ACTIVITY RECOGNITION USING OPTIMIZED REDUCED KERNEL EXTREME LEARNING MACHINE (OPT-RKELM)

YANG DONG RUI

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

ACTIVITY RECOGNITION USING OPTIMIZED REDUCED KERNEL EXTREME LEARNING MACHINE (OPT-RKELM)

YANG DONG RUI

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE DEGREE

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: YANG DONGRUI

Registration/Matric No.:

Name of Degree: MASTER OF COMPUTER SCIENCE

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

ACTIVITY RECOGNITION USING OPTIMIZED REDUCED KERNEL EXTREME LEARNING MACHINE(OPT-RKELM)

Field of Study:

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:
Designation:

ACTIVITY RECOGNITION USING OPTIMIZED REDUCED KERNEL EXTREME LEARNING MACHINE (OPT-RKELM)

ABSTRACT

In the past decade, research related to Human Activity Recognition (HAR) based on devices embedded sensors has shown good overall recognition performance. As a consequence, HAR has been identified as a potential topic for healthcare assessment systems. One of the major research problems is the computation resources required by machine learning algorithm used for classification for HAR. Numerous researchers have tried different methods to enhance the algorithm to improve performance, some of these methods include Support Vector Machine (SVM), Decision Trees, Extreme Learning Machine (ELM), Kernel Extreme Learning Machine (KELM), and Deng's Reduced Kernel Extreme Learning Machine (RKELM). However, unsatisfactory accuracy, slow learning speed, and stability is still a problem. In this study, we have purposed a model named as Optimized Reduced Kernel Extreme Learning Machine (Opt-RKELM). It applies the characteristic of ReliefF algorithm to rank and select top scoring features for feature selection. ReliefF can solve the problem of large feature dimension in the existing RKELM. By using clustering method K-Means, we have found the best center point position to calculate Kernel matrix. at last, we have employed Quantum-behaved Particle Swarm Optimization (QPSO) to get the optimal kernel parameter in the proposed model. To evaluate the effectiveness of Opt-RKELM, two benchmark datasets related to human activity recognition problems are used. The notable advantages of the proposed model are excellent recognition accuracy, fast learning speed, stable prediction ability, and good generalization ability.

Keywords: Human Activity Recognition, Extreme Learning Machine, ReliefF, K-Means,

Optimized Reduced Kernel Extreme Learning Machine, Quantum-behaved Particle Swarm Optimization.

ABSTRAK

Dalam dekad yang lalu, penyelidikan yang berkaitan dengan Human Activity Recognition (HAR) berdasarkan peranti sensor terbenam telah menunjukkan prestasi pengiktirafan keseluruhan yang baik. Sebagai akibatnya, HAR telah dikenal pasti sebagai topik yang berpotensi untuk sistem penilaian kesihatan. Satu masalah penyelidikan utama adalah sumber perhitungan yang diperlukan oleh machine learning algoritma yang digunakan untuk klasifikasi HAR. Pelbagai penyelidik telah mencuba kaedah untuk meningkatkan algoritma untuk meningkatkan prestasi. Beberapa kaedah ini termasuk Support Vector Machine (SVM), Decision Trees, Extreme Learning Machine (ELM), Kernel Extreme Learning Machine (KELM), dan Deng's Reduced Kernel Extreme Learning Machine (RKELM). Walau bagaimanapun, kerumitan komputasi tinggi, pembelajaran perlahan kelajuan, dan kestabilan masih menjadi masalahnya. Dalam kajian ini, kami telah membuat satu model bernama Optimized Reduced Kernel Extreme Learning Machine (Opt-RKELM). Ia terpakai ciri ReliefF Algorithm untuk menentukan dan memilih ciri pemarkahan atas untuk ciri pemilihan. ReliefF dapat menyelesaikan masalah dimensi besar dalam RKELM yang sedia ada. Dengan menggunakan kaedah clustering K-Means, kami telah menemui kedudukan titik pusat terbaik untuk hitung matriks Kernel. Pada yang terakhir, kami telah menggunakan Quantum-behaved Particle Swarm Optimization (QPSO) untuk mendapatkan parameter kernel optimum dalam model yang dicadangkan. Dua kumpulan penanda aras yang berkaitan dengan Human Activity Recognition digunakan untuk menilai keberkesanan Opt-RKELM. Kelebihan dari model yang dicadangkan adalah prestasi pengiktirafan yang baik, kos pengiraan yang rendah, kelajuan pembelajaran yang pantas, stabil keupayaan ramalan, kurang ketergantungan parameter, dan keupayaan generalisasi yang baik.

