartphones to count steps has increased significantly and benefits society because its applications are numerous. It is used in HAD for better detection of walking. Consequently, a reliable step counting technique was introduced in this stage of the work. The accelerometer in the Samsung Galaxy J7 was used and nine volunteers performed three types of walking (normal, slow, and fast) in four phone usage modes (swinging, texting, in a waist-bag, and in a hand-bag). The results of step counting showed the reliability of the proposed algorithm.

### 1.3 Research Contributions and Objectives

Due to the importance of accuracy in ML problems, feature selection is a key component of HAD that reduces the size of the high-dimensional dataset. Selection of relevant features can help accomplish the two most challenging research problems: the high-dimensionality of data and redundancy of features. Despite the fact that there are many feature selection methods, the most appropriate methods to use with predictive problems have not yet been identified.

Identifying the best feature selection method makes possible the main goal of this dissertation, which is to evaluate a better performance with respect to specific classifiers not only for the development of HAD but also the improvement of ML. To this end, the feature selection method needs to produce a well-defined determination with multi-way dependencies and correlations, which implies that simply selecting the best independent features may not create the best feature set.

In this dissertation, a novel feature selection method is first proposed using a linearly dependent concept on a homogeneous linear combination system, known as Linearly Dependent Concept (LDC). LDC is intended to define useful features determining human activity. It can appropriately reduce the number of irrelevant features and also offers advantages for the classification process. Even though this method can solve some challenges in feature redundancy problems, it still has some weak points. Due to the transformation requirement for a homogeneous linear combination system, it cannot identify more useful features. To compensate for this, a new technique of feature selection is tried, based on the Cyclic Attribute Technique (CAT), using group theory and the fundamental properties of the cyclic group.

The quality of the selected features enables the classifier to correctly identify informatics features. ANN is a more robust classifier that reduced the number of false positives in HAD [24]. Therefore, by creating a formula for discovering the number of hidden neuron nodes, this research intends to evaluate the performance of selected features using a new MLP model with hidden layers and learning rates.

To assess the effectiveness of the HAD system, this dissertation investigates the performance of two feature selection methods with a specific classifier. Both feature selection methods process three different datasets: i) a UCI-HAR dataset [25], ii) a UCI-HAPT dataset [26], iii) a DATASET-UCI dataset [27]. These are the most well-known public datasets of the UCI repository (available from http://archive.ics.uci.edu/ml) for HAR. Then, the ANN classifier model determines the level of accuracy that can be achieved using the two feature selection methods. Performances of classification tasks are analysed and compared using Support Vector Machine (SVM), Bagging (BAG), K-Nearest Neighbors (KNN), the Classification and Regression Tree (CART), and Bayesian (BAYES) analysis.

Finally, this research targets the following five objectives:

i) Reducing the high dimensionality data size of human activity determination

ii) Determining the best-suited features in human activity determination

iii) Creating new feature selection methods, LDC and CAT, for human activity determination

iv) Expanding a new precise ANN classifier model for human activity determination

v) Developing a more accurate human activity determination system

## 1.4 Outline

The dissertation is organized into the following five chapters.

Chapter 1 presents an overview which introduces study issues, general knowledge of HAD, research contributions, and objectives for the development of the HAD system.

Chapter 2 discusses certain techniques and motivations related to this dissertation that appeared in previous work on HAR. It also supports the background understanding of theories and techniques to be applied in this dissertation.

Chapter 3 explains the detailed methodologies and designs of two different feature selection methods (LDC and CAT) and a novel classifier model (ANN) for the HAD system.

Chapter 4 describes the outcomes of the experiments and analyzes the acquired results in order to assess performances after implementing the two feature selection methods on 4 datasets.

Chapter 5 concludes the report of the dissertation by examining the findings of the outcomes acquired and presents future work.

# CHAPTER 2

# BACKGROUNDS AND REVIEWS

This chapter provides detailed literature on the current HAD. The motivation, methods, data collection, step counting, feature selection, and classifications of current HAD work are discussed here. It also supports background understanding of significant theories, definitions, techniques, principles, and characteristics in this dissertation. The intention of this chapter is analyzing the existing HAD system and highlighting the gap in the current work. The gap establishing in this chapter allows the choice for this dissertation to be created.

This chapter consists of four major sections. In the first stage, section 2.1 is for covering the basic requirement of the HAD process. It investigates the primary HAD process, the role of sensors in HAD, and some everyday activities of humans. The second part is section 2.2, which approaches some HAD-related real-time applications and provides some methods for collecting data and steps counting. Based on this, the dissertation performs data collection and step counting to understand more in HAD.

The third part is section 2.3, which focuses on knowledge and methods for selecting features and classifying them. This section is the central part of this dissertation. Based on this, this dissertation proposes two new techniques of feature selection and a new technique of classification. The last part is section 2.4, which discusses some popular datasets from the UCI-HAR dataset repository lab for HAD, and introduces background knowledge of linearly dependent concepts, group theory and fundamental properties of the cyclic group that are intended to be used in this research.

## 2.1 Review on Human Activity Determination (HAD)

### 2.1.1 Main HAD Process

HAD refers to the automatic recognition of human physical activities using wearable sensors with the ML technique. It determines whether a single user or

multiple users performs physical activities. HAD is an essential research area in AI, due to the most vital applications in many fields, especially the healthcare system, security system, and also military [28]. Many researchers have developed in HAD with several techniques, methods, principals, characteristics, and methodologies to improve the HAD system. Although many researchers have developed many types of HAD system, there are four primary procedures with three levels in the ML techniques to determine personal human behaviors as shown in Figure 2.1 [14].

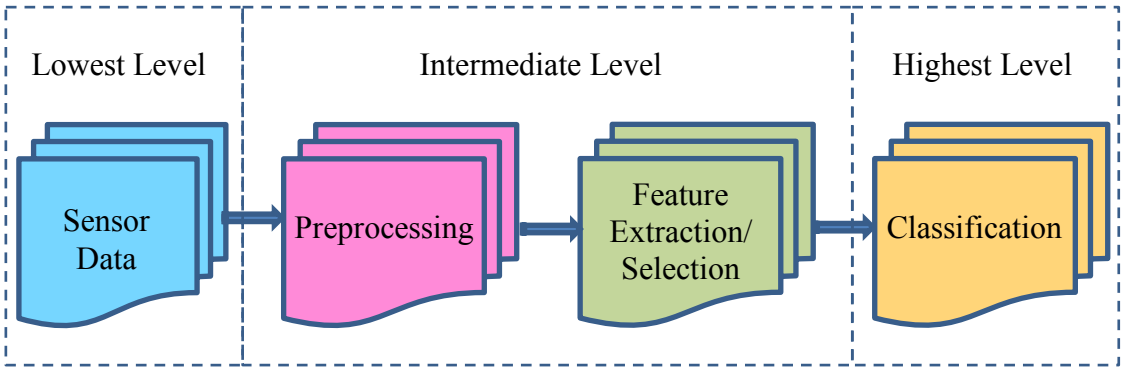



**Figure 2.1** Four main processes of HAD system

In Figure 2.1, the HAD system consists of three levels to determine human activity. The lowest level includes retrieving the sensor data are generated with the mathematical ideas and language. The intermediate level involves testing hypotheses and deriving estimates by preprocessing, feature extraction, and feature selection with assuming properties. The highest level concerns with determining the overall goal of HAD by using classifiers for the activity sequences [29].

Basically, the main target of the HAD is to identify the activities of the user with the help of computing techniques [30]. Although HAD is a wider environment in AI, the primary work is recognizing the human physical activity [31]. HAD has great impacts for not only the healthcare system but also other areas such as entertainment, industry, operational areas and sport [32]–[34].

### 2.1.2 Sensors for HAD

Mobile phone technologies are increasing quickly year after year and offering many features. In addition, through networks such as Wi-Fi, Bluetooth, Skype, Facebook, Gmail, Viber, these mobile phones can communicate all over the world.

Being smarter day by day with multi-tasking and high precision, these mobile phones are known as smartphones. As today's smartphones offer a variety of built-in sensors and allow users to operate the software, HAD using built-in sensors on mobile phones has gradually become more popular and significant.

Although the previous smartphones had a single sensor (accelerometer) [22], the present mobile phones came with many embedded sensors such as light sensors, camera sensors, accelerometer, gyroscope, magnetometer, etc. Most sensors are small, compact and portable for retrieving and responding to input information from the physical environment such as light, sound, heat, moisture, motion, pressure, human activity, etc. In general, the sensor output is one that can convert for reading or further processing to human-readable information. Among many built-in sensors, accelerometer, gyroscope, and magnetometer are very common in the study fields of HAD due to the efficient tools for data retrieval and user activity tracking. Among them, the accelerometer is one of the most popular sensors due to an electromechanical device to measure both static and dynamic acceleration. Table 2.1 shows a brief overview of some built-in mobile phone sensors.

**Table 2.1** A brief description of some integrated mobile phone sensors

| **Built-in Sensors** | **Uses** |
|---|---|
| Light Sensor | Screen dimming |
| Camera Sensor | Image/ Video capture |
| Microphone Sensor | Audio capture |
| GPS Sensor | Global location |
| Accelerometer Sensor | Acceleration/ Position |
| Gyroscope Sensor | Local Orientation |
| Magnetometer Sensor | Motion |
| Linear Sensor | Displacement |

HAD used built-in sensors of mobile phone is a popular method among many researchers. Although they have considered HAD with different methods of data collection, various analyzing procedures, many feature selection methods, and different classification techniques, there is still a lack of better results in HAD [35]. As shown in Table 2.1, the accelerometer sensor is a motion sensor capable of measuring the acceleration and position along three axes. The gyroscope sensor can also evaluate and place the rotational forces along three axes. The magnetometer is a sensor of movement that can assess the parameters of physical motion. Moreover, the use of other sensors is also helpful in diffident ways for HAD.

According to the development of sensor technology, several researchers are focusing on creating HAD systems using the mobile phone's built-in sensors. The authors used the accelerometer as a motion sensor and they produced excellent outcomes for their studies [36], [37]. The researchers used a built-in accelerometer of a cell phone to infer physical activity for a fundamental problem of place [38]. The study also expended a tri-axial accelerometer of a mobile phone on a robust HAD classification model [39]. In this study, three people wore the device in front of two trouser pockets and performed the five daily activities (static, walking, running, upstairs walking, and downstairs walking). In [40], the authors used the Android accelerometer sensor for the device orientation. They transformed accelerometer data using Euler Angle Conversion (EAC) and showed an improvement in precision for five activities (standing, walking, running, ascending-stairs, and descending-stairs).

The authors showed that the intended sensors for extended recording and monitoring in computing are flexible to use in HAD [8]. Some studies concentrated on the sensor types that used to determine human activity. A recent review on this topic found that the performance of HAD mainly depends on choosing the sensors type to correctly retrieve the sensor data. An increasing number of studies have discovered HAD with a combination of sensors to be better than signal sensor detection. Due to the effectiveness of accelerometers and gyroscopes in mobile phones, they are widely utilized as the combination sensors to fulfill the task of recognizing human activity [41]. The authors used the built-in accelerometer and gyroscope sensors to detect user motions [12], [42]. Using 3 sensors (accelerometer, gyroscope, and magnetometer), an

effective classifier model was established for better precision [43]. In HAD, Dernbach, et al. [44] observed in HAD that it is better to use more than one sensor than to use a single sensor. They demonstrated the reliability of performance using combined sensors (accelerometer + gyroscope + magnetometer) to determine six activities (walking, running, sitting, walking-upstairs, walking-downstairs, and standing).

### 2.1.3 Human Activity

Human activity is a multi-dimensional behaviour of a human's everyday routine work. Regular physical activity is very essential for everyone because it can assist their ability to make as an enjoyable daily life and can provide long-term health advantages [45]. In addition, regular exercise and physical activity can reduce the risks of some diseases such as heart disease, diabetes, etc. [46]. Every person should be active in daily activities to maintain their health. There are many physical activities such as sitting, standing, laying, walking, walking up-stairs, walking down-stairs, waving, running, jogging, cycling, etc. As shown in Figure 2.2, human physical activities in daily life can be split into three types: static, transitions, and dynamic [47]-[48].

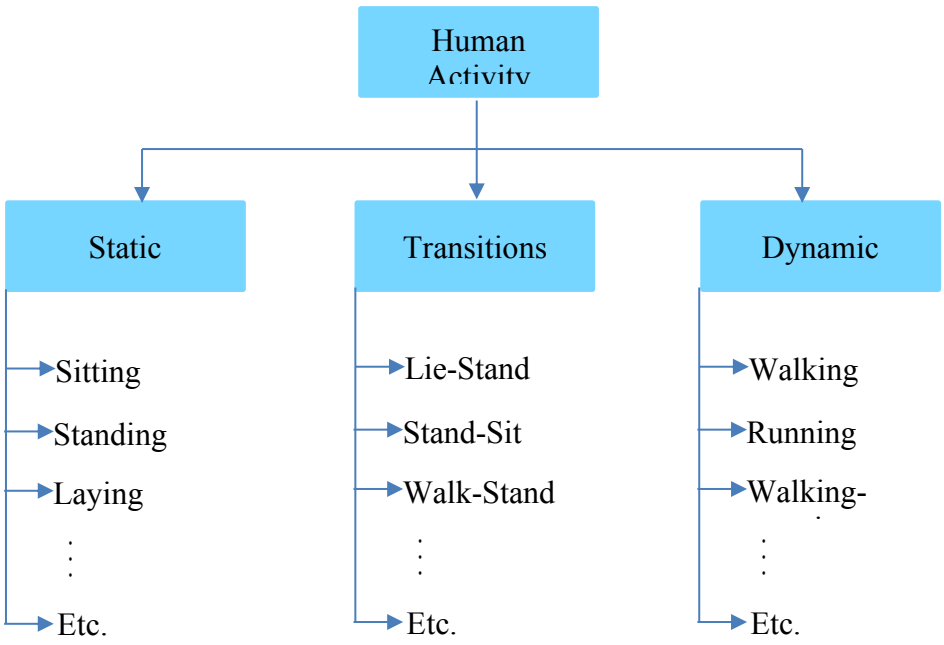



**Figure 2.2** Some examples of physical activities of humans

In real life, human activities are very complicated and miscellaneous. In some cases, it could perform simultaneously. For example, having dinner can be viewed as sitting down together. Monitoring the activities of patients improve the treatment of medical diagnosis and rehabilitation [8]. Although many researchers attempted to control the physical activities in interference studies, the early results generated unreliable outcomes. The authors identified 3 walking styles (slow, normal, and fast) of 21 users based on user movements. Android accelerometer was utilized and placed into the user' pockets of a trouser [49]. They investigated the performance between them and gave a high performance in the fast-walking activity. The preliminary work in this field showed that the location of the sensor and its orientation carried out the complexity of the dataset. In [50], the authors examined a phone's location problem by carrying a mobile phone in different ways during human activities. An overview of some HAD using built-in sensors of a mobile phone is provided in Table 2.2.

**Table 2.2** A brief review of HAD system using built-in sensors of mobile phone

| No. | Cite No. | Year | No. users | No. Activities | Activities | | | | | | | | Sensors | | | Accuracy% |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Sitting | Standing | Lying | Walking | Walking-upstairs | Walking-downstairs | Jogging | Running | Accelerometer | Gyroscope | Magnetometer | |
| 1 | [38] | 2011 | 29 | 6 | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | 91.70 |
| 2 | [42] | 2013 | 25 | 1 | | | | ✓ | | | | | ✓ | | | 89.30 |
| 3 | [32] | 2014 | 6 | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | | | 98.00 |
| 4 | [39] | 2015 | 12 | 1 | | | | ✓ | | | | | ✓ | | | 90.80 |
| 5 | [41] | 2015 | 3 | 5 | ✓ | | | ✓ | ✓ | ✓ | | ✓ | ✓ | | | 93.40 |
| 6 | [6] | 2016 | 5 | 5 | | ✓ | | ✓ | ✓ | ✓ | | ✓ | ✓ | | | 84.77 |
| 7 | [26] | 2016 | 13 | 4 | ✓ | ✓ | | ✓ | | | ✓ | | ✓ | | | 92.32 |
| 8 | [35] | 2016 | 44 | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | 94.00 |
| 9 | [40] | 2016 | 29 | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | 94.00 |
| 10 | [53] | 2016 | 5 | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | 96.00 |
| 11 | [27] | 2018 | 30 | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | 98.80 |
| 12 | [29] | 2015 | 44 | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | 90.00 |
| 13 | [49] | 2013 | 30 | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | 96.00 |
| 14 | [44] | 2012 | 1 | 6 | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | 97.00 |

Although many researchers have developed several methodologies for HAD, there is still considerable with some drawbacks of failing to recognize every activity. The authors attempted to recognize the physical activities for elderly people (age≥70) by using the accelerometer sensor in the mobile phone [51]. By maintaining daily records of food intake and physical activity, the total energy expenditure and resting metabolic rate was tried to report [52]. However, their results failed to measure an accurate amount of physical activity due to the overestimation of physical activity. Among many human physical activities, walking is a common activity. Moreover, it directly associates with reducing the risk of some diseases. A new method for evaluating transportation-related physical activity was introduced and demonstrated the meaningfulness of daily levels of physical activity [53].

## 2.2 A Review on Real-Time Recognition of Human Activity

There are two categories available for the HAD system: 1) online recognition system and 2) offline recognition system. In which, data collection and step counting are online experiments with real-time response. The processing of these experiments provides the possibility of understanding the capabilities of the sensors to build the HAD system. This section overviews the previous data collection and step counting in order to improve the HAD system.

### 2.2.1 Data Collection

Data retrieving is a preliminary work of the HAD system. Due to the rich sources sensing, accelerometer and gyroscope are widely used to determine human physical activities. The data were collected from the tri-axial accelerometer on human physical activity and used a quasi-periodic time series to verify the better performance of the activity [54]. To achieve robust determination in human activity, a HAR dataset was created using accelerometer and gyroscope sensors of a Samsung Galaxy S2 to recognize six activities of 30 users [25]. The raw data was sampled with a constant rate of 50Hz, further analyzed and extracted features. Then they released this dataset to UCI Machine Learning Repository, namely the ***UCI-HAR dataset***. In [39], the authors used Samsung Galaxy Ace GT-S5830M to collect data using of accelerometer sensor. In their experiment, three users performed five activities (sitting, walking, running, up-

stairs-walking, and down-stairs-walking). For data retrieving, they used the sampling rate around 40Hz. They considered a recognition system of a position-independent activity [55].

In [56], the authors collected data from 10 volunteers performing six different activities (sitting, standing, walking, upstairs, downstairs, and lying) on three sensors (accelerometer, gyroscope, linear). The data were recorded at a frequency rate of 50 Hz and 10,939 samples were collected. Their technique was focused on the Support Vector Machine (SVM) classifier for system development and 89.59% of accuracy has been accomplished. The study compared the device recognition and offline recognition with seven volunteers performing five activities (walking, running, cycling, idling, and driving a car) [57]. They demonstrated that the accuracy of device recognition was better than the accuracy of offline recognition. In [58], the authors considered many locations for sensors and introduced a USC-HAD (University of Southern California-Human Activity Dataset) dataset. They described their dataset in detail and compared it to the other datasets and collected the data from human activities using an accelerometer sensor of a mobile phone [59-62]. A brief description of the data acquisition system with built-in sensors of mobile phones is shown in Table 2.3.

**Table 2.3** Overview of some data retrieving techniques for HAD system

| No. | Cite. | Year | No. subjects | No. activities | Sensor type | Android position | Sampling Rate |
|---|---|---|---|---|---|---|---|
| 1 | [50] | 2014 | 29 | 6 | AC | Trouser pocket | 20 |
| 2 | [57] | 2012 | 7 | 5 | AC | Trouser pocket | 40 |
| 3 | [58] | 2012 | 15 | 7 | AC | Chest | 52 |
| 4 | [54] | 2016 | 4 | 6 | AC | Hand pocket | 100 |
| 5 | [60] | 2015 | 9 | 6 | AC | Arm, Waist | 50-200 |
| 6 | [56] | 2017 | 30 | 9 | AC | Trouser pocket | 50-200 |

| No. | Cite. | Year | No. subjects | No. activities | Sensor type | Android position | Sampling Rate |
|---|---|---|---|---|---|---|---|
| 7 | [63] | 2013 | 30 | 6 | AC+ GY | Waist | 50 |
| 8 | [57] | 2012 | 14 | 12 | AC+ GY+ MG | Hip | 100 |
| 9 | [60] | 2015 | 3 | 5 | AC+ GY+ MG | Trouser pocket | 40 |
| 10 | [56] | 2017 | 10 | 6 | AC+ GY+ LN | Waist, Angle, Arm | 50 |

In Table 2.3, the word ***AC*** of the column *Sensor Type* means accelerometer, the word ***GY*** refers to the gyroscope, the word ***MG*** offers magnetometer, and the word ***LN*** gives the linear. In the next section, due to efficiency in computing environments, accelerometer-based step counting will be discussed.

### 2.2.2 Step Counting

Step counting is the automatic detection of walking steps taken by a user. Being a key point to understand user's motions, it is a critical part of HAD research [63]. It can provide many applications such as medical fields, fitness tracking centre [64], as well as step counting is essential to monitor in human daily activity. As the current smartphones with a variety of sensors and great processing skills, step counting using built-in sensors of a smartphone is increasingly becoming a vital factor among many researchers [65]. However, the step counting with sensors on a smartphone is still interesting research issues in dissimilar walking actions and mobile phone locations.

In the former, the pedometers are used for counting steps as well as many other applications. Because they are portable and special devices to attach on waist, foot, or arms for step counting. In the present, due to the effective sensing of mobile phones, accelerometer and gyroscope sensors of mobile phones become to track the walking steps as pedometers. To obtain precise measures, step counting with built-in sensors of mobile phones, especially accelerometer and gyroscope, has become an increasing research area [66]. Both sensors are suitable for analyzing movements because accelerometer can detect the rate of change in motion and gyroscope can measure the angular velocity. Indeed, sensor location and orientation are very

significant factors in the daily activities of humans to be more efficient step counting [67]. Based on threshold data and Fast Fourier Transform, a new step counting technique for the accelerometer in the android phone was introduced [68]. Six volunteers were carrying the smartphone in one of their front pants pockets and performing three activities (sitting, standing, and walking).

Step counting is essential for the rehabilitation improvement of a person. Many researchers used the accelerometer to study gait detection and walking step detection. The accelerometer was considered to count the daily walks and compared with the steps of a pedometer [69]. In order to solve the false walking, a new robust and precise step counting technique was suggested [70]. A novel pedometer by combining both accelerometer and gyroscope sensors for various walking conditions was designed [71]. To support the health of elderly patients, the authors investigated the accuracy of accelerometer type slow walking speeds [72]. A novel step counter using accelerometer and gyroscope was concentrated [73]. The authors attempted the accelerometer to monitor the natural walking for pulmonary patients with a step counter [74]. An overview of some studies on the methods of step counting using built-in sensors of mobile phones is expressed in Table 2.4.

**Table 2.4** Overview of Some Step Detection using Mobile's Sensors

| No. | Cite. | Year | Methods | Android sensors |
|---|---|---|---|---|
| 1 | [72] | 2006 | FFT algorithm | AC |
| 2 | [69] | 2010 | Convergent validity | AC |
| 3 | [66] | 2012 | Kalman filter method | AC |
| 4 | [68] | 2014 | FFT algorithm | AC |
| 5 | [74] | 2016 | A fixed stride length | AC |
| 6 | [70] | 2017 | Peak detection-based method | AC |
| 7 | [67] | 2018 | Internal and ecological validity | AC+ GY |
| 8 | [73] | 2015 | Test's inherent conditions | AC+ GY |
| 9 | [71] | 2017 | Random motion detection algorithm | AC+ GY |

## 2.3 A Review on Feature Selection and Classification

This section offers an overview of the related work to the current techniques and criteria for selecting features. This overview provides the research ideas and methodologies for studies to improve the HAD system. Finally, it makes this dissertation a challenge.

### 2.3.1 Feature Selection Techniques

Feature selection is a significant issue for improving the efficiency of human activity in determination. Because, HAD data is a huge amount of data containing relevant and irrelevant, or redundant features together [75]. These irrelevant or redundant features make the classification model complex. Those features make decreasing the performance of the model, but they make the evaluation time increase [76]. There are three general classes for feature selection algorithms: filter methods, wrapper methods, and embedded methods in ML [77]. In the filter feature selection method, a statistical measure is used to evaluate each feature with respect to learning methods [78]. The score of features from the dataset is ordered either by selecting or removing with statistical methods such as information gain and decision tree method [79]. The Chi-squared test, information gain, and correlation coefficient score are some examples of filter techniques.

The wrapper method is a search process as a best-first search algorithm that considers selecting a set of features based on different combinations of assigning a score to model precision. This strategy removes redundant or meaningless features and chooses helpful features to decrease the predictive model overfitting [77]. The stochastic search, heuristic method, and recursive feature elimination algorithms are examples of wrapper methods. The embedded method performs as the model creates in the training process [16]. This approach studies are offering the top features to provide the model's elevated precision. One of the most popular kinds of integrating feature selection is the regularization technique. It is also known as the technique of penalization, such as a regression algorithm, towards a predictive algorithm's low complexity. Examples of regularization algorithms are the Least Absolute Shrinkage and Selection Operator (LASSO), Elastic Net and Ridge Regression [77].

Many feature selection methods are available automatically, such as Decision Tree (DT), Gradient Boosting (GB), Random Forest (RF), R-Square (RS), Partial Least Square (PLS), etc. Selecting features in supervised learning increases the efficiency of classification [80]. DT is a popular classifier in ML and one of the common important supervised learning algorithms [81]. The DT splits the data according to a set of laws into sections. In which, the general segment technique is called a tree, a whole data comprising all sections is labeled as root, and the remaining sections are labelled as leaves.

GB is one of the best satisfying classifiers of ML. The model has the extra benefit of selecting the most significant features [82]. GB offers two approaches for assessing a feature's significance: 1) split-based approach and 2) sample-based approach. By dividing a node, the split-based technique decreases characteristics and sums over all nodes [83]. The sample-based rule improves a fit statistic from an uninformative variable according to the sample values. Based on the transformation method of feature selection, Principle Component Analysis (PCA) combines the features without reducing the cost and storage of the original feature set [84]. To fill in the lack of determination of physical activity, the authors removed the redundant features and reduced the time complexity using the built-in accelerometer of mobile phones [85]. The threshold-based condition box for minimizing feature vectors using accelerometer and gyroscope was designed [6].

Feature selection is an essential component of predictive modeling in classification. Although the researchers have discussed several methods for selecting features, there are two types of attributes of selections: feature selection method and dimension reduction method [86]. Though both techniques attempt to reduce the number of attributes in the dataset, these are the different purposes. Dimension reduction method by combinations of original attributes is creating new attributes such as PCA, Singular Value Decomposition (SVD), Sammon Mapping (SM), etc. The feature selection method is searching for useful attributes such as filter methods. The study introduced a similarity feature selection method for unsupervised learning [87]. In their work, only one feature was selected for each group according to similar features. They approved the performance result of the number of the selecting features group and

the similarity of features. Feature selection has become an important role in many ML problems. In order to develop the recognition of human activities with wearable sensors, the generalize discriminant analysis feature reduction method was studied [88].

One significant feature selection strategy is to reduce the vast amount of data on determining human activity. Based on Generalized Discriminant Analysis (GDA), the authors suggested a strong feature dimensional reduction technique and demonstrated that their technique significantly reduced the dimension of the feature space for large datasets [89]. Tensor Manifold Discriminant Projection (TMDP) was proposed as a new tensor-based feature selection method by demonstrating its effectiveness [90].

Due to an important technique of feature selection in the healthcare system, the Dynamic Time Warping (DTW) for automatic heartbeat classification was used to introduce a novel disease-specific feature selection method [91]. To perform feature and instance selection, an evolutionary model using a co-operative co-evolutionary algorithm for instance and feature selection (IFS-CoCo) was investigated and approved this method in many computational problems [92].

Based on re-sampling of raw data and principal component analysis (PCA), the data dimensionality reduction technique was presented to reduce incomplete data dimensionality in HAD [93]. A new feature selection algorithm was suggested for supervised classification issues through simultaneous feature selection and Gaussian mixture model estimation by [94]. By using Kernel Principal Component Analysis (KPCA) and Linear Discriminant Analysis (LDA), was extracted the effective features [95]. Deep Belief Network (DBN) was trained to evaluate the performance and compared with traditional methods for classifying ANN and SVM.

Feature selection can reduce not only original features but also the data dimensionality. Some feature selection techniques such as PCA, LDA can reduce the data dimensionality. However, those methods do not reduce the original feature sizes. In [9], the authors designed recognition of activity using the minimum redundancy maximum relevant feature selection method. Using correlation-based feature selection, fast correlation-based filter, and Relief-F algorithms, the authors discussed the importance of feature selection for elderly fall risk and faller classification [96]. In order

to reduce potentially large data, a feature selection method using an accelerometer and gyroscope was presented for elderly and stroke patients [97]. A brief description of some feature selection methods followed in this work is shown in Table 2.5.

**Table 2.5** A brief overview on some feature selection methods

| No. | Cite. | Year | Methods | Classifier | Accuracy% |
|---|---|---|---|---|---|
| 1 | [67] | 2010 | IFS-CoCo | Nearest Neighbor | 87.69 |
| 3 | [66] | 2014 | DSF | SVM | 86.66 |
| 4 | [69] | 2014 | EM | SVM | 88.5 |
| 2 | [64] | 2016 | GDA | RVM | 99.2 |
| 5 | [65] | 2016 | TMDP | KNN | 85.42 |
| 6 | [93] | 2017 | DDR | ANN, SVM | 82.00 |
| 7 | [ 21] | 2018 | LDC | ANN | 98.8 |

In the fourth column ***Methods*** of Table 2.5, the word ***IFS-CoCo*** means a Cooperative Co-evolutionary algorithm for Instance and Feature Selection method, the word ***DSF*** comes Disease-Specific Feature selection method, the word ***EM*** deals Expectation-Maximization feature selection model, the word ***GDA*** refers to the Generalized Discriminant Analysis, the word ***TMDP*** gives Tensor Manifold Discriminant Projection feature selection , the word ***DDR*** offers Data Dimensionality Reduction technique and the word ***LDC*** means Linearly Dependent Concept. In the fifth column ***Classifier*** of Table 2.5, the word ***RVM*** denotes Relevance Vector Machine classifier. The word ***SVM*** means Support Vector Machine and the word ***KNN*** gives K-Nearest Neighbour. Although many feature selections techniques have been developed in ML issues in recants years, feature selection techniques in HAD are still challenging to enhance.

### 2.3.2 Classification Techniques

The main goal of the classification stage is to accurately identify the class labels to solve the classification issues. The classification technique is building a model that maps input feature space to one or more class output space by a function [98]. There are two stages in classification: 1) the training stage in which each feature vector refers to a set of associated class labels and 2) the performance evaluation stage in which to decide the system output.

The Artificial Neural Network (ANN) model is built by training and testing data. Although ANN is an efficient tool in machine learning, it is a complicated network with many hidden layers of neurons. In ANN, neurons are arranged in various layers and interconnected. Each neuron is conducted by mathematical computations to receive processes and send information. The effectiveness of this network mainly depends on the training data and its model [86]. Many researchers have suggested improving the performance of neural networks classifiers. A backpropagation neural network is used to implement a pattern recognition system. The current approach to the recognition problem is based on a feed-forward neural network with backpropagation learning. The sample schematic representation of a single artificial neuron with two input neurons is described in Figure 2.3.

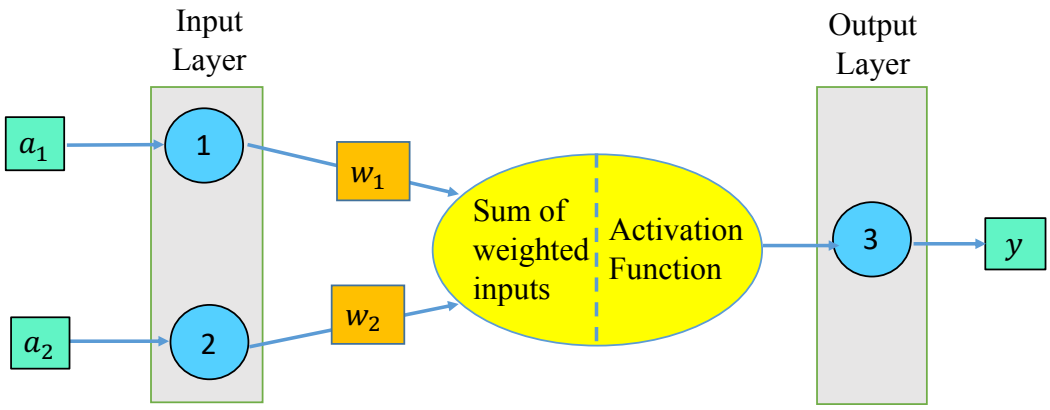



**Figure 2.3** A single artificial neuron model

In Figure 2.3, input components are $a_1$ and $a_2$ and $y$ is an output component. When the signal $a_1$ and $a_2$ apply to the artificial neuron network, the input $a_1$ goes to neuron 1 and $a_2$ goes to neuron 2 in the input layer. The activation function sends the signal to the neuron 3 in the output layer to generate the output $y$ by passing

the weight $w_1$ and $w_2$. Multiple efforts have been made to assess classifier efficiency using many supervised learning algorithms. Neural networks with one or more hidden layers are called multi-layer neural networks or MLP. Due to the robust performance of Multi-Layer Perceptron (MLP), it is widely applied to recognize human activity processes [99]. The MLP classifier can divide into three groups: 1) the input layer, 2) the hidden layer, 3) the output layer. A layer of input units is connected to a layer of hidden units and a layer of hidden units also connected to a layer of the output unit.

Due to a link between each of the neurons in any specified layer and each of the neurons in the next layer, ANN is known as a fully connected network. It is called a feed-forward-network when there is a link between neurons in a specified layer and there is none back to previous layers. This network is said to be a feed-forward-backward-network when the neurons are required to adjust the weight of the previous layer [100]. A multi-layer perceptron model with two hidden layers of supervised learning is shown in Figure 2.4.

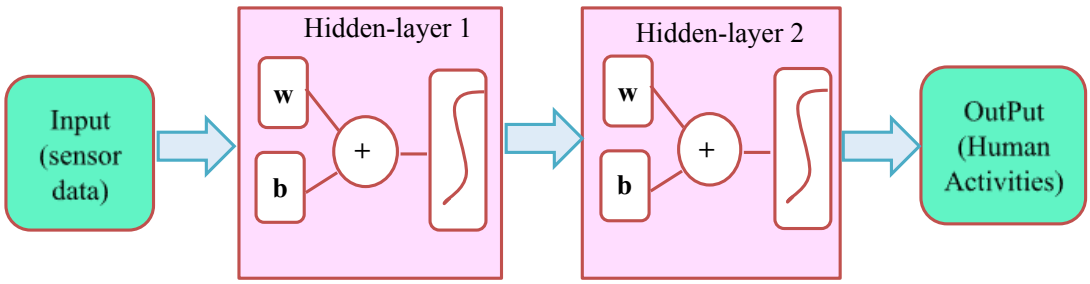



**Figure 2.4** Multi-layer perceptron model with two-hidden layers

In Figure 2.4, ***w*** and ***b*** are weight and bias for activation function in hidden layers. In ML, a classifier is an important role for learning and predicting of input data. Among many classifiers, ANN, DT, and KNN are well-known classifiers in HAD according to the literature reviews. Three classification algorithms (J48, Logistic Regression and Multilayer Perceptron) were proposed to fill the lack of classification by creating a model for recognizing six daily activities (walking, jogging, walking-upstairs, walking-downstairs, sitting and standing) [86]. Their results showed that MLP performed well. They talked about that it was simpler to define jogging activity than walking, and it was much harder to distinguish the two stairs climbing operations. There are many ML methods to carry out many kinds of studies on determining human

activity. In [101], the authors suggested an effective group-based context-aware classification technique for HAD. They compared to other famous classifiers and achieved the best accuracy. They trained a deep belief network and compared traditional techniques of classification of ANN and SVM. Their technique achieved better accuracy than other methods.

Several researchers used different classification models to examine HAD. In [102], the authors explored a hierarchical recognizer for classifying physical activity using artificial neural networks because of the significance of classification in HAR. Their technique gave an average precision of 97.9% in HAR. In [86], the authors developed a model to recognize six daily physical activities (walking, jogging, upstairs-walking, downstairs-walking, sitting and standing) using three classification algorithms (J48, Logistic Regression, and Multilayer Perceptron). They concentrated on simple time-domain features and reached classification precision of 98%. Their findings suggested the highest general precision conducted by the MLP. In [103], the authors concentrated on sensitivity, specificity, and F-scores in MLP classification to achieve more efficiency.

Many researchers have tried to achieve better efficiency in the determination of classification criteria in human activity. A quick and precise model of the Hidden Markov Model (HMM) classifier was proposed that used accelerometer and gyroscope in the mobile phone for various activities of a single user [104]. The threshold-based condition box was also designed to minimize feature vectors using the accelerometer and gyroscope [105].

A neural network classifier was used to discuss a robust HAD model [106]. They tried this model on time-domain features. They gathered information carrying the phone into the pant pockets of the front and back pant s. Five daily activities (static, walking, running, upstairs-walking, and downstairs-walking) were conducted by three volunteers. Their technique had accomplished a total precision of 92%.

An automatic function extractor and classifier were introduced by using built-in sensors on a mobile phone [107]. Using deep learning neural networks, they conducted classification of UCI-HAR dataset and analyzed classification using other

state-of-the-art techniques. They accomplished performance precision of 94.79% by considering a wider time span of temporal local score. A Multi Class HF-SVM (MC-HF SVM) model was introduced for monitoring disabled and elderly patients using portable accelerometer sensor and gyroscope sensor in mobile phone [108]. In [109], the authors mentioned HAD's requirements for ML approach criteria. Using Neural Networks (NN), Random Forests (RF), Nearest Centroid (NC), and multi-class SVM(MC-SVM), they demonstrated the significance of classification methods.

## 2.4 A Review on Datasets and Main Ideas on Feature Selection

### 2.4.1 Dataset

Despite the high efficiency of the HAD system among various applications, it consists of very complicated sensor information. The greater dimensional dataset makes the system process more complicated. Among many datasets to determine the most appropriate human activity classification, there are three distinct HAR datasets: i) UCI-HAR dataset [25], ii) UCI-HAPT dataset [26], iii) DATASET-UCI dataset [27], which are the most popular UCI repository datasets (available from http:/archive.ics.uci.edu/ml). Table 2.6 shows the overall description of the three distinct datasets.

**Table 2.6** General details of three datasets used in this dissertation

| Dataset | No of. Activities | No. of. Instances | No. of. Attributes | Released Date |
|---|---|---|---|---|
| **UCI-HAR** | 6 | 10299 | 561 | 2012-12-10 |
| **UCI-HAPT** | 12 | 10929 | 561 | 2015-07-29 |
| **DATASET-UCI** | 6 | 5744 | 561 | 2016-03-09 |

In Table 2.6, all data sets can be found in the UCI-HAR dataset repository. In the first line, the UCI-HAR dataset was developed and published in 2012 by Irvine University of California researchers [44]. In the second line, the UCI-HAPT dataset was an extended dataset of the UCI-HAR dataset released in 2015. Following

the UCI-HAR dataset, the DATASET-UCI dataset was released in 2016. These three datasets had already been split into a training set of 70% and a test set of 30%. Table 2.7 describes the list of activities and the percentage of each activity for three datasets.