Keywords: Human Activity Recognition of Daily Living, Reduced Kernel Extreme Learning Machine, ReliefF, K-Means, Optimized Reduced Kernel Extreme Learning Machine, Quantum Particle Swarm Optimization.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor, Prof Dr. Loo Chu Kiong for his guidance, patience and many invaluable comments and suggestions throughout the whole Master's study and related research. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better supervisor and mentor for my Master's study.

I want to acknowledge University of Malaya and also thanks my friends and laboratory members for their insightful comments and encouragement, but also for the hard questions which inspired me to widen my research from various perspectives.

Finally, I must express my very profound gratitude to my family especially my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ABCO	:	Ant Colony Optimization.
BPNN	:	Back-Propagation Neural Network.
ELM	:	Extreme Learning Machine.
FNN	:	Feedforward Neural Network.
GA	:	Genetic Algorithm.
HAPT	:	Smartphone-Based Recognition of Human Activities and Postural Transitions.
HAR	:	Human Activity Recognition.
HARUS	:	Human Activity Recognition Using Smartphones.
K-Means	:	K-Means.
k-NN	:	k-Nearest Neighbor.
KELM	:	Kernel Extreme Learning Machine.
Opt-RKELM	:	Optimized Reduced Kernel Extreme Learning Machine.
PSO	:	Particle Swarm Optimization.
QPSO	:	Quantum-behaved Particle Swarm Optimization.
ReliefF	:	ReliefF.
RKELM	:	Reduced Kernel Extreme Learning Machine.
SLFN	:	Single Hidden Layer Feedforward Neural Network.
SVM	:	Support Vector Machine.

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Background

Human activity recognition includes a series of limb movements with rich meanings. It is a way of expressing people's behavioral intentions or completing the information transmission between people and the environment. Human activity recognition refers to the process of computer automatic detection, analysis and understanding of various sports and behaviors of the human body, and even take actions or responses. It has broad application prospects in human-computer interaction, rehabilitation engineering, education, game, teleconference, sports, healthcare, elderly care, and personalized recommendation.

A number of activity identification methods have been established that use specific purposes hardware devices such as Dijkstra, Kamsma, and Zijlstra (2010) or sensor body networks (Mannini & Sabatini, 2010; Altun, Barshan, & Tunçel, 2010). Although the use of numerous sensors can improve the performance of recognition algorithms, it is unrealistic to expect the general public to use them for everyday activities due to the difficulty and time required to wear them.

With the development of microelectronics systems, accelerometers have been miniaturized so that they can be embedded in small mobile devices, This integration affords researchers with a platform for the classification of a group of daily activities such as standing, walking, laying, walking, walking up, and downstairs. In this process, the inertial body signal is processed using a supervised machine learning algorithm of limited resource hardware.

In recent years, with the rapid development of smartphone technology and the maturity of sensor technology, the application of sensor technology has become more and more extensive, and the role played by activity recognition has become increasingly important.

For example, sensor technology can be easily applied to smart home environments: activity reminders, fall detection, rehabilitation guidance, health assessment, and more. In health assessment, a person's ability to perform activities in daily life is usually closely related to his or her health. For example, people with Alzheimer's disease are characterized by sitting or lying for a long time and having sleep disorders. Therefore, capturing the location of these activities, the duration, frequency of occurrence, etc. can be used to preliminarily infer whether the active person has a certain condition. Compared to traditional wearable activity recognition (Mital, Smith, Hill, & Henderson, 2011), the development of an activity recognition application using a smartphone has several advantages, such as device portability, no additional fixed equipment, and lack of discomfort.