**Table 2.7** The percentage of each activity per dataset

| No. | Activities | Percentage of each activity | | |
|---|---|---|---|---|
| | | UCI-HAPT | UCI-HAR | DATASET-UCI |
| 1 | Walking | 15.8% | 18.5% | 17.6% |
| 2 | Walking-Upstairs | 14.1% | 17.3% | 14.9% |
| 3 | Walking-Downstairs | 12.9% | 18.9% | 16.2% |
| 4 | Sitting | 16.5% | 16.7% | 19.6% |
| 5 | Standing | 18.1% | 13.7% | 17.9% |
| 6 | Laying | 17.9% | 15.0% | 13.8% |
| 7 | Stand-to-Sit | 0.6% | - | - |
| 8 | Sit-to-Stand | 0.3% | - | - |
| 9 | Sit-to-Lie | 1.0% | - | - |
| 10 | Lie-to-Sit | 0.8% | - | - |
| 11 | Stand-to-Lie | 1.3% | - | - |
| 12 | Lie-to-Stand | 0.8% | - | - |

In Table 2.7, all datasets include six basic types of activity such as walking, walking-upstairs, walking-downstairs, sitting, standing, and laying. However, there are more than six postural transition activities in the UCI-HAPT dataset, such as stand to sit, sit to stand, sit to lie, lie to sit, stand to lie, and lie to stand. The size of the UCI-HAR dataset is 10299 × 561 (5,777,739), the size of the UCI-HAPT dataset is 10929 × 561 (6,131,169), and the size of the DATASET-UCI dataset is 5744 × 561 (3,222,384). For all datasets, the accelerometer and gyroscope sensor of Samsung Galaxy SII mobile phone recorded 3-axial acceleration and 3-axial angular velocity

with the 50Hz sampling rate. These raw signals were removed from the noise and extracted the data from the fixed-width sliding window by 50% overlapping. From each window, the time-domain features and frequency-domain features were acquired. Table 2.8 shows the extracted features and detailed descriptions of variable types.

**Table 2.8** Description of variable types of features

| No. | Variable Types | Description |
|---|---|---|
| 1 | Mean () | Average value |
| 2 | Std () | Standard Deviation |
| 3 | Mad () | Median absolute deviation |
| 4 | Max () | Maximum value |
| 5 | Min () | Minimum value |
| 6 | Sma () | Signal magnitude area |
| 7 | Energy () | Energy value |
| 8 | Iqr () | Interquartile range |
| 9 | Entropy () | Signal entropy value |
| 10 | arCoeff () | Autoregression coefficient |
| 11 | Correlation () | Correlation coefficient |
| 12 | maxInds () | Index of the largest magnitude frequency component |
| 13 | meanFreq () | A weighted average of the frequency component |
| 14 | Skewness () | The skewness of the frequency domain signal |
| 15 | Kurtosis () | Kurtosis of the frequency domain signal |
| 16 | bandsEnergy () | Energy of the frequency by FFT of each window |
| 17 | Angle () | The angle between the vectors |

Table 2.8 denotes the acceleration signals of the user body and gravity as tBodyAcc-XYZ and tGravityAcc-XYZ. TBodyAccJerk-XYZ and tBodyGyroJerk-XYZ define the jerk signals the rate of acceleration change over time obtained from raw signals and angular velocity. Table 2.9 describes the extracted features of raw signals from the time-domain of three datasets.

**Table 2.9** Description of the types of raw signal in the time domain

| No. | Raw signals in the time domain | Description |
|---|---|---|
| 1 | tBody Acc-XYZ | Body acceleration in time |
| 2 | tGravityAcc-XYZ | Gravity acceleration in time |
| 3 | tBodyAccJerk-XYZ | Jerk in body acceleration in time |
| 4 | tBodyGyro-X YZ | Body Gyroscope measure in time |
| 5 | tBodyGyroJerk-XYZ | Jerk in body Gyroscope measure in time |
| 6 | tBody AccMag | Magnitude of body acceleration in time |
| 7 | tGravity AcMag | Magnitude of gravity acceleration in time |
| 8 | tBody AccJerkMag | Magnitude of jerk in body acceleration in time |
| 9 | tBodyGyroMag | Magnitude of body Gyroscope measure in time |
| 10 | tBodyGyroJerkMag | Magnitude of jerk in body Gyroscope measure in time |

There are 561 features in total, including 272 time-domain features, 289 frequency-domain features. Initially, 3-axial raw signals are referred to as ' XYZ ' from the three-axis directions X, Y, Z. To represent raw signals from the accelerometer and gyroscope, the feature names are expressed as tAcc-XYZ, tGyro-XYZ, etc. In which, the prefix ‘t’ is pointed out to express the time domain variables. Thirty volunteers participated in the same age group within 19-48 years in two datasets (UCI-HAR

dataset, UCI-HAPT). Table 2.10 shows in the extracted feature types of raw signals from the frequency domain of three datasets.

**Table 2.10** Description of the types of raw signal in the frequency domain

| No. | Raw signals in frequency domain | Description |
|---|---|---|
| 1 | fBody Acc-XYZ | Body acceleration in frequency |
| 2 | fBodyAccJerk-XYZ | Jerk in body acceleration in frequency |
| 3 | fBodyGyro-X YZ | Body Gyroscope measure in frequency |
| 4 | fBody AccMag | Magnitude of body acceleration in frequency |
| 5 | fBody AccJerkMag | Magnitude of jerk in body acceleration in frequency |
| 6 | fBodyGyroMag | Magnitude of body Gyroscope measure in frequency |
| 7 | fBodyGyroJerkMag | Magnitude of jerk in body Gyroscope measure |

However, 30 people performed 6 activities in the UCI-HAR dataset (walking, walking-upstairs, walking-downstairs, sitting, standing and laying) and 30 people performed 12 activities in the UCI-HAPT dataset (walking, walking-upstairs, walking-downstairs, sitting, standing, laying, stand to sit, sit to stand, sit to lie, lie to sit, stand to lie, and lie to stand). On the other side, 30 volunteers in the age group of 22-79 years performed 6 activities (walking, walking-upstairs, walking-downstairs, sitting, standing and laying) and gathered the DATASET-UCI dataset.

The magnitude of three-dimensional signals is designated with tBodyAccMag, tBodygyroMag, tBodyAccJerkMag, tBodyGyroJerkMag, and tGravityAccMag, tGravitygyroMag specify the magnitude of three-dimensional gravity signals. By implementing a Fast Fourier Transform (FFT), all above three-dimensional signals generated fBodyAcc-XYZ, fBodyGyro-XYZ, fBodyAccJerk-XYZ, fBodyGyroMag, fBodyAccJerkMag, and fBodyGyroJerkMag, where prefix ‘f’ was

indicated to show frequency domain variables. Variable type ‘angle’ of 7 features from time-domain is calculated for tBodyAccMean, tBodyAccJerkMean, tBodyGyroMean, tBodyGyroJerkMean, and gravityMean. Table 2.11 shows a detailed description of all features of three datasets as well as the total number of features.

**Table 2.11** Feature categorization and number of features from the specified datasets

| Category | Raw Signals Types | # Features | Variable Types |
|---|---|---|---|
| Time (272) | tBodyAcc-XYZ | 40 | Mean (), Std (), Mad (), Max (), Min (), Sma (), Energy (), Iqr (), Entropy (), arCorf (), Corelaion () |
| | tGravityAcc-XYZ | 40 | |
| | tBodyAccJerk-XYZ | 40 | |
| | tBodyGyro-XYZ | 40 | |
| | tBodyGyroJerk-XYZ | 40 | |
| | tBodyAccMag | 13 | |
| | tGravityAccMag | 13 | |
| | tBodyAccJerkMag | 13 | |
| | tBodyGyroMag | 13 | |
| | tBodyGyroJerkMag | 13 | |
| | (tBodyAccMean, Gravity)<br>(tBodyAccJerkMean, Gravity)<br>(tBodyGyroMean, Gravity)<br>(tBodyGyroJerkMean, Gravity)<br>(XYZ,GravityMean) | 7 | angle () |
| Frequency (289) | fBodyAcc-XYZ | 79 | maxInds (), meanFreq (), Skewness (), Kurtosis (), bandEnergy () |
| | fBodyAccJerk-XYZ | 79 | |
| | fBodyGyro-XYZ | 79 | |
| | fBodyAccMag | 13 | |
| | fBodyAccJerkMag | 13 | |
| | fBodyGyroMag | 13 | |
| | fBodyGyroJerkMag | 13 | |
| **Total features** | | **561** | |

Finally, each dataset was given with a total of 561 features performed by 30 volunteers for physical activities. The next section is to support the main idea of feature selection intended for use in this dissertation.

### 2.4.2 Linearly Dependent Concepts

Large datasets regularly consist of hundreds of millions of separable pieces of data information. Working with and operating on this data is easier when it is described in the form of vectors and matrices [110]. Linear equations are especially essential in modern science to create models with linear approximations. Therefore, linear algebra is also an important requirement in machine learning and information processing for algorithms [111]. Linear algebra is a branch of mathematics dealing with vectors and vector operations. There are three fundamental concepts: a linear combination; linearly dependence; and linearly independence. The following definitions and concepts are discussed in this dissertation with some examples.

***(i) Vector***

Vectors can be regarded as a number array. Usually, they are described by a bold letter as $\boldsymbol{x}$ in the lowercase. By subscription, the individual numbers are marked indicating the individual member's status. For instance, $x_1$is the first vector, $x_2$ is the second vector and so on. With the individual elements in square brackets, a vector can be expressed as an array,

$$\boldsymbol{x} = [\, x_1, x_2, \ldots, x_m \,].$$

where *m* is the last element of an array.

In a matrix of $\boldsymbol{m} \times \boldsymbol{n}$, a column vector or column matrix is a matrix of $\boldsymbol{m} \times \mathbf{1}$ which implies a matrix composed of a single column with m elements and a row vector or row matrix is a matrix of $\mathbf{1} \times \boldsymbol{n}$, that is to say a matrix composed of a single row with n elements as shown in Table 2.12.

**Table 2.11** Row vector and column vector

| Row vector | Column vector |
|---|---|
| $\boldsymbol{x} = [\, x_1, x_2, \ldots, x_m \,]$ | $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_m} \end{bmatrix}$ |

***(ii) Vector Addition and Multiplication***

Adding two vectors is completed by adding each respective component of a vector as shown in the example.

$$\boldsymbol{x} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \qquad y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$x + y = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$x + y = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix}$$

$$x + y = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

Vector multiplication can be described by multiplying a vector with a scalar for each respective vector component as shown below.

$$If\ \boldsymbol{x} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix},$$

$$then \qquad 2x = 2 \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$2x = \begin{bmatrix} 1 \times 2 & 2 \times 2 \\ 3 \times 2 & 4 \times 2 \end{bmatrix}$$

$$2x = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

***(iii) Matrix***

A matrix is an array of elements in two dimensions. It is represented by an uppercase letter such as A, and each element is represented by a small letter with two indices such as $a_{ij}$, where I represent the row and j represents the column. The following matrix is an m × n matrix. A matrix m x n implies that the matrix consists of vectors of m row and vectors of n column.

$$A = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ & & \vdots & \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

***(iv) Vector Space***

A *vector space* is a collection of vectors on which two operations + and · are defined, called *vector addition* and *scalar multiplication.* The $R^2$ space and $R^n$ space are examples of vector space as shown in Table 2.13.

**Table 2.13** Description of $R^2$ vector space and $R^n$ vector space

| Vector Space | Description |
|---|---|
| $R^2$ space | The xy plane represents the vector space $R^2$. There are two components in each vector v. |
| | $\boldsymbol{R^2 = \{v = (x_1, x_2) / x_1, x_2 \in R^2\}}$ |
| $R^n$ space | The $R^n$ space is a set of all vectors containing *n* components in each vector v. |
| | $\boldsymbol{R^n = \{v = (x_1, x_2, \ldots, x_n) / x_1, x_2, \ldots, x_n \in R^n\}}$ |

***(v) Linear Combination***

If a vector $v = (x_1, x_2, \ldots, x_n) \in R^n$ can be written as below:

$$v = c_1x_1 + c_2x_2 + \ldots + c_nx_n \quad (2.1)$$

$$\text{or} \quad v = \sum_i^n c_i\, x_i \quad (2.2)$$

Where $c_1, c_2, \ldots, c_n$ are scalars in *R*.

Then *v* is called a linear combination of the vectors $x_1, x_2, \ldots, x_n$.

### *(vi) Homogeneous Linear Combination*

If a linear equation $c_1x_1 + c_2x_2 + ...,+ c_nx_n$ is equal to zero, it is called a linear homogeneous equation. In symbol,

$$\sum_i^n c_i\, x_i = 0 \tag{2.3}$$

where $x_1, x_2, ..., x_n$ *are vectors in* $R^n$ and $c_1, c_2, ..., c_n$ are scalars in *R*.

### *(vii) Homogeneous Linear Combination System*

If more than one homogeneous linear equation exists, it can be called a homogeneous linear system of mixture. For example, below is described a homogeneous linear combination system with ***m*** homogeneous linear equations.

$$\left.\begin{array}{r} c_1x_{11} + c_2x_{12} + \ \ldots + c_nx_{1n} = 0 \\ c_1x_{21} + c_2x_{22} + \ \ldots, + c_nx_{2n} = 0 \\ \vdots \\ c_1x_{m1} + c_2x_{m2} + \ \ldots, + c_nx_{mn} = 0 \end{array}\right\} \tag{2.4}$$

where $x_{11}, x_{12}, \ldots, x_{1n}$ *are vectors in* $R^n$ and $c_1, c_2, ..., c_n$ are scalars in *R*.
Then this system (2.4) is called as a homogeneous linear combination system. This system can be written in a matrix form as illustrated below.

$$\begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ & & \vdots & \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Then,

$$A = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ & & \vdots & \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}, \quad 0 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{2.5}$$

Then the system of linear combination can be written as a matrix notation

$$Ac = 0 \quad . \tag{2.6}$$

where $A$ is an $m \times n$ matrix and '0' is a zero matrix.

### *(viii) Linear Independence and Linear Dependence*

If a homogeneous linear equation of the vectors $(x_1, x_2, \dots, x_n) \in R^n$ is

$$\sum_j^m \sum_i^n c_i \, x_{ji} = 0 \tag{2.7}$$

then we can find all values of $c_1, c_2, \dots, c_n$.

(a) If each $\boldsymbol{c_i}$ is equivalent to zero, then the vectors $x_1, x_2, \dots, x_n$ are linearly independence.

(b) If there exists a $\boldsymbol{c_i}$ is not equivalent to zero, then the vectors $x_1, x_2, \dots, x_n$ are linearly dependence.

### *(ix) Trivial Solution and Nontrivial Solution*

A homogeneous linear equation of the vectors $(x_1, x_2, \dots, x_n) \in R^n$ has two kinds of solutions trivial solution and nontrivial solution.

(a) If every $\boldsymbol{c_i}$ is equal to zero, the equation (2.7) has a trivial solution.

(b) If there exists a $\boldsymbol{c_i}$ is not equal to zero, the equation (2.7) has a nontrivial solution.

### *(x) Gauss Jordan Elimination Method*

Gauss Jordan elimination method is very popular in many engineering and science for solving linear systems [112]. The main technique of this method is reducing to row echelon form of the augmented matrix of the linear system. The method uses the following elementary row operations to solve the system.

(a) Interchanging two rows.

(b) Multiplying one of the rows by a nonzero constant.

(c) Adding a multiple of one row to another row.

### *(xi) Example of Trivial Solution*

Given: $v = \{(x_1, x_2, x_3\}$

where $\boldsymbol{x_1} = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, x_2 = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix}, x_3 = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}.$

Then homogeneous linear combination system is as follows.

$$c_1x_1 + c_2x_2 + c_3x_3 = 0$$

where $c_1$, $c_2$, $c_3$ are any scalar numbers.

The trivial solution of that system by using Gauss Jordan elimination method is as below.

$$c_1x_1 + c_2x_2 + c_3x_3 = 0$$

Here

$$c_1 \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} + c_2 \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} + c_3 \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} .$$

By vector addition, we can construct a linear combination system with four equations as follows.

$$\left.\begin{array}{rrrrr} 2c_1 + & 3c_2 + & c_3 & = 0 \\ c_1 & & & = 0 \\ & 2c_2 + & 2c_3 & = 0 \\ c_1 + & c_2 & & = 0 \end{array}\right\}$$

Then we can find all values of scalars by using Gauss Jordan elimination method.

$$\left[\begin{array}{cccc} 2 & 3 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 1 & 1 & 0 & 0 \end{array}\right] \xrightarrow[\text{Elimination}]{\text{Gauss Jordan}} \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Here, $c_1 = 0$.

$c_2 = 0$.

$c_3 = 0$.

Therefore, $v$ is a linear independence.

And the system has a trivial solution.

### *(xii) Example of Nontrivial Solution*

Given: v = {($x_1$, $x_2$, $x_3$}

where $x_1 = \begin{bmatrix} -3 & 0 \\ 5 & 3 \end{bmatrix}, x_2 = \begin{bmatrix} 12 & 0 \\ 4 & -18 \end{bmatrix}, x_3 = \begin{bmatrix} 6 & 0 \\ -2 & -8 \end{bmatrix}.$

Then homogeneous linear combination system is as follows.

$$c_1x_1 + c_2x_2 + c_3x_3 = 0$$

where $c_1$, $c_2$, $c_3$ are any scalar numbers.

It can be determined by the nontrivial solution of that system by using the Gauss Jordan elimination method as below.

$$c_1x_1 + c_2x_2 + c_3x_3 = 0$$

Here,

$$c_1 \begin{bmatrix} -3 & 0 \\ 5 & 3 \end{bmatrix} + c_2 \begin{bmatrix} 12 & 0 \\ 4 & -18 \end{bmatrix} + c_3 \begin{bmatrix} 6 & 0 \\ -2 & -8 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

By vector addition, a linear combination system can be constructed with three equations as follows.

$$\begin{aligned} -3\boldsymbol{c_1} + 12\boldsymbol{c_2} + \boldsymbol{6c_3} &= 0 \\ 5\boldsymbol{c_1} + 4\boldsymbol{c_2} - 2\boldsymbol{c_3} &= 0 \\ \boldsymbol{3c_1} - 18\boldsymbol{c_2} - 8\boldsymbol{c_3} &= 0 \end{aligned}$$

Then, using Gauss Jordan elimination technique, all scalar values can discover.

$$\left[\begin{array}{ccc|c} -3 & 12 & 6 & 0 \\ 5 & 4 & -2 & 0 \\ 3 & -18 & -8 & 0 \end{array}\right] \xrightarrow[\text{Elimination}]{\text{Gauss Jordan}} \left[\begin{array}{ccc|c} -3 & 12 & 6 & 0 \\ 5 & 4 & -2 & 0 \\ 3 & -18 & -8 & 0 \end{array}\right]$$

Hence, $c_1 = 2$.

$c_2 = -1$.

$c_3 = 3$.

The values show that all scalar values are not zeros.

Therefore, v is a linear dependence.

And the system has a non-trivial solution.

### 2.4.3 Group Theory

Group theory, a branch of abstract algebra, is an essential tool in some scientific fields such as mathematics, physics, chemistry, etc. It is a symmetrical analysis that remains invariant under certain transformations [113]. Group theory supports to predict the existence of group elements. Different structures and behaviors of objects have different symmetry. For the asymmetry of polygons, group theory is very essential in geometry. Group theory is applied in number theory and Elliptic groups are widely applied in modern cryptography fields.

A cyclic group is a single-element abelian group. In group theory, the fundamental properties of cyclic groups play a vital role. The authors viewed the fundamental of a cyclic group [114]. The writers discussed the significance of polygon stars in art and culture [115]. The study of abstract groups is very challenging to get the algebraic concepts. The authors attempted to understand the structure of the cyclic group for algebra concepts using stars polygons [116]. The study suggested a technique for finding expectation numbers of cyclic groups [117].

There are many complicated classes with different kinds of elements in a group. A classification of that group's affiliation is one of the most significant points in a group. Sometimes, groups contain some arbitrary subgroups without specific properties. Sometimes, groups with some binary operations could also come up with special properties. In [118], the authors investigated the relationship between cyclic soft groups and classical groups. The authors proposed a method to find an average order of the elements in a cyclic group of order $n$ and proved that an arbitrary group G is cyclic if and only if distinct subgroups of G have distinct indexes in G [119]. The reason why this section is interested in the group theory and fundamental properties of the cyclic group with a binary operation involving some special properties.

(1) **Group**

Let $G= \{g_1, g_2, \ldots ,g_n \}$ be a finite set of elements. Then $G$ is called a group if it fulfills four properties (closure, associativity, unity, and inverse element) under addition operation as follows:

| | | | |
|---|---|---|---|
| (i) | $\forall a, b \in G,$ | $c = a + b \in G$ | (Closure) |
| (ii) | $\forall a, b, c \in G,$ | $(a + b) + c = a (b + c) \in G$ | (Associativity) |
| (iii) | $\forall \boldsymbol{a} \in \boldsymbol{G}, \exists \boldsymbol{e} \in G,$ | $a + e = a$ | (Identity) |
| (iv) | $\forall \boldsymbol{a} \in \boldsymbol{G}, \exists \boldsymbol{b} \in G,$ | $a + b = e$ | (Inverse element) |

**(2) Order of Group *G***

Let $G$ be a group. Then the order of a group $G$ is the total number of elements in $G$ denoted by either $|G|$ or $O(G)$.

**For example:**

Given: $G = \{1,2,3,4,5,6,7\}$.

Then order of a group $G$ is $|G| = O(G) = 7$

**(3) Cyclic Group**

A group $G$ is a cyclic group if that is generated by a single element of it. Instantly, if $G = \langle a \rangle = \{na / a \in G, n \in \boldsymbol{Z}\}$ is called a cyclic group generated by '$a$'.

**For example:**

Let $G = \{1,2,3,4,5,6,7,\}$ and $Z_5 = \{2,3,4,5,6\}$, then

$Z_5 = \langle 1 \rangle$ with $|1| = 5$

Here, we call $Z_5 = \langle 1 \rangle$ is a cyclic group generated by 1.

**(4) Fundamental Properties of Cyclic Group**

Every cyclic group satisfies the following fundamental properties:

(a) Every group is a cyclic group itself with generator 1.

(b) Every subgroup of a cyclic group is also a cyclic group.

(c) The order of every subgroup is a divisor of the order of group $G$.

(d) There exists exactly one subgroup of order k which is divisor of the order of $G$.

**For example:**

Let $G = Z_8 = \{1,2,3,4,5,6,7,8\}$

Then $G$ is a cyclic group generated by 1 with $|G| = 8$.

The divisors of 8 are 2 and 4.

Since, each divisor has exactly one subgroup of *G.*

<2> = {2,4,6,8} with order 4 and

<4>= {4,8} to order 2 are subgroups of *G.*

(5) **Euler Phi Function**

The function of Euler Phi $\boldsymbol{\varphi}$ is described as the total number of elements in a cyclic group. If there is a divisor *d* of the order of group *G*, then the number of elements of that devisor d can be expressed as

$\boldsymbol{\varphi}$ = the number of positive integers less than d and relatively prime to *d*

Some examples of total number of elements of cyclic group *G* are shown in Table 2.14.

**Table 2.14** Example of the total number of elements of cyclic subgroup G

| $\|G\|$ (Order of G) | $d$ (Divisor of \|G\|) | $\boldsymbol{\varphi}$ (Euler Phi) |
|---|---|---|
| 4 | 2, 4 | 1, 2 |
| 6 | 2, 3 | 1, 2 |
| 8 | 2, 4, 8 | 1, 2, 4 |
| 10 | 2, 5 | 1, 4 |

In Table 2.14, the word ***G*** means a cyclic group *G* generated by 1 and the word $\boldsymbol{|G|}$ is the order of group *G*. The word ***d*** refers to the divisor of the order of group *G*. The word $\boldsymbol{\varphi}$ is the total number of elements of the cyclic subgroup of *G.*

# CHAPTER 3

# METHODOLOGY

The main goal of this work is to develop a HAD system using high-quality features and a best classifier. New feature selection techniques and a new classification design are the focal points of the work. This chapter of the dissertation is organized into two main sections. An initial assessment of the HAD system was made based on data collection and step counting. The first section of the chapter deals with the performance and evaluation of these two tasks. The second section discusses the five primary concerns of this research: data selection, data preprocessing, feature selection, classification, and performance evaluation.

## 3.1 Task_1 (Understanding Work in HAD)

### 3.1.1 Data Collection

To understand how a HAD system works, it is first necessary to collect human activity data. The collected data are needed to control real activity recognition [120]. The creation of a new human activity dataset is described in this section. The main goal is to identify the key characteristics of sensor data that determine the activity. The data collection process is the preliminary step in determining human activity and uses the built-in sensors on a mobile phone. Many types of sensors can be implemented to retrieve sensor data for human activity recognition. However, retrieved sensor datasets differ from each other due to the following factors:

- Different types of sensors
- Different integrated devices
- Different sensor locations
- Different activities
- Different type of data collection protocols
- Different sampling rates
- Different types of users

In this work, a Samsung Galaxy Note 4 device using the Android 6.0.1 operating system was selected as the data collection tool. Three sensors (accelerometer, gyroscope, and magnetometer) were used to retrieve data. A new data collection tool using version 3.1.4 of Android Studio was developed and installed on the Samsung Galaxy Note 4 to retrieve data produced by physical activity.

Thirty healthy volunteers of various heights, weights, and ages participated. Most were international students at Prince of Songkla University from countries such as Pakistan, the Philippines, Cambodia, Vietnam, Bangladesh, Thailand, and Myanmar. Data collection took place at the Department of Computer Science, Faculty of Science, Prince of Songkla University. Participants followed guidance on how to perform the data collection process. Before the tests began, the characteristics of participants such as age, sex, height, and weight were recorded. To collect data produced by eight human activities, the Samsung Galaxy Note 4 was connected to the waist-bag of the users. The three sensors collected data of the user's movements every 100 milliseconds. Every 2 minutes, each user performed a sequence of activities. This collected dataset was named Testing Activity Data (TAD). There were eight activities to perform in this data collection as shown in Table 3.1.

**Table 3.1** Descriptions of Activities

| Class | Activity | Description |
|---|---|---|
| 1 | Sitting | Sitting on a chair |
| 2 | Standing | Standing straight |
| 3 | Walking | Normal walking forward on a flat floor |
| 4 | Walking- upstairs | walking up the stairs |
| 5 | Walking-downstairs | walking down the stairs |
| 6 | Laying | Laying on the bed |
| 7 | Stand-walking | Walking with keeping the position |
| 8 | Stand-jogging | Jogging with keeping the position |

A series of instances were collected as raw data containing a timestamp, three accelerometer values along the X-axis, Y-axis, and Z-axis, three gyroscope values along the X-axis, Y-axis, Z-axis, and three magnetometer values along the X-axis, Y-axis, and Z-axis. Detailed information for TED data collection is shown in Table 3.2.

**Table 3.2** Detail descriptions of TED data collection

| Type | Description |
|---|---|
| Mobile phone | Samsung Galaxy Note 4 |
| Number of sensors | Accelerometer, Gyroscope, Magnetometer |
| Number of subjects | 30 |
| Number of activities | 8 |
| Mobile phone location | User's waist-bag |
| Time stamp | 100ms |
| Total number of samples | 150,982 |
| Tagging raw data | [Timestamp], [Ac-X], [Ac-Y], [Ac-Z], [Gy-X], [Gy-Y], [Gy-Z], [MX], [MY], [MZ], [Steps], |

In Table 3.2, the word ***Timestamp*** means the occurrence of time to record the specific user's activity. The word ***Ac*** means an accelerometer value. The word ***Gy*** refers to gyroscope value, and the ***M*** is the magnetometer value. The words ***X, Y, and Z*** mention along the X-axis, Y-axis, and Z-axis. The word ***Steps*** implies the number of counting steps for walking by each user.

The collected TAD dataset was accomplished and compared with UCI-HAR [25] dataset that was planned to use in this research. It is a common dataset widely used by current researchers. This dataset includes 6 activities of 30 subjects wearing the smartphone at their waist. The activity data were recorded by the accelerometer and gyroscope with a sampling rate of 50Hz. A comparison of a brief description of UCI-HAR dataset and TAD dataset is given in Table 3.3.

Once obtained, the collected TAD dataset was compared with the UCI-HAR dataset [25] chosen for use in this research. The UCI-HAR dataset is widely used by current researchers and comprises data of six activities performed by 30 subjects wearing a smartphone at their waist. The activity data were recorded by the accelerometer and gyroscope at a sampling rate of 50Hz. Brief descriptions of the UCI-HAR dataset and the TAD dataset are given for comparison in Table 3.3.

**Table 3.3** A comparison of UCI-HAR dataset and TAD dataset

| Description | UCI-HAR dataset | TAD-Dataset |
|---|---|---|
| Mobile phone | Samsung Galaxy SII | Samsung Galaxy Note4 |
| Sensors type | Accelerometer, Gyroscope | Accelerometer, Gyroscope, magnetometer |
| Number of subjects | 30 | 30 |
| Sampling rate | 50Hz | 10Hz |
| Sensor location | User's waist bag | User's waist bag |
| Number of activities | 6 | 8 |
| Number of samples | 10299 | 150982 |

The comparative results in Table 3.3 show that the production of the datasets was characterized by several different factors:

- Different devices
- Different types of sensors
- Different types of activities
- Different sampling rates

While a Samsung Galaxy Note 4 was the embedded TAD data collection tool, a Samsung Galaxy SII was the embedded UCI-HAR data collection tool. Where the TAD dataset was retrieved by 3 sensors (accelerometer, gyroscope, and magnetometer), the UCI-HAR dataset was collected by 2 sensors (accelerometer, and

gyroscope). While 30 users performed 8 activities (walking, walking upstairs, walking downstairs, sitting, standing, lying down, walking on the spot, and jogging on the spot) to provide the TAD data, 30 volunteers performed 6 activities (walking, walking upstairs, walking downstairs, sitting, standing, lying down) to provide data for the UCI-HAR dataset.

In addition, sensing frequency for the TAD dataset was 10HZ and 50Hz frequency was given by the UCI-HAR dataset. In these factors of views, the TAD dataset and UCI-HAR dataset differ from each other. Due to difficulty to compare different datasets together. Therefore, 3 benchmark datasets (UCI-HAR, DATASET-UCI, HAPT) intended to use in this dissertation will be discussed in the next section. As a benefit of collecting human activity data, the knowledge of sensing data and its characteristics is to be supported in finding new techniques for feature selections.

Moreover, the sensing frequency to collect the TAD dataset was 10HZ and the sensing frequency of UCI-HAR dataset was 50Hz. In view of these factors, the TAD dataset and UCI-HAR dataset differ from each other. Due to the difficulty of comparing these different datasets, this study used three benchmark datasets (UCI-HAR, DATASET-UCI, and UCI-HAPT), which will be discussed in the next section. The knowledge gained from the collected human activity data and its characteristics will support the development of new techniques for feature selection.

### 3.1.2 Step Counting

Beyond the data collection aspect, further understanding of HAD systems followed from step counting using the built-in accelerometer of the mobile phone. This sensor provided data from real activity that was applied in the assessment of HAD before development began.

In this task, the Samsung Galaxy Note 4 operating with Android 6.0.1 was used to count the walking steps of the user. An Android studio application was introduced to collect the data. Data were recorded with the acquisition protocol. The sensor recorded the user's horizontal movement along the X-axis, vertical movements along the Y-axis, and the forward and backward movements along the Z-axis. Under instructions, nine participants (6 males, 3 females) performed walking activities. The

experiment involved three types of walking (normal walking, slow walking, and fast walking) in four sensor location modes (in-hand swinging-mode, texting mode, waist-bag mode, and arm-bag mode). Brief descriptions of walking styles for step counting are listed in Table 3.4.

**Table 3.4** Three types of walking during step counting

| Walking Types | Description | Notation |
|---|---|---|
| Normal-Walking | Walking at ordinary pace | N-walk |
| Slow-Walking | Walking more slowly than ordinary walking | S-walk |
| Fast-Walking | Walking faster than ordinary walking | F-walk |

Four different sensor usage modes indicating where the participants carried the mobile phone are shown in Table 3.5.

**Table 3.5** Four usage modes for mobile location

| Usage Modes | Description | Notation |
|---|---|---|
| In-Hand Swinging mode | Carrying the mobile phone in the hand with a swinging motion | Swinging-M |
| Texting mode | Holding the mobile phone in the hand while text messaging | Texting-M |
| Waist-bag mode | Carrying the mobile phone in a waist-bag | Waist-bag-M |
| Arm-bag mode | Carrying the mobile phone in an arm-bag | Arm-bag-M |

The walking data collected by the accelerometer were recorded with the subject ID, activity name, and number of steps. To prevent mislabeling, the participants were required to stop and change the labels on the phone after each activity. The participants had to follow the experimental rules precisely and were not permitted to alter the position of the mobile phone during the process.

The accelerometer collected walking data at a frequency of 10 Hz. After collection, data were adjusted to cancel signal noise. The gravitational force was separated from the raw signal by a high-pass filter of 0.8. After gravitational isolation,

a low-pass filter of 0.3 reduced unwanted noise signals to make acceleration signals smoother for each walking activity. In addition to performing three modes of walking, participants had to position the phone in four places for each mode of walking. Therefore, there were four categorizations of each walking mode. Hence, there were twelve activity categories for each user in this study, as shown in Figure 3.1.

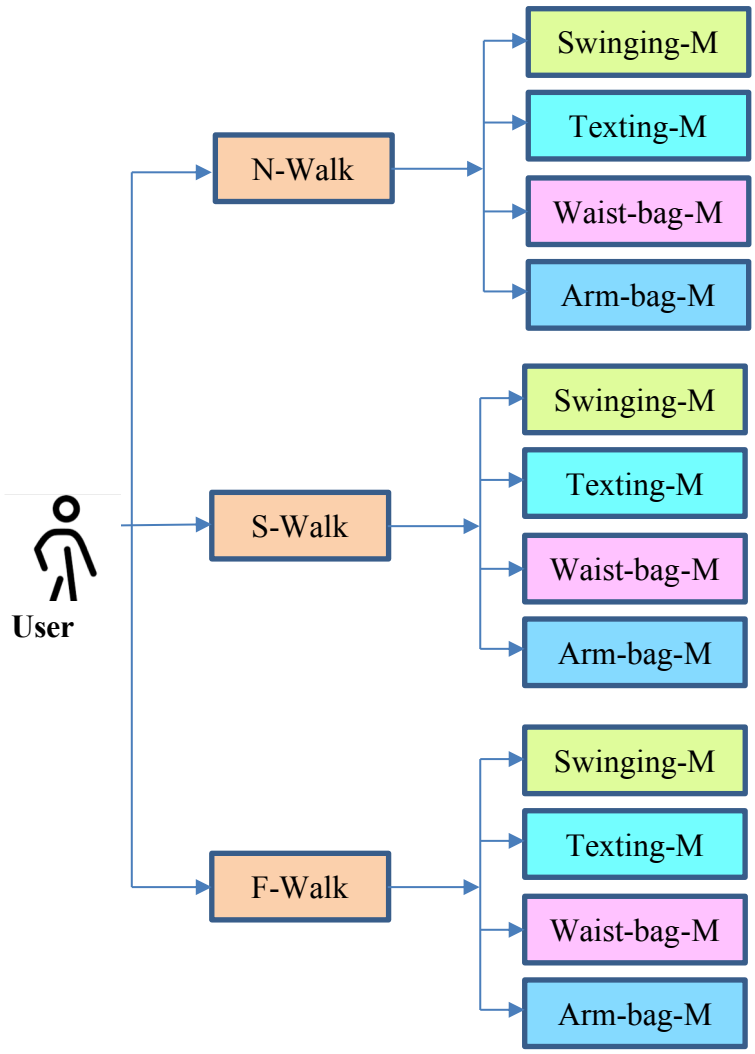



**Figure 3.1** Twelve categorizations of walking activity for each user

The step counting algorithm used in this task mainly relied on the acceleration value, described in equation (3.1), with a sinusoidal pattern.

$$a = \sqrt{Ac_x{}^2 + Ac_y{}^2 + Ac_z{}^2} \tag{3.1}$$

where $a$ is the magnitude of acceleration on sensor data and

$Ac_x$, $Ac_y$, and $Ac_z$ are acceleration values along the X-axis, Y-axis, and Z-axis.

**Step detection algorithm**

Input: $a$ // Magnitude of acceleration
Output: $Total$ // Number of walking steps

1. $initialize \;\; w \leftarrow \phi$ , $step \leftarrow 0, i \leftarrow 2, m$
2. $while\ i < m \;\; do$
3. $\quad if\ a_i > a_{i-1}\ \&\ a_i > a_{i+1}\ then$
4. $\quad\quad w\,[i] \leftarrow 1$
5. $\quad else$
6. $\quad\quad w\,[i] \leftarrow 0$
7. $j \leftarrow 0, k \leftarrow 0$
8. $while\ \ j < m \;\; do$
9. $\quad if\ w[j] = \ 1\ then$
10. $\quad\quad if\ k \ \neq 0\ then$
11. $\quad\quad\quad \boldsymbol{B} \leftarrow j - k - 1$
12. $\quad\quad\quad if\ B > 2\ then$
13. $\quad\quad\quad\quad step \leftarrow \ step + 1$
14. $\quad\quad k \leftarrow j$
15. $\quad j + +$
16. $if\ j = m \;\; then$
17. $\quad B \leftarrow m - k$
18. $\quad if\ B > 2 \ \ then$
19. $\quad\quad step \leftarrow \ step + 1$
20. $Total \leftarrow step$
21. $Return\ Total$

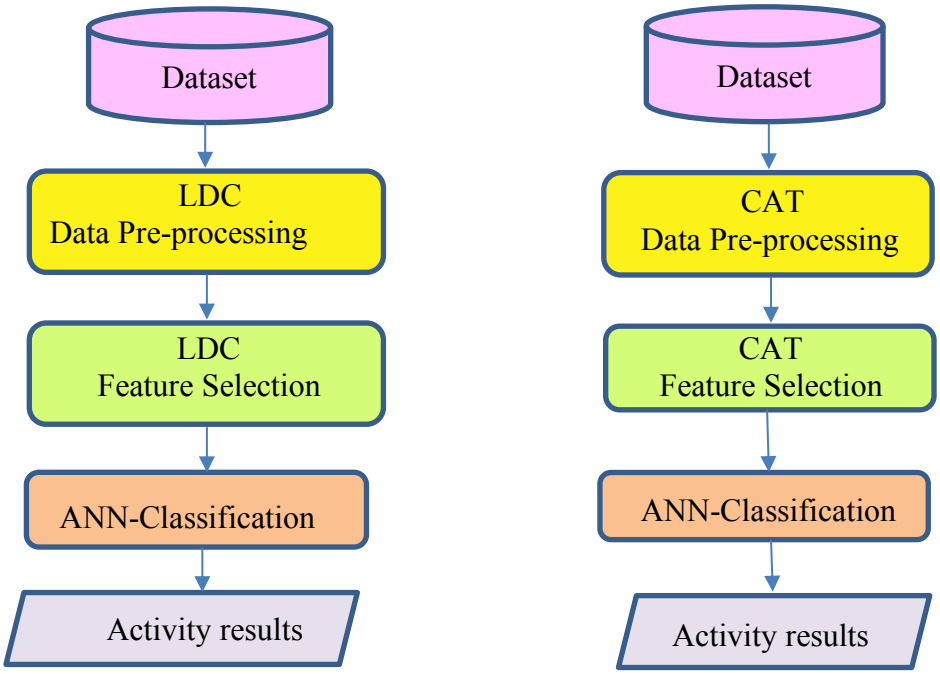
**Figure 3.2** Step detection algorithm

In Figure 3.2, the term *a* refers to the magnitude of the acceleration value calculated from equation 3.1. The term *Total* means the total number of steps. The term *w* refers to the peak point, the term *m* refers to the total number of time-stamps, the term *step* refers to the detection step point calculated by the algorithm, and the term *B* refers to step occurrences that the algorithm detects at the step points. Based on the information acquired from the data collection task, two new feature selection

methods and a new classifier technique will be designed in the second task, system design.

### 3.2 Task_2 (Implementing the system design)

The second focus of this work, the main system design, requires the introduction of some fundamental theories. The design of this HAD system is based on feature selection and classification. The key objective of this research is the identification of the most suitable classifier using a feature selection algorithm for ML problems. Two different types of feature selection techniques were applied to find the best classifier for HAD. The first technique used in effective HAD system construction was the Linear Dependent Concept (LDC) and the second technique was the Cyclic Attribution Technique (CAT). Different types of feature selection techniques entail different procedures for testing human activities. The system designs in this study are shown in Figure 3.3.




(a) LDC system design (b) CAT system design

**Figure 3.3** System designs of HAD based on LDC and CAT feature selections

In figure 3.3, the system has three main functions: 1) data preprocessing, 2) feature selection, and 3) classification. The input of the system is three datasets from UCI-Repository datasets and output is human activities.

### 3.2.1 Dataset

This section deals with the three datasets of human activity recognition that were obtained from the UCI repository for use in this dissertation, namely the UCI-HAR dataset [25], the UCI-HAPT dataset [26], and the DATASET-UCI dataset [27]. These datasets were used through pre-processing, feature selection, implementation, and testing to determine the best-suited classifier for human activities. The general description of the three datasets is shown in Table 3.6.

**Table 3.6** General Information of three datasets

| Dataset | # Activities | # Instances | # Attributes | Released Date |
|---|---|---|---|---|
| UCI-HAR | 6 | 10299 | 561 | 2012-12-10 |
| UCI-HAPT | 12 | 10929 | 561 | 2015-07-29 |
| DATASET-UCI | 6 | 5744 | 561 | 2016-03-09 |

The datasets in Table 3.6 are the most well-known publicly accessible UCI repository datasets from the field of human activity recognition (available from http://archive.ics.uci.edu/ml). Although the total number of attributes for all datasets is the same, the total number of instances is different. Moreover, while the UCI-HAR dataset and DATASET-UCI were derived from the same activities, the UCI-HAPT dataset was derived from six more activities. The activities of each dataset are described in Table 3.7 and a brief description of each dataset is shown in Table 3.8.

In Table 3.7, although the HAPT dataset contains six more activities, the first six are the same in all three datasets. In order to compare the performance results in the HAPT dataset with the other two datasets, it was divided into two datasets, namely HAPT_1 and HAPT_2. The HAPT_1 dataset comprised the first six activities which are the same as the activities in the UCI-HAR and DATASET-UCI

datasets and HAPT_2 contained the complete original UCI-HAPT dataset. Therefore, there were four datasets to apply when evaluating the performances of HAD feature selection with the designed classifiers.

**Table 3.7** Activities of each dataset

| No. | Activities | UCI-HAR | DATASET-UCI | UCI-HAPT |
|---|---|---|---|---|
| 1 | Walking | √ | √ | √ |
| 2 | Walking-Upstairs | √ | √ | √ |
| 3 | Walking-Downstairs | √ | √ | √ |
| 4 | Sitting | √ | √ | √ |
| 5 | Standing | √ | √ | √ |
| 6 | Lying down | √ | √ | √ |
| 7 | Standing-to-Sitting | - | - | √ |
| 8 | Sitting-to-Standing | - | - | √ |
| 9 | Sitting-to-Lying | - | - | √ |
| 10 | Lying-to-Sitting | - | - | √ |
| 11 | Standing-to-Lying | - | - | √ |
| 12 | Lying-to-Standing | - | - | √ |

Table 3.7 provides a short overview of these four datasets, which will be used throughout this research in pre-processing, feature selection, implementation, and classification tasks.

**Table 3.8** Four datasets to use in system

| No. | Dataset | No. Activities | No. Instances | No. Features |
|---|---|---|---|---|
| 1. | UCI-HAR | 6 | 10299 | 561 |
| 2. | DATASET-UCI | 6 | 5744 | 561 |
| 3. | HAPT_1 | 6 | 10411 | 561 |
| 4. | HAPT_2 | 12 | 10929 | 561 |

### 3.2.2 Data pre-processing

Two new feature selection methods (LDC and CAT) are introduced in this chapter. Since different feature selection methods have different data pre-processing protocols, data pre-processing protocols for the two different feature selection methods are also addressed in this section.

#### 3.2.2.1 Data pre-processing for LDC

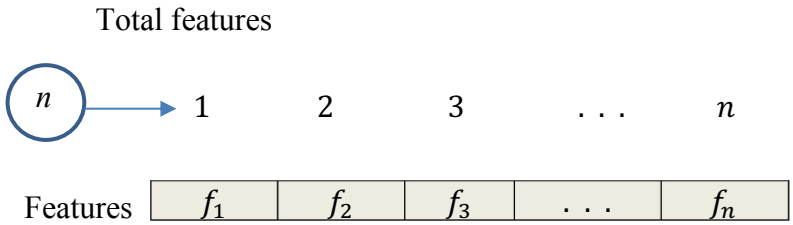
The four datasets were split into training sets (70%) and testing sets (30%). In the first step of data pre-processing for LDC, these training and testing sets were combined as a single dataset and stored in a two-dimensional array as shown below.

| Instance index $i$ \ Feature index $j$ | 1 | 2 | . . . | $n$ |
|---|---|---|---|---|
| 1 | $a_{11}$ | $a_{12}$ | . . . | $a_{1n}$ |
| 2 | $a_{21}$ | $a_{22}$ | . . . | $a_{2n}$ |
| 3 | $a_{31}$ | $a_{32}$ | . . . | $a_{3n}$ |
| . | . | . | . .. | . |
| . | . | . | | . |
| $k$ | . | . | | . |
| | $a_{k1}$ | $a_{k2}$ | $a_{k3}$ | $a_{kn}$ |

(3.2)

where $a_{ij}$ is a sensor value, $1 \leq i \leq k$ , and $1 \leq j \leq n$.

The term $n$ is the total number of features, $k$ is the total number of instances, $i$ is the row index, and $j$ is the column index. Then, the sequence of feature $f_i$ , where $1 \leq i \leq n$, is stored in a one-dimensional array indexed by $n$ as described below,




where $f_i$ s the given feature, $1 \leq i \leq n$, and $n$ is the total number of features.

This method prepares data for transformation into linear combination space. Due to the basic concepts of LDC, given data cannot be considered if divided into smaller groups. The CAT was taken into consideration for feature selection to try and overcome this challenge to more efficient feature selection.

### 3.2.2.2 Data pre-processing for CAT

Since the use of CAT aims to overcome some of the shortcomings of the LDC method, the same datasets used in the LDC method were also used in the CAT method. In the first stage, pre-processing for CAT is the same as pre-processing for LDC. The CAT method then continues with further pre-processing. This method first separates data into appropriate domains (time domain and frequency domain), as illustrated in Figure 3.4.

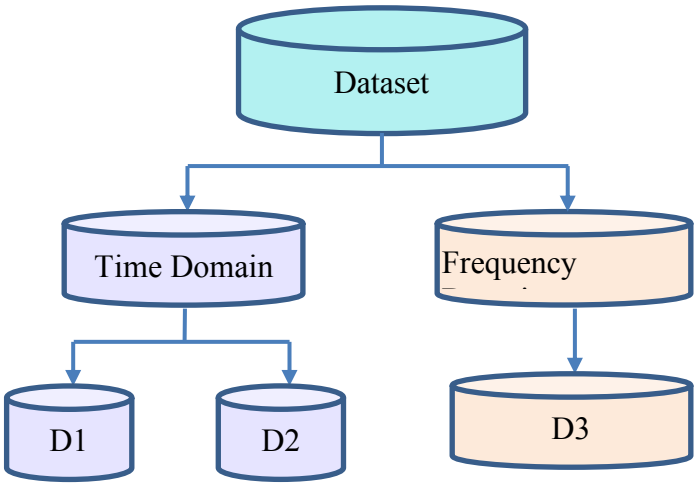



**Figure 3.4** Dividing data into three feature subgroups

In Figure 3.4, D1 is data with time domain-relevant features based on variable values; D2 is data with time-domain features based on angle values; D3 is data with frequency domain-relevant features. For each dataset, the total numbers of features placed in D1, D2, and D3 are shown in Table 3.9.

**Table 3.9** Total numbers of features for each subgroup of data in each dataset

| Dataset | D1 | D2 | D3 | Total features |
|---|---|---|---|---|
| UCI-HAR | 265 | 7 | 289 | 561 |
| DATASET-UCI | 265 | 7 | 289 | 561 |
| HAPT_1 | 265 | 7 | 289 | 561 |
| HAPT_2 | 265 | 7 | 289 | 561 |

At the first stage of implementing CAT, the three different feature subgroups (D1, D2, D3) are divided into two group-sets of input features ($G$) and input instances ($H$). The detailed sets of input features and instances are listed in Table 3.10. There are three input feature groups: $G_1$ comes from $D_1$ of the time domain, $G_2$ from $D_2$ of the time domain, and $G_3$ comes from $D_3$ of the frequency domain.

**Table 3.10** Total numbers of features for each subgroup of data in each dataset

| Dataset | No. Input Instances ($H$) | No. Input Features ($G$) | | |
|---|---|---|---|---|
| | | $G_1$ | $G_2$ | $G_3$ |
| UCI-HAR | 10299 | 265 | 7 | 289 |
| DATASET-UCI | 5744 | 265 | 7 | 289 |
| HAPT_1 | 10411 | 265 | 7 | 289 |
| HAPT_2 | 10929 | 265 | 7 | 289 |

### 3.2.3 Feature Selection

Due to the importance of the relevance of features for classification, accurate feature selection is key to solving ML problems. Although there are many feature selection methods, the most appropriate features have not yet been identified for predicting problems with human activity. The objective of this section is to find the best-suited features for the determination of human activity.

LDC is intended to define the useful features in determining human activity. LDC can properly reduce the irrelevant features of each dataset [120]. There would remain some advantages for the feature selection of human activity recognition LDC feature selection method can solve some problems of feature redundancy. However, it cannot identify which features are more useful because it uses all features to compute for transformation. Therefore, the concept of a new feature selection technique based on the Cyclic Attribute Technique (CAT) is introduced not only to determine key features but also to further reduce irrelevant features in HAD.

#### 3.2.3.1 Linearly Dependent Concept (LDC) feature selection method [i]

To remove redundant features, the LDC feature selection method uses linearly dependent concepts. Six new statements using linear dependent ideas are introduced in this section before a novel LDC feature selection method is developed.

**Statement (1): Dataset Expression in Matrix Form**

Every two-dimensional dataset can be expressed in a matrixed form.

As an example, a two-dimensional dataset is as described below.

| | 1 | 2 | . . . | $n$ |
|---|---|---|---|---|
| 1 | $a_{11}$ | $a_{12}$ | . . . | $a_{1n}$ |
| 2 | $a_{21}$ | $a_{22}$ | . . . | $a_{2n}$ |
| 3 | $a_{31}$ | $a_{32}$ | . . . | $a_{3n}$ |
| . | . | . | . .. | . |
| . | . | . | | . |
| . | . | . | | . |
| $k$ | $a_{k1}$ | $a_{k2}$ | $a_{k3}$ | $a_{kn}$ |

(3.3)

Then, this two-dimensional dataset can also be expressed as matrix ***A***.

$$A = \begin{array}{c} \\ i=1 \\ i=2 \\ i=3 \\ \vdots \\ i=k \end{array} \begin{array}{c} \begin{array}{ccccc} j=1 & j=2 & j=3 & \cdots & j=n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ & & \vdots & & \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kn} \end{bmatrix} \end{array} \tag{3.4}$$

Here, ***A*** is *(**k** × **n**) matrix,* $i$ is the row index, $j$ is the column index, $k$ is the total number of instances, and $n$ is the total number of features.

**Statement (2):** **Feature Set expression to a Matrix Form**

Every feature set can be illustrated as a one-dimensional array in a matrix form.

As an example, a feature set with a one-dimensional dataset is shown below.

Features:

| $f_1$ | $f_2$ | $f_3$ | . . . | $f_n$ |
|---|---|---|---|---|

(3.5)

This feature set can be expressed as matrix $F$,

$$F = [f_1 \quad f_2 \quad f_3 \ \ldots \ f_n] \tag{3.6}$$

This is a $(1 \times n)$ matrix and $n$ is the total number of features and $f_1, f_2, f_3, \ldots, f_n$ are vectors or attributes or features.

**Statement (3):** **Feature Expression as a Column Matrix**

If a dataset containing ***k*** samples and ***n*** features can be expressed as *(**k** × **n**)* matrix ***A*** then each feature can also be expressed as a column matrix of given input values containing ***k*** samples.

For example, if a dataset in a *(**k** × **n**)* matrix ***A*** may be displayed as below, where ***k*** is the total number of instances and ***n*** is the total number of features,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ & & \vdots & & \\ a_{k1} & a_{k2} & a_{k3} & \cdots & a_{kn} \end{bmatrix} \tag{3.7}$$

then each feature can be expressed in a column matrix as below:

$$f_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{k1} \end{bmatrix}, f_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ \vdots \\ a_{k2} \end{bmatrix}, \ldots, f_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ a_{3n} \\ \vdots \\ a_{kn} \end{bmatrix} \tag{3.8}$$

where $f_1$, $f_2$, ... $f_n$ are the features.

**Statement (4):** **Linear Combination System**

If a two-dimensional dataset can be expressed in a matrix *A (k* $\times$ $n$ matrix) and $c_i$ is a constant, where $1 \leq i \leq n$ , then this two-dimensional dataset can be constructed as a linear combination system.

In symbols, it can be expressed as follows:

$$X = \sum_{j}^{m} \sum_{i}^{n} c_i \, x_{ji} \tag{3.9}$$

where *X* is the linear combination name, $f_{ji}$ is a feature and $c_i$ is a constant (*1<i <n, 1<j<m*).

**Statement (5):** **Linear Homogeneous System**

If a linear combination system ***X*** of a two-dimensional array dataset is equal to ***0*** (***0*** is a zero-column matrix), it can be called a linear homogeneous system.

In symbols,

$$X = \sum_{j}^{m} \sum_{i}^{n} c_i \, x_{ji} = 0 \qquad (3.10)$$

where $f_1$, $f_2$ , $f_3$, ... , $f_n$ are features, $c_1$, $c_2$ , $c_3$, ... , $c_n$ are constants and $\boldsymbol{0}$ is a zero matrix.

**Statement (6): Determining Useful or Redundant feature**

For solving a linear combination system in which '*X= 0*', there are two types of solution constant: a group with $c_i$ that is equal to zero, and a group with $c_i$ that is not equal to zero. Then, the features can be identified as follows:

*i*) If $c_i$= 0, an associated feature is a redundant feature.

ii) If $c_i \neq 0$, an associated feature is a useful feature.

The above statements underpin the LDC feature selection method described in this dissertation. The four primary procedures involved in the design of the LDC feature selection method are set out below.

1) Set each feature $\boldsymbol{f_i}$ as a column matrix.
2) Solve a homogeneous linear combination system of the matrix (*X=0)*.
3) Check zeros or non-zeros for all constant values.
4) Determine useful features of a given dataset.

Therefore, the LDC feature selection method can select the useful features from every dataset containing ***k*** samples and ***n*** features, referred to previously as a matrix ***A (k* x *n*** matrix).

In figure 3.5, the feature $f_i$ ($i$= 1,2,3, ... , $n$) expressed as a column matrix of the given input values containing ***k*** samples and ***n*** features. A constant $c_i$ ($i$= 1,2,3, ... , $n$) is a value for building a linear combination system. The **SF** is the set of newly selected features set using LDC feature selection method.

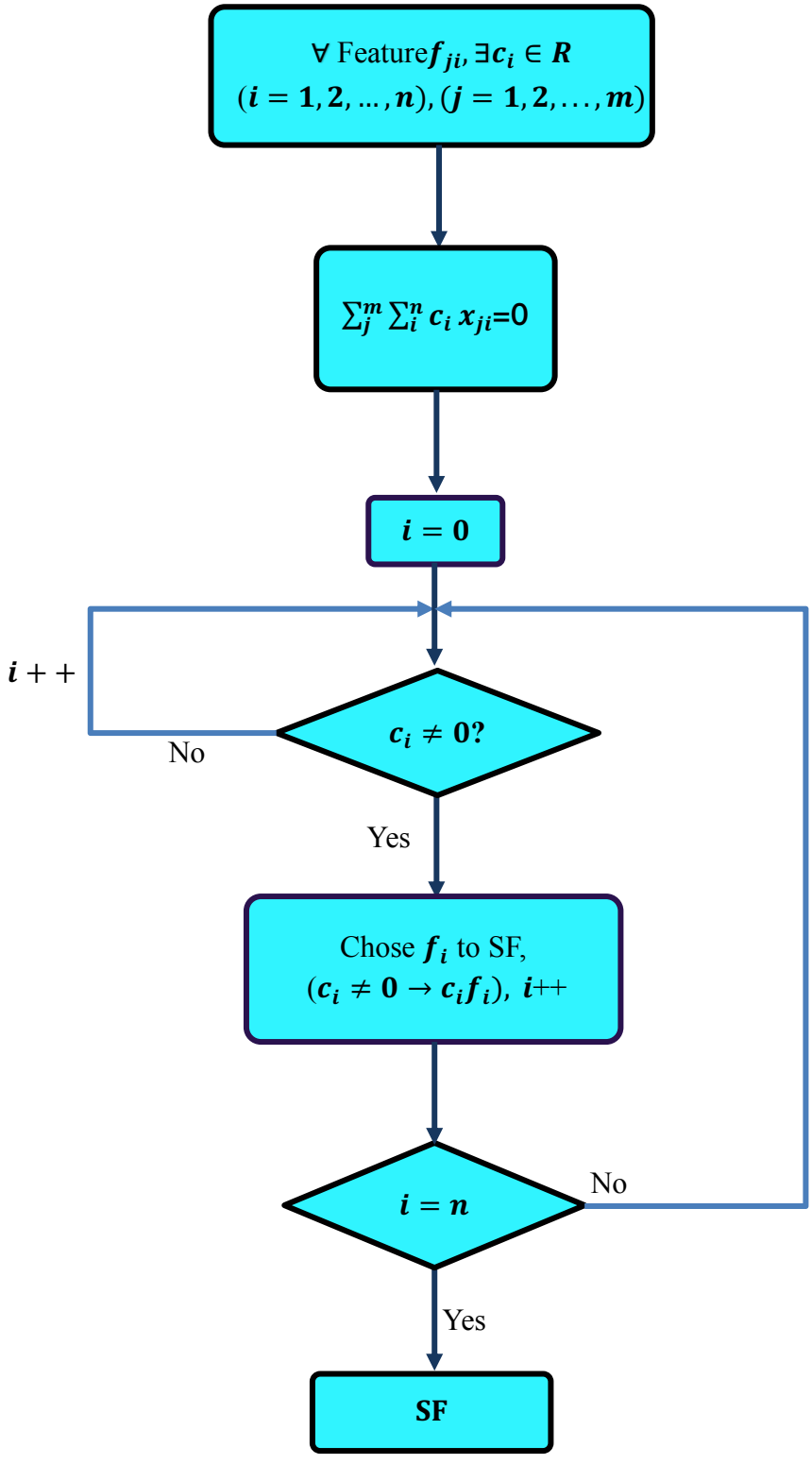



**Figure 3.5** Algorithm design for LDC feature selection method

### 3.2.3.2 Cyclic Attribution Technique (CAT) feature selection method

This technique investigates a different aspect of selected features and outputs. The LDC selection method can reduce only around half of the datasets. The CAT feature selection technique looks more carefully at the most important features [121]. Before describing the development of the CAT selection method, four new definitions are introduced that use group theory and cyclic group characteristics, which are interesting concepts and methods for feature selection.

**Definition (1): Cyclic Group of Features**

If each feature set is a group *G* containing *n* features, then this group is a cyclic group itself with a generator 1.

**Definition (2): Cyclic Subgroup of Features**

Let ***G*** be a cyclic feature group generated by 1 containing ***n*** features. There exists a quotient *Q* for every prime divisor *D* of *n*. Therefore, a cyclic group generated by quotient *Q*, denoted by <*Q*>, is a cyclic subset of feature group *G*, satisfying the properties of feature group *G* and is a newly selected feature set containing relevant or useful features of a specified dataset.

**Definition (3): Total Number of Features of Cyclic Subgroup**

Let a cyclic group <Q> be a newly selected feature subgroup containing relevant or useful features. Then, in this feature set <*Q*>, the total number of features ($\varphi$) is

$$\varphi = Q\left(1-\frac{1}{q_1}\right)\left(1-\frac{1}{q_2}\right)\dots(1-\frac{1}{q_h}) \qquad (3.11)$$

where $q_1, q_2, \dots, q_h$ are prime divisors of Q and h <*Q*

**Definition (4): Important Features**

Let D be a divisor of $n$, the total number of selected features $\varphi = m$ and <Q> be a set of new selected features. Then the best selected features of <Q> are:

$$<Q> = \{D(1,2,..\ .,m)\} \quad (3.12)$$

The CAT feature selection technique was developed using the above-proposed statements and definitions. Figure 3.6 illustrates the algorithm design for the CAT method.

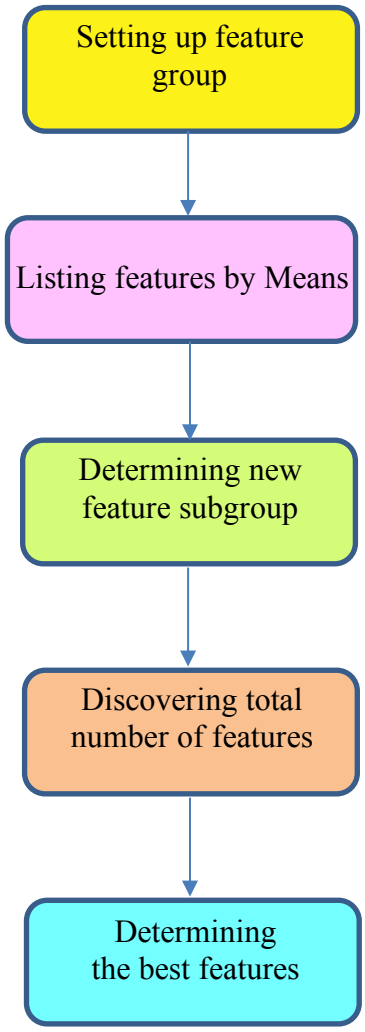



**Figure 3.6** Algorithm design for CAT feature selection method

As shown in Figure 3.6, the CAT feature selection process is created through five primary procedures. Assuming a dataset referring to a matrix $A$ ($k \times n$ matrix) containing $k$ samples and $n$ features, then the procedures can be expressed as follows.

(1) **Setting up feature group**

$$G = \{f_i / i = 1,2,\ldots,n\}$$

$$H = \{x_j / j = 1,2,\ldots,k\}$$

where $G =$ the given feature set

$H =$ the set of the time series value of each feature

$n$ = the total number of features, $k$ = the total number of given instants

(2) **Listing features by means**

$$m_i = \frac{\sum_{j=1}^{k} x_j}{k}, where\ i = 1,2,\ldots,n$$

$$M = \{m_i / m_{i-1} < m_i < m_{i+1}\}$$

where $New\_G = \{i / m_i \in M\}$

$m_i$ = the mean value of each feature, $New\_G$ = the new list of features

M = the set of all mean values in ascending order

**(3) Determining new feature subgroup**

$$D = \{d_i / \ d_i = \ prime\ divisors\ of\ n\}$$

$$Q = \{q_i / \ q_i = quotients\ related\ with\ d_i\}$$

$$< Q > = New\_F$$

where $D$ = the set of all prime divisors $d_i$

$Q$ = the set of all quotients $q_i$

<$Q$> = the new features subgroup

(4) **Discovering total number of features**

$$\varphi = \{\varphi_i / \ \varphi_i \ = \ q_i(1 - \frac{1}{p_i}),$$

$$p_i = prime\ divisors\ of\ q_i, 1 < i < q_i$$

where $\varphi = total\ elements\ of\ New\_F$

$\varphi =$ the set of the total number of features of the new feature subgroup

$p_i =$ the prime divisor of the quotient $q$

(5) **Determining important features**

$G_i = \{d_i(1,2,..\ .,\varphi_i)\}$,

CAT_F= $\cup\ (G_i)$

where

$G_i$ = the new selected feature subgroup

CAT_F = the set of features newly selected by the CAT method

The description of this algorithm concludes the design of the CAT feature selection method. Designing a selection method that produces newly selected features was the first major aim of this dissertation. Designing the ANN classifier model-based feature selection method to achieve correct predicted values is the second major aim.

### 3.2.4 Designing ANN Classifier [ii]

ANN is designed to classify HAD system data based on LDC and CAT. For a more accurate and robust identification of features in a HAD system, MLP is expected to produce a better classifier [122]. In this work, the design of the MLP model focused on four categories: activation function, determination of hidden neuron sizes, decision of the learning rate, and determination of the number of hidden layers.

1) **Activation Function**

The proposed MLP uses a Feed Forward Back Propagation algorithm. The logistic (sigmoid) function is used as an activation function for each hidden layer. It is shown in following equation.

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

2) **Determination of the Hidden Neuron Sizes**

This strategy is the key point in this research. It can identify the number of hidden neurons for classifier design. Indeed, hidden neuron nodes and hidden layers are significant components that control the complexity of an artificial neural network model. Although ANN has no technique and no stable condition to determine numbers of hidden neurons, the model's property is convergence to a balanced state [123]. As a

consequence, it was decided to approach the formula of hidden neuron nodes of the MLP model as outlined below:

**Formula of Number of Hidden Neuron Nodes**

If $n$ is the finite total number of input neurons to be classified in the ANN model, the total number of hidden neurons for each hidden layer can be determined as follows:

$$\text{Total number of hidden neurons} = \frac{2n^2+1}{3n+1} \quad (3.13)$$

where $n$ is the total number of input neurons.

Example: **To find the total number of hidden neuron nodes for input neurons $n$ = 561 (the total number of input neurons in the UCI-HAR dataset):**

Given the $t$otal hidden neuron nodes $= \frac{2n^2+1}{3n+1}$

when $n$ = 561, rounding to the nearest integer,

total hidden neuron nodes $= \frac{2(561)^2+1}{3(561)+1}$ = 373.79 = 374

Therefore, when 561 neurons are input to the ANN model, 374 hidden neurons are possible to use for each hidden layer.

**(3) Decision of the learning rate**

In a backpropagation neural network, the learning rate is important because it can reduce training time and learning error. If the learning rate is small, the network is approaching the best solution. Consequently, an experiment to determine a better learning rate with the MLP classifier was tested on the UCI-HAR dataset with a learning rate set between 0.0001 and 0.001. The neural network was trained three times, with one hidden layer, with two hidden layers, and with three hidden layers and the Means Square Errors (MSE) were compared, as illustrated in Figure 3.7.

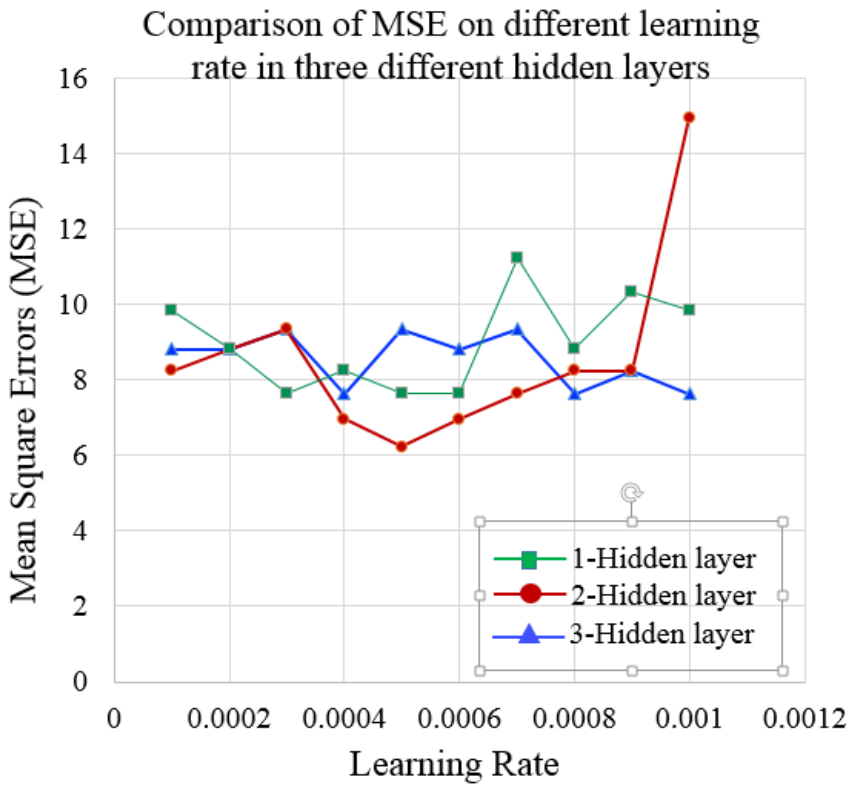



**Figure 3.7** Comparison of learning rates and MSE of UCI-HAR dataset

According to the results shown in Figure 3.5, using the one hidden layer network, the learning rate 0.0005 gave a small error and using the two hidden layers network, the learning rate of 0.0005 provided the lowest error recorded. When using the three hidden layers network, the learning rate 0.0008 produced a small error. With these results in mind, the learning rate of 0.0005 was considered the best to apply in this network.

4) **Determination of the hidden layers**

MLP models with 1 hidden layer and 2 hidden layers classified three datasets (UCI-HAR dataset, DATASET-UCI, HAPT_1) using 25%, 50%, and a proposed number of neurons notes. Accuracies of the models are presented in Table 3.11.

**Table 3.11** Accuracy of MLP models with 1 hidden layer and 2 hidden layers used with 25%, 50%, and proposed numbers of neurons

| Dataset | MLP model with 1 hidden layer | | | MLP model with 2 hidden layers | | |
|---|---|---|---|---|---|---|
| | Number of Neurons | | | Number of Neurons | | |
| | 25% | 50% | Proposed | 25% | 50% | Proposed |
| UCI-HAR | 98.32 | 98.38 | 98.67 | 98.48 | 98.54 | 98.61 |
| DATASET-UCI. | 90.95 | 91.07 | 91.01 | 90.55 | 90.43 | 91.3 |
| HAPT_1 | 98.24 | 98.5 | 98.56 | 97.82 | 98.5 | 98.56 |

As shown in Table 3.11, the MLP model with 1 hidden layer achieved the best accuracy with the UCI-HAR dataset using the proposed number of neurons. The best accuracy with the DATASET-UCI dataset was achieved using 50% of the neurons, and the best accuracy with the HAPT_1 was achieved with the proposed number of neurons. The MLP model with 2 hidden layers, achieved the best accuracy with every dataset using the proposed number of neurons. Therefore, the best MLP model was considered to be the two hidden layer network using a proposed number of neuron nodes.

### 3.2.5 MLP model

To develop a HAD system based on the proposed feature selection method, a good classifier model is required. In section 3.2.4, the specific requirements were investigated for constructing an exact MLP model. These specific requirements are described in Table 3.12.

**Table 3.12** Specific requirements to build an MLP model

| No. | Categories | Specific Requirements |
|---|---|---|
| 1. | Activation function | Logistic (sigmoid) |
| 2. | Splitting the dataset | 70% for training and 30% for testing |
| 3. | Hidden neuron sizes | $\frac{2n^2+1}{3n+1}$ , ( $n$ = total number of input) |
| 4. | Learning rate | 0.0005 |
| 5. | Number of hidden layers | 2 (using the same hidden neuron sizes) |

Finally, the successful design of the new effective MLP classifier model was as shown in Figure 3.8.

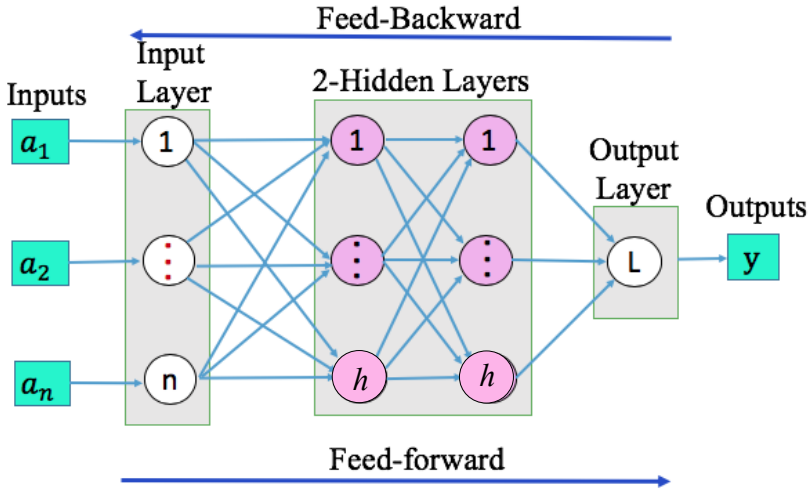



**Figure 3.8** A new MLP model design

In Figure 3.8, the $a_1, a_2, \ldots, a_n$ are the input patterns and **y** is the output pattern. In this case, $n$ is the total number of input neurons and $h$ is the total number of hidden neuron nodes in each hidden layer.

### 3.2.6 Performance Evaluation

To meet the main goal of this study, the MLP performed classifications on four datasets, namely UCI-HAR, DATASET-UCI, HAPT 1 and HAPT-2. Performance was evaluated with a confusion matrix. The outcomes of each classification were represented as a matrix with real positive (TP), real negative (TN), false positive (FP), or false negative (FN) information. These values define how well the classifier determines measurements of real and predicted activities. The parts of the confusion matrix are shown in Table 3.13 [124].

**Table 3.13** Confusion matrix

| | | **Predicted Value** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| Actual Value | Positive | TP | FN |
| | Negative | FP | TN |

where

| | |
|---|---|
| TP (True Positive) | - The number of positive records correctly predicted as positive by the model |
| TN (True Negative) | - The number of negative records correctly predicted as negative by the model |
| FP (False Positive) | - The number of negative records incorrectly predicted as positive by the model |
| FN (False Negative) | - The number of positive records incorrectly predicted as negative by the model |

Accuracy was determined by the total number of correctly predicted results from the classifier that included both positive and negative predictions [125]. The performance result was evaluated with the equation

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.14}$$

The activity classes for UCI-HAR dataset, DATASET -UCI dataset, and HAPT_1 dataset were described in Table 3.14 and activity classes for HAPT_2 were shown in Table 3.15.

**Table 3.14** Activity classes of UCI-HAR, DATASET -UCI, and HAPT_1

| No. | Activity | Notation |
|---|---|---|
| 1. | Walking | WK |
| 2. | Walking-Upstairs | WKU |
| 3. | Walking-Downstairs | WKD |
| 4. | Sitting | SIT |
| 5. | Standing | STD |
| 6. | Laying | LY |

**Table 3.15** Activity Classes for HAPT_2

| No. | Activity | Notation |
|---|---|---|
| 1. | Walking | WK |
| 2. | Walking-Upstairs | WKU |
| 3. | Walking-Downstairs | WKD |
| 4. | Sitting | SIT |
| 5. | Standing | STD |
| 6. | Laying | LY |
| 7. | Stand-to-Sit | STD-SIT |
| 8. | Sit-to-Stand | SIT-STD |
| 9. | Sit-to-Lie | SIT-LY |
| 10. | Lie-to-Sit | LY-SIT |
| 11. | Stand-to-Lie | STD-LY |
| 12. | Lie-to-Stand | LY-STD |

Concepts, principles, theories, techniques, and designs central to this research were presented in this section. The initial objective of data collection and step counting was well realized and understood. A further objective was the development of a new feature selection method using LDC to provide a better classifier for the HAD system. Although LDC could properly reduce the irrelevant features, it faced some challenges to specify more important features. Therefore, the CAT feature selection method was included in this dissertation to identify the key features and to reduce more irrelevant features. Furthermore, a new classifier using MLP was designed to determine the quality of new feature selection methods and support the HAD system. Three UCI datasets (UCI-HAR, DATASET-UCI, UCI-HAPT) were integrated into the research to provide the HAD system with suitable datasets for the development of feature selection and classification. Therefore, the next section presents the practical implementation of the proposed strategy with the obtained results.

# CHAPTER 4

# RESULTS AND DISCUSSIONS

Having achieved the first objective of this work, accurate step counting, the second objective comes into focus: the design of a specific classifier for HAD feature selection using the LDC and CAT techniques. The main task of the specific classifier is to identify the outcomes of each feature selection method.

Five sections are presented in this chapter. Section 4.1 presents the results of step counting from actual walking. Section 4.2 explains a measure to determine the performance of the MLP classifier. Section 4.3 presents results and performance evaluation of feature selection from datasets using the LDC feature selection method. Section 4.4 focuses on the results of feature selection and performance using the CAT method. In section 4.5, analysis and discussion of the LDC and the CAT feature selection methods is presented.

## 4.1 Step counting

The step counting technique using the accelerometer sensor in the Samsung Galaxy Note 4 to detect the walking steps of the user was outlined in section 3.2. The accelerometer sensor measured acceleration along the X-axis, Y-axis, and Z-axis. The number of steps in each activity was saved and video capture was recorded for experimental elaboration. The number of steps collected by the algorithm was compared to actual walking steps. This experiment involved three types of walking (normal walking, slow walking, and fast walking) in four smartphone usage modes (in-hand swinging, in-hand texting, in a waist-bag, and in an arm-bag). Significant results for the average successful steps percentage of three walking styles for each mode are shown in Figure 4.1. The results obtained from normal walking were more accurate than those obtained from other walking styles.

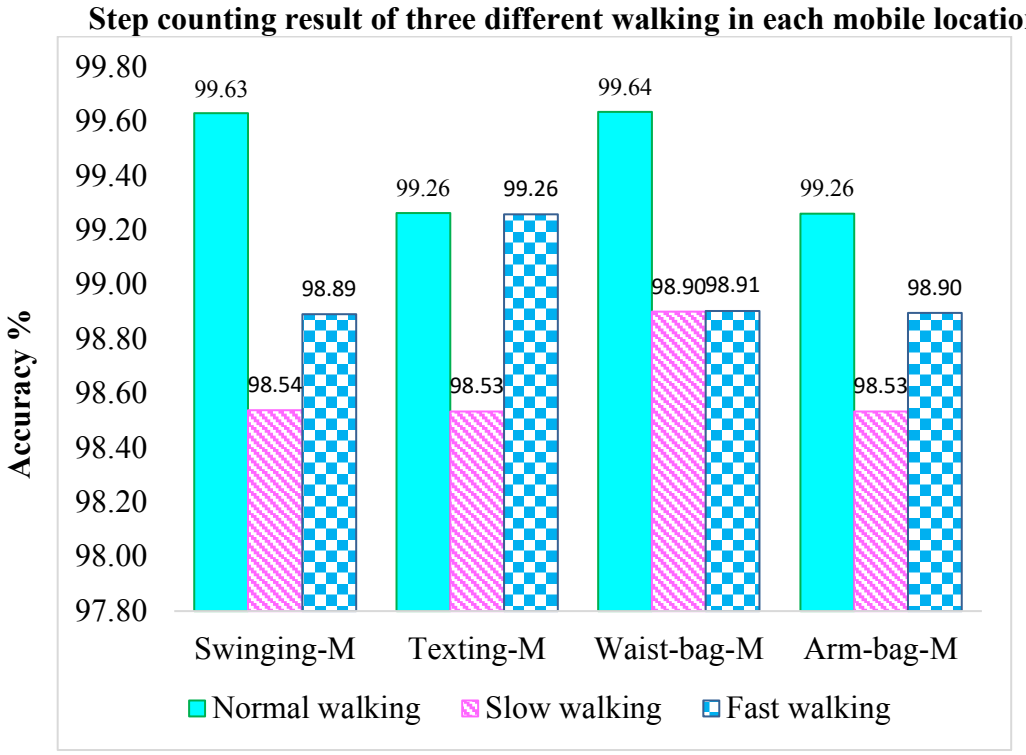



**Figure 4.1** Comparison of step counting results for 3 walking styles in each mode

* Swinging-M: Hand-in Swinging mode, Texting-M: Texting mode, Waist-bag-M: Waist-bag mode, and Arm-bag-M: Arm-bag mode.

Figure 4.1 clearly showed that the result of normal-walking is better than other walking styles. That means the results of routine walking were stable and accurate in every usage mode. Due to the complicated movement involved in the slow walking activity, the accuracy achieved in every usage mode was less than the accuracy achieved in both normal and fast walking activities. During normal and fast walking activities, the smartphone was stable and the steps were correctly picked up in the texting mode and waist-bag mode. Normal walking gave the best outcome in each usage mode. In each mode, the outcomes of fast-walking were the second best. Slow-walking gave the worst result in every mode. Table 4.1 compares the results of the four mobile phone modes between algorithm counting and actual walking steps.

**Table 4.1** Comparison results between algorithm counting and actual walking steps in four mobile locations

| User | Hand-in swinging mode | | | | | |
|---|---|---|---|---|---|---|
| | N-walk | | S-walk | | F-walk | |
| | R | P | R | P | R | P |
| 1 | 30 | 30 | 31 | 30 | 30 | 30 |
| 2 | 30 | 30 | 31 | 31 | 30 | 29 |
| 3 | 31 | 31 | 31 | 31 | 30 | 30 |
| 4 | 30 | 30 | 30 | 30 | 30 | 30 |
| 5 | 30 | 30 | 31 | 30 | 30 | 30 |
| 6 | 30 | 30 | 30 | 30 | 30 | 30 |
| 7 | 30 | 29 | 30 | 29 | 31 | 30 |
| 8 | 30 | 30 | 30 | 29 | 29 | 29 |
| 9 | 30 | 30 | 30 | 30 | 31 | 30 |

| User | Text mode | | | | | |
|---|---|---|---|---|---|---|
| | N-walk | | S-walk | | F-walk | |
| | R | P | R | P | R | P |
| 1 | 30 | 30 | 31 | 30 | 30 | 30 |
| 2 | 31 | 31 | 31 | 30 | 30 | 29 |
| 3 | 30 | 30 | 30 | 30 | 29 | 29 |
| 4 | 30 | 29 | 31 | 31 | 30 | 30 |
| 5 | 31 | 31 | 31 | 30 | 31 | 31 |
| 6 | 30 | 30 | 30 | 30 | 30 | 30 |
| 7 | 30 | 29 | 29 | 29 | 30 | 29 |
| 8 | 30 | 30 | 30 | 30 | 30 | 30 |
| 9 | 30 | 30 | 30 | 29 | 30 | 30 |

| User | Waist-bag mode | | | | | |
|---|---|---|---|---|---|---|
| | N-walk | | S-walk | | F-walk | |
| | R | P | R | P | R | P |
| 1 | 30 | 30 | 30 | 30 | 31 | 29 |
| 2 | 31 | 31 | 31 | 30 | 30 | 30 |
| 3 | 31 | 30 | 31 | 31 | 30 | 30 |
| 4 | 31 | 31 | 29 | 29 | 30 | 30 |
| 5 | 31 | 31 | 30 | 31 | 31 | 30 |
| 6 | 30 | 30 | 30 | 29 | 30 | 30 |
| 7 | 31 | 31 | 30 | 30 | 30 | 30 |
| 8 | 31 | 31 | 31 | 31 | 31 | 31 |
| 9 | 30 | 30 | 30 | 29 | 31 | 31 |

| User | Arm-bag mode | | | | | |
|---|---|---|---|---|---|---|
| | N-walk | | S-walk | | F-walk | |
| | R | P | R | P | R | P |
| 1 | 30 | 29 | 30 | 29 | 30 | 30 |
| 2 | 31 | 31 | 31 | 31 | 30 | 30 |
| 3 | 30 | 30 | 30 | 30 | 30 | 29 |
| 4 | 30 | 30 | 30 | 29 | 30 | 29 |
| 5 | 31 | 31 | 31 | 31 | 31 | 30 |
| 6 | 30 | 30 | 30 | 29 | 30 | 30 |
| 7 | 30 | 30 | 29 | 31 | 30 | 30 |
| 8 | 30 | 30 | 30 | 30 | 31 | 31 |
| 9 | 29 | 29 | 29 | 29 | 30 | 30 |

* N-walk: normal walking, S-walk: slow walking, and F-walk: fast walking,

* R: Actual walking steps and P: Predicted walking steps

Table 4.1 demonstrates that the step counting algorithm can exactly detect the steps of a user walking for three walking styles in four phone position modes. However, all user walking can be into activeness level because everyone has their own walking pace relying on weight and height. Every step counting could not exactly get an accurate target. It is a challenge in the counting of steps because the step counting depends mainly on the walking styles of the user. The overall results for the stability of four usage modes (in-hand swing mode, texting mode, waist-bag mode, arm-bag mode) on three different walking styles are shown in Figure 4.2.

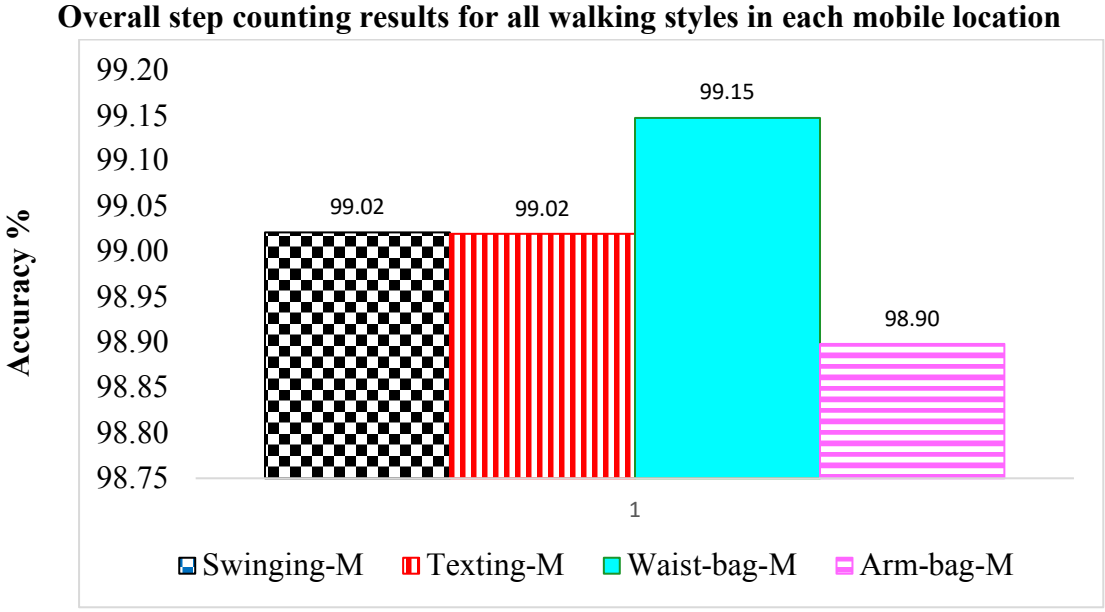



**Figure 4.2** Overall results of step counting in four different modes

As a result of Figure 4.2, the step counting technique performed well in the waist-bag mode for all walking styles due to the stability of the sensor. This meant that the waist-bag mode was the best mode for counting steps to retrieve sensor data. For all walking styles, the step counting algorithm had the same quality in swinging mode and texting mode. These three modes were the best modes to apply the step counting algorithm. The step counting results show that the arm-bag mode was unstable in each walking style. The step counting experiment indicated how the sensor operates to provide data in determining human activity. This study contributed knowledge of the sensor that helped in designing effective feature selection methods.

## 4.2 MLP Model

The efficiency of classifiers relies on a number of factors including the type of classifier, the input features and the required output. An MLP classifier was applied to classify the data from human activity recognition. The MLP model was designed with a logistic sigmoid activation function, a learning rate of 0.0005, 2 hidden layers, and 375 hidden neuron nodes. MLP, UCI-HAR, DATASET-UCI, and HAPT 1 datasets were used to evaluate the performance of the MLP classifier. The performance results were presented in comparison with results of five other classifiers (Support Vector Machine (SVM), Bagging (BAG), K-Nearest Neighbors (KNN), Classification and Regression Tree (CART), and Bayesian (BAYES). The comparative accuracy rates and running times are shown in Table 4.2.

**Table 4.2** Accuracy rates and running times of six classification methods

| | UCI-HAR | | DATASET-UCI. | | *HAPT_1* | |
|---|---|---|---|---|---|---|
| Classifier | Accuracy (%) | Running Time (min:sec) | Accuracy (%) | Running Time (min:sec) | Accuracy (%) | Running Time (min:sec) |
| MLP | **98.61** | **0:48.27** | **91.30** | **0:44.62** | **98.56** | **0:54.25** |
| SVM | 97.41 | 0:19.45 | 88.69 | 0:12.52 | 97.50 | 0:20.17 |
| BAG | 96.44 | 4:25.73 | 91.35 | 3:35.72 | 97.06 | 6:56.25 |
| KNN | 96.25 | 0:27.36 | 86.83 | 0:12.91 | 95.99 | 0:28.22 |
| CART | 92.49 | 0:7.03 | 84.22 | 0:5.35 | 93.31 | 0:10.47 |
| BAYES | 79.42 | 0:4.19 | 53.6 | 0:1.88 | 75.77 | 0:4.43 |

As shown in Table 4.2, among all the classifiers, the best accuracy was achieved by MLP with UCI-HAR and HAPT_1 dataset: 98.61% and 98.56%, respectively. In DATASET-UCI, the proposed MLP classifier was 0.05% less accurate than the BAG classifier, although the BAG classifier took 3 minutes and 35.72 seconds to evaluating the data: 2 minutes and 51 seconds longer than the proposed MLP model. The MLP classifier was clearly faster than the BAG classifier. Although the BAYES classifier was the fastest of all the classifiers, it was the least accurate. The running time of the MLP classifier was longer than SVM, KNN, CART, and BAYES classifiers due

to the use of feed-forward neural networks to achieve optimized accuracy. However, the MLP classifier performed better than other classifiers. The MLP classifier was clearly one of the most robust classifiers for all datasets.

The MLP model was designed with a logistic sigmoid activation function, a learning rate of 0.0005, 2 hidden layers and a proposed number of hidden neuron nodes. This approach was chosen to investigate the most suitable number of hidden neuron nodes for the MLP model. To evaluate accuracy rates and running times, the UCI-HAR, DATASET-UCI, and HAPT 1 datasets were generated to the MLP model with different percentages of hidden neuron nodes. The four main pillars of the MLP architecture were datasets with 25%, 50%, 100%, and a proposed number of hidden neuron nodes, as shown in Table 4.3.

**Table 4.3** Accuracy results of MLP model with different number of neurons

| Dataset | Feature (%) | No. Hidden Neuron Nodes | Accuracy (%) | Running Time (Sec) |
|---|---|---|---|---|
| UCI-HAR | 25% | 140 | 98.48 | 17 |
| | 50% | 280 | 98.54 | 35 |
| | **Proposed** | **376** | **98.67** | **43** |
| | 100% | 561 | 98.51 | 60 |
| DATASET-UCI | 25% | 140 | 90.55 | 18 |
| | 50% | 280 | 90.43 | 30 |
| | **Proposed** | **376** | **91.30** | **36** |
| | 100% | 561 | 90.02 | 60 |
| HAPT_1 | 25% | 140 | 97.82 | 19 |
| | 50% | 280 | 98.50 | 33 |
| | **Proposed** | **376** | **98.56** | **41** |
| | 100% | 561 | 98.27 | 60 |

Using the proposed number of hidden neuron nodes clearly improved the performance of the MLP model with each dataset. It appears that the performance of MLP has a positive dependence on the number of hidden neuron nodes. Moreover, using the proposed number of hidden neuron nodes proved the MLP model to be one of the most robust classifiers not only for the issue of recognition of human activity but also any issues of machine learning. As a side note, the calculations took more running time when more neuron nodes were added due to the increased difficulty of learning the task for the classifier. The suitability of the MLP classifier was compared to the other five classifiers and a number of hidden neuron nodes as compared to 25%, 50%, and 100% of hidden neuron nodes. This approach used hidden layers to check the architecture and maintain the performance of the proposed MLP classifier.

The two-hidden layers MLP model was described in section 3.2.4. This approach reduced the two hidden layers of the proposed MLP model to one and compared the accuracy results and running times. After passing the three datasets, the accuracy rates and running times were presented in Table 4.4.

**Table 4.4** Accuracy results of MLP with different numbers of neurons

| Dataset | 1 Hidden Layer | | 2 Hidden Layers | |
|---|---|---|---|---|
| | Accuracy (%) | Running Time (Sec) | Accuracy (%) | Running Time (Sec) |
| UCI-HAR | 98.61 | 48 | **98.67** | **43** |
| DATASET-UCI | 91.01 | 44 | **91.30** | **36** |
| HAPT_1 | 98.56 | 47 | **98.56** | **41** |

The results clearly show that the proposed MLP model using two hidden layers achieved higher overall accuracy from each dataset with less running time than the MLP model using one hidden layer. There are many reasons why the runtime of the 2-hidden layer model was less than the runtime of the 1-hidden layer architecture. To train neural networks, there are now very large and complex HAD datasets. ANNs are the very core of Deep Learning. There are so many kinds of ANNs. When an ANN has two or more hidden layers, it is called a deep neural network (DNN). In this study, the

ANN with a 1-hidden layer architecture normally learned from inputs to outputs with a single layer mapping that could be used to provide input and output sequences of different lengths by taking more time to run. However, ANN architecture with two hidden layers is very flexible and can generally be used as a DNN to learn from inputs to outputs. Therefore, it is possible to reduce the running time by feeding the data into an MLP through 2-hidden layers networks. It clearly indicates that the proposed architecture of two hidden layers is a better MLP model than an architecture of one hidden layer with supposedly optimal hidden neuron nodes.

### 4.3 LDC Feature Selection Method

Feature selection reduces the number of features in the input. Reducing the number of features not only reduces the dimensionality of the dataset but also improves classification efficiency.

#### 4.3.1 Results of Feature Selection by LDC

When the LDC feature selection method is run on a dataset, it executes the values of constants correlated with useful features from the original dataset [124]. A brief example of the executed result of constant values by the LDC to UCI-HAR dataset is shown in Table 4.5.

**Table 4.5** Constant values related with features of LDC

| No. | Feature Name | Constant-values |
|---|---|---|
| 0 | tBodyAcc-mean ()-X | 0 |
| 1 | tBodyAcc-mean ()-Y | -0.984916859 |
| 2 | tBodyAcc-mean ()-Z | -0.76404401 |
| 3 | tBodyAcc-std ()-X | 2.852157208 |
| 4 | tBodyAcc-std ()-Y | 0 |
| 5 | tBodyAcc-std ()-Z | 0 |
| 6 | tBodyAcc-mad ()-X | 0 |
| 7 | tBodyAcc-mad ()-Y | 0 |
| 8 | tBodyAcc-mad ()-Z | 0 |

| No. | Feature Name | Constant-values |
|---|---|---|
| 9 | tBodyAcc-max ()-X | 0.160728141 |
| 10 | tBodyAcc-max ()-Y | 0.146233619 |
| 11 | tBodyAcc-max ()-Z | 0.254649998 |
| ⋮ | ⋮ | ⋮ |

In Table 4.5, there are two types of constant values correlated to given features. Some are equal to zero and some are not. To reduce the original dataset, some features with zero constant values are assumed to be redundant or irrelevant. The remaining features are useful features that produce accurate data on human activity. After implementing the LDC, the selected and reduced features of the total 561 features of each dataset are illustrated in Table 4.6.

**Table 4.6** Selected and removed features from each dataset by the LDC

| Dataset | Original features | Removed features | Selected Features |
|---|---|---|---|
| UCI-HAR | 561 | 316 | 245 |
| DATASET-UCI | 561 | 292 | 269 |
| HAPT_1 | 561 | 317 | 244 |
| HAPT_2 | 561 | 269 | 292 |

The high dimensionality of the data also makes data visualization complex and difficult during data processing. In this experiment, the LDC method not only determined the essential features but also reduced data dimensionality. Table 4.7 shows the reduced dimensionality of data implemented by the LDC method for each dataset.

**Table 4.7** Dimensionality reductions on each dataset by the LDC

| Dataset | Original data | LDC data | Reduced data |
|---|---|---|---|
| UCI-HAR | 5,777,739 | 2,523,255 | 3,254,484 |
| DATASET-UCI | 3,222,384 | 1,545,136 | 1,677,248 |
| HAPT_1 | 5,840,571 | 2,540,284 | 3,300,287 |
| HAPT_2 | 6,131,169 | 3,191,268 | 2,939,901 |

As shown in Table 4.6 and Table 4.7, LDC not only identified the important features of the original dataset but can also reduce data dimensionality. In doing so, it fulfilled research questions 1 and 2. In the next section, to complete research question 3, feature selection with the LDC method is evaluated with the ANN classifier. Performance evaluation identified the effectiveness of the LDC feature selection method applied to the whole dataset and to three groups (t, t', f) derived from the dataset. Accuracy rates were compared with the whole dataset.

### 4.3.2 Performance Evaluations to the Whole Dataset Based on LDC

The performances of MLP are measured with and without feature selection with the LDC method. Four datasets (UCI-HAR, DATASET-UCI, HAPT-1, HAPT-2) and five classifiers (Support Vector Machine (SVM), Bagging (BAG), K-Nearest Neighbors (KNN) were applied for all experiments.

#### (1) Performance Results to UCI-HAR Dataset with LDC

Results of the evaluation with (245 features) and without LDC (561 features) of the UCI-HAR dataset are presented with confusion matrix.

**Table 4.8** Confusion matrix of MLP with and without LDC to UCI-HAR dataset

| **UCI-HAR dataset** | | | **Predicted Active** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | WK | WKU | WKD | SIT | STD | LY |
| With LDC / (245) features | **Actual activity** | WK | 548 | 19 | 0 | 0 | 0 | 0 |
| | | WKU | 13 | 515 | 1 | 0 | 0 | 0 |
| | | WKD | 0 | 0 | 609 | 0 | 1 | 1 |
| | | SIT | 0 | 0 | 0 | 495 | 1 | 0 |
| | | STD | 0 | 0 | 0 | 1 | 430 | 2 |
| | | LY | 0 | 0 | 0 | 0 | 0 | 454 |
| **UCI-HAR dataset** | | | **Predicted Active** | | | | | |
| | | | WK | WKU | WKD | SIT | STD | LY |
| without LDC/ (561) features | **Actual Activity** | WK | 544 | 23 | 0 | 0 | 0 | 0 |
| | | WKU | 16 | 513 | 0 | 0 | 0 | 0 |
| | | WKD | 0 | 1 | 609 | 0 | 0 | 1 |
| | | SIT | 0 | 0 | 0 | 495 | 1 | 0 |
| | | STD | 0 | 0 | 0 | 1 | 431 | 1 |
| | | LY | 0 | 0 | 0 | 0 | 0 | 454 |

*WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying

With the UCI-HAR dataset, MLP achieved an overall accuracy of 98.74% with LDC feature selection and 98.54% without. Although classification of the walking and walking upstairs activities was wrong, the predictions of the remaining activities were good both with and without LDC. This implies that the specific MLP classifier performed very well on the data with and without LDC and the accuracy of the MLP model was 0.2% better with LDC than without.

To determine the reliability of the LDC feature selection method with the MLP classifier, the UCI-HAR output dataset was classified by five different classification algorithms. Figure 4.3 shows all classification results of six classifiers based on features selected by the LDC method.

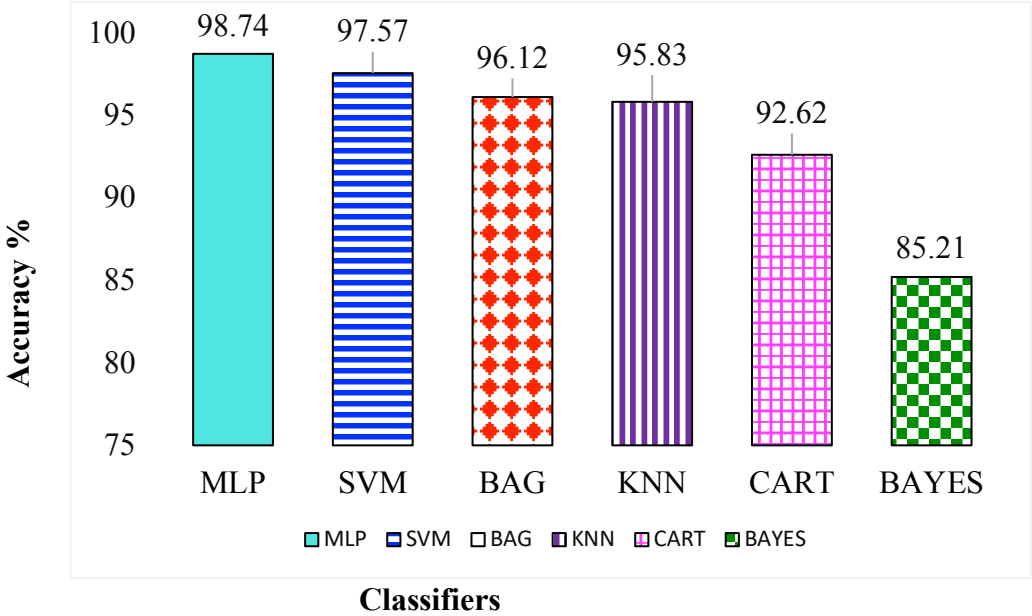



**Figure 4.3** Performances of 6 classifiers to UCI-HAR dataset after feature selection with LDC

The results of Figure 4.3 show that the MLP classifier performed well in comparison to the other classification algorithms. MLP achieved the best accuracy of 98.74%. The next best classifier was SVM with an accuracy rate of 97.57% while BAYES classification achieved the lowest accuracy of 85.21%. According to the results, the LDC feature selection method successfully identified the beneficial features

of the UCI-HAR dataset and offered the highest performance with the specific classifier for HAD.

**(2) Result Accuracies of MLP to DATASET-UCI by LDC**

The MLP classifier was applied to the DATASET-UCI dataset with (269 features) and without (561 features) LDC feature selection. Table 4.9 presents the performance values of the DATASET-UCI dataset with and without LDC feature selection.

**Table 4.9** Confusion matrix of MLP with and without LDC to the DATASET-UCI

| **DATASET-UCI** | | | **Predicted Activity** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | WK | WKU | WKD | SIT | STD | LY |
| With LDC / (269) features | **Actual Activity** | WK | 273 | 1 | 9 | 0 | 2 | 0 |
| | | WKU | 8 | 212 | 14 | 1 | 1 | 7 |
| | | WKD | 15 | 13 | 236 | 0 | 1 | 0 |
| | | SIT | 0 | 1 | 2 | 310 | 53 | 4 |
| | | STD | 0 | 0 | 0 | 29 | 290 | 0 |
| | | LY | 0 | 1 | 0 | 0 | 0 | 241 |
| **DATASET-UCI** | | | **Predicted Activity** | | | | | |
| | | | WK | WKU | WKD | SIT | STD | LY |
| without LDC/ (561) features | **Actual Activity** | WK | 262 | 5 | 17 | 0 | 1 | 0 |
| | | WKU | 19 | 203 | 13 | 2 | 2 | 4 |
| | | WKD | 17 | 21 | 227 | 0 | 0 | 0 |
| | | SIT | 0 | 6 | 0 | 299 | 64 | 1 |
| | | STD | 0 | 0 | 0 | 43 | 276 | 0 |
| | | LY | 0 | 0 | 0 | 0 | 0 | 242 |

*WK: walking, WKU: walking-upstairs, WKD: walking-downstairs,
SIT: sitting, STD: standing, LY: laying

The MLP model applied to the DATASET-UCI after LDC feature selection (269 features) achieved an overall accuracy of 90.66%. The overall accuracy of the MLP model on the original DATASET-UCI dataset (561 features) without LDC was 87.53%. While the dataset without the LDC shows some confusion between walking upstairs and walking downstairs, and between sitting and standing, this confusion decreased after feature selection with LDC. Therefore, the overall accuracy of the dataset after LDC selection was 3.13% more than the overall accuracy of the dataset without LDC feature selection.

The DATASET-UCI dataset after feature selection was then classified by five different classifiers to provide a comparison with MLP classifier. The results are presented in Figure 4.4.

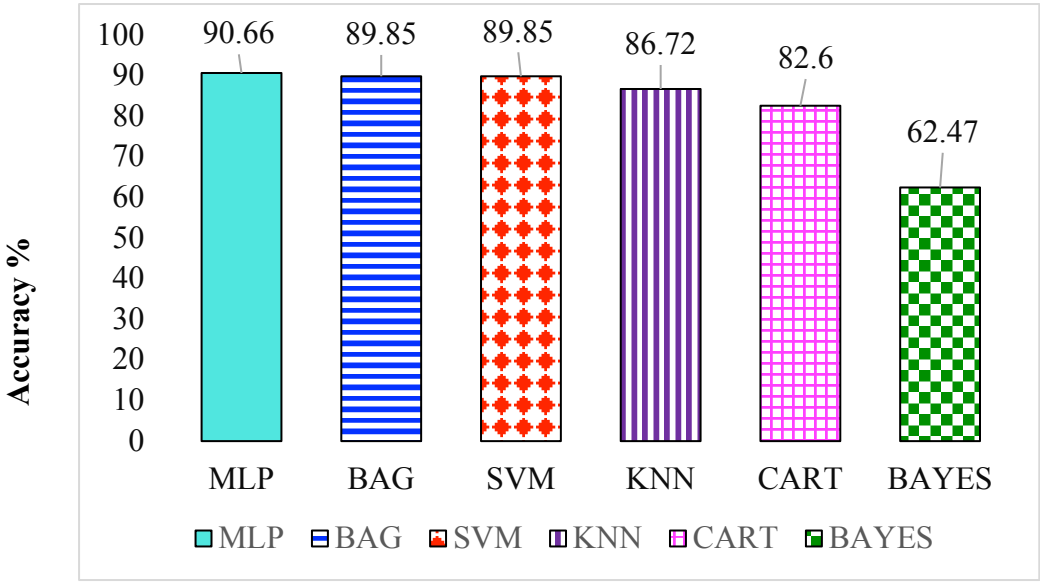



**Figure 4.4** Performance results of six classifiers on DATASET-UCI dataset

Although the scheme tests human activity datasets, the core of the scheme is independent of the features of other activity datasets; therefore, it is applicable to any dataset. In Figure 4.4, the MLP achieved the best accuracy of 90.66%, while BAG and SVM were second-best with the same accuracy rate of 89.85%. BAYES was the least accurate classifier with 62.47% accuracy. MLP was 0.81% more accurate than the BAG and SVM classifiers. MLP performed better than the other classification algorithms.

### (3) Discussion on Result Accuracy of HAPT_1 Dataset by LDC

The MLP classification results on the HAPT_1 dataset with (244 features) and without feature selection with LDC (561 features) are presented in Table 4.10.

**Table 4.10** Confusion matrix of MLP on the HAPT_1 dataset with and without LDC

| **HAPT_1** | | | **Predicted Activity** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | WK | WKU | WKD | SIT | STD | LY |
| **With LDC / (244) features** | **Actual Activity** | WK | 483 | 0 | 0 | 0 | 0 | 0 |
| | | WKU | 3 | 478 | 3 | 0 | 0 | 0 |
| | | WKD | 0 | 0 | 385 | 0 | 0 | 0 |
| | | SIT | 0 | 1 | 0 | 534 | 24 | 0 |
| | | STD | 1 | 0 | 0 | 6 | 600 | 0 |
| | | LY | 0 | 0 | 0 | 0 | 0 | 606 |
| **HAPT_1** | | | **Predicted Activity** | | | | | |
| | | | WK | WKU | WKD | SIT | STD | LY |
| **With LDC / (561) features** | **Actual Activity** | WK | 481 | 1 | 1 | 0 | 0 | 0 |
| | | WKU | 2 | 480 | 2 | 0 | 0 | 0 |
| | | WKD | 0 | 1 | 384 | 0 | 0 | 0 |
| | | SIT | 0 | 1 | 0 | 537 | 21 | 0 |
| | | STD | 0 | 0 | 0 | 12 | 595 | 0 |
| | | LY | 0 | 0 | 0 | 1 | 0 | 605 |

*WK: walking, WKU: walking-upstairs, WKD: walking-downstairs,
SIT: sitting, STD: standing, LY: laying

In Table 4.10, the overall accuracy of MLP was 98.75% after feature selection with LDC (244 features), and 98.66% before (561 features). Despite the fact that both datasets had a false classification between sitting and standing activity, the overall accuracy of the dataset was 0.09% better after LDC feature selection.

To determine the reliability of the results of the MLP classifier, five different classifiers were implemented on the HAPT_1 dataset after LDC feature selection (244 features). The classification accuracies of all six different classifiers are entered in Figure 4.5.

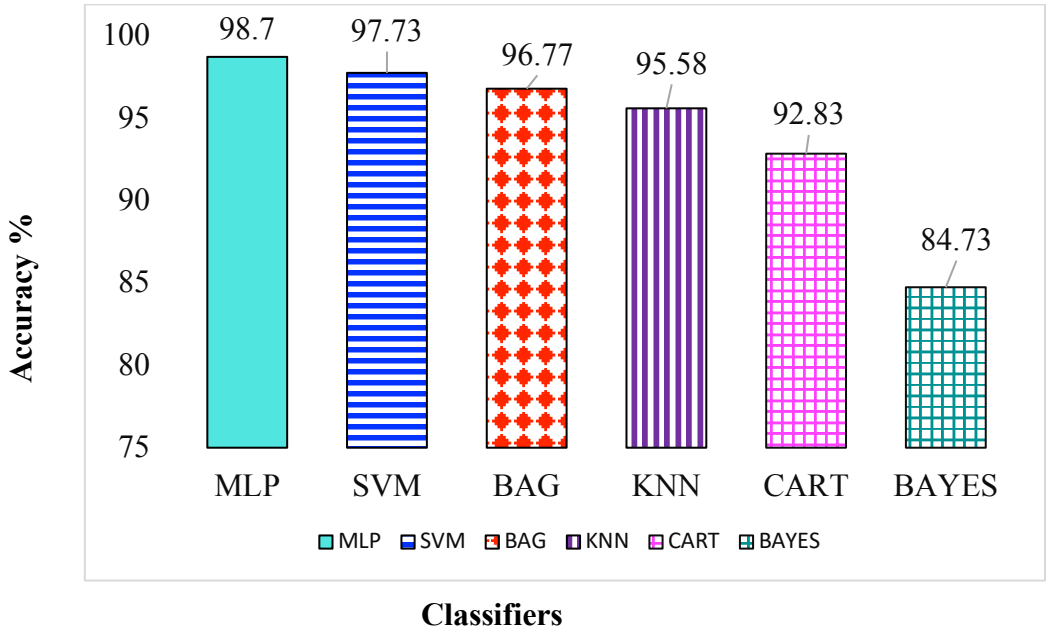



**Figure 4.5** Performances comparison of 6 classifiers to HAPT_1 dataset using LDC

Figure 4.5 illustrates that MLP worked best, SVM worked second best, BAG third best, KNN fourth best, CART fifth best, and BAYES performed the worst. That implies that the MLP model based on the LDC feature selection method was strong on the HAPT_1 dataset.

**(4) Discussion on Result Accuracy of HAPT_2 Dataset by LDC**

The dataset HAPT_2 included 12 activities (walking, walking upstairs, walking downstairs, sitting, standing, lying, standing-to-sitting, sitting-to-standing, sitting-to-lying, lying-to-sitting, standing-to-lying, lying-to-standing). The

performance evaluation of the MLP model applied to the HAPT_2 dataset after LDC feature selection (292 features) is shown in Table 4.11.

**Table 4.11** Confusion matrix of MLP on HAPT_2 dataset with LDC

| **HAPT_2 with the LDC (292-features)** | | **Predicted Activity** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | WK | WKU | WKD | SIT | STD | LY | STD-SIT | SIT-STD | SIT-LY | LY-SIT | STD-LY | LY-STD |
| **Actual Activity** | WK | 503 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | WKU | 1 | 479 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | WKD | 2 | 0 | 434 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SIT | 0 | 0 | 0 | 495 | 29 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | STD | 0 | 0 | 0 | 16 | 577 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | LY | 0 | 0 | 0 | 0 | 0 | 594 | 0 | 0 | 0 | 0 | 0 | 0 |
| | STD-SIT | 0 | 0 | 1 | 1 | 0 | 0 | 16 | 0 | 2 | 0 | 0 | 0 |
| | SIT-STD | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 8 | 0 | 1 | 0 | 0 |
| | SIT-LY | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 9 | 0 |
| | LY-SIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 6 |
| | STD-LY | 1 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 3 | 0 | 35 | 0 |
| | LY-STD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 8 | 0 | 18 |

* WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying, STD-SIT: stand-to-sit, SIT-STD: sit-to-stand, SIT-LY: sit-to-lie, LY-SIT: lie-to-sit, STD-LY: stand-to-lie, LY-STD: lie to stand.

In Table 4.11, MLP achieved an accuracy of 97.52%. Although sitting and standing activities were falsely classified, MLP performed better on each activity. In order to determine the efficiency of the MLP model, the original HAPT_2 dataset before LDC feature selection method was also evaluated. The performance results of the MLP model without LDC are presented in Table 4.12.

**Table 4.12** Confusion matrix of MLP on HAPT_2 dataset without LDC

| **HAPT_2 Without the LDC (561-features)** | | **Predicted Activity** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | WK | WKU | WKD | SIT | STD | LY | STD-SIT | SIT-STD | SIT-LY | LY-SIT | STD-LY | LY-STD |
| **Actual Activity** | WK | 504 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | WKU | 0 | 480 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | WKD | 2 | 0 | 434 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SIT | 0 | 0 | 0 | 503 | 21 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| | STD | 0 | 0 | 0 | 16 | 577 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | LY | 0 | 0 | 0 | 0 | 0 | 594 | 0 | 0 | 0 | 0 | 0 | 0 |
| | STD-SIT | 0 | 0 | 0 | 2 | 1 | 0 | 16 | 0 | 0 | 0 | 1 | 0 |
| | SIT-STD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 1 | 0 | 1 |
| | SIT-LY | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 17 | 0 | 9 | 0 |
| | LY-SIT | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 14 | 0 | 4 |
| | STD-LY | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 0 | 3 | 0 | 35 | 0 |
| | LY-STD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 9 | 1 | 16 |

* WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying, STD-SIT: stand-to-sit, SIT-STD: sit-to-stand, SIT-LY: sit-to-lie, LY-SIT: lie-to-sit, STD-LY: stand-to-lie, LY-STD: lie to stand

As a result of Table 4.12, the overall accuracy of HAPT_2 before LDC feature selection (561 features) was 97.53%. The confusion table demonstrates that there is a little bit of false classification between sitting and standing activities. According to the results of Table 4.11 and 4.12, the overall accuracy of HAPT_2 before LDC feature selection was 0.01% higher than HAPT_2 after LDC selection. Besides

this difference, the performance values of every activity are virtually the same in both models.

To compare the results of the MLP model, five other classifiers were applied to HAPT-2 after LDC feature selection (292 features). Experimental results of dataset HAPT_2 after LDC feature selection (292 features) from all six classifiers are shown in Figure 4.6.

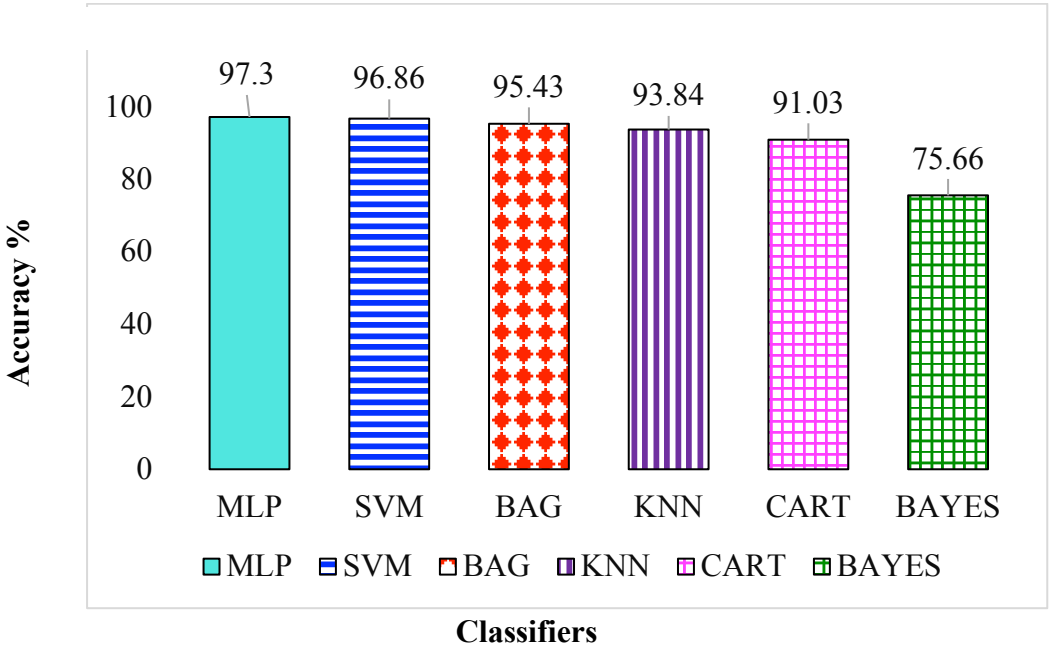



**Figure 4.6** Performances of 6 classifiers to HAPT_2 dataset with the LDC feature selection

In Figure 4.6, MLP achieved the best accuracy with 97.30%. SVM was the next best classifier with 96.86% accuracy. BAG performed third best, KNN fourth-best, CART fifth-best, and BAYES the worst. The results indicate that the MLP classifier was robust on the HAPT_2 dataset with the LDC features.

### 4.3.2 Performance evaluations by applying the LDC method to three groups (t, t', f) of the datasets

The LDC feature selection method was introduced as the main contribution of this dissertation in section 3.4. Three distinct datasets (UCI-HAR dataset, DATASET-UCI dataset, and HAPT-1 datasets) were used to further examine the

effectiveness of the proposed LDC feature selection method. Each dataset was divided into three groups (t = time-domain features grouped by angle values, t'= another time-domain features group of raw data, and f = frequency-domain features). The LDC method was individually applied to these three groups of features. By combining these three case features from each group, the performance results were evaluated and compared to the proposed system based on LDC feature selections of the whole dataset. The comparative performance results from the UCI-HAR dataset are illustrated in Table 4.13, Table 4.14, and Table 4.15.

**Table 4.13** Evaluation comparisons for UCI-HAR dataset: Performance results by individual features selected with LDC to form three data groups (t, t', f) and performance results based on proposed LDC method

| UCI-HAR dataset | | | | | | |
|---|---|---|---|---|---|---|
| Dataset | **t** | **t'** | **f** | Total Features | Accuracy % | Running Time (Sec) |
| UCI-HAR | 7 | 265 | 289 | 561 | 96.02 | 17 |
| LDC Data | 3 | 158 | 145 | 306 | 98.45 | 17 |

**Table 4.14** Evaluation comparison for DATASET-UCI dataset: Performance results by individual features selected with LDC to form three data groups (t, t', f) and performance result based on proposed LDC method

| DATASET-UCI dataset | | | | | | |
|---|---|---|---|---|---|---|
| Dataset | **t** | **t'** | **f** | Total Features | Accuracy % | Running Time (Sec) |
| DATASET-UCI | 7 | 265 | 289 | 561 | 87.53 | 16 |
| LDC Data | 2 | 162 | 150 | 314 | 90.07 | 16 |

**Table 4.15** Evaluation comparisons for HAPT-1 dataset: Performance results by individual features selected with LDC to form three data groups (t, t', f) and performance results based on proposed LDC method

| HAPT-1 dataset | | | | | | |
|---|---|---|---|---|---|---|
| Dataset | **t** | **t'** | **f** | Total Feature | Accuracy% | Running Time (Sec) |
| HAPT-1 | 7 | 265 | 289 | 561 | 98.66 | 19 |
| LDC Data | 3 | 156 | 143 | 302 | 98.56 | 25 |

* t = a time domain feature group by angle values,
* $t'$= another time domain features group by raw data,
* f = frequency domain features
*LDC Data = the data selecting the features by LDC method

As shown in Table 4.13, Table 4.14, and Table 4.15, although the proposed model performed slightly better with the entire dataset than with the individually selected feature groups, the performance calculation reduced running time more than the performance of the individually selected feature sets from the three different data groups. Therefore, the proposed method maintained the capacity of the suggested LDC feature selection method to achieve the necessary level of component accuracy by satisfying the most significant selection features for human activity recognition.

### 4.4 CAT Feature Selection Method

Feature selection methods were developed using LDC and CAT. The aim was to identify the best features for human activity recognition data. The CAT feature selection method selects the key features of a specified dataset [120].

#### 4.4.1 Implemented results of features of the CAT method

Data preprocessing was outlined in section 3.2.2.2. Each dataset was divided into three groups: $G_1$ from the time-domain raw signals, $G_2$ from the time-

domain angle values, and $G_3$ from the frequency domain. The CAT feature selection method identified the important features of each group to produce newly selected feature groups ($New\_G_1$, $New\_G_2$, $New\_G_3$) of each dataset. The total numbers are presented in Table 4.16.

**Table 4.16** Selected new features group and the total numbers of features

| Original feature group | New selected feature group | Total number of original features | Total number of selected features |
|---|---|---|---|
| $G_1$ | $New\_G_1$ | 265 | 56 |
| $G_2$ | $New\_G_2$ | 7 | 6 |
| $G_3$ | $New\_G_3$ | 289 | 16 |
| Total number of features | | **561** | **78** |

As a result of Table 4.16, the CAT feature selection method identifies the total number of selected features comprised 14% of the original dataset. The subgroups of the selected features, the total number of these features, and a list of all newly selected features are shown in Table 4.17.

**Table 4.17** New feature groups, total numbers of features, and CAT features

| $G_i$ | No. new features | CAT features |
|---|---|---|
| $G_1$ | 56 | {5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 53, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 106, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 159, 160, 165, 170, 175, 180, 185, 190, 195, 200, 205, 210, 212, 215, 220, 225, 230, 235, 240, 245, 250, 255, 260} |
| $G_2$ | 6 | {1,2,3,4,5,6} |
| $G_3$ | 16 | {17, 34, 51, 68, 85, 102, 119, 136, 153, 170, 187, 204, 221, 238, 255, 272} |

In Table 4.17, $\boldsymbol{G_1}$ is a newly selected feature subgroup from the time domain $D_1$, $\boldsymbol{G_2}$ is a newly selected feature subgroup from the time-domain $D_2$, and $\boldsymbol{G_3}$

is a newly selected feature subgroup from the frequency domain $D_3$. The numbers **1,2,3, . . . , 255** under the heading **CAT features** are the features newly selected by CAT.

The CAT feature selection method not only determined essential features but also reduced data dimensionality. Table 4.18 indicates the reduced dimensionality of each dataset implemented by the CAT method.

**Table 4.18** Dimensionality reductions on four datasets by CAT

| Data | UCI-HAR | DATERSET-UCI | HAPT_1 | HAPT_2 |
|---|---|---|---|---|
| **Original Data** | **5,777,739** | **3,222,384** | **5,840,571** | **6,131,169** |
| **CAT Data** | **803,322** | **448,032** | **812,058** | **852,462** |
| **Reduced Data** | **4,974,417** | **2,774,352** | **5,028,513** | **5,278,707** |

As shown in Table 4.18, CAT feature selection operated well on each dataset to reduce dimensionality. This result supported the classifier model to make classification faster and more effective.

In this section, performance of the CAT feature selection method was evaluated to identify the improvements due to the CAT feature selection method. The CAT method was applied to the whole dataset, and to three feature groups (t, t', f), picking up 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50% of features, and the performances were also evaluated without CAT.

### 4.4.2 Performances evaluations on the whole data by applying the CAT method

Four datasets (UCI-HAR, DATASET-UCI, HAPT_1, HAPT-2) were used in this experiment in the same way as in the LDC method. The MLP classifier was applied to each data set before and after feature selection by the CAT method.

#### (1) Discussion on Result Accuracy of UCI-HAR Dataset by CAT

The MLP classifier was applied to the UCI-HAR dataset before (78 features) and after feature selection by the CAT method (561 features). Results are given in Table 4.19.

**Table 4.19** Confusion matrix of MLP to UCI-HAR dataset with and without CAT

| **UCI-HAR dataset** | | | **Predicted Activity** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | WK | WKU | WKD | SIT | STD | LY |
| With CAT / (78 features) | **Actual Activity** | WK | 518 | 49 | 0 | 0 | 0 | 0 |
| | | WKU | 47 | 482 | 0 | 0 | 0 | 0 |
| | | WKD | 0 | 0 | 609 | 0 | 2 | 0 |
| | | SIT | 0 | 0 | 0 | 491 | 2 | 3 |
| | | STD | 0 | 0 | 0 | 1 | 431 | 1 |
| | | LY | 0 | 0 | 0 | 1 | 0 | 453 |
| UCI-HAR dataset | | | **Predicted Activity** | | | | | |
| | | | WK | WKU | WKD | SIT | STD | LY |
| without CAT/ (561 features) | **Actual Activity** | WK | 544 | 23 | 0 | 0 | 0 | 0 |
| | | WKU | 16 | 513 | 0 | 0 | 0 | 0 |
| | | WKD | 0 | 1 | 609 | 0 | 0 | 1 |
| | | SIT | 0 | 0 | 0 | 495 | 1 | 0 |
| | | STD | 0 | 0 | 0 | 1 | 431 | 1 |
| | | LY | 0 | 0 | 0 | 0 | 0 | 454 |

*WK: walking, WKU: walking-upstairs, WKD: walking-downstairs,
SIT: sitting, STD: standing, LY: laying

As a result of Table 4.19, the MLP classifier achieved an overall accuracy of 96.67% on the UCI-HAR dataset after feature selection with CAT and 98.56% before. In the MLP model, while walking and walking upstairs activities were falsely classified, the predictions of the remaining activities were good.

To determine the capability of the CAT model, SVM, BAG, KNN, CART, and BAYES classifiers were also implemented on the UCI-HAR dataset after CAT feature selection. The performance results of five classifiers applied to the

UCI-HAR dataset are compared in Figure 4.7.

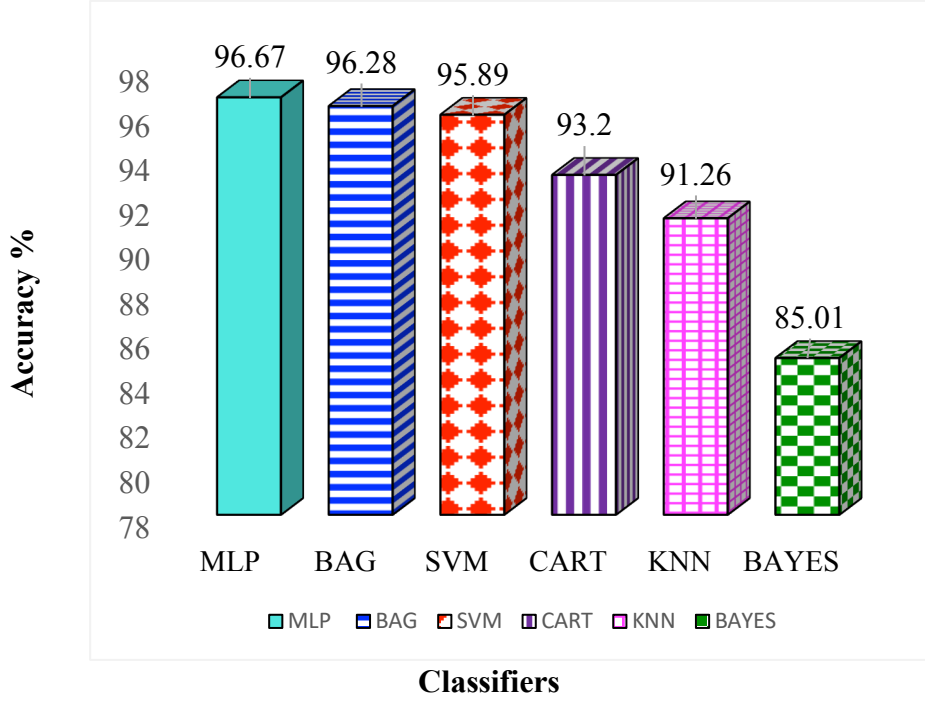



**Figure 4.7** Performance values of six classifiers for UCI-HAR data with CAT

As shown in Figure 4.8, MLP achieved the best overall accuracy of 96.67% followed by BAG with 96.28% accuracy. The BAYES classifier produced the worst accuracy of 85.01%. The MLP classifier for UCI-HAR dataset based on the CAT feature selection performed well in comparison with the other five classification algorithms. This result is a product of the successful identification by CAT of the key features of the UCI-HAR dataset. The MLP classifier supplied the best output for the CAT model.

**(2) Discussion on Result Accuracy of DATASET-UCI Dataset by CAT**

The DATASET-UCI is invoked as the next experiment to create the MLP model. Table 4.20 presents performance values of the MLP model applied to the DATASET-UCI dataset with (78 features) and without (561 features) CAT feature selection.

**Table 4.20** Confusion matrix of MLP to DATASET-UCI with and without CAT

| **DATASET-UCI dataset** | | | **Predicted Activity** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | WK | WKU | WKD | SIT | STD | LY |
| With CAT / (78 features) | **Actual Activity** | WK | 257 | 6 | 20 | 0 | 2 | 0 |
| | | WKU | 7 | 208 | 13 | 2 | 3 | 10 |
| | | WKD | 19 | 11 | 233 | 0 | 2 | 0 |
| | | SIT | 0 | 3 | 0 | 297 | 63 | 7 |
| | | STD | 1 | 0 | 0 | 76 | 242 | 0 |
| | | LY | 0 | 2 | 0 | 0 | 0 | 240 |
| UCI-HAR dataset | | | **Predicted Activity** | | | | | |
| | | | WK | WKU | WKD | SIT | STD | LY |
| without CAT/ (561 features) | **Actual Activity** | WK | 262 | 5 | 17 | 0 | 1 | 0 |
| | | WKU | 19 | 203 | 13 | 2 | 2 | 4 |
| | | WKD | 17 | 21 | 227 | 0 | 0 | 0 |
| | | SIT | 0 | 6 | 0 | 299 | 64 | 1 |
| | | STD | 0 | 0 | 0 | 43 | 276 | 0 |
| | | LY | 0 | 0 | 0 | 0 | 0 | 242 |

*WK: walking, WKU: walking-upstairs, WKD: walking-downstairs,
SIT: sitting, STD: standing, LY: laying

Table 4.20 presents the MLP applied to the DATASET-UCI before CAT feature selection (561 features) was 87.53% accurate. After CAT feature selection (78 features) the classifier was 85.67% accurate. Although the confusion table shows that the standing activity was wrongly categorized as sitting activity, the overall accuracy

of the MLP model on that dataset after feature selection with the CAT method (78 features) still met the aims of this study.

To establish the efficiency of the MLP model, five different classifiers were applied to DATASET-UCI after CAT feature selection (78 features). The comparison of performance results of different classifiers is shown in Figure 4.

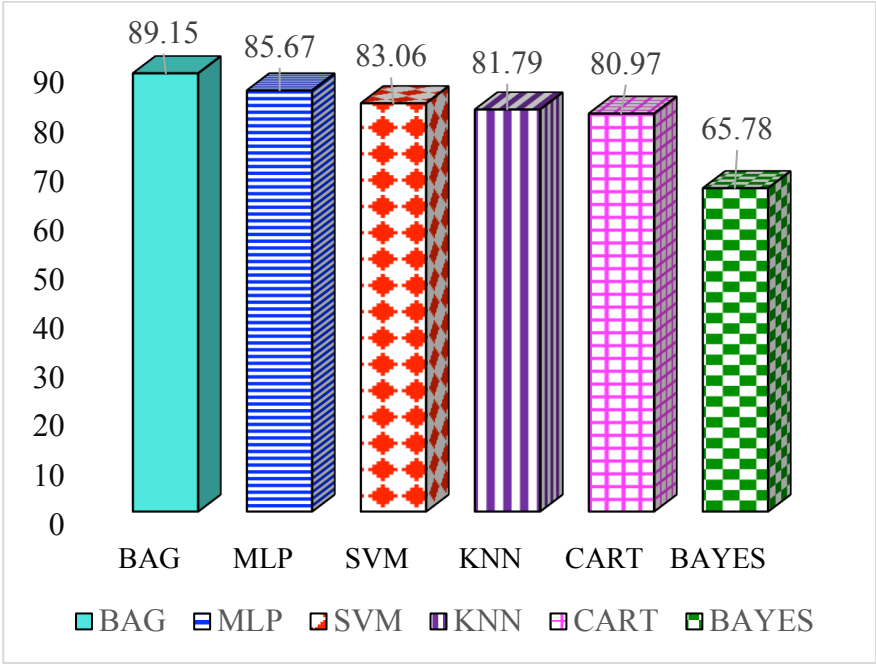



**Figure 4.8** Performances of six classifiers on DATASET-UCI with the CAT feature selection

As shown in Figure 4.8, due to false classifications between sitting and standing activities, the accuracy of the MLP classifier was 3.48% lower than the accuracy of the BAG classifier. However, MLP still operated well in classification among the remaining four algorithms. The MLP model achieved the second-best accuracy with 85.67%, SVM the third best and the BAYES classifier the lowest accuracy with 65.78%.

**(3) Discussion on Result Accuracy of HAPT_1 Dataset by CAT**

The MLP model was applied to the HAPT_1 dataset with (78 features) and without feature selection with the CAT method (561 features). The results are shown in Table 4.21.

**Table 4.21** Classification results to HAPT_1 with and without CAT

| **HAPT_1 dataset** | | | **Predicted Activity** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | WK | WKU | WKD | SIT | STD | LY |
| With CAT / (78 features) | **Actual Activity** | WK | 478 | 1 | 4 | 0 | 0 | 0 |
| | | WKU | 6 | 474 | 4 | 0 | 0 | 0 |
| | | WKD | 3 | 3 | 378 | 0 | 0 | 0 |
| | | SIT | 0 | 1 | 0 | 494 | 64 | 0 |
| | | STD | 1 | 0 | 0 | 31 | 576 | 0 |
| | | LY | 0 | 0 | 0 | 0 | 0 | 606 |
| **HAPT_1 dataset** | | | **Predicted Activity** | | | | | |
| | | | WK | WKU | WKD | SIT | STD | LY |
| without CAT/ (561 features) | **Actual Activity** | WK | 481 | 1 | 1 | 0 | 0 | 0 |
| | | WKU | 2 | 480 | 2 | 0 | 0 | 0 |
| | | WKD | 0 | 1 | 384 | 0 | 0 | 0 |
| | | SIT | 0 | 1 | 0 | 537 | 21 | 0 |
| | | STD | 0 | 0 | 0 | 12 | 595 | 0 |
| | | LY | 0 | 0 | 0 | 1 | 0 | 605 |

*WK: walking, WKU: walking-upstairs, WKD: walking-downstairs,
SIT: sitting, STD: standing, LY: laying

In Table 4.21, MLP achieved a classification accuracy of 96.22% on the HAPT_1 dataset after feature selection with the CAT method. Before CAT feature selection, MLP was 98.66% accurate. Having falsely classified sitting and standing activities, the overall accuracy of MLP after CAT selection was 2.44% less than it was before feature selection.

To compare the performance of the MLP classifier, five different classifiers were applied to the HAPT_1 dataset after feature selection with the CAT method (78 features). Figure 4.9 displays the outcomes of the different classifiers.

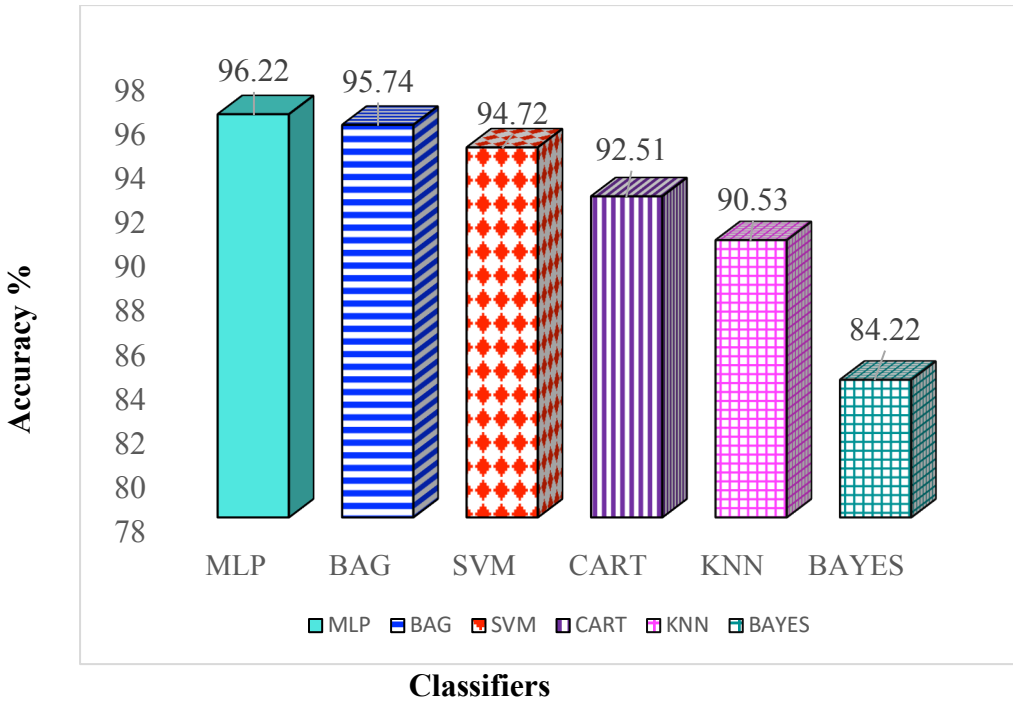



**Figure 4.9** Performances of six classifiers on HAPT_1 with the CAT feature selection (78 features)

As it can be seen in Figure 4.9, the MLP classifier achieved the best accuracy of 96.22%. BAG achieved the second-best accuracy with 95.74% and the BAYES classifier the lowest accuracy with 84.22%. The CAT feature selection method successfully identified the useful features of the HAPT_1 dataset.

### (4) Discussion on Result Accuracy of HAPT_2 Dataset by CAT

The HAPT_2 dataset contained 12 activities (walking, walking upstairs, walking downstairs, sitting, standing, lying, standing-to-sitting, sitting-to-standing, sitting-to-lying, lying-to-sitting, standing-to-lying, lying-to-standing). The MLP classification results of each activity in HAPT_2 after CAT feature selection method are given in Table 4.22.

**Table 4.22** Classification results of MLP to HAPT_2 with CAT features selection

| HAPT_2 with CAT (78-features) | | Predicted Activity | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | WK | WKU | WKD | SIT | STD | LY | STD-SIT | SIT-STD | SIT-LY | LY-SIT | STD-LY | LY-STD |
| Actual Activity | WK | 497 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | WKU | 3 | 474 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | WKD | 9 | 3 | 424 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SIT | 0 | 0 | 0 | 479 | 43 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| | STD | 1 | 0 | 0 | 31 | 561 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | LY | 0 | 0 | 0 | 0 | 0 | 593 | 0 | 0 | 0 | 0 | 0 | 1 |
| | STD-SIT | 0 | 0 | 0 | 1 | 1 | 0 | 16 | 0 | 1 | 0 | 1 | 0 |
| | SIT-STD | 0 | 0 | 0 | 0 | 0 | 0 | | 8 | 0 | 1 | 0 | 1 |
| | SIT-LY | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 18 | 0 | 8 | 0 |
| | LY-SIT | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 13 | 0 | 4 |
| | STD-LY | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 3 | 0 | 36 | 0 |
| | LY-STD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 6 | 0 | 20 |

* WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying, STD-SIT: stand-to-sit, SIT-STD: sit-to-stand, SIT-LY: sit-to-lie, LY-SIT: lie-to-sit, STD-LY: stand-to-lie, LY-STD: lie to stand.

In Table 4.22, the overall accuracy of the MLP model to HAPT_2 with 78 features was 95.73%. The specific MLP classifier model was more efficient on the first six activities. There were some misclassifications in the remaining 6 activities. In order to determine the effectiveness of the CAT model, the original HAPT_2 dataset before CAT feature selection was also classified by MLP. The performance results for the HAPT_2 dataset without the CAT (561 features) are presented in Table 4.23.

**Table 4.23** Classification results of MLP to HAPT_2 without CAT feature selection

| HAPT_2 Without CAT (561-features) | | Predicted Activity | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | WK | WKU | WKD | SIT | STD | LY | STD-SIT | SIT-STD | SIT-LY | LY-SIT | STD-LY | LY-STD |
| Actual Activity | WK | 504 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | WKU | 0 | 480 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | WKD | 2 | 0 | 434 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SIT | 0 | 0 | 0 | 503 | 21 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| | STD | 0 | 0 | 0 | 16 | 577 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | LY | 0 | 0 | 0 | 0 | 0 | 594 | 0 | 0 | 0 | 0 | 0 | 0 |
| | STD-SIT | 0 | 0 | 0 | 2 | 1 | 0 | 16 | 0 | 0 | 0 | 1 | 0 |
| | SIT-STD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 1 | 0 | 1 |
| | SIT-LY | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 17 | 0 | 9 | 0 |
| | LY-SIT | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 14 | 0 | 4 |
| | STD-LY | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 0 | 3 | 0 | 35 | 0 |
| | LY-STD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 9 | 1 | 16 |

* WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying, STD-SIT: stand-to-sit, SIT-STD: sit-to-stand, SIT-LY: sit-to-lie, LY-SIT: lie-to-sit, STD-LY: stand-to-lie, LY-STD: lie to stand.

Table 4.23 is shown that the overall accuracy of HAPT_2 without CAT (561 features) is 97.53%. The specific MLP classifier model performed better on the first six activities, which excluded the transition activities. Five different classifiers were also run on HAPT_2 after feature selection with the CAT method to compare the performance values with MLP. The comparison of performance results is shown in Figure 4.10.

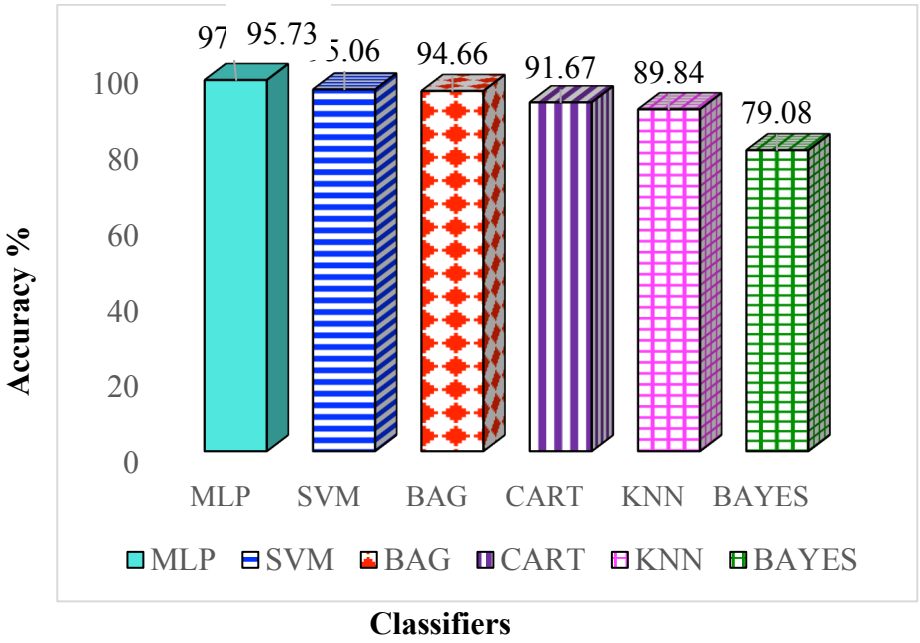



**Figure 4.10** Comparison of performance values with the MLP model and other classifiers using HAPT_2 dataset with CAT

In Figure 4.10, the MLP classifier produced the best accuracy of 97.53%. SVM was the next most accurate with 95.06% and the BAYES classifier was the least accurate with 79.08%. The MLP classifier successfully identified features of the HAPT_2 dataset with the CAT feature selection method.

### 4.4.3 Performances evaluations by dividing the data into three groups (t, t', f), picking up 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50% of features, and evaluation the performances without CAT

The main goal of this work was to effectively identify six human activities with a feature selection-based MLP classifier. The CAT feature selection method was implemented as one contribution of this dissertation. The CAT case approach was validated and assessed to show the effectiveness of its selection of the most significant features. The experiment passed through three datasets: UCI-HAR, DATASET-UCI, and HAPT-1.

These datasets were divided into three groups (t = time-domain features grouped by angle values, t'= another time-domain features group of raw data, and f = frequency-domain features. The experimental design included a total of ten (10) test

operations that picked up 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50% of each group of data before feature selection by CAT to evaluate the performance results by combining selected features from these three groups. The evaluation results are listed in Table 4.24, Table 4.25, and Table 4.26. The performance results are compared with the evaluation result of the proposed CAT feature set.

**Table 4.24** Performance results of data divided into three groups (t, t', f) with picked-up data of 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50% without feature selection by the CAT method and performance results based on feature selection by CAT method for the UCI-HAR dataset

| **UCI-HAR dataset** | | | | | |
|---|---|---|---|---|---|
| **Data** | **t** | **t'** | **f** | **Total Feature** | **Accuracy%** |
| 5% without CAT | 0 | 13 | 14 | 27 | 68.28 |
| 10% without CAT | 1 | 26 | 29 | 56 | 79.58 |
| 15% without CAT | 1 | 40 | 43 | 84 | 80.39 |
| 20% without CAT | 1 | 53 | 58 | 112 | 83.37 |
| 25% without CAT | 2 | 66 | 72 | 140 | 84.56 |
| 30% without CAT | 2 | 80 | 87 | 169 | 86.05 |
| 35% without CAT | 2 | 93 | 101 | 196 | 86.25 |
| 40% without CAT | 3 | 106 | 116 | 225 | 87.02 |
| 45% without CAT | 3 | 119 | 130 | 252 | 88.03 |
| 50% without CAT | 4 | 132 | 144 | 280 | 91.26 |
| **CAT Data** | **6** | **56** | **16** | **78** | **96.57** |

**Table 4.25** Performance results of data divided into three groups (t, t', f) with picked-up data of 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50% without feature selection by the CAT method and performance results based on feature selection by CAT method for the DATASET-UCI dataset

| **DATASET-UCI dataset** | | | | | |
|---|---|---|---|---|---|
| **Data** | **t** | **t'** | **f** | **Total Feature** | **Accuracy%** |
| 5% without CAT | 0 | 13 | 14 | 27 | 61.77 |
| 10% without CAT | 1 | 26 | 29 | 56 | 71.00 |
| 15% without CAT | 1 | 40 | 43 | 84 | 73.49 |
| 20% without CAT | 1 | 53 | 58 | 112 | 76.39 |
| 25% without CAT | 2 | 66 | 72 | 140 | 76.74 |
| 30% without CAT | 2 | 80 | 87 | 169 | 79.81 |
| 35% without CAT | 2 | 93 | 101 | 196 | 82.89 |
| 40% without CAT | 3 | 106 | 116 | 225 | 84.57 |
| 45% without CAT | 3 | 119 | 130 | 252 | 84.11 |
| 50% without CAT | 4 | 132 | 144 | 280 | 83.87 |
| **CAT Data** | **6** | **56** | **16** | **78** | **85.67** |

**Table 4.26** Performance results of data divided into three groups (t, t', f) with picked-up data of 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50% without feature selection by the CAT method and performance results based on feature selection by CAT method for the HAPT_1 dataset

| HAPT_1 dataset | | | | | |
|---|---|---|---|---|---|
| **Data** | **t** | **t'** | **f** | **Total Feature** | **Accuracy%** |
| 5% without CAT | 0 | 13 | 14 | 27 | 65.91 |
| 10% without CAT | 1 | 26 | 29 | 56 | 77.11 |
| 15% without CAT | 1 | 40 | 43 | 84 | 80.12 |
| 20% without CAT | 1 | 53 | 58 | 112 | 83.51 |
| 25% without CAT | 2 | 66 | 72 | 140 | 87.42 |
| 30% without CAT | 2 | 80 | 87 | 169 | 87.00 |
| 35% without CAT | 2 | 93 | 101 | 196 | 89.08 |
| 40% without CAT | 3 | 106 | 116 | 225 | 90.01 |
| 45% without CAT | 3 | 119 | 130 | 252 | 90.24 |
| 50% without CAT | 4 | 132 | 144 | 280 | 91.01 |
| **CAT Data** | **6** | **56** | **16** | **78** | **96.22** |

In Table 4.24, Table 4.25, and Table 4.26, results of classifying data without feature selection by CAT show that accuracy was higher when the percentage of data without feature selection was higher. However, as shown in this table, the MLP classifier performed better on a dataset with 78 selected features than on the divided dataset. The MLP classifier was more effective based on complete CAT feature

selection than on data which was not completely selected by CAT. As expected, the performance of the system using CAT was better than all testing using random percentage methods. The performance results were 96.57% accurate on the UCI-HAR dataset, 85.67% accurate on the DATASET-UCI dataset, and 96.22% accurate on the HAPT-1 dataset. In conclusion, the effectiveness of determining human activity was improved by the ability of CAT to identify the most significant features.

## 4.5 Comparative analysis of the LDC method and the CAT method

This section presents a comparative analysis of the impact of feature selection by LDC and the CAT on four datasets.

### 4.5.1 Dimensionality reduction

High dimensionality makes data visualization difficult and data processing complex. This section addresses objective 1 of this research to reduce the dimensionality of datasets in HAD. To determine the minimum number of features that could be selected for HAD, two feature selection methods were applied to four datasets. All the above results show that the LDC and the CAT methods accurately identified the more prominent features from each dataset. Based on the experimental results, these feature selection techniques could reduce the dimensionality of each dataset by determining the best features in each dataset. The comparison of the original features with a number of selected features by the LDC and the CAT for each dataset is presented in Table 4.27.

**Table 4.27** Comparison of original features, selected features using LDC and CAT

| Dataset | Original features | LDC features | CAT features | CAT reduced more features than LDC |
|---|---|---|---|---|
| **UCI-HAR** | 561 | 245 | 78 | 167 |
| **DATASET-UCI** | 561 | 269 | 78 | 191 |
| **HAPT_1** | 561 | 244 | 78 | 166 |
| **HAPT_2** | 561 | 292 | 78 | 214 |

As shown in Table 4.27, The ***LDC features*** column gives the important features selected by the LDC. The ***CAT features*** column gives the most significant features selected by the CAT. The column ***CAT reduced more features than LDC*** indicates that the number of more reduced features of the CAT.

Depending on the experimental results of the LDC and the CAT, these techniques can well determine the best feature from each dataset. However, the number of features selected by each method was significantly different. The data sizes of the selected feature sets and the original feature datasets are compared in Figure 4.11.

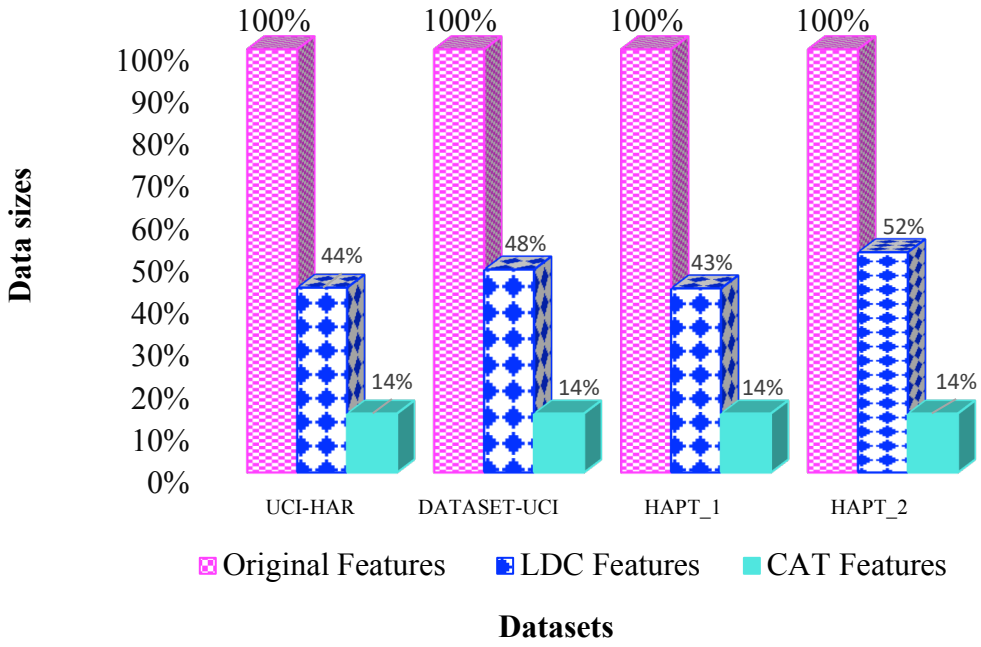



**Figure 4.11** Comparison of data percentage size of original, LDC and CAT features

Figure 4.11 shows that both the LDC and CAT techniques performed quite well at selecting features from each HAD dataset. However, the CAT feature selection method selected significantly more of the best features. These selected features reduced the high dimension of the data. Both feature selection methods performed well in choosing the most important features and in reducing the dimensionality of the data. The reduction in data dimensionality was the same as the reduction in feature data. The CAT method reduced the features selected from the original dataset by 86%, and also the dimensionality of the dataset by 86%. The

comparison of the data dimensionality applied by the CAT and the LDC to four different datasets is shown in Table 4.28.

**Table 4.28** Dimensionality data sizes of original, the CAT and LDC data

| Dataset | Original Data size | Data size by LDC | Data size by CAT |
|---|---|---|---|
| **UCI-HAR** | **5,777,739** | **2,523,255** | **803,322** |
| **DATASET-UCI** | **3,222,384** | **1,545,136** | **448,032** |
| **HAPT_1** | **5,840,571** | **2,540,284** | **812,058** |
| **HAPT_2** | **6,131,169** | **3,191,268** | **852,462** |

In Table 4.28, the column ***Dataset*** corresponds to the names of 4 datasets used throughout this research. The ***Original Data size*** column refers to the size of the original dataset from each original dataset. ***Data size by the LDC*** column gives the dataset sizes of the best feature subsets selected by the LDC. ***Data size by the CAT*** column shows the dataset sizes of the best feature subsets selected by the CAT.

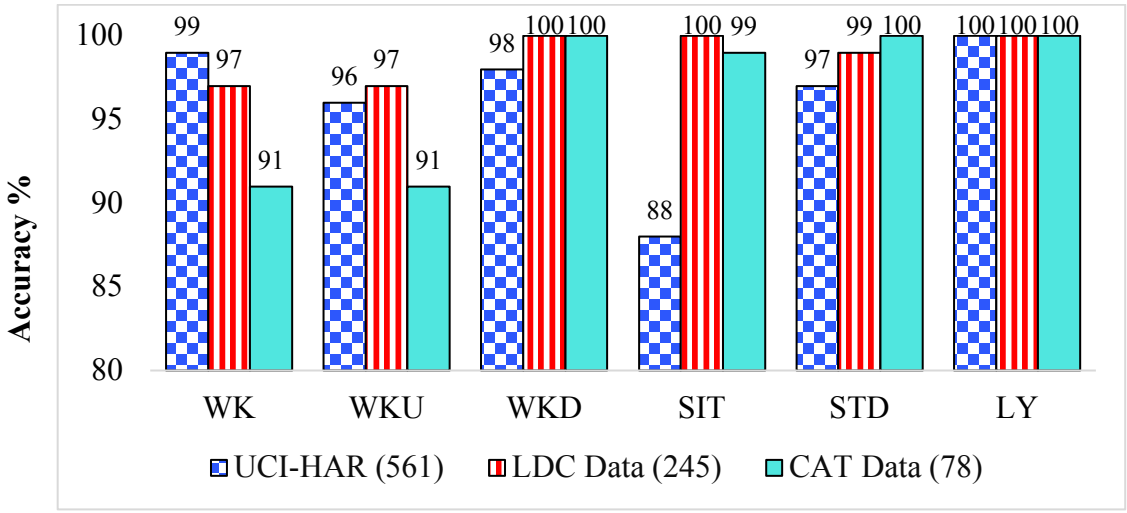
Based on the results, the CAT method more significantly reduced data size than the LDC method: by 86% compared with 48% to 57%. This result supports the achievement of Objective 1 effectively. The following section analyses in detail the performance results of the MLP classifier on four HAD datasets based on both the LDC and CAT feature selection methods.

### 4.5.2 Performance Comparison on each activity with and without LDC & CAT

This chapter deals with Objective 4 of this study: to build a more precise recognition system. The MLP classifier performed well in evaluating the performance results of datasets derived from the LDC and the CAT feature selection methods. The MLP classifier was applied to the original UCI-HAR dataset, the DATASET-UCI dataset, and the HAPT-1 dataset to compare the performance values of newly selected feature datasets.

Firstly, the performance results of classifications of each activity were considered using the original UCI-HAR dataset (561features), and using datasets from feature selection with the LDC method (245 features), and the CAT method (78

features). The performance results of the original data are available from the researchers who published the UCI-HAR dataset [27]. The UCI-HAR datasets from feature selection with LDC and CAT methods were assessed by the MLP classifier designed in this research. Figure 4.12 displays a comparison of all performance results for these three datasets for each activity.




* WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying

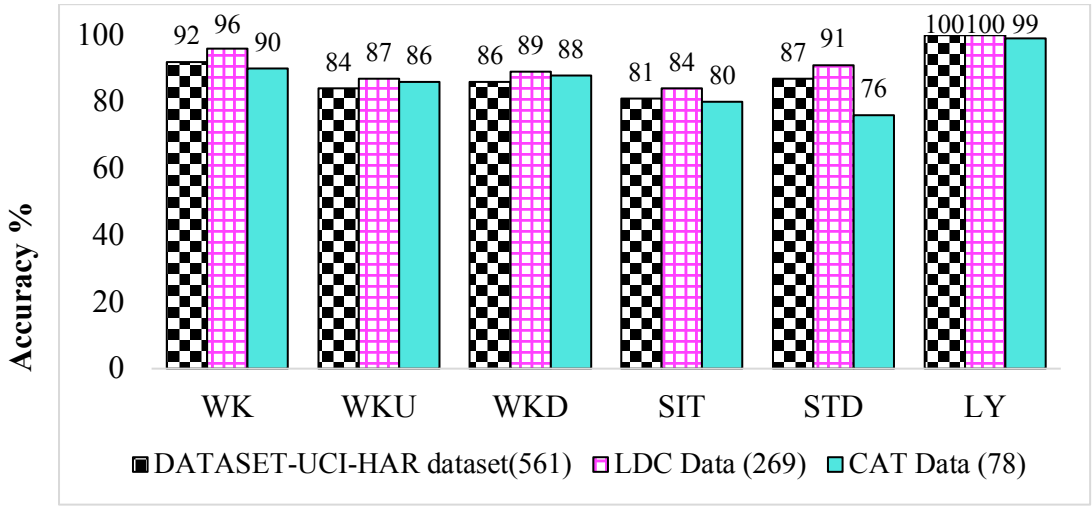
**Figure 4.12** Performance comparison of each activity from UCI-HAR dataset with original data, LDC data, and CAT data

As the results show, the MLP classifier performed well after feature selection with LDC and CAT for walking downstairs, sitting, standing, and lying down activities. Without feature selection, the performance of MLP on the UCI-HAR dataset was only better for walking. For the activity of walking upstairs, the MLP classifier performed best on the LDC selected feature set. For walking downstairs, standing, and lying down, MLP was most accurate with the CAT selected feature set.

The overall accuracy of MLP applied to the UCI-HAR dataset without feature selection was 96.02%, the overall accuracy of MLP applied to the CAT selected feature set was 96.67%, and the overall accuracy of MLP applied to the LDC feature set was 98.74%. While the MLP model applied to the LDC feature set used 44% of the UCI-HAR dataset, it achieved an accuracy that was 2.74% better than it achieved with

the original feature dataset. While the MLP model after feature selection with CAT used only 14% of the UCI-HAR dataset, its overall accuracy was still 0.65% better than with the complete, original dataset. Therefore, feature selection with the LDC and CAT methods effectively achieved Objective 4 of this study.

Secondly, the DATASET-UCI dataset was classified without feature selection, and after feature selection with the LDC and the CAT methods. The LDC and CAT data refer to the newly selected features datasets from the DATASET-UCI dataset. The original DATASET-UCI dataset contained 561 features. The LDC dataset contained 269 features, and the CAT dataset 78 features. The DATASET-UCI, the LDC data, and the CAT data were evaluated by the MLP classifier designed in this research. The comparison of all performance results of MLP for these datasets is given in Figure 4.13.



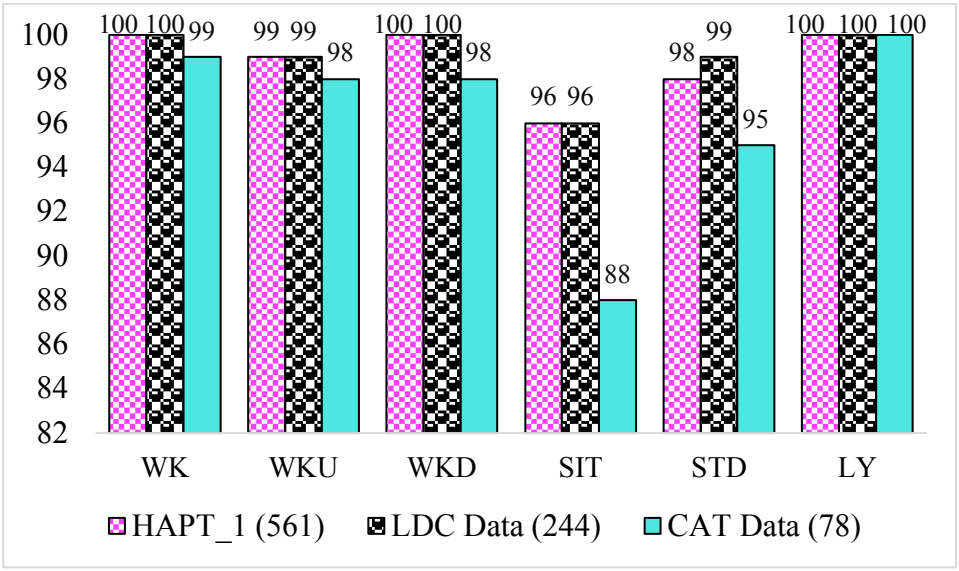

* WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying

**Figure 4.13** Performance comparison of each activity to DATASET-UCI dataset with original data, LDC data, and CAT data

The LDC data produced the best results, with an achieved accuracy rate of 90.66%. The DATASET-UCI produced an accuracy rate of 87.53% and the CAT dataset produced an accuracy rate of 85.67%. While the accuracy produced by the CAT data was 4.99% less than the LDC data produced and 1.86% less than the DATASET-UCI dataset produced, it was achieved using only 14% of the original dataset.

Therefore, the CAT feature selection method still performed well in selecting the best features from the DATASET-UCI dataset.

Finally, the classification of the HAPT_1 dataset and its derived LDC data and CAT data were considered. The LDC and CAT data from the HAPT_1 dataset refer to the datasets of features newly selected by LDC and CAT feature selection from the HAPT_1 dataset. The original HAPT_1 dataset contained 561 features, the LDC dataset 244 features, and the CAT dataset 78 features. The HAPT_1 dataset, the LDC dataset, and the CAT dataset were evaluated by the MLP classifier created in this research. The comparison of all performance results of datasets in each activity is shown in Figure 4.14.




* WK: walking, WKU: walking-upstairs, WKD: walking-downstairs, SIT: sitting, STD: standing, LY: laying

**Figure 4.14** Performances of HAPT_1, LDC data, CAT data in each activity

MLP performed best on the LDC dataset with an accuracy of 98.7%. The classifier achieved 98.66% accuracy with the HAPT_1 dataset and 96.22% accuracy with the CAT dataset. It misclassified between standing and sitting activities. Even though overall accuracy with the CAT dataset was 2.48% less than the accuracy achieved with the LDC dataset and 2.44% less than the overall accuracy achieved with the HAPT_1 dataset, the MLP generally produced quite good results from the CAT dataset.

# CHAPTER 5

# CONCLUSIONS

## 5.1 Summary

Evaluating recognition performance is crucial for HAD in many study fields, especially in healthcare systems. Although many researchers have developed HAD systems with machine learning methods, there remains a difficult challenge to develop a more general and precise system. Some of the problems include the high dimensionality of the data, extreme slowness of the model, and the complexity of detection. The main objectives of this research were to reduce the dimensionality of the data, to determine the best-suited features, to create new feature selection methods using LDC and CAT, and to develop a new, precise ANN classifier model and a more accurate HAD system. This research was undertaken in two tasks.

Task_1 of this study aimed to understand the HAD system. Data collection is one process of the HAD system. To get a background understanding of mobile sensing data, the TAD dataset was created. Before creating the main system design, step counting increased understanding of the system and helped to manage the high dimensionality of the dataset. Then a step counting technique using a smartphone accelerometer was introduced as a basic preliminary work. Task_2 of this research was the work on the main system. Two different feature selection methods and a novel classifier were proposed to fulfill the primary goal. It was decided to implement a system where the advantages of each technique could be used. Feature selection is the process of selecting a subset of relevant features or variables. Redundant or irrelevant features are useless features that do not have any information for the output. Ignoring irrelevant features improves the accuracy and decreases the dimensionality of the data.

An excellent feature selection method requires a better mathematical model to determine the best-suited features. A good feature selection technique is directly correlated to the dimensionality of the data and the classification model. Therefore, two different feature selection methods were investigated. One method,

Linearly Dependent Concepts (LDC) feature selection, used linearly dependent concepts, and the other, Cyclic Attribution Technique (CAT) feature selection, used group theory and fundamental properties of cyclic groups.

In order to achieve better performance in ML, it is essential when modeling to choose a suitable classifier with appropriate parameters. ANN is an effective tool in the field of pattern recognition. Hence, an artificial neural network model using Multi-Layer Perception (MLP) was designed with a new formula of hidden neuron nodes for systematic development of the HAD system based on a novel feature selection method.

This ANN model examined the level of accuracy in HAD by using a new feature selection method. The accuracy of the MLP classifier was measured on four datasets (UCI-HAR, DATASET-UCI, HAPT_1, HAPT_2). The performance indicators were the identification of six activities (walking, walking upstairs, walking downstairs, sitting, standing, and lying down) before and after feature selection by LDC and CAT methods. To compare the classification outcomes, SVM, BAG, KNN, CART, and BAYES algorithms were also used to classify the feature sets. This section presents a model comparison of MLP classification between LDC and CAT datasets. Table 5.1 summarizes the primary precision of the results of running time obtained in the previous section.

**Table 5.1** Performance of MLP on three original datasets before and after feature selection methods (LDC vs. CAT)

| Input Data | UCI-HAR dataset | | | DATASET-UCI | | | HAPT_1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | No. Features | Accuracy % | Runing time (sec) | No. Features | Accuracy % | Runing time (sec) | No. Features | Accuracy % | Runing time (sec) |
| **Original** | **561** | **96.67** | **43** | **561** | **87.53** | **36** | **561** | **98.56** | **41** |
| **LDC Data** | **245** | **98.74** | **10** | **269** | **90.66** | **15** | **244** | **98.75** | **18** |
| **CAT Data** | **78** | **96.57** | **12** | **78** | **85.67** | **10** | **78** | **96.22** | **13** |

In Table 5.1, the word ***Original*** refers to the original data from the UCI-HAR dataset, DATASET-UCI dataset, and HAPT_1. Word ***LDC data*** give the newly selected features dataset by using LDC feature selection method from these datasets. The ***CAT data*** related to the newly selected feature dataset by using the CAT feature selection method from these datasets. Depending on the LDC and CAT experimental outcomes, the best feature of each dataset can be determined by these methods. However, there is a significant difference in the number of features selected by the LDC and CAT.

The main goal was to identify the most efficient feature selection method with a specific MLP classifier for HAD and to demonstrate the advantages and disadvantages of feature selection by the LDC and CAT methods. These two feature selection methods had been developed with different methodologies and techniques. The main aim of feature selection methods is to identify the most important features. Although the sizes of the three datasets used in this system were significantly different, accuracies among them were not much different. The findings showed a quite satisfactory level of accuracy.

Based on the LDC and CAT experimental outcomes, the best features of each dataset were determined by both these methods. However, there was a significant difference in the number of features selected by the two methods. Implementing the ANN model with LDC reduced 561 features to 245 (44%) from the UCI-HAR dataset, 269 (48%) from the DATASET-UCI dataset, 244 (43%) from the HAPT_1 dataset, and 292 (52%) from the HAPT_2 dataset. Running the model with CAT reduced 561 features to 78 (14%) for all datasets.

The results in Table 5.1 show that feature selection with LDC generated the best overall accuracy of the model with all datasets, achieving a score of 96.41%, the original datasets produced an overall accuracy of 94.93%, and feature selection with the CAT method generated the third best accuracy of 93.57% from all the datasets.

The two feature selection methods have different approaches and solutions. Possible solutions can be found in this study. The benefits of these proposed

feature selection methods are the generation of a small set of features, the reduced dimensionality of the data, and a more effective and quicker classifier. The disadvantages of these feature selection methods are the dependence on feature sets and some loss of information. Nonetheless, our proposed system successfully fulfilled the main objectives of this research and, therefore, a specific feature selection method and specific classifier might be able to play important roles in the determination of human activity.

As shown in Table 5.1, the MLP model-based LDC is a very comprehensive methodology and evaluation technique. Although CAT could reduce the features 30% more than LDC, it was not more accurate. When comparing running time, LDC not only had a better accuracy but also a shorter running time than CAT, although CAT running time was a few seconds quicker than LDC runtime on the DATASET-UCI and HAPT datasets. Therefore, the MLP model-based LDC approach was the most comprehensively applicable and effective methodology for HAD systems development and the MLP model-based CAT approach was the second-best methodology.

## 5.2 Future Work

The present MLP model-based LDC and CAT feature selections methods were generated to produce a more reliable and precise system for the determination of human activity. In the future, implemented in many machine learning problems, these models will be beneficial for any ML problems and support in healthcare systems. In addition, these proposed classifier and feature selection methods can be compared with different feature selection methods and create real-time applications to support many human healthcare systems.

# APPENDIX

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 1 | tBodyAcc-mean()-X | tBodyAcc-mean()-X | tBodyAcc-Mean-1 |
| 2 | tBodyAcc-mean()-Y | tBodyAcc-mean()-Y | tBodyAcc-Mean-2 |
| 3 | tBodyAcc-mean()-Z | tBodyAcc-mean()-Z | tBodyAcc-Mean-3 |
| 4 | tBodyAcc-std()-X | tBodyAcc-std()-X | tBodyAcc-STD-1 |
| 5 | tBodyAcc-std()-Y | tBodyAcc-std()-Y | tBodyAcc-STD-2 |
| 6 | tBodyAcc-std()-Z | tBodyAcc-std()-Z | tBodyAcc-STD-3 |
| 7 | tBodyAcc-mad()-X | tBodyAcc-mad()-X | tBodyAcc-Mad-1 |
| 8 | tBodyAcc-mad()-Y | tBodyAcc-mad()-Y | tBodyAcc-Mad-2 |
| 9 | tBodyAcc-mad()-Z | tBodyAcc-mad()-Z | tBodyAcc-Mad-3 |
| 10 | tBodyAcc-max()-X | tBodyAcc-max()-X | tBodyAcc-Max-1 |
| 11 | tBodyAcc-max()-Y | tBodyAcc-max()-Y | tBodyAcc-Max-2 |
| 12 | tBodyAcc-max()-Z | tBodyAcc-max()-Z | tBodyAcc-Max-3 |
| 13 | tBodyAcc-min()-X | tBodyAcc-min()-X | tBodyAcc-Min-1 |
| 14 | tBodyAcc-min()-Y | tBodyAcc-min()-Y | tBodyAcc-Min-2 |
| 15 | tBodyAcc-min()-Z | tBodyAcc-min()-Z | tBodyAcc-Min-3 |
| 16 | tBodyAcc-sma() | tBodyAcc-sma() | tBodyAcc-SMA-1 |
| 17 | tBodyAcc-energy()-X | tBodyAcc-energy()-X | tBodyAcc-Energy-1 |
| 18 | tBodyAcc-energy()-Y | tBodyAcc-energy()-Y | tBodyAcc-Energy-2 |
| 19 | tBodyAcc-energy()-Z | tBodyAcc-energy()-Z | tBodyAcc-Energy-3 |
| 20 | tBodyAcc-iqr()-X | tBodyAcc-iqr()-X | tBodyAcc-IQR-1 |
| 21 | tBodyAcc-iqr()-Y | tBodyAcc-iqr()-Y | tBodyAcc-IQR-2 |
| 22 | tBodyAcc-iqr()-Z | tBodyAcc-iqr()-Z | tBodyAcc-IQR-3 |
| 23 | tBodyAcc-entropy()-X | tBodyAcc-entropy()-X | tBodyAcc-ropy-1 |
| 24 | tBodyAcc-entropy()-Y | tBodyAcc-entropy()-Y | tBodyAcc-ropy-1 |
| 25 | tBodyAcc-entropy()-Z | tBodyAcc-entropy()-Z | tBodyAcc-ropy-1 |
| 26 | tBodyAcc-arCoeff()-X,1 | tBodyAcc-arCoeff()-X,1 | tBodyAcc-ARCoeff-1 |
| 27 | tBodyAcc-arCoeff()-X,2 | tBodyAcc-arCoeff()-X,2 | tBodyAcc-ARCoeff-2 |
| 28 | tBodyAcc-arCoeff()-X,3 | tBodyAcc-arCoeff()-X,3 | tBodyAcc-ARCoeff-3 |
| 29 | tBodyAcc-arCoeff()-X,4 | tBodyAcc-arCoeff()-X,4 | tBodyAcc-ARCoeff-4 |
| 30 | tBodyAcc-arCoeff()-Y,1 | tBodyAcc-arCoeff()-Y,1 | tBodyAcc-ARCoeff-5 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 31 | tBodyAcc-arCoeff()-Y,2 | tBodyAcc-arCoeff()-Y,2 | tBodyAcc-ARCoeff-6 |
| 32 | tBodyAcc-arCoeff()-Y,3 | tBodyAcc-arCoeff()-Y,3 | tBodyAcc-ARCoeff-7 |
| 33 | tBodyAcc-arCoeff()-Y,4 | tBodyAcc-arCoeff()-Y,4 | tBodyAcc-ARCoeff-8 |
| 34 | tBodyAcc-arCoeff()-Z,1 | tBodyAcc-arCoeff()-Z,1 | tBodyAcc-ARCoeff-9 |
| 35 | tBodyAcc-arCoeff()-Z,2 | tBodyAcc-arCoeff()-Z,2 | tBodyAcc-ARCoeff-10 |
| 36 | tBodyAcc-arCoeff()-Z,3 | tBodyAcc-arCoeff()-Z,3 | tBodyAcc-ARCoeff-11 |
| 37 | tBodyAcc-arCoeff()-Z,4 | tBodyAcc-arCoeff()-Z,4 | tBodyAcc-ARCoeff-12 |
| 38 | tBodyAcc-correlation()-X,Y | tBodyAcc-correlation()-X,Y | tBodyAcc-Correlation-1 |
| 39 | tBodyAcc-correlation()-X,Z | tBodyAcc-correlation()-X,Z | tBodyAcc-Correlation-2 |
| 40 | tBodyAcc-correlation()-Y,Z | tBodyAcc-correlation()-Y,Z | tBodyAcc-Correlation-3 |
| 41 | tGravityAcc-mean()-X | tGravityAcc-mean()-X | tGravityAcc-Mean-1 |
| 42 | tGravityAcc-mean()-Y | tGravityAcc-mean()-Y | tGravityAcc-Mean-2 |
| 43 | tGravityAcc-mean()-Z | tGravityAcc-mean()-Z | tGravityAcc-Mean-3 |
| 44 | tGravityAcc-std()-X | tGravityAcc-std()-X | tGravityAcc-STD-1 |
| 45 | tGravityAcc-std()-Y | tGravityAcc-std()-Y | tGravityAcc-STD-2 |
| 46 | tGravityAcc-std()-Z | tGravityAcc-std()-Z | tGravityAcc-STD-3 |
| 47 | tGravityAcc-mad()-X | tGravityAcc-mad()-X | tGravityAcc-Mad-1 |
| 48 | tGravityAcc-mad()-Y | tGravityAcc-mad()-Y | tGravityAcc-Mad-2 |
| 49 | tGravityAcc-mad()-Z | tGravityAcc-mad()-Z | tGravityAcc-Mad-3 |
| 50 | tGravityAcc-max()-X | tGravityAcc-max()-X | tGravityAcc-Max-1 |
| 51 | tGravityAcc-max()-Y | tGravityAcc-max()-Y | tGravityAcc-Max-2 |
| 52 | tGravityAcc-max()-Z | tGravityAcc-max()-Z | tGravityAcc-Max-3 |
| 53 | tGravityAcc-min()-X | tGravityAcc-min()-X | tGravityAcc-Min-1 |
| 54 | tGravityAcc-min()-Y | tGravityAcc-min()-Y | tGravityAcc-Min-2 |
| 55 | tGravityAcc-min()-Z | tGravityAcc-min()-Z | tGravityAcc-Min-3 |
| 56 | tGravityAcc-sma() | tGravityAcc-sma() | tGravityAcc-SMA-1 |
| 57 | tGravityAcc-energy()-X | tGravityAcc-energy()-X | tGravityAcc-Energy-1 |
| 58 | tGravityAcc-energy()-Y | tGravityAcc-energy()-Y | tGravityAcc-Energy-2 |
| 59 | tGravityAcc-energy()-Z | tGravityAcc-energy()-Z | tGravityAcc-Energy-3 |
| 60 | tGravityAcc-iqr()-X | tGravityAcc-iqr()-X | tGravityAcc-IQR-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 61 | tGravityAcc-iqr()-Y | tGravityAcc-iqr()-Y | tGravityAcc-IQR-2 |
| 62 | tGravityAcc-iqr()-Z | tGravityAcc-iqr()-Z | tGravityAcc-IQR-3 |
| 63 | tGravityAcc-entropy()-X | tGravityAcc-entropy()-X | tGravityAcc-ropy-1 |
| 64 | tGravityAcc-entropy()-Y | tGravityAcc-entropy()-Y | tGravityAcc-ropy-1 |
| 65 | tGravityAcc-entropy()-Z | tGravityAcc-entropy()-Z | tGravityAcc-ropy-1 |
| 66 | tGravityAcc-arCoeff()-X,1 | tGravityAcc-arCoeff()-X,1 | tGravityAcc-ARCoeff-1 |
| 67 | tGravityAcc-arCoeff()-X,2 | tGravityAcc-arCoeff()-X,2 | tGravityAcc-ARCoeff-2 |
| 68 | tGravityAcc-arCoeff()-X,3 | tGravityAcc-arCoeff()-X,3 | tGravityAcc-ARCoeff-3 |
| 69 | tGravityAcc-arCoeff()-X,4 | tGravityAcc-arCoeff()-X,4 | tGravityAcc-ARCoeff-4 |
| 70 | tGravityAcc-arCoeff()-Y,1 | tGravityAcc-arCoeff()-Y,1 | tGravityAcc-ARCoeff-5 |
| 71 | tGravityAcc-arCoeff()-Y,2 | tGravityAcc-arCoeff()-Y,2 | tGravityAcc-ARCoeff-6 |
| 72 | tGravityAcc-arCoeff()-Y,3 | tGravityAcc-arCoeff()-Y,3 | tGravityAcc-ARCoeff-7 |
| 73 | tGravityAcc-arCoeff()-Y,4 | tGravityAcc-arCoeff()-Y,4 | tGravityAcc-ARCoeff-8 |
| 74 | tGravityAcc-arCoeff()-Z,1 | tGravityAcc-arCoeff()-Z,1 | tGravityAcc-ARCoeff-9 |
| 75 | tGravityAcc-arCoeff()-Z,2 | tGravityAcc-arCoeff()-Z,2 | tGravityAcc-ARCoeff-10 |
| 76 | tGravityAcc-arCoeff()-Z,3 | tGravityAcc-arCoeff()-Z,3 | tGravityAcc-ARCoeff-11 |
| 77 | tGravityAcc-arCoeff()-Z,4 | tGravityAcc-arCoeff()-Z,4 | tGravityAcc-ARCoeff-12 |
| 78 | tGravityAcc-correlation()-X,Y | tGravityAcc-correlation()-X,Y | tGravityAcc-Correlation-1 |
| 79 | tGravityAcc-correlation()-X,Z | tGravityAcc-correlation()-X,Z | tGravityAcc-Correlation-2 |
| 80 | tGravityAcc-correlation()-Y,Z | tGravityAcc-correlation()-Y,Z | tGravityAcc-Correlation-3 |
| 81 | tBodyAccJerk-mean()-X | tBodyAccJerk-mean()-X | tBodyAccJerk-Mean-1 |
| 82 | tBodyAccJerk-mean()-Y | tBodyAccJerk-mean()-Y | tBodyAccJerk-Mean-2 |
| 83 | tBodyAccJerk-mean()-Z | tBodyAccJerk-mean()-Z | tBodyAccJerk-Mean-3 |
| 84 | tBodyAccJerk-std()-X | tBodyAccJerk-std()-X | tBodyAccJerk-STD-1 |
| 85 | tBodyAccJerk-std()-Y | tBodyAccJerk-std()-Y | tBodyAccJerk-STD-2 |
| 86 | tBodyAccJerk-std()-Z | tBodyAccJerk-std()-Z | tBodyAccJerk-STD-3 |
| 87 | tBodyAccJerk-mad()-X | tBodyAccJerk-mad()-X | tBodyAccJerk-Mad-1 |
| 88 | tBodyAccJerk-mad()-Y | tBodyAccJerk-mad()-Y | tBodyAccJerk-Mad-2 |
| 89 | tBodyAccJerk-mad()-Z | tBodyAccJerk-mad()-Z | tBodyAccJerk-Mad-3 |
| 90 | tBodyAccJerk-max()-X | tBodyAccJerk-max()-X | tBodyAccJerk-Max-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 91 | tBodyAccJerk-max()-Y | tBodyAccJerk-max()-Y | tBodyAccJerk-Max-2 |
| 92 | tBodyAccJerk-max()-Z | tBodyAccJerk-max()-Z | tBodyAccJerk-Max-3 |
| 93 | tBodyAccJerk-min()-X | tBodyAccJerk-min()-X | tBodyAccJerk-Min-1 |
| 94 | tBodyAccJerk-min()-Y | tBodyAccJerk-min()-Y | tBodyAccJerk-Min-2 |
| 95 | tBodyAccJerk-min()-Z | tBodyAccJerk-min()-Z | tBodyAccJerk-Min-3 |
| 96 | tBodyAccJerk-sma() | tBodyAccJerk-sma() | tBodyAccJerk-SMA-1 |
| 97 | tBodyAccJerk-energy()-X | tBodyAccJerk-energy()-X | tBodyAccJerk-Energy-1 |
| 98 | tBodyAccJerk-energy()-Y | tBodyAccJerk-energy()-Y | tBodyAccJerk-Energy-2 |
| 99 | tBodyAccJerk-energy()-Z | tBodyAccJerk-energy()-Z | tBodyAccJerk-Energy-3 |
| 100 | tBodyAccJerk-iqr()-X | tBodyAccJerk-iqr()-X | tBodyAccJerk-IQR-1 |
| 101 | tBodyAccJerk-iqr()-Y | tBodyAccJerk-iqr()-Y | tBodyAccJerk-IQR-2 |
| 102 | tBodyAccJerk-iqr()-Z | tBodyAccJerk-iqr()-Z | tBodyAccJerk-IQR-3 |
| 103 | tBodyAccJerk-entropy()-X | tBodyAccJerk-entropy()-X | tBodyAccJerk-ropy-1 |
| 104 | tBodyAccJerk-entropy()-Y | tBodyAccJerk-entropy()-Y | tBodyAccJerk-ropy-1 |
| 105 | tBodyAccJerk-entropy()-Z | tBodyAccJerk-entropy()-Z | tBodyAccJerk-ropy-1 |
| 106 | tBodyAccJerk-arCoeff()-X,1 | tBodyAccJerk-arCoeff()-X,1 | tBodyAccJerk-ARCoeff-1 |
| 107 | tBodyAccJerk-arCoeff()-X,2 | tBodyAccJerk-arCoeff()-X,2 | tBodyAccJerk-ARCoeff-2 |
| 108 | tBodyAccJerk-arCoeff()-X,3 | tBodyAccJerk-arCoeff()-X,3 | tBodyAccJerk-ARCoeff-3 |
| 109 | tBodyAccJerk-arCoeff()-X,4 | tBodyAccJerk-arCoeff()-X,4 | tBodyAccJerk-ARCoeff-4 |
| 110 | tBodyAccJerk-arCoeff()-Y,1 | tBodyAccJerk-arCoeff()-Y,1 | tBodyAccJerk-ARCoeff-5 |
| 111 | tBodyAccJerk-arCoeff()-Y,2 | tBodyAccJerk-arCoeff()-Y,2 | tBodyAccJerk-ARCoeff-6 |
| 112 | tBodyAccJerk-arCoeff()-Y,3 | tBodyAccJerk-arCoeff()-Y,3 | tBodyAccJerk-ARCoeff-7 |
| 113 | tBodyAccJerk-arCoeff()-Y,4 | tBodyAccJerk-arCoeff()-Y,4 | tBodyAccJerk-ARCoeff-8 |
| 114 | tBodyAccJerk-arCoeff()-Z,1 | tBodyAccJerk-arCoeff()-Z,1 | tBodyAccJerk-ARCoeff-9 |
| 115 | tBodyAccJerk-arCoeff()-Z,2 | tBodyAccJerk-arCoeff()-Z,2 | tBodyAccJerk-ARCoeff-10 |
| 116 | tBodyAccJerk-arCoeff()-Z,3 | tBodyAccJerk-arCoeff()-Z,3 | tBodyAccJerk-ARCoeff-11 |
| 117 | tBodyAccJerk-arCoeff()-Z,4 | tBodyAccJerk-arCoeff()-Z,4 | tBodyAccJerk-ARCoeff-12 |
| 118 | tBodyAccJerk-correlation()-X,Y | tBodyAccJerk-correlation()-X,Y | tBodyAccJerk-Correlation-1 |
| 119 | tBodyAccJerk-correlation()-X,Z | tBodyAccJerk-correlation()-X,Z | tBodyAccJerk-Correlation-2 |
| 120 | tBodyAccJerk-correlation()-Y,Z | tBodyAccJerk-correlation()-Y,Z | tBodyAccJerk-Correlation-3 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 121 | tBodyGyro-mean()-X | tBodyGyro-mean()-X | tBodyGyro-Mean-1 |
| 122 | tBodyGyro-mean()-Y | tBodyGyro-mean()-Y | tBodyGyro-Mean-2 |
| 123 | tBodyGyro-mean()-Z | tBodyGyro-mean()-Z | tBodyGyro-Mean-3 |
| 124 | tBodyGyro-std()-X | tBodyGyro-std()-X | tBodyGyro-STD-1 |
| 125 | tBodyGyro-std()-Y | tBodyGyro-std()-Y | tBodyGyro-STD-2 |
| 126 | tBodyGyro-std()-Z | tBodyGyro-std()-Z | tBodyGyro-STD-3 |
| 127 | tBodyGyro-mad()-X | tBodyGyro-mad()-X | tBodyGyro-Mad-1 |
| 128 | tBodyGyro-mad()-Y | tBodyGyro-mad()-Y | tBodyGyro-Mad-2 |
| 129 | tBodyGyro-mad()-Z | tBodyGyro-mad()-Z | tBodyGyro-Mad-3 |
| 130 | tBodyGyro-max()-X | tBodyGyro-max()-X | tBodyGyro-Max-1 |
| 131 | tBodyGyro-max()-Y | tBodyGyro-max()-Y | tBodyGyro-Max-2 |
| 132 | tBodyGyro-max()-Z | tBodyGyro-max()-Z | tBodyGyro-Max-3 |
| 133 | tBodyGyro-min()-X | tBodyGyro-min()-X | tBodyGyro-Min-1 |
| 134 | tBodyGyro-min()-Y | tBodyGyro-min()-Y | tBodyGyro-Min-2 |
| 135 | tBodyGyro-min()-Z | tBodyGyro-min()-Z | tBodyGyro-Min-3 |
| 136 | tBodyGyro-sma() | tBodyGyro-sma() | tBodyGyro-SMA-1 |
| 137 | tBodyGyro-energy()-X | tBodyGyro-energy()-X | tBodyGyro-Energy-1 |
| 138 | tBodyGyro-energy()-Y | tBodyGyro-energy()-Y | tBodyGyro-Energy-2 |
| 139 | tBodyGyro-energy()-Z | tBodyGyro-energy()-Z | tBodyGyro-Energy-3 |
| 140 | tBodyGyro-iqr()-X | tBodyGyro-iqr()-X | tBodyGyro-IQR-1 |
| 141 | tBodyGyro-iqr()-Y | tBodyGyro-iqr()-Y | tBodyGyro-IQR-2 |
| 142 | tBodyGyro-iqr()-Z | tBodyGyro-iqr()-Z | tBodyGyro-IQR-3 |
| 143 | tBodyGyro-entropy()-X | tBodyGyro-entropy()-X | tBodyGyro-ropy-1 |
| 144 | tBodyGyro-entropy()-Y | tBodyGyro-entropy()-Y | tBodyGyro-ropy-1 |
| 145 | tBodyGyro-entropy()-Z | tBodyGyro-entropy()-Z | tBodyGyro-ropy-1 |
| 146 | tBodyGyro-arCoeff()-X,1 | tBodyGyro-arCoeff()-X,1 | tBodyGyro-ARCoeff-1 |
| 147 | tBodyGyro-arCoeff()-X,2 | tBodyGyro-arCoeff()-X,2 | tBodyGyro-ARCoeff-2 |
| 148 | tBodyGyro-arCoeff()-X,3 | tBodyGyro-arCoeff()-X,3 | tBodyGyro-ARCoeff-3 |
| 149 | tBodyGyro-arCoeff()-X,4 | tBodyGyro-arCoeff()-X,4 | tBodyGyro-ARCoeff-4 |
| 150 | tBodyGyro-arCoeff()-Y,1 | tBodyGyro-arCoeff()-Y,1 | tBodyGyro-ARCoeff-5 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 151 | tBodyGyro-arCoeff()-Y,2 | tBodyGyro-arCoeff()-Y,2 | tBodyGyro-ARCoeff-6 |
| 152 | tBodyGyro-arCoeff()-Y,3 | tBodyGyro-arCoeff()-Y,3 | tBodyGyro-ARCoeff-7 |
| 153 | tBodyGyro-arCoeff()-Y,4 | tBodyGyro-arCoeff()-Y,4 | tBodyGyro-ARCoeff-8 |
| 154 | tBodyGyro-arCoeff()-Z,1 | tBodyGyro-arCoeff()-Z,1 | tBodyGyro-ARCoeff-9 |
| 155 | tBodyGyro-arCoeff()-Z,2 | tBodyGyro-arCoeff()-Z,2 | tBodyGyro-ARCoeff-10 |
| 156 | tBodyGyro-arCoeff()-Z,3 | tBodyGyro-arCoeff()-Z,3 | tBodyGyro-ARCoeff-11 |
| 157 | tBodyGyro-arCoeff()-Z,4 | tBodyGyro-arCoeff()-Z,4 | tBodyGyro-ARCoeff-12 |
| 158 | tBodyGyro-correlation()-X,Y | tBodyGyro-correlation()-X,Y | tBodyGyro-Correlation-1 |
| 159 | tBodyGyro-correlation()-X,Z | tBodyGyro-correlation()-X,Z | tBodyGyro-Correlation-2 |
| 160 | tBodyGyro-correlation()-Y,Z | tBodyGyro-correlation()-Y,Z | tBodyGyro-Correlation-3 |
| 161 | tBodyGyroJerk-mean()-X | tBodyGyroJerk-mean()-X | tBodyGyroJerk-Mean-1 |
| 162 | tBodyGyroJerk-mean()-Y | tBodyGyroJerk-mean()-Y | tBodyGyroJerk-Mean-2 |
| 163 | tBodyGyroJerk-mean()-Z | tBodyGyroJerk-mean()-Z | tBodyGyroJerk-Mean-3 |
| 164 | tBodyGyroJerk-std()-X | tBodyGyroJerk-std()-X | tBodyGyroJerk-STD-1 |
| 165 | tBodyGyroJerk-std()-Y | tBodyGyroJerk-std()-Y | tBodyGyroJerk-STD-2 |
| 166 | tBodyGyroJerk-std()-Z | tBodyGyroJerk-std()-Z | tBodyGyroJerk-STD-3 |
| 167 | tBodyGyroJerk-mad()-X | tBodyGyroJerk-mad()-X | tBodyGyroJerk-Mad-1 |
| 168 | tBodyGyroJerk-mad()-Y | tBodyGyroJerk-mad()-Y | tBodyGyroJerk-Mad-2 |
| 169 | tBodyGyroJerk-mad()-Z | tBodyGyroJerk-mad()-Z | tBodyGyroJerk-Mad-3 |
| 170 | tBodyGyroJerk-max()-X | tBodyGyroJerk-max()-X | tBodyGyroJerk-Max-1 |
| 171 | tBodyGyroJerk-max()-Y | tBodyGyroJerk-max()-Y | tBodyGyroJerk-Max-2 |
| 172 | tBodyGyroJerk-max()-Z | tBodyGyroJerk-max()-Z | tBodyGyroJerk-Max-3 |
| 173 | tBodyGyroJerk-min()-X | tBodyGyroJerk-min()-X | tBodyGyroJerk-Min-1 |
| 174 | tBodyGyroJerk-min()-Y | tBodyGyroJerk-min()-Y | tBodyGyroJerk-Min-2 |
| 175 | tBodyGyroJerk-min()-Z | tBodyGyroJerk-min()-Z | tBodyGyroJerk-Min-3 |
| 176 | tBodyGyroJerk-sma() | tBodyGyroJerk-sma() | tBodyGyroJerk-SMA-1 |
| 177 | tBodyGyroJerk-energy()-X | tBodyGyroJerk-energy()-X | tBodyGyroJerk-Energy-1 |
| 178 | tBodyGyroJerk-energy()-Y | tBodyGyroJerk-energy()-Y | tBodyGyroJerk-Energy-2 |
| 179 | tBodyGyroJerk-energy()-Z | tBodyGyroJerk-energy()-Z | tBodyGyroJerk-Energy-3 |
| 180 | tBodyGyroJerk-iqr()-X | tBodyGyroJerk-iqr()-X | tBodyGyroJerk-IQR-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 181 | tBodyGyroJerk-iqr()-Y | tBodyGyroJerk-iqr()-Y | tBodyGyroJerk-IQR-2 |
| 182 | tBodyGyroJerk-iqr()-Z | tBodyGyroJerk-iqr()-Z | tBodyGyroJerk-IQR-3 |
| 183 | tBodyGyroJerk-entropy()-X | tBodyGyroJerk-entropy()-X | tBodyGyroJerk-ropy-1 |
| 184 | tBodyGyroJerk-entropy()-Y | tBodyGyroJerk-entropy()-Y | tBodyGyroJerk-ropy-1 |
| 185 | tBodyGyroJerk-entropy()-Z | tBodyGyroJerk-entropy()-Z | tBodyGyroJerk-ropy-1 |
| 186 | tBodyGyroJerk-arCoeff()-X,1 | tBodyGyroJerk-arCoeff()-X,1 | tBodyGyroJerk-ARCoeff-1 |
| 187 | tBodyGyroJerk-arCoeff()-X,2 | tBodyGyroJerk-arCoeff()-X,2 | tBodyGyroJerk-ARCoeff-2 |
| 188 | tBodyGyroJerk-arCoeff()-X,3 | tBodyGyroJerk-arCoeff()-X,3 | tBodyGyroJerk-ARCoeff-3 |
| 189 | tBodyGyroJerk-arCoeff()-X,4 | tBodyGyroJerk-arCoeff()-X,4 | tBodyGyroJerk-ARCoeff-4 |
| 190 | tBodyGyroJerk-arCoeff()-Y,1 | tBodyGyroJerk-arCoeff()-Y,1 | tBodyGyroJerk-ARCoeff-5 |
| 191 | tBodyGyroJerk-arCoeff()-Y,2 | tBodyGyroJerk-arCoeff()-Y,2 | tBodyGyroJerk-ARCoeff-6 |
| 192 | tBodyGyroJerk-arCoeff()-Y,3 | tBodyGyroJerk-arCoeff()-Y,3 | tBodyGyroJerk-ARCoeff-7 |
| 193 | tBodyGyroJerk-arCoeff()-Y,4 | tBodyGyroJerk-arCoeff()-Y,4 | tBodyGyroJerk-ARCoeff-8 |
| 194 | tBodyGyroJerk-arCoeff()-Z,1 | tBodyGyroJerk-arCoeff()-Z,1 | tBodyGyroJerk-ARCoeff-9 |
| 195 | tBodyGyroJerk-arCoeff()-Z,2 | tBodyGyroJerk-arCoeff()-Z,2 | tBodyGyroJerk-ARCoeff-10 |
| 196 | tBodyGyroJerk-arCoeff()-Z,3 | tBodyGyroJerk-arCoeff()-Z,3 | tBodyGyroJerk-ARCoeff-11 |
| 197 | tBodyGyroJerk-arCoeff()-Z,4 | tBodyGyroJerk-arCoeff()-Z,4 | tBodyGyroJerk-ARCoeff-12 |
| 198 | tBodyGyroJerk-correlation()-X,Y | tBodyGyroJerk-correlation()-X,Y | tBodyGyroJerk-Correlation-1 |
| 199 | tBodyGyroJerk-correlation()-X,Z | tBodyGyroJerk-correlation()-X,Z | tBodyGyroJerk-Correlation-2 |
| 200 | tBodyGyroJerk-correlation()-Y,Z | tBodyGyroJerk-correlation()-Y,Z | tBodyGyroJerk-Correlation-3 |
| 201 | tBodyAccMag-mean() | tBodyAccMag-mean() | tBodyAccMag-Mean-1 |
| 202 | tBodyAccMag-std() | tBodyAccMag-std() | tBodyAccMag-STD-1 |
| 203 | tBodyAccMag-mad() | tBodyAccMag-mad() | tBodyAccMag-Mad-1 |
| 204 | tBodyAccMag-max() | tBodyAccMag-max() | tBodyAccMag-Max-1 |
| 205 | tBodyAccMag-min() | tBodyAccMag-min() | tBodyAccMag-Min-1 |
| 206 | tBodyAccMag-sma() | tBodyAccMag-sma() | tBodyAccMag-SMA-1 |
| 207 | tBodyAccMag-energy() | tBodyAccMag-energy() | tBodyAccMag-Energy-1 |
| 208 | tBodyAccMag-iqr() | tBodyAccMag-iqr() | tBodyAccMag-IQR-1 |
| 209 | tBodyAccMag-entropy() | tBodyAccMag-entropy() | tBodyAccMag-ropy-1 |
| 210 | tBodyAccMag-arCoeff()1 | tBodyAccMag-arCoeff()1 | tBodyAccMag-ARCoeff-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 211 | tBodyAccMag-arCoeff()2 | tBodyAccMag-arCoeff()2 | tBodyAccMag-ARCoeff-2 |
| 212 | tBodyAccMag-arCoeff()3 | tBodyAccMag-arCoeff()3 | tBodyAccMag-ARCoeff-3 |
| 213 | tBodyAccMag-arCoeff()4 | tBodyAccMag-arCoeff()4 | tBodyAccMag-ARCoeff-4 |
| 214 | tGravityAccMag-mean() | tGravityAccMag-mean() | tGravityAccMag-Mean-1 |
| 215 | tGravityAccMag-std() | tGravityAccMag-std() | tGravityAccMag-STD-1 |
| 216 | tGravityAccMag-mad() | tGravityAccMag-mad() | tGravityAccMag-Mad-1 |
| 217 | tGravityAccMag-max() | tGravityAccMag-max() | tGravityAccMag-Max-1 |
| 218 | tGravityAccMag-min() | tGravityAccMag-min() | tGravityAccMag-Min-1 |
| 219 | tGravityAccMag-sma() | tGravityAccMag-sma() | tGravityAccMag-SMA-1 |
| 220 | tGravityAccMag-energy() | tGravityAccMag-energy() | tGravityAccMag-Energy-1 |
| 221 | tGravityAccMag-iqr() | tGravityAccMag-iqr() | tGravityAccMag-IQR-1 |
| 222 | tGravityAccMag-entropy() | tGravityAccMag-entropy() | tGravityAccMag-ropy-1 |
| 223 | tGravityAccMag-arCoeff()1 | tGravityAccMag-arCoeff()1 | tGravityAccMag-ARCoeff1 |
| 224 | tGravityAccMag-arCoeff()2 | tGravityAccMag-arCoeff()2 | tGravityAccMag-ARCoeff2 |
| 225 | tGravityAccMag-arCoeff()3 | tGravityAccMag-arCoeff()3 | tGravityAccMag-ARCoeff3 |
| 226 | tGravityAccMag-arCoeff()4 | tGravityAccMag-arCoeff()4 | tGravityAccMag-ARCoeff4 |
| 227 | tBodyAccJerkMag-mean() | tBodyAccJerkMag-mean() | tBodyAccJerkMag-Mean-1 |
| 228 | tBodyAccJerkMag-std() | tBodyAccJerkMag-std() | tBodyAccJerkMag-STD-1 |
| 229 | tBodyAccJerkMag-mad() | tBodyAccJerkMag-mad() | tBodyAccJerkMag-Mad-1 |
| 230 | tBodyAccJerkMag-max() | tBodyAccJerkMag-max() | tBodyAccJerkMag-Max-1 |
| 231 | tBodyAccJerkMag-min() | tBodyAccJerkMag-min() | tBodyAccJerkMag-Min-1 |
| 232 | tBodyAccJerkMag-sma() | tBodyAccJerkMag-sma() | tBodyAccJerkMag-SMA-1 |
| 233 | tBodyAccJerkMag-energy() | tBodyAccJerkMag-energy() | tBodyAccJerkMag-Energy1 |
| 234 | tBodyAccJerkMag-iqr() | tBodyAccJerkMag-iqr() | tBodyAccJerkMag-IQR-1 |
| 235 | tBodyAccJerkMag-entropy() | tBodyAccJerkMag-entropy() | tBodyAccJerkMag-ropy-1 |
| 236 | tBodyAccJerkMag-arCoeff()1 | tBodyAccJerkMag-arCoeff()1 | tBodyAccJerkMag-ARCoeff-1 |
| 237 | tBodyAccJerkMag-arCoeff()2 | tBodyAccJerkMag-arCoeff()2 | tBodyAccJerkMag-ARCoeff-2 |
| 238 | tBodyAccJerkMag-arCoeff()3 | tBodyAccJerkMag-arCoeff()3 | tBodyAccJerkMag-ARCoeff-3 |
| 239 | tBodyAccJerkMag-arCoeff()4 | tBodyAccJerkMag-arCoeff()4 | tBodyAccJerkMag-ARCoeff-4 |
| 240 | tBodyGyroMag-mean() | tBodyGyroMag-mean() | tBodyGyroMag-Mean-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 241 | tBodyGyroMag-std() | tBodyGyroMag-std() | tBodyGyroMag-STD-1 |
| 242 | tBodyGyroMag-mad() | tBodyGyroMag-mad() | tBodyGyroMag-Mad-1 |
| 243 | tBodyGyroMag-max() | tBodyGyroMag-max() | tBodyGyroMag-Max-1 |
| 244 | tBodyGyroMag-min() | tBodyGyroMag-min() | tBodyGyroMag-Min-1 |
| 245 | tBodyGyroMag-sma() | tBodyGyroMag-sma() | tBodyGyroMag-SMA-1 |
| 246 | tBodyGyroMag-energy() | tBodyGyroMag-energy() | tBodyGyroMag-Energy-1 |
| 247 | tBodyGyroMag-iqr() | tBodyGyroMag-iqr() | tBodyGyroMag-IQR-1 |
| 248 | tBodyGyroMag-entropy() | tBodyGyroMag-entropy() | tBodyGyroMag-ropy-1 |
| 249 | tBodyGyroMag-arCoeff()1 | tBodyGyroMag-arCoeff()1 | tBodyGyroMag-ARCoeff-1 |
| 250 | tBodyGyroMag-arCoeff()2 | tBodyGyroMag-arCoeff()2 | tBodyGyroMag-ARCoeff-2 |
| 251 | tBodyGyroMag-arCoeff()3 | tBodyGyroMag-arCoeff()3 | tBodyGyroMag-ARCoeff-3 |
| 252 | tBodyGyroMag-arCoeff()4 | tBodyGyroMag-arCoeff()4 | tBodyGyroMag-ARCoeff-4 |
| 253 | tBodyGyroJerkMag-mean() | tBodyGyroJerkMag-mean() | tBodyGyroJerkMag-Mean-1 |
| 254 | tBodyGyroJerkMag-std() | tBodyGyroJerkMag-std() | tBodyGyroJerkMag-STD-1 |
| 255 | tBodyGyroJerkMag-mad() | tBodyGyroJerkMag-mad() | tBodyGyroJerkMag-Mad-1 |
| 256 | tBodyGyroJerkMag-max() | tBodyGyroJerkMag-max() | tBodyGyroJerkMag-Max-1 |
| 257 | tBodyGyroJerkMag-min() | tBodyGyroJerkMag-min() | tBodyGyroJerkMag-Min-1 |
| 258 | tBodyGyroJerkMag-sma() | tBodyGyroJerkMag-sma() | tBodyGyroJerkMag-SMA-1 |
| 259 | tBodyGyroJerkMag-energy() | tBodyGyroJerkMag-energy() | tBodyGyroJerkMagEnergy1 |
| 260 | tBodyGyroJerkMag-iqr() | tBodyGyroJerkMag-iqr() | tBodyGyroJerkMag-IQR-1 |
| 261 | tBodyGyroJerkMag-entropy() | tBodyGyroJerkMag-entropy() | tBodyGyroJerkMag-ropy-1 |
| 262 | tBodyGyroJerkMag-arCoeff()1 | tBodyGyroJerkMag-arCoeff()1 | tBodyGyroJerkMag-ARCoeff-1 |
| 263 | tBodyGyroJerkMag-arCoeff()2 | tBodyGyroJerkMag-arCoeff()2 | tBodyGyroJerkMag-ARCoeff-2 |
| 264 | tBodyGyroJerkMag-arCoeff()3 | tBodyGyroJerkMag-arCoeff()3 | tBodyGyroJerkMag-ARCoeff-3 |
| 265 | tBodyGyroJerkMag-arCoeff()4 | tBodyGyroJerkMag-arCoeff()4 | tBodyGyroJerkMag-ARCoeff-4 |
| 266 | fBodyAcc-mean()-X | fBodyAcc-mean()-X | fBodyAcc-Mean-1 |
| 267 | fBodyAcc-mean()-Y | fBodyAcc-mean()-Y | fBodyAcc-Mean-2 |
| 268 | fBodyAcc-mean()-Z | fBodyAcc-mean()-Z | fBodyAcc-Mean-3 |
| 269 | fBodyAcc-std()-X | fBodyAcc-std()-X | fBodyAcc-STD-1 |
| 270 | fBodyAcc-std()-Y | fBodyAcc-std()-Y | fBodyAcc-STD-2 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 271 | fBodyAcc-std()-Z | fBodyAcc-std()-Z | fBodyAcc-STD-3 |
| 272 | fBodyAcc-mad()-X | fBodyAcc-mad()-X | fBodyAcc-Mad-1 |
| 273 | fBodyAcc-mad()-Y | fBodyAcc-mad()-Y | fBodyAcc-Mad-2 |
| 274 | fBodyAcc-mad()-Z | fBodyAcc-mad()-Z | fBodyAcc-Mad-3 |
| 275 | fBodyAcc-max()-X | fBodyAcc-max()-X | fBodyAcc-Max-1 |
| 276 | fBodyAcc-max()-Y | fBodyAcc-max()-Y | fBodyAcc-Max-2 |
| 277 | fBodyAcc-max()-Z | fBodyAcc-max()-Z | fBodyAcc-Max-3 |
| 278 | fBodyAcc-min()-X | fBodyAcc-min()-X | fBodyAcc-Min-1 |
| 279 | fBodyAcc-min()-Y | fBodyAcc-min()-Y | fBodyAcc-Min-2 |
| 280 | fBodyAcc-min()-Z | fBodyAcc-min()-Z | fBodyAcc-Min-3 |
| 281 | fBodyAcc-sma() | fBodyAcc-sma() | fBodyAcc-SMA-1 |
| 282 | fBodyAcc-energy()-X | fBodyAcc-energy()-X | fBodyAcc-Energy-1 |
| 283 | fBodyAcc-energy()-Y | fBodyAcc-energy()-Y | fBodyAcc-Energy-2 |
| 284 | fBodyAcc-energy()-Z | fBodyAcc-energy()-Z | fBodyAcc-Energy-3 |
| 285 | fBodyAcc-iqr()-X | fBodyAcc-iqr()-X | fBodyAcc-IQR-1 |
| 286 | fBodyAcc-iqr()-Y | fBodyAcc-iqr()-Y | fBodyAcc-IQR-2 |
| 287 | fBodyAcc-iqr()-Z | fBodyAcc-iqr()-Z | fBodyAcc-IQR-3 |
| 288 | fBodyAcc-entropy()-X | fBodyAcc-entropy()-X | fBodyAcc-ropy-1 |
| 289 | fBodyAcc-entropy()-Y | fBodyAcc-entropy()-Y | fBodyAcc-ropy-1 |
| 290 | fBodyAcc-entropy()-Z | fBodyAcc-entropy()-Z | fBodyAcc-ropy-1 |
| 291 | fBodyAcc-maxInds-X | fBodyAcc-maxInds-X | fBodyAcc-MaxInds-1 |
| 292 | fBodyAcc-maxInds-Y | fBodyAcc-maxInds-Y | fBodyAcc-MaxInds-2 |
| 293 | fBodyAcc-maxInds-Z | fBodyAcc-maxInds-Z | fBodyAcc-MaxInds-3 |
| 294 | fBodyAcc-meanFreq()-X | fBodyAcc-meanFreq()-X | fBodyAcc-MeanFreq-1 |
| 295 | fBodyAcc-meanFreq()-Y | fBodyAcc-meanFreq()-Y | fBodyAcc-MeanFreq-2 |
| 296 | fBodyAcc-meanFreq()-Z | fBodyAcc-meanFreq()-Z | fBodyAcc-MeanFreq-3 |
| 297 | fBodyAcc-skewness()-X | fBodyAcc-skewness()-X | fBodyAcc-Skewness-1 |
| 298 | fBodyAcc-kurtosis()-X | fBodyAcc-kurtosis()-X | fBodyAcc-Kurtosis-1 |
| 299 | fBodyAcc-skewness()-Y | fBodyAcc-skewness()-Y | fBodyAcc-Skewness-1 |
| 300 | fBodyAcc-kurtosis()-Y | fBodyAcc-kurtosis()-Y | fBodyAcc-Kurtosis-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 301 | fBodyAcc-skewness()-Z | fBodyAcc-skewness()-Z | fBodyAcc-Skewness-1 |
| 302 | fBodyAcc-kurtosis()-Z | fBodyAcc-kurtosis()-Z | fBodyAcc-Kurtosis-1 |
| 303 | fBodyAcc-bandsEnergy()-1,8 | fBodyAcc-bandsEnergy()-1,8 | fBodyAcc-BandsEnergyOld1 |
| 304 | fBodyAcc-bandsEnergy()-9,16 | fBodyAcc-bandsEnergy()-9,16 | fBodyAccBandsEnergyOld2 |
| 305 | fBodyAcc-bandsEnergy()-17,24 | fBodyAcc-bandsEnergy()-17,24 | fBodyAccBandsEnergyOld3 |
| 306 | fBodyAcc-bandsEnergy()-25,32 | fBodyAcc-bandsEnergy()-25,32 | fBodyAccBandsEnergyOld4 |
| 307 | fBodyAcc-bandsEnergy()-33,40 | fBodyAcc-bandsEnergy()-33,40 | fBodyAccBandsEnergyOld5 |
| 308 | fBodyAcc-bandsEnergy()-41,48 | fBodyAcc-bandsEnergy()-41,48 | fBodyAccBandsEnergyOld6 |
| 309 | fBodyAcc-bandsEnergy()-49,56 | fBodyAcc-bandsEnergy()-49,56 | fBodyAccBandsEnergyOld7 |
| 310 | fBodyAcc-bandsEnergy()-57,64 | fBodyAcc-bandsEnergy()-57,64 | fBodyAccBandsEnergyOld8 |
| 311 | fBodyAcc-bandsEnergy()-1,16 | fBodyAcc-bandsEnergy()-1,16 | fBodyAcc-BandsEnergyOld-9 |
| 312 | fBodyAcc-bandsEnergy()-17,32 | fBodyAcc-bandsEnergy()-17,32 | fBodyAcc-BandsEnergyOld-10 |
| 313 | fBodyAcc-bandsEnergy()-33,48 | fBodyAcc-bandsEnergy()-33,48 | fBodyAcc-BandsEnergyOld-11 |
| 314 | fBodyAcc-bandsEnergy()-49,64 | fBodyAcc-bandsEnergy()-49,64 | fBodyAcc-BandsEnergyOld-12 |
| 315 | fBodyAcc-bandsEnergy()-1,24 | fBodyAcc-bandsEnergy()-1,24 | fBodyAcc-BandsEnergyOld-13 |
| 316 | fBodyAcc-bandsEnergy()-25,48 | fBodyAcc-bandsEnergy()-25,48 | fBodyAcc-BandsEnergyOld-14 |
| 317 | fBodyAcc-bandsEnergy()-1,8 | fBodyAcc-bandsEnergy()-1,8 | fBodyAcc-BandsEnergyOld-15 |
| 318 | fBodyAcc-bandsEnergy()-9,16 | fBodyAcc-bandsEnergy()-9,16 | fBodyAcc-BandsEnergyOld-16 |
| 319 | fBodyAcc-bandsEnergy()-17,24 | fBodyAcc-bandsEnergy()-17,24 | fBodyAcc-BandsEnergyOld-17 |
| 320 | fBodyAcc-bandsEnergy()-25,32 | fBodyAcc-bandsEnergy()-25,32 | fBodyAcc-BandsEnergyOld-18 |
| 321 | fBodyAcc-bandsEnergy()-33,40 | fBodyAcc-bandsEnergy()-33,40 | fBodyAcc-BandsEnergyOld-19 |
| 322 | fBodyAcc-bandsEnergy()-41,48 | fBodyAcc-bandsEnergy()-41,48 | fBodyAcc-BandsEnergyOld-20 |
| 323 | fBodyAcc-bandsEnergy()-49,56 | fBodyAcc-bandsEnergy()-49,56 | fBodyAcc-BandsEnergyOld-21 |
| 324 | fBodyAcc-bandsEnergy()-57,64 | fBodyAcc-bandsEnergy()-57,64 | fBodyAcc-BandsEnergyOld-22 |
| 325 | fBodyAcc-bandsEnergy()-1,16 | fBodyAcc-bandsEnergy()-1,16 | fBodyAcc-BandsEnergyOld-23 |
| 326 | fBodyAcc-bandsEnergy()-17,32 | fBodyAcc-bandsEnergy()-17,32 | fBodyAcc-BandsEnergyOld-24 |
| 327 | fBodyAcc-bandsEnergy()-33,48 | fBodyAcc-bandsEnergy()-33,48 | fBodyAcc-BandsEnergyOld-25 |
| 328 | fBodyAcc-bandsEnergy()-49,64 | fBodyAcc-bandsEnergy()-49,64 | fBodyAcc-BandsEnergyOld-26 |
| 329 | fBodyAcc-bandsEnergy()-1,24 | fBodyAcc-bandsEnergy()-1,24 | fBodyAcc-BandsEnergyOld-27 |
| 330 | fBodyAcc-bandsEnergy()-25,48 | fBodyAcc-bandsEnergy()-25,48 | fBodyAcc-BandsEnergyOld-28 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 331 | fBodyAcc-bandsEnergy()-1,8 | fBodyAcc-bandsEnergy()-1,8 | fBodyAcc-BandsEnergyOld-29 |
| 332 | fBodyAcc-bandsEnergy()-9,16 | fBodyAcc-bandsEnergy()-9,16 | fBodyAcc-BandsEnergyOld-30 |
| 333 | fBodyAcc-bandsEnergy()-17,24 | fBodyAcc-bandsEnergy()-17,24 | fBodyAcc-BandsEnergyOld-31 |
| 334 | fBodyAcc-bandsEnergy()-25,32 | fBodyAcc-bandsEnergy()-25,32 | fBodyAcc-BandsEnergyOld-32 |
| 335 | fBodyAcc-bandsEnergy()-33,40 | fBodyAcc-bandsEnergy()-33,40 | fBodyAcc-BandsEnergyOld-33 |
| 336 | fBodyAcc-bandsEnergy()-41,48 | fBodyAcc-bandsEnergy()-41,48 | fBodyAcc-BandsEnergyOld-34 |
| 337 | fBodyAcc-bandsEnergy()-49,56 | fBodyAcc-bandsEnergy()-49,56 | fBodyAcc-BandsEnergyOld-35 |
| 338 | fBodyAcc-bandsEnergy()-57,64 | fBodyAcc-bandsEnergy()-57,64 | fBodyAcc-BandsEnergyOld-36 |
| 339 | fBodyAcc-bandsEnergy()-1,16 | fBodyAcc-bandsEnergy()-1,16 | fBodyAcc-BandsEnergyOld-37 |
| 340 | fBodyAcc-bandsEnergy()-17,32 | fBodyAcc-bandsEnergy()-17,32 | fBodyAcc-BandsEnergyOld-38 |
| 341 | fBodyAcc-bandsEnergy()-33,48 | fBodyAcc-bandsEnergy()-33,48 | fBodyAcc-BandsEnergyOld-39 |
| 342 | fBodyAcc-bandsEnergy()-49,64 | fBodyAcc-bandsEnergy()-49,64 | fBodyAcc-BandsEnergyOld-40 |
| 343 | fBodyAcc-bandsEnergy()-1,24 | fBodyAcc-bandsEnergy()-1,24 | fBodyAcc-BandsEnergyOld-41 |
| 344 | fBodyAcc-bandsEnergy()-25,48 | fBodyAcc-bandsEnergy()-25,48 | fBodyAcc-BandsEnergyOld-42 |
| 345 | fBodyAccJerk-mean()-X | fBodyAccJerk-mean()-X | fBodyAccJerk-Mean-1 |
| 346 | fBodyAccJerk-mean()-Y | fBodyAccJerk-mean()-Y | fBodyAccJerk-Mean-2 |
| 347 | fBodyAccJerk-mean()-Z | fBodyAccJerk-mean()-Z | fBodyAccJerk-Mean-3 |
| 348 | fBodyAccJerk-std()-X | fBodyAccJerk-std()-X | fBodyAccJerk-STD-1 |
| 349 | fBodyAccJerk-std()-Y | fBodyAccJerk-std()-Y | fBodyAccJerk-STD-2 |
| 350 | fBodyAccJerk-std()-Z | fBodyAccJerk-std()-Z | fBodyAccJerk-STD-3 |
| 351 | fBodyAccJerk-mad()-X | fBodyAccJerk-mad()-X | fBodyAccJerk-Mad-1 |
| 352 | fBodyAccJerk-mad()-Y | fBodyAccJerk-mad()-Y | fBodyAccJerk-Mad-2 |
| 353 | fBodyAccJerk-mad()-Z | fBodyAccJerk-mad()-Z | fBodyAccJerk-Mad-3 |
| 354 | fBodyAccJerk-max()-X | fBodyAccJerk-max()-X | fBodyAccJerk-Max-1 |
| 355 | fBodyAccJerk-max()-Y | fBodyAccJerk-max()-Y | fBodyAccJerk-Max-2 |
| 356 | fBodyAccJerk-max()-Z | fBodyAccJerk-max()-Z | fBodyAccJerk-Max-3 |
| 357 | fBodyAccJerk-min()-X | fBodyAccJerk-min()-X | fBodyAccJerk-Min-1 |
| 358 | fBodyAccJerk-min()-Y | fBodyAccJerk-min()-Y | fBodyAccJerk-Min-2 |
| 359 | fBodyAccJerk-min()-Z | fBodyAccJerk-min()-Z | fBodyAccJerk-Min-3 |
| 360 | fBodyAccJerk-sma() | fBodyAccJerk-sma() | fBodyAccJerk-SMA-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 361 | fBodyAccJerk-energy()-X | fBodyAccJerk-energy()-X | fBodyAccJerk-Energy-1 |
| 362 | fBodyAccJerk-energy()-Y | fBodyAccJerk-energy()-Y | fBodyAccJerk-Energy-2 |
| 363 | fBodyAccJerk-energy()-Z | fBodyAccJerk-energy()-Z | fBodyAccJerk-Energy-3 |
| 364 | fBodyAccJerk-iqr()-X | fBodyAccJerk-iqr()-X | fBodyAccJerk-IQR-1 |
| 365 | fBodyAccJerk-iqr()-Y | fBodyAccJerk-iqr()-Y | fBodyAccJerk-IQR-2 |
| 366 | fBodyAccJerk-iqr()-Z | fBodyAccJerk-iqr()-Z | fBodyAccJerk-IQR-3 |
| 367 | fBodyAccJerk-entropy()-X | fBodyAccJerk-entropy()-X | fBodyAccJerk-ropy-1 |
| 368 | fBodyAccJerk-entropy()-Y | fBodyAccJerk-entropy()-Y | fBodyAccJerk-ropy-1 |
| 369 | fBodyAccJerk-entropy()-Z | fBodyAccJerk-entropy()-Z | fBodyAccJerk-ropy-1 |
| 370 | fBodyAccJerk-maxInds-X | fBodyAccJerk-maxInds-X | fBodyAccJerk-MaxInds-1 |
| 371 | fBodyAccJerk-maxInds-Y | fBodyAccJerk-maxInds-Y | fBodyAccJerk-MaxInds-2 |
| 372 | fBodyAccJerk-maxInds-Z | fBodyAccJerk-maxInds-Z | fBodyAccJerk-MaxInds-3 |
| 373 | fBodyAccJerk-meanFreq()-X | fBodyAccJerk-meanFreq()-X | fBodyAccJerk-MeanFreq-1 |
| 374 | fBodyAccJerk-meanFreq()-Y | fBodyAccJerk-meanFreq()-Y | fBodyAccJerk-MeanFreq-2 |
| 375 | fBodyAccJerk-meanFreq()-Z | fBodyAccJerk-meanFreq()-Z | fBodyAccJerk-MeanFreq-3 |
| 376 | fBodyAccJerk-skewness()-X | fBodyAccJerk-skewness()-X | fBodyAccJerk-Skewness-1 |
| 377 | fBodyAccJerk-kurtosis()-X | fBodyAccJerk-kurtosis()-X | fBodyAccJerk-Kurtosis-1 |
| 378 | fBodyAccJerk-skewness()-Y | fBodyAccJerk-skewness()-Y | fBodyAccJerk-Skewness-1 |
| 379 | fBodyAccJerk-kurtosis()-Y | fBodyAccJerk-kurtosis()-Y | fBodyAccJerk-Kurtosis-1 |
| 380 | fBodyAccJerk-skewness()-Z | fBodyAccJerk-skewness()-Z | fBodyAccJerk-Skewness-1 |
| 381 | fBodyAccJerk-kurtosis()-Z | fBodyAccJerk-kurtosis()-Z | fBodyAccJerk-Kurtosis-1 |
| 382 | fBodyAccJerk-bandsEnergy()-1,8 | fBodyAccJerk-bandsEnergy()-1,8 | fBodyAccJerk-BandsEnergyOld-1 |
| 383 | fBodyAccJerk-bandsEnergy()-9,16 | fBodyAccJerk-bandsEnergy()-9,16 | fBodyAccJerk-BandsEnergyOld-2 |
| 384 | fBodyAccJerk-bandsEnergy()-17,24 | fBodyAccJerk-bandsEnergy()-17,24 | fBodyAccJerk-BandsEnergyOld-3 |
| 385 | fBodyAccJerk-bandsEnergy()-25,32 | fBodyAccJerk-bandsEnergy()-25,32 | fBodyAccJerk-BandsEnergyOld-4 |
| 386 | fBodyAccJerk-bandsEnergy()-33,40 | fBodyAccJerk-bandsEnergy()-33,40 | fBodyAccJerk-BandsEnergyOld-5 |
| 387 | fBodyAccJerk-bandsEnergy()-41,48 | fBodyAccJerk-bandsEnergy()-41,48 | fBodyAccJerk-BandsEnergyOld-6 |
| 388 | fBodyAccJerk-bandsEnergy()-49,56 | fBodyAccJerk-bandsEnergy()-49,56 | fBodyAccJerk-BandsEnergyOld-7 |
| 389 | fBodyAccJerk-bandsEnergy()-57,64 | fBodyAccJerk-bandsEnergy()-57,64 | fBodyAccJerk-BandsEnergyOld-8 |
| 390 | fBodyAccJerk-bandsEnergy()-1,16 | fBodyAccJerk-bandsEnergy()-1,16 | fBodyAccJerk-BandsEnergyOld-9 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 391 | fBodyAccJerk-bandsEnergy()-17,32 | fBodyAccJerk-bandsEnergy()-17,32 | fBodyAccJerk-BandsEnergyOld10 |
| 392 | fBodyAccJerk-bandsEnergy()-33,48 | fBodyAccJerk-bandsEnergy()-33,48 | fBodyAccJerk-BandsEnergyOld11 |
| 393 | fBodyAccJerk-bandsEnergy()-49,64 | fBodyAccJerk-bandsEnergy()-49,64 | fBodyAccJerk-BandsEnergyOld12 |
| 394 | fBodyAccJerk-bandsEnergy()-1,24 | fBodyAccJerk-bandsEnergy()-1,24 | fBodyAccJerk-BandsEnergyOld13 |
| 395 | fBodyAccJerk-bandsEnergy()-25,48 | fBodyAccJerk-bandsEnergy()-25,48 | fBodyAccJerk-BandsEnergyOld14 |
| 396 | fBodyAccJerk-bandsEnergy()-1,8 | fBodyAccJerk-bandsEnergy()-1,8 | fBodyAccJerk-BandsEnergyOld15 |
| 397 | fBodyAccJerk-bandsEnergy()-9,16 | fBodyAccJerk-bandsEnergy()-9,16 | fBodyAccJerk-BandsEnergyOld16 |
| 398 | fBodyAccJerk-bandsEnergy()-17,24 | fBodyAccJerk-bandsEnergy()-17,24 | fBodyAccJerk-BandsEnergyOld17 |
| 399 | fBodyAccJerk-bandsEnergy()-25,32 | fBodyAccJerk-bandsEnergy()-25,32 | fBodyAccJerk-BandsEnergyOld18 |
| 400 | fBodyAccJerk-bandsEnergy()-33,40 | fBodyAccJerk-bandsEnergy()-33,40 | fBodyAccJerk-BandsEnergyOld19 |
| 401 | fBodyAccJerk-bandsEnergy()-41,48 | fBodyAccJerk-bandsEnergy()-41,48 | fBodyAccJerk-BandsEnergyOld20 |
| 402 | fBodyAccJerk-bandsEnergy()-49,56 | fBodyAccJerk-bandsEnergy()-49,56 | fBodyAccJerk-BandsEnergyOld21 |
| 403 | fBodyAccJerk-bandsEnergy()-57,64 | fBodyAccJerk-bandsEnergy()-57,64 | fBodyAccJerk-BandsEnergyOld22 |
| 404 | fBodyAccJerk-bandsEnergy()-1,16 | fBodyAccJerk-bandsEnergy()-1,16 | fBodyAccJerk-BandsEnergyOld23 |
| 405 | fBodyAccJerk-bandsEnergy()-17,32 | fBodyAccJerk-bandsEnergy()-17,32 | fBodyAccJerk-BandsEnergyOld24 |
| 406 | fBodyAccJerk-bandsEnergy()-33,48 | fBodyAccJerk-bandsEnergy()-33,48 | fBodyAccJerk-BandsEnergyOld25 |
| 407 | fBodyAccJerk-bandsEnergy()-49,64 | fBodyAccJerk-bandsEnergy()-49,64 | fBodyAccJerk-BandsEnergyOld26 |
| 408 | fBodyAccJerk-bandsEnergy()-1,24 | fBodyAccJerk-bandsEnergy()-1,24 | fBodyAccJerk-BandsEnergyOld27 |
| 409 | fBodyAccJerk-bandsEnergy()-25,48 | fBodyAccJerk-bandsEnergy()-25,48 | fBodyAccJerk-BandsEnergyOld28 |
| 410 | fBodyAccJerk-bandsEnergy()-1,8 | fBodyAccJerk-bandsEnergy()-1,8 | fBodyAccJerk-BandsEnergyOld29 |
| 411 | fBodyAccJerk-bandsEnergy()-9,16 | fBodyAccJerk-bandsEnergy()-9,16 | fBodyAccJerk-BandsEnergyOld30 |
| 412 | fBodyAccJerk-bandsEnergy()-17,24 | fBodyAccJerk-bandsEnergy()-17,24 | fBodyAccJerk-BandsEnergyOld31 |
| 413 | fBodyAccJerk-bandsEnergy()-25,32 | fBodyAccJerk-bandsEnergy()-25,32 | fBodyAccJerk-BandsEnergyOld32 |
| 414 | fBodyAccJerk-bandsEnergy()-33,40 | fBodyAccJerk-bandsEnergy()-33,40 | fBodyAccJerk-BandsEnergyOld33 |
| 415 | fBodyAccJerk-bandsEnergy()-41,48 | fBodyAccJerk-bandsEnergy()-41,48 | fBodyAccJerk-BandsEnergyOld34 |
| 416 | fBodyAccJerk-bandsEnergy()-49,56 | fBodyAccJerk-bandsEnergy()-49,56 | fBodyAccJerk-BandsEnergyOld35 |
| 417 | fBodyAccJerk-bandsEnergy()-57,64 | fBodyAccJerk-bandsEnergy()-57,64 | fBodyAccJerk-BandsEnergyOld36 |
| 418 | fBodyAccJerk-bandsEnergy()-1,16 | fBodyAccJerk-bandsEnergy()-1,16 | fBodyAccJerk-BandsEnergyOld37 |
| 419 | fBodyAccJerk-bandsEnergy()-17,32 | fBodyAccJerk-bandsEnergy()-17,32 | fBodyAccJerk-BandsEnergyOld38 |
| 420 | fBodyAccJerk-bandsEnergy()-33,48 | fBodyAccJerk-bandsEnergy()-33,48 | fBodyAccJerk-BandsEnergyOld39 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 421 | fBodyAccJerk-bandsEnergy()-49,64 | fBodyAccJerk-bandsEnergy()-49,64 | fBodyAccJerk-BandsEnergyOld40 |
| 422 | fBodyAccJerk-bandsEnergy()-1,24 | fBodyAccJerk-bandsEnergy()-1,24 | fBodyAccJerk-BandsEnergyOld41 |
| 423 | fBodyAccJerk-bandsEnergy()-25,48 | fBodyAccJerk-bandsEnergy()-25,48 | fBodyAccJerk-BandsEnergyOld42 |
| 424 | fBodyGyro-mean()-X | fBodyGyro-mean()-X | fBodyGyro-Mean-1 |
| 425 | fBodyGyro-mean()-Y | fBodyGyro-mean()-Y | fBodyGyro-Mean-2 |
| 426 | fBodyGyro-mean()-Z | fBodyGyro-mean()-Z | fBodyGyro-Mean-3 |
| 427 | fBodyGyro-std()-X | fBodyGyro-std()-X | fBodyGyro-STD-1 |
| 428 | fBodyGyro-std()-Y | fBodyGyro-std()-Y | fBodyGyro-STD-2 |
| 429 | fBodyGyro-std()-Z | fBodyGyro-std()-Z | fBodyGyro-STD-3 |
| 430 | fBodyGyro-mad()-X | fBodyGyro-mad()-X | fBodyGyro-Mad-1 |
| 431 | fBodyGyro-mad()-Y | fBodyGyro-mad()-Y | fBodyGyro-Mad-2 |
| 432 | fBodyGyro-mad()-Z | fBodyGyro-mad()-Z | fBodyGyro-Mad-3 |
| 433 | fBodyGyro-max()-X | fBodyGyro-max()-X | fBodyGyro-Max-1 |
| 434 | fBodyGyro-max()-Y | fBodyGyro-max()-Y | fBodyGyro-Max-2 |
| 435 | fBodyGyro-max()-Z | fBodyGyro-max()-Z | fBodyGyro-Max-3 |
| 436 | fBodyGyro-min()-X | fBodyGyro-min()-X | fBodyGyro-Min-1 |
| 437 | fBodyGyro-min()-Y | fBodyGyro-min()-Y | fBodyGyro-Min-2 |
| 438 | fBodyGyro-min()-Z | fBodyGyro-min()-Z | fBodyGyro-Min-3 |
| 439 | fBodyGyro-sma() | fBodyGyro-sma() | fBodyGyro-SMA-1 |
| 440 | fBodyGyro-energy()-X | fBodyGyro-energy()-X | fBodyGyro-Energy-1 |
| 441 | fBodyGyro-energy()-Y | fBodyGyro-energy()-Y | fBodyGyro-Energy-2 |
| 442 | fBodyGyro-energy()-Z | fBodyGyro-energy()-Z | fBodyGyro-Energy-3 |
| 443 | fBodyGyro-iqr()-X | fBodyGyro-iqr()-X | fBodyGyro-IQR-1 |
| 444 | fBodyGyro-iqr()-Y | fBodyGyro-iqr()-Y | fBodyGyro-IQR-2 |
| 445 | fBodyGyro-iqr()-Z | fBodyGyro-iqr()-Z | fBodyGyro-IQR-3 |
| 446 | fBodyGyro-entropy()-X | fBodyGyro-entropy()-X | fBodyGyro-ropy-1 |
| 447 | fBodyGyro-entropy()-Y | fBodyGyro-entropy()-Y | fBodyGyro-ropy-1 |
| 448 | fBodyGyro-entropy()-Z | fBodyGyro-entropy()-Z | fBodyGyro-ropy-1 |
| 449 | fBodyGyro-maxInds-X | fBodyGyro-maxInds-X | fBodyGyro-MaxInds-1 |
| 450 | fBodyGyro-maxInds-Y | fBodyGyro-maxInds-Y | fBodyGyro-MaxInds-2 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 451 | fBodyGyro-maxInds-Z | fBodyGyro-maxInds-Z | fBodyGyro-MaxInds-3 |
| 452 | fBodyGyro-meanFreq()-X | fBodyGyro-meanFreq()-X | fBodyGyro-MeanFreq-1 |
| 453 | fBodyGyro-meanFreq()-Y | fBodyGyro-meanFreq()-Y | fBodyGyro-MeanFreq-2 |
| 454 | fBodyGyro-meanFreq()-Z | fBodyGyro-meanFreq()-Z | fBodyGyro-MeanFreq-3 |
| 455 | fBodyGyro-skewness()-X | fBodyGyro-skewness()-X | fBodyGyro-Skewness-1 |
| 456 | fBodyGyro-kurtosis()-X | fBodyGyro-kurtosis()-X | fBodyGyro-Kurtosis-1 |
| 457 | fBodyGyro-skewness()-Y | fBodyGyro-skewness()-Y | fBodyGyro-Skewness-1 |
| 458 | fBodyGyro-kurtosis()-Y | fBodyGyro-kurtosis()-Y | fBodyGyro-Kurtosis-1 |
| 459 | fBodyGyro-skewness()-Z | fBodyGyro-skewness()-Z | fBodyGyro-Skewness-1 |
| 460 | fBodyGyro-kurtosis()-Z | fBodyGyro-kurtosis()-Z | fBodyGyro-Kurtosis-1 |
| 461 | fBodyGyro-bandsEnergy()-1,8 | fBodyGyro-bandsEnergy()-1,8 | fBodyGyro-BandsEnergyOld-1 |
| 462 | fBodyGyro-bandsEnergy()-9,16 | fBodyGyro-bandsEnergy()-9,16 | fBodyGyro-BandsEnergyOld-2 |
| 463 | fBodyGyro-bandsEnergy()-17,24 | fBodyGyro-bandsEnergy()-17,24 | fBodyGyro-BandsEnergyOld-3 |
| 464 | fBodyGyro-bandsEnergy()-25,32 | fBodyGyro-bandsEnergy()-25,32 | fBodyGyro-BandsEnergyOld-4 |
| 465 | fBodyGyro-bandsEnergy()-33,40 | fBodyGyro-bandsEnergy()-33,40 | fBodyGyro-BandsEnergyOld-5 |
| 466 | fBodyGyro-bandsEnergy()-41,48 | fBodyGyro-bandsEnergy()-41,48 | fBodyGyro-BandsEnergyOld-6 |
| 467 | fBodyGyro-bandsEnergy()-49,56 | fBodyGyro-bandsEnergy()-49,56 | fBodyGyro-BandsEnergyOld-7 |
| 468 | fBodyGyro-bandsEnergy()-57,64 | fBodyGyro-bandsEnergy()-57,64 | fBodyGyro-BandsEnergyOld-8 |
| 469 | fBodyGyro-bandsEnergy()-1,16 | fBodyGyro-bandsEnergy()-1,16 | fBodyGyro-BandsEnergyOld-9 |
| 470 | fBodyGyro-bandsEnergy()-17,32 | fBodyGyro-bandsEnergy()-17,32 | fBodyGyro-BandsEnergyOld-10 |
| 471 | fBodyGyro-bandsEnergy()-33,48 | fBodyGyro-bandsEnergy()-33,48 | fBodyGyro-BandsEnergyOld-11 |
| 472 | fBodyGyro-bandsEnergy()-49,64 | fBodyGyro-bandsEnergy()-49,64 | fBodyGyro-BandsEnergyOld-12 |
| 473 | fBodyGyro-bandsEnergy()-1,24 | fBodyGyro-bandsEnergy()-1,24 | fBodyGyro-BandsEnergyOld-13 |
| 474 | fBodyGyro-bandsEnergy()-25,48 | fBodyGyro-bandsEnergy()-25,48 | fBodyGyro-BandsEnergyOld-14 |
| 475 | fBodyGyro-bandsEnergy()-1,8 | fBodyGyro-bandsEnergy()-1,8 | fBodyGyro-BandsEnergyOld-15 |
| 476 | fBodyGyro-bandsEnergy()-9,16 | fBodyGyro-bandsEnergy()-9,16 | fBodyGyro-BandsEnergyOld-16 |
| 477 | fBodyGyro-bandsEnergy()-17,24 | fBodyGyro-bandsEnergy()-17,24 | fBodyGyro-BandsEnergyOld-17 |
| 478 | fBodyGyro-bandsEnergy()-25,32 | fBodyGyro-bandsEnergy()-25,32 | fBodyGyro-BandsEnergyOld-18 |
| 479 | fBodyGyro-bandsEnergy()-33,40 | fBodyGyro-bandsEnergy()-33,40 | fBodyGyro-BandsEnergyOld-19 |
| 480 | fBodyGyro-bandsEnergy()-41,48 | fBodyGyro-bandsEnergy()-41,48 | fBodyGyro-BandsEnergyOld-20 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 481 | fBodyGyro-bandsEnergy()-49,56 | fBodyGyro-bandsEnergy()-49,56 | fBodyGyro-BandsEnergyOld-21 |
| 482 | fBodyGyro-bandsEnergy()-57,64 | fBodyGyro-bandsEnergy()-57,64 | fBodyGyro-BandsEnergyOld-22 |
| 483 | fBodyGyro-bandsEnergy()-1,16 | fBodyGyro-bandsEnergy()-1,16 | fBodyGyro-BandsEnergyOld-23 |
| 484 | fBodyGyro-bandsEnergy()-17,32 | fBodyGyro-bandsEnergy()-17,32 | fBodyGyro-BandsEnergyOld-24 |
| 485 | fBodyGyro-bandsEnergy()-33,48 | fBodyGyro-bandsEnergy()-33,48 | fBodyGyro-BandsEnergyOld-25 |
| 486 | fBodyGyro-bandsEnergy()-49,64 | fBodyGyro-bandsEnergy()-49,64 | fBodyGyro-BandsEnergyOld-26 |
| 487 | fBodyGyro-bandsEnergy()-1,24 | fBodyGyro-bandsEnergy()-1,24 | fBodyGyro-BandsEnergyOld-27 |
| 488 | fBodyGyro-bandsEnergy()-25,48 | fBodyGyro-bandsEnergy()-25,48 | fBodyGyro-BandsEnergyOld-28 |
| 489 | fBodyGyro-bandsEnergy()-1,8 | fBodyGyro-bandsEnergy()-1,8 | fBodyGyro-BandsEnergyOld-29 |
| 490 | fBodyGyro-bandsEnergy()-9,16 | fBodyGyro-bandsEnergy()-9,16 | fBodyGyro-BandsEnergyOld-30 |
| 491 | fBodyGyro-bandsEnergy()-17,24 | fBodyGyro-bandsEnergy()-17,24 | fBodyGyro-BandsEnergyOld-31 |
| 492 | fBodyGyro-bandsEnergy()-25,32 | fBodyGyro-bandsEnergy()-25,32 | fBodyGyro-BandsEnergyOld-32 |
| 493 | fBodyGyro-bandsEnergy()-33,40 | fBodyGyro-bandsEnergy()-33,40 | fBodyGyro-BandsEnergyOld-33 |
| 494 | fBodyGyro-bandsEnergy()-41,48 | fBodyGyro-bandsEnergy()-41,48 | fBodyGyro-BandsEnergyOld-34 |
| 495 | fBodyGyro-bandsEnergy()-49,56 | fBodyGyro-bandsEnergy()-49,56 | fBodyGyro-BandsEnergyOld-35 |
| 496 | fBodyGyro-bandsEnergy()-57,64 | fBodyGyro-bandsEnergy()-57,64 | fBodyGyro-BandsEnergyOld-36 |
| 497 | fBodyGyro-bandsEnergy()-1,16 | fBodyGyro-bandsEnergy()-1,16 | fBodyGyro-BandsEnergyOld-37 |
| 498 | fBodyGyro-bandsEnergy()-17,32 | fBodyGyro-bandsEnergy()-17,32 | fBodyGyro-BandsEnergyOld-38 |
| 499 | fBodyGyro-bandsEnergy()-33,48 | fBodyGyro-bandsEnergy()-33,48 | fBodyGyro-BandsEnergyOld-39 |
| 500 | fBodyGyro-bandsEnergy()-49,64 | fBodyGyro-bandsEnergy()-49,64 | fBodyGyro-BandsEnergyOld-40 |
| 501 | fBodyGyro-bandsEnergy()-1,24 | fBodyGyro-bandsEnergy()-1,24 | fBodyGyro-BandsEnergyOld-41 |
| 502 | fBodyGyro-bandsEnergy()-25,48 | fBodyGyro-bandsEnergy()-25,48 | fBodyGyro-BandsEnergyOld-42 |
| 503 | fBodyAccMag-mean() | fBodyAccMag-mean() | fBodyAccMag-Mean-1 |
| 504 | fBodyAccMag-std() | fBodyAccMag-std() | fBodyAccMag-STD-1 |
| 505 | fBodyAccMag-mad() | fBodyAccMag-mad() | fBodyAccMag-Mad-1 |
| 506 | fBodyAccMag-max() | fBodyAccMag-max() | fBodyAccMag-Max-1 |
| 507 | fBodyAccMag-min() | fBodyAccMag-min() | fBodyAccMag-Min-1 |
| 508 | fBodyAccMag-sma() | fBodyAccMag-sma() | fBodyAccMag-SMA-1 |
| 509 | fBodyAccMag-energy() | fBodyAccMag-energy() | fBodyAccMag-Energy-1 |
| 510 | fBodyAccMag-iqr() | fBodyAccMag-iqr() | fBodyAccMag-IQR-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 511 | fBodyAccMag-entropy() | fBodyAccMag-entropy() | fBodyAccMag-ropy-1 |
| 512 | fBodyAccMag-maxInds | fBodyAccMag-maxInds | fBodyAccMag-MaxInds-1 |
| 513 | fBodyAccMag-meanFreq() | fBodyAccMag-meanFreq() | fBodyAccMag-MeanFreq-1 |
| 514 | fBodyAccMag-skewness() | fBodyAccMag-skewness() | fBodyAccMag-Skewness-1 |
| 515 | fBodyAccMag-kurtosis() | fBodyAccMag-kurtosis() | fBodyAccMag-Kurtosis-1 |
| 516 | fBodyBodyAccJerkMag-mean() | fBodyBodyAccJerkMag-mean() | fBodyAccJerkMag-Mean-1 |
| 517 | fBodyBodyAccJerkMag-std() | fBodyBodyAccJerkMag-std() | fBodyAccJerkMag-STD-1 |
| 518 | fBodyBodyAccJerkMag-mad() | fBodyBodyAccJerkMag-mad() | fBodyAccJerkMag-Mad-1 |
| 519 | fBodyBodyAccJerkMag-max() | fBodyBodyAccJerkMag-max() | fBodyAccJerkMag-Max-1 |
| 520 | fBodyBodyAccJerkMag-min() | fBodyBodyAccJerkMag-min() | fBodyAccJerkMag-Min-1 |
| 521 | fBodyBodyAccJerkMag-sma() | fBodyBodyAccJerkMag-sma() | fBodyAccJerkMag-SMA-1 |
| 522 | fBodyBodyAccJerkMag-energy() | fBodyBodyAccJerkMag-energy() | fBodyAccJerkMag-Energy-1 |
| 523 | fBodyBodyAccJerkMag-iqr() | fBodyBodyAccJerkMag-iqr() | fBodyAccJerkMag-IQR-1 |
| 524 | fBodyBodyAccJerkMag-entropy() | fBodyBodyAccJerkMag-entropy() | fBodyAccJerkMag-ropy-1 |
| 525 | fBodyBodyAccJerkMag-maxInds | fBodyBodyAccJerkMag-maxInds | fBodyAccJerkMag-MaxInds-1 |
| 526 | fBodyBodyAccJerkMag-meanFreq() | fBodyBodyAccJerkMag-meanFreq() | fBodyAccJerkMag-MeanFreq-1 |
| 527 | fBodyBodyAccJerkMag-skewness() | fBodyBodyAccJerkMag-skewness() | fBodyAccJerkMag-Skewness-1 |
| 528 | fBodyBodyAccJerkMag-kurtosis() | fBodyBodyAccJerkMag-kurtosis() | fBodyAccJerkMag-Kurtosis-1 |
| 529 | fBodyBodyGyroMag-mean() | fBodyBodyGyroMag-mean() | fBodyGyroMag-Mean-1 |
| 530 | fBodyBodyGyroMag-std() | fBodyBodyGyroMag-std() | fBodyGyroMag-STD-1 |
| 531 | fBodyBodyGyroMag-mad() | fBodyBodyGyroMag-mad() | fBodyGyroMag-Mad-1 |
| 532 | fBodyBodyGyroMag-max() | fBodyBodyGyroMag-max() | fBodyGyroMag-Max-1 |
| 533 | fBodyBodyGyroMag-min() | fBodyBodyGyroMag-min() | fBodyGyroMag-Min-1 |
| 534 | fBodyBodyGyroMag-sma() | fBodyBodyGyroMag-sma() | fBodyGyroMag-SMA-1 |
| 535 | fBodyBodyGyroMag-energy() | fBodyBodyGyroMag-energy() | fBodyGyroMag-Energy-1 |
| 536 | fBodyBodyGyroMag-iqr() | fBodyBodyGyroMag-iqr() | fBodyGyroMag-IQR-1 |
| 537 | fBodyBodyGyroMag-entropy() | fBodyBodyGyroMag-entropy() | fBodyGyroMag-ropy-1 |
| 538 | fBodyBodyGyroMag-maxInds | fBodyBodyGyroMag-maxInds | fBodyGyroMag-MaxInds-1 |
| 539 | fBodyBodyGyroMag-meanFreq() | fBodyBodyGyroMag-meanFreq() | fBodyGyroMag-MeanFreq-1 |
| 540 | fBodyBodyGyroMag-skewness() | fBodyBodyGyroMag-skewness() | fBodyGyroMag-Skewness-1 |

| No. | FEATURE-NAMES | | |
|---|---|---|---|
| | UCI-HAR | DATASET_UCI | HAPT |
| 541 | fBodyBodyGyroMag-kurtosis() | fBodyBodyGyroMag-kurtosis() | fBodyGyroMag-Kurtosis-1 |
| 542 | fBodyBodyGyroJerkMag-mean() | fBodyBodyGyroJerkMag-mean() | fBodyGyroJerkMag-Mean-1 |
| 543 | fBodyBodyGyroJerkMag-std() | fBodyBodyGyroJerkMag-std() | fBodyGyroJerkMag-STD-1 |
| 544 | fBodyBodyGyroJerkMag-mad() | fBodyBodyGyroJerkMag-mad() | fBodyGyroJerkMag-Mad-1 |
| 545 | fBodyBodyGyroJerkMag-max() | fBodyBodyGyroJerkMag-max() | fBodyGyroJerkMag-Max-1 |
| 546 | fBodyBodyGyroJerkMag-min() | fBodyBodyGyroJerkMag-min() | fBodyGyroJerkMag-Min-1 |
| 547 | fBodyBodyGyroJerkMag-sma() | fBodyBodyGyroJerkMag-sma() | fBodyGyroJerkMag-SMA-1 |
| 548 | fBodyBodyGyroJerkMag-energy() | fBodyBodyGyroJerkMag-energy() | fBodyGyroJerkMag-Energy-1 |
| 549 | fBodyBodyGyroJerkMag-iqr() | fBodyBodyGyroJerkMag-iqr() | fBodyGyroJerkMag-IQR-1 |
| 550 | fBodyBodyGyroJerkMag-entropy() | fBodyBodyGyroJerkMag-entropy() | fBodyGyroJerkMag-ropy-1 |
| 551 | fBodyBodyGyroJerkMag-maxInds | fBodyBodyGyroJerkMag-maxInds | fBodyGyroJerkMag-MaxInds-1 |
| 552 | fBodyBodyGyroJerkMag-meanFreq() | fBodyBodyGyroJerkMag-meanFreq() | fBodyGyroJerkMag-MeanFreq-1 |
| 553 | fBodyBodyGyroJerkMag-skewness() | fBodyBodyGyroJerkMag-skewness() | fBodyGyroJerkMag-Skewness-1 |
| 554 | fBodyBodyGyroJerkMag-kurtosis() | fBodyBodyGyroJerkMag-kurtosis() | fBodyGyroJerkMag-Kurtosis-1 |
| 555 | angle(tBodyAccMean,gravity) | angle(tBodyAccMean,gravity) | tBodyAcc-AngleWRTGravity-1 |
| 556 | angle(tBodyAccJerkMean),gravityMean | angle(tBodyAccJerkMean),gravityMean | tBodyAccJerkAngleWRTGravity1 |
| 557 | angle(tBodyGyroMean,gravityMean) | angle(tBodyGyroMean,gravityMean) | tBodyGyro-AngleWRTGravity-1 |
| 558 | angle(tBodyGyroJerkMean,gravityMean | angle(tBodyGyroJerkMean,gravityMean | tBodyGyroJerkAngleWRTGravity1 |
| 559 | angle(X,gravityMean) | angle(X,gravityMean) | tXAxisAcc-AngleWRTGravity-1 |
| 560 | angle(Y,gravityMean) | angle(Y,gravityMean) | tYAxisAcc-AngleWRTGravity-1 |
| 561 | angle(Z,gravityMean) | angle(Z,gravityMean) | tZAxisAcc-AngleWRTGravity-1 |

## REFERENCES


[1] S. Gonz, M. Chen, S. Member, and V. C. M. Leung, "A Survey on Activity Detection and Classification Using Wearable Sensors," *Technology*, vol. 15, no. 4, pp. 539–549, 2017.

[2] D. J. Cook, "Pervasive Computing at Scale : Transforming the State of the Art," *Mobile Computing*, vol. 8, pp. 22–35, 2012.

[3] L. Chen *et al.*, "Sensor-based Activity Recognition," *IEEE Transactions on Systems, Man*, and *Cybernetics. Part C Applied Review*, vol. 42, no. 6, pp. 790–808, 2012.

[4] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey," *architecture* of *computer system*. *(ARCS), 2010 23rd International Conference*, pp. 1–10, 2010.

[5] O. Durmaz and I. Mustafa, "A Review and Taxonomy of Activity Recognition on Mobile Phones," *Bionanoscience*, vol.3, pp. 145-171, June, 2013.

[6] D. Acharjee, A. Mukherjee, J. K. Mandal, and N. Mukherjee, "Activity recognition system using inbuilt sensors of smart mobile phone and minimizing feature vectors," *Microsystem Technologies.*, vol. 22, no. 11, pp. 2715–2722, Nov. 2016.

[7] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors.," *IEEE Communication Survey and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[8] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Survey*, vol. 46, no. 3, pp. 1–33, 2014.

[9] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," *Proc. - BSN 2006 Int. Work. Wearable Implant. Body Sens. Networks*, pp. 113–116, 2006.

[10] M. Arif, M. Bilal, and A. Kattan, "Better Physical Activity Classification using Smartphone Acceleration Sensor," *Journal of Medical System,* vol. 38, pp.1–10, 2014.

[11] I. Suarez, A. Jahn, C. Anderson, and K. David, "Improved Activity Recognition by Using Enriched Acceleration Data," *Proceeding: 2015 ACM international* joint *conference* on *pervasive* and *ubiquitous computing*, pp. 1011–1015, 2015.

[12] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity Recognition using Cell Phone Accelerometers," *ACM SIGKDD Explorer Newsletter*, vol.12, no.2, pp. 74–82, 2010.

[13] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Tr, "Wearable Sensing to Annotate Meeting Recordings," *Personal and Ubiquitous Computing*, vol. 7, pp. 263–274, 2003.

[14] A. Avci, S. Bosch, M. Marin-perianu, R. Marin-perianu, and P. Havinga, "Activity Recognition Using Inertial Sensing for Healthcare , Wellbeing and Sports Applications : A Survey Activity Recognition Using Inertial Sensing for Healthcare , Wellbeing and Sports Applications : A Survey," *ACM Computing Survey*, pp.167-176, no.23, 2010.

[15] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003.

[16] I. Guyon, "An Introduction to Variable and Feature Selection 1 Introduction," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[17] M. Ramaswami and R. Bhaskaran, "A Study on Feature Selection Techniques in Educational Data Mining," *Journal of Computing,* vol. 1, no. 1, pp. 7–11, 2009.

[18] E. Kim, S. Helal, and D. Cook, "Human Activity Recognition and Pattern Discovery," *Pervasive Computing IEEE*, vol. 9, no. 1, pp. 48–53, 2010.

[19] X. Su, H. Tong, and P. Ji, "Activity recognition with smartphone sensors," *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 235–249, 2014.

[20] L. Zhen and L. Qiong, "A New Feature Selection Method for Internet Traffic Classification Using ML," *Procedial Physics: International Conference on Medical Physics, Biomedical physics and Engineering*, vol. 33, pp. 1338–1345, 2012.

[21] W. WinMyo, P. Aiyarak, and W. Wettayaprasit, "A Noble Feature Selection Method for Human Activity Recognition using Linearly Dependent Concept

( LDC)," in *the 7th International Conference on Software and Computer Applications*, pp. 173–177, 2018.

[22] M. B. del Rosario, S. J. Redmond, and N. H. Lovell, "Tracking the evolution of smartphone sensing for monitoring human movement," *Sensors (Switzerland)*, vol. 15, no. 8, pp. 18901–18933, 2015.

[23] B. Najafi, K. Aminian, A. Paraschiv-ionescu, F. Loew, and C. J. Büla, "Ambulatory System for Human Motion Analysis Using a Kinematic Sensor : Monitoring of Daily Physical Activity in the Elderly," *IEEE Transition and Biomedical Engineering*, vol. 50, no.6, pp. 711-723, 2014, 2003.

[24] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks : The state of the art," *International Journal of Forecasting*, vol. 14, pp. 35–62, 1998.

[25] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," *Proceeding: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning,* pp. 21-26, 2013.

[26] J.-L. Reyes-Ortiz, L. Oneto, A. Sama, X. Parra, and D. A. I.-O. http://orcid. org/000.-0002-4943-3021 Anguita, "Transition-Aware Human Activity Recognition using smartphones.," *Neurocomputing An International Journal*, vol. 171, pp. 754–767, 2016.

[27] E. De-la-hoz-franco, P. Ariza-colpas, and J. M. Quero, "Sensor-based datasets for Human Activity Recognition – A Systematic Review of Literature," *IEEE Access,* vol. 6, pp. 59192-59120, October, 2018.

[28] O. D. Lara and M. a. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," *IEEE Communication: Survey and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[29] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "A Survey of Online Activity Recognition Using Mobile Phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, Jan. 2015.

[30] G. D. Abowd, A. Dey, R. Orr, and J. Brotherton, "Context-awareness in wearable and ubiquitous computing," *ACM Computing Survey*, pp. 179-184, Oct1997,

[31] L. Rizzo, P. Dondio, S. J. Delany, and L. Longo, "Modeling Mental Workload Via Rule-Based Expert System : A Comparison with NASA-TLX and Workload Profile Modeling mental workload via rule-based expert system : a comparison with NASA-TLX & Workload Profile," *12th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI),* vol. 475, p-215-229 September 2017.

[32] P. T. Kirisci, "A Wearable Computing Prototype for supporting training activities in Automotive Production A Wearable Computing Prototype for supporting training activities in Automotive Production," *Engineering,* vol.1 pp. 1-13, May 2014.

[33] L. Wang, "Recognition of Human Activities with Wearable Sensors," *EURASIP Journal of Advance Signal Processing*, vol. 2012, no. 108, pp. 1–13, 2012.

[34] N. Oukrich, E. B. Cherraqi, and A. Maach, "Human Daily Activity Recognition Using Neural Networks and Ontology-Based Activity Representation," *ACM Computing Survey*, pp. 622–633, 2018.

[35] D. Das Dawn and S. H. Shaikh, "A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector," *Visual Computer*, vol. 32, no. 3, pp. 289–306, 2016.

[36] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *ACM Computing Survey*, pp. 1–17, 2004.

[37] M. Lee, J. Kim, K. Kim, I. Lee, S. H. Jee, and S. K. Yoo, "Physical Activity Recognition Using a Single Tri-Axis Accelerometer," *ACM Computing Survey*, vol. I, pp. 20–23, 2009.

[38] J. Kwapisz, G. Weiss, and S. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorer* vol. 12, no. 2, pp. 74–82, 2011.

[39] C. Torres-Huitzil and M. Nuno-Maganda, "Robust smartphone-based human activity recognition using a tri-axial accelerometer," *2015 IEEE 6th Latin American Symposium Circuits and System*, pp. 1–4, 2015.

[40] X. Heng, Z. Wang, and J. Wang, "Human activity recognition based on transformed accelerometer data from a mobile phone," *International Journal of Communication System*, vol. 29, no. 13, pp. 1981–1991, Sep. 2016.

[41] T. Brezmes, J. Gorricho, and J. Cotrina, “Activity Recognition from Accelerometer Data on a Activity Recognition from Accelerometer Data on a mobile phone,” *ACM Computing Survey*, pp. 796-799, 2015.

[42] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A Public Domain Dataset for Human Activity Recognition Using Smartphones,” *Proceeding: Europen Symposium on Artificial Neural Networks, Computer Intellegent and Maching Learning*, pp. 437–442, 2013.

[43] A. Olivares, J. Ramirez, J. M. Gorriz, G. Olivares, and M. Damas, “Detection of (In)activity Periods in Human Body Motion Using Inertial Sensors: A Comparative Study,” *Sensors*, vol. 12, no. 5, pp. 5791–5814, 2012.

[44] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook, “Simple and Complex Activity Recognition through Smartphones,” *Intelligent Environment (IE), 2012 8th Int. Conf.*, pp. 214–221, 2012.

[45] J. Binder, “Impact of Physical Fitness on Cognitive Performance in Patients at a Mesmory Clinic,” *Denebtia Open Acess,* March, vol. 9, no.1, p-129-135, 2019.

[46] M. B. May, “Physical activity : Evidence for the Anti-ageing effects in elderly Physical activity : Evidence for the Anti-ageing effects in elderly,” *African Union of Sport Medicine,* no. 3, pp. 29-33, January, 2018.

[47] J. K. Aggarwal and M. S. Ryoo, “Human Activity Analysis ," *ACM Computing Survey*,,” vol. 43, no. 3, 2011.

[48] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, D. Samaras, and S. Brook, “Two-person Interaction Detection Using Body-Pose Features and Multiple Instance Learning.” *Stony Brook,* pp. 16-21, *2017.*

[49] M. Derawi and P. Bours, “Gait and activity recognition using commercial phones,” *Computing Security*, vol. 39, pp. 137–144, 2013.

[50] S. A. Antos, M. V. Albert, and K. P. Kording, “Hand, belt, pocket or bag: Practical activity tracking with mobile phones,” *Journal of Neuroscience Methods*, vol. 231, pp. 22–30, Jul. 2014.

[51] M. G. Davis and K. R. Fox, “Physical activity patterns assessed by accelerometry in older people,” *Europen Journal Applied Physiology*, vol. 100, no. 5, pp. 581–589, 2007.

[52] S.W.Lichtman, K. Pisaraka, E. R. Berman, M.Pestone, H.Dowling, E. Offen "Discrepancy between self-reported and actual caloric intake and exercise in obese subjects," *New England Journal of Medicine,* vol. 327, no. 27, pp. 1893-1898, 1992.

[53] C. Voss, J. Sims-Gould, M. C. Ashe, H. A. McKay, C. Pugh, and M. Winters, "Public transit use and physical activity in community-dwelling older adults: Combining GPS and accelerometry to assess transportation-related physical activity," *J. Transport Health*, vol. 3, no. 2, pp. 191–199, 2016.

[54] A. D. Ignatov and V. V. Strijov, "Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer," *Multimedia Tools Application*, vol. 75, no. 12, pp. 7257–7270, 2016.

[55] "A Public Domain Dataset for Human Activity Recognition using Smartphones - Semantic Scholar." [Online]. Available: /paper/A-Public-Domain-Dataset-for-Human-Activity-Anguita Ghio/83de43bc849ad3d9579cc f540e6fe566 ef90a58e, [Accessed: 01-Dec-2016].

[56] D. N. Tran and D. D. Phan, "Human Activities Recognition in Android Smartphone Using Support Vector Machine," *Proceeding of the 2nd Intional Confefence Intelligent System for Model Simulation, ISMS*, pp. 64–68, 2017.

[57] P. Siirtola and J. Röning, "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data," *International Journal Interact Multimedia Artificient Intelligence*, vol. 1, no. 5, p. 38, 2012.

[58] M. Zhang and A. A. Sawchuk, "Usc-Had," *Proceeding: 2012 ACM Conference Ubiquitous Computing*, pp. 1036-1043, 2012.

[59] P. Casale, O. Pujol, and P. Radeva, "Personalization and user verification in wearable systems using biometric walking patterns," *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 563–580, 2012.

[60] A. Stisen *et al.*, "Smart Devices are Different: Assessing and MitigatingMobile Sensing Heterogeneities for Activity Recognition with RealWorld HAR Dataset," *Proceeding of 13th ACM Conference Embedded Networked Sensing System*, pp. 127–140, 2015.

[61] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell

phone accelerometers," *ACM SIGKDD Explorer News*, vol. 12, no. 2, p. 74, 2011.

[62] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: a new dataset for human activity recognition using acceleration data from smartphones," *Applied Science,*vol. 5*,* pp. 12-16, 2016.

[63] K. Tran, T. Le, and T. Dinh, "A high-accuracy step counting algorithm for iPhones using Accelerometer," *2012 IEEE International Symposium Signal Processing and Information Technology, ISSPIT 2012*, pp. 213–217, 2013.

[64] W. Jiang and Z. Yin, "Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks," *Proceeding on 23rd ACM International Conference of Multimedia*, pp. 1307–1310, 2015.

[65] A. A. L. Foudery, A. A. Alkandari, and N. M. Almutairi, "Trash basket sensor notification using arduino with android application," *Indonesia Journal of Electrical Engineering and Computer Science*, vol. 10, no. 1, pp. 120–128, 2018.

[66] M. Naqvi and a Kumar, "Step Counting Using Smartphone-Based Accelerometer," *International Journal on Computer Science and Engineering*, vol. 4, no. 05, pp. 675–681, 2012.

[67] X. Kang, B. Huang, and G. Qi, "A novel walking detection and step counting algorithm using unconstrained smartphones," *Sensors (Switzerland)*, Sensors, vol. 18, no. 1, 2018.

[68] P. Imunisasi, D. Pada, B. Di, and W. Kerja, "Step counting using smartphone accelerometer and fast fourier transform," *ACM survey*, vol. 8, no. 2, pp. 1–12, 2014.

[69] R. Cuberek, W. El Ansari, K. Fromel, K. Skalik, and E. Sigmund, "A comparison of two motion sensors for the assessment of free-living physical activity of adolescents," *International Journal of Environmental Reserch for Public Health*, vol. 7, no. 4, pp. 1558–1576, 2010.

[70] F. Gu, K. Khoshelham, J. Shang, F. Yu, and Z. Wei, "Robust and accurate smartphone-based step counting for indoor localization," *IEEE Sensor Journal*, vol. 17, no. 11, pp. 3453–3460, 2017.

[71] A. K. Siddanahalli Ninge Gowda, S. R. Babu, and D. C. Sekaran, "UMOISP:

Usage Mode and Orientation Invariant Smartphone Pedometer," *IEEE Sensor Journal*, vol. 17, no. 3, pp. 869–881, 2017.

[72] N. Ichinoseki-Sekine, Y. Kuwae, Y. Higashi, T. Fujimoto, M. Sekine, and T. Tamura, "Improving the accuracy of pedometer used by the elderly with the FFT algorithm," *Medical & Science in Sports & Exercise*, vol. 38, no. 9, pp. 1674–1681, 2006.

[73] N. A. Capela, E. D. Lemaire, and N. Baddour, "Novel algorithm for a smartphone-based 6-minute walk test application: algorithm, application development, and evaluation.," *Journal of Neuroengineering and Rehabilitation.*, vol. 12, no. 1, p. 19, 2015.

[74] C. J. Schneider, "A Natural Walking Monitor for Pulmonary Patients Using Mobile Phones," *Pop Culture as Everyday Life*, vol. 19, no. 4, pp. 47–56, 2016.

[75] P. Alto, "Selection of Relevant Features and Examples in Machine Learning." *Artificial Intelligent,* vol. 97., no.1, pp. 245-271, 1997.

[76] M. A. M. Hasan, M. Nasser, S. Ahmad, and K. I. Molla, "Feature Selection for Intrusion Detection Using Random Forest," *J. Inf. Secur.*, vol. 6, no. 3, pp. 1319–1324, 2013.

[77] W. N. Gansterer and G. F. Ecker, "On the Relationship Between Feature Selection and Classification Accuracy," pp. 90–105. *ACM Computing Survey*, June 2015.

[78] S. Beniwal and J. Arora, "Classification and Feature Selection Techniques in Data Mining," *International Journal of Engineering and Research techonology*, vol. 1, no. 6, pp. 1–6, 2012.

[79] Y.-P. Chen *et al.*, "A novel bacterial foraging optimization algorithm for feature selection," *Expert System ane Application*, vol. 83, pp. 1–17, 2017.

[80] R. Setiono, "Feature Selection : An Ever Evolving Frontier in Data Mining," *The Fourth Workshop on Feature Selection and Data Mining,* pp. 4–13.

[81] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Supervised Machine Learning: A Review of Classification Techniques," *Artificien Intelligent Review*, vol. 26, pp. 159–190, 2006.

[82] D. Duling, W. Thompson, and S. Schubert, "SAS Global Forum 2008 Data

Mining and Predictive Modeling SAS Global Forum 2008 Data Mining and Predictive Modeling," *SAS Globalization Forum*, pp. 1–12, 2008.

[83] N. Salehi-Moghaddami, H. S. Yazdi, and H. Poostchi, "Correlation based splitting criterionin multi branch decision tree," *Open Computer Science*, vol. 1, no. 2, pp. 205–220, 2011.

[84] R. X. Gao, "PCA-Based Feature Selection Scheme for Machine Defect Classification," *IEEE Transation ane Instrum Measurements*, vol. 53, no. 6, December, 2013.

[85] M. Arif, M. Bilal, A. Kattan, and S. I. Ahamed, "Better physical activity classification using smartphone acceleration sensor," *Journal of Medical System*, vol. 38, no. 9, 2014.

[86] Akashdeep, I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert System ane Application*, vol. 88, pp. 249–257, 2017.

[87] P. Mitra, C. D. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transation on Pattern Analysis and Machine Intelligent*, vol. 24, no. 3, pp. 301-312, 2002.

[88] L. Wang, "Recognition of Human Activities with Wearable Sensors," *EURASIP Journal on Advance Signal Processing*, vol. 2012, no. 108, pp. 1–13, 2012.

[89] L. Wang, Y. Wang, and Q. Chang, "Feature selection methods for big data bioinformatics: A survey from the search perspective," *Methods*, vol. 111, pp. 21–31, 2016.

[90] Y. Guo *et al.*, "Tensor Manifold Discriminant Projections for Acceleration-Based Human Activity Recognition," *IEEE Transational of Multimed.*, vol. 18, no. 10, pp. 1977–1987, 2016.

[91] Z. Zhang, J. Dong, X. Luo, K. S. Choi, and X. Wu, "Heartbeat classification using disease-specific feature selection," *Computing, Biolical Medicine*, vol. 46, no. 1, pp. 79–89, 2014.

[92] J. Derrac, S. Garca`, and F. Herrera, "IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule," *Pattern Recognition*, vol. 43, no. 6, pp. 2082–2105, 2010.

[93] M. Simão, P. Neto, and O. Gibaru, "Using data dimensionality reduction for recognition of incomplete dynamic gestures," *Pattern Recognition Letters*, vol. 99, pp. 32–38, 2017.

[94] J. Kersten, "Simultaneous feature selection and Gaussian mixture model estimation for supervised classification problems," *Pattern Recognition*, vol. 47, no. 8, pp. 2582–2595, 2014.

[95] 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 Computing System*, vol. 81, pp. 307–313, 2018.

[96] J. Howcroft, J. Kofman, and E. D. Lemaire, "Feature selection for elderly faller classification based on wearable sensors," *J. Neuroengineering ane Rehabilitation.*, vol. 14, no. 1, pp. 1–11, 2017.

[97] N. A. Capela, E. D. Lemaire, and N. Baddour, "Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients," *PLoS One*, vol. 10, no. 4, pp. 1–18, 2015.

[98] S. Beniwal and J. Arora, "Classification and Feature Selection Techniques in Data Mining," *International Journal of Engineering Research and Technologies (IJER*, vol. 1, no. 6, pp. 1–6, 2012.

[99] A. Gholami, H. Bonakdari, A. H. Zaji, D. G. Michelson, and A. A. Akhtari, "Improving the performance of multi-layer perceptron and radial basis function models with a decision tree model to predict flow variables in a sharp 90° bend," *Application on Soft Computing Journal*, vol. 48, pp. 563–583, 2016.

[100] Y. Kong, M. Owusu-Akomeah, H. A. Antwi, X. Hu, and P. Acheampong, "Evaluation of the robusticity of mutual fund performance in Ghana using Enhanced Resilient Backpropagation Neural Network (ERBPNN) and Fast Adaptive Neural Network Classifier (FANNC)," *Finance ane Innovatice*, vol. 5, no. 1, pp. 1–12, 2019.

[101] L. Wang, X. Zhao, Y. Si, L. Cao, and Y. Liu, "Context-associative Hierarchical Memory Model for Human Activity Recognition and Prediction," *IEEE Transational ane Multimedia*, vol. 20, no. 10, pp. 1–1, 2016.

[102] T. K. Padma Shri and N. Sriraam, "Spectral entropy feature subset selection

using SEPCOR to detect alcoholic impact on gamma sub band visual event related potentials of multichannel electroencephalograms (EEG)," *Application on Soft Computing Journal*, vol. 46, pp. 441–451, 2016.

[103] F. Sikder and D. Sarkar, "Log-Sum Distance Measures and Its Application to Human-Activity Monitoring and Recognition Using Data from Motion Sensors," *IEEE Sensors Jornal*, vol. 17, no. 14, pp. 4520-4533, 2017.

[104] L. Cao, Y. Wang, B. Zhang, Q. Jin, and A. V. Vasilakos, "GCHAR: An efficient Group-based Context-aware human activity recognition on smartphone," *Journal of Parallel Distribution and Computing*, vol. 18, no.1, pp. 67-80, 2018.

[105] D. Acharjee, S. P. Maity, and A. Mukherjee, "Hidden Markov model a tool for recognition of human contexts using sensors of smart mobile phone," *Microsystem Technology*, vol. 23, no. 3, pp. 571–582, August. 2016.

[106] M. B. Del Rosario *et al.*, "A comparison of activity classification in younger and older cohorts using a smartphone.," *Physiol Measurements*, vol. 35, no. 11, pp. 2269–86, 2014.

[107] X. Zhu and H. Qiu, "High Accuracy Human Activity Recognition Based on Sparse Locality Preserving Projections," *PLoS One*, vol. 11, no. 11, p.166-567, Nov. 2016.

[108] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *Journal of University Computer Science*, vol. 19, no. 9, pp. 1295–1314, 2013.

[109] F. Cruciani, I. Cleland, C. Nugent, P. McCullagh, K. Synnes, and J. Hallberg, "Automatic annotation for human activity recognition in free living using a smartphone," *Sensors (Switzerland)*, vol. 18, no. 7, pp. 1–20, 2018.

[110] N. A. Atasoy, B. Sen, and B. Selcuk, "Using gauss - Jordan elimination method with CUDA for linear circuit equation systems," *Procedia Technology*, vol. 1, no. 1, pp. 31–35, 2012.

[111] L. Valby, "Linear (In)dependence," *Linear Algebra*, pp. 1–4, 2012.

[112] E. Eriksen, "Linear Systems and Gaussian Elimination," *School of Engineering Science*, pp-1-34, 2011.

[113] J. a Gallian, J. G. Rainbolt, and B. Cole, "Abstract Algebra with GAP for

Contemporary'' *Abstract Algebra,* pp. 1-87, 2010.

[114] M. Eie and S.-T. Chang, "Cyclic Groups," *A Course Abstract Algebra*, pp. 53–64, 2012.

[115] A. Cyclic and G. Structures, "As Cyclic Group Structures," *Applied Descrete Structure,* Vol. 1.1, 2012.

[116] S. Spitzer, "Using Star Polygons to Understand Cyclic Group Structure," Mathematice, Musics, Art, Culture, pp. 479–480, 2012.

[117] M. M. El-farrah, "Expectation Numbers of Cyclic Groups," *TopSCHOLAR*, Vol, 1, pp. 1-49, 2015.

[118] H. Aktaş and Ş. Özlü, "Cyclic soft groups and their applications on groups," *Science World Journal*, vol. 2014, 2014.

[119] B. Ikenaga, "Almost cyclic groups," math.GR, vol.1, pp. 1–7, 2005.

[120] B. Sloan, "Multiplicative Groups," *Advance Science and Enviromental Science*, vol. I, pp. 1–9, 2003.

[121] W. WinMyo, W. Wettayaprasit, P. Aiyarak, "A cyclic Attribution Technique Feature Selection Method for Human Activity Recognition", *International Journal of Intelligent Systems and Applications (IJISA)*, vol.11, pp.25-32, 2019.

[122] W. WinMyo, W. Wettayaprasit, and P. Aiyarak, "Designing Classifier for Human Activity Recognition Using Artificial Neural Network," *2019 IEEE 4th International Conference on Computing ane Communication System*, pp. 81–85, 2019.

[123] R. Brookey, "Convergence: Facilitating Transdisciplinary Integration of Life Sciences, Physical Sciences, Engineering, and Beyond," *Science Engineering Medicine*, vol. 1, pp. 285–292, 2014.

[124] Notes, Lecture, "Convergence and divergence across construction," vol. 1, pp. 1–7, 2000.

[125] G. Hanrahan, "MODELLING OF POLLUTANTS IN COMPLEX Edited by IJLMpub," *Advance Science anr Enviromental Science*, vol. I, pp. 1–329, 2009.

# VITAE

| | |
|---|---|
| Name | Mrs. Win Win Myo |
| Student ID | 5810230007 |

## Educational Attainment

| N0. | Degree | Name of Institution | Year of Graduation |
|---|---|---|---|
| 1 | Master of Science (Computer Science) | Yangon University (Myanmar) | 2004 |
| 2 | Master of Science (Mathematics) | Yangon University (Myanmar) | 1998 |
| 3 | Bachelor of Science (Honors) (Mathematics) | Yangon University (Myanmar) | 1996 |

## Scholarship Awards

PSU.GS financial support for dissertation, Fiscal year 2018 from Graduate School, Prince of Songkla University, 2018- present.

PSU President Scholarship (PhD Degree), 2015 Academic year.

## Work - Position and Address

| | |
|---|---|
| Associate Professor | Faculty of computing, University of Information Technology (Yangon), Myanmar, 2018-Present. |
| Lecturer | University of Dagon, Yangon, Myanmar, 2013-2018. |
| Lecturer | University of Taungoo, Myanmar, 2010-2013 |
| Asst. Lecturer | University of Dagon, Yangon, Myanmar, 2003-2010. |
| Tutor | University of Sittway, Myanmar, 2002-2003 |
| Tutor | University of Dagon, Yangon, Myanmar, 1998-2003 |

**List of Proceedings**

[i] W. WinMyo, W. Wettayaprasit, P. Aiyarak, "A Noble Feature Selection Method for Human Activity Recognition using Linearly Dependent Concept ( LDC )," in *the 7th International Conference on Software and Computer Applications*, 2018, no. Ldc, pp. 173–177.

[ii] W. WinMyo, W. Wettayaprasit, P. Aiyarak, "Designing Classifier for Human Activity Recognition Using Artificial Neural Network", 2019 IEEE 4th International Conference on Computer and Communication Systems, 2019, IEEE Catalog Number: CFP19D48, pp. 81-85.

**List of Publication**

[iii] W. WinMyo, W. Wettayaprasit, P. Aiyarak, "A more reliable step counter using built-in accelerometer in smartphone," *Indones. Journal of Electrical. Engineering and Computer Science.*, vol. 12, no. 2, pp. 775–782, 2018.