1.2 Problem Statements

One drawback of the smartphone-based approach is that energy and services on the mobile phone are shared with other applications. This problem becomes critical in mobile phone devices which have limited resources. Y. Chen, Zhao, Wang, and Chen (2012) uses the ELM algorithm based fast and robust human activity recognition model. Huang, Zhou, Ding, and Zhang (2012) proposed KELM, can achieve very high accuracy in classification problem. W.-Y. Deng, Zheng, and Wang (2014) introduced RKELM as one solution to this problem. RKELM solves the problem of huge computation of Kernel by randomly selecting 10% of the input data to calculate the Kernel Matrix. This process was able to reduce the use of computational resources, and computational time. But, randomly selecting the 10% input data also creates an unstable issue that reduces the recognition rate. Because activity datasets can sometimes have a lot of data. If these data are not properly selected for classifier, the activity recognition accuracy will be severely degraded. Furthermore, optimal parameter choice is required for the RKELM model to recognize activity with high accuracy. In view of this, the following research problems have been identified as follows:

- Less relevant features reduce accuracy and slow down the performance Opt-RKELM in activity recognition.
- The random sampling used in RKELM degrades the overall performance of the activity recognition model.
- Kernel parameter dependency problem of the RKELM affects the performance of the model.

1.3 Research Objective

Base on the problem statements list in section 1.2, the research objective of this study are given as follows:

- To reduce high-dimensional feature space from large datasets by using an appropriate feature selection method ReliefF.
- To replace random sampling in the RKELM by using an unsupervised machine learning technique K-Means in order to solve the lack of deterministic of classification.
- To determine the best parameter for kernel using an optimization technique Quantum-behaved Particle Swarm Optimization.

The three objectives listed above are matched with the three problem statements respectively. Therefore, the main aim of this study is to implement a system called Optimized Reduced Kernel Extreme Learning Machine (Opt-RKELM) for Human Activity Recognition

1.4 Proposed Method

The proposed method will be split into three stages, pre-processing stage, machine learning stage and optimization stage. In the first stage is the pre-processing stage. There have two steps in the preprocessing stage. The first step is using the ReliefF to select the important features from the available input features. After these input features had been filtered out, the next step is to perform the clustering method by using k-mean. This clustering process can find out the important data points for the Kernel matrix function calculation.

After the pre-processing stage, the next stage is the classification stage. Opt-RKELM used to train and recognize the human activities. compare to Deng's RKELM, the input data points for the kernel matrix function calculation will be replaced by the k-mean data points. By doing this replacement able to reduce the training speed and improve the recognition stability. The final stage is the optimization stage. In the optimization stage, QPSO is proposed and implemented to adjust the RKELM parameters. By doing this, it able to solve the RKELM parameter dependencies problems and improve the high recognition rate.

1.5 Research Contributions

The contributions of this thesis can be summarized as follow: In the first place, an Optimized-Reduced Kernel Extreme Learning Machine (Opt-RKELM) for Human Activity Recognition (HAR) system was proposed. To achieving this, at-first, ReliefF were implemented to select the relevant and important features from the available input features. These input features are used to train the Reduce Kernel-Extreme Learning Machine (RKELM) for the human activity recognition.

In order to speed up the training process and increase the stability of the recognizing result, another level of pre-processing was proposed and implemented. Before the input features pass to the RKELM for training, these input features will go through a clustering method, K-mean, to select the important data to build-up the kernel matrix in the RKELM. By doing this, the training process will become more faster and stable. In-order to optimize the proposed system and solve the RKELM parameter dependencies issues, QPSO were proposed and implemented.

To validate the proposed Opt-RKELM, two benchmark datasets were selected for validation. Experiments have shown that the proposed Opt-RKELM model can achieve better performance than other existing models.

As a summary, the main contribution for this thesis is the proposed Opt-RKELM model is performs better than the RKELM model (W.-Y. Deng et al., 2014), and use even less training time. Opt-RKELM model can run very fast using less computation resource. This research is prepare for produce a Journal publication and a draft is being prepared under the title "Activity recognition using optimized reduced kernel extreme learning machine (Opt-RKELM)".

1.6 Thesis Outline

The rest of the thesis is organized as follows: This thesis is organized into six parts: Introduction and Literature Review, Methodology and Experiments, Result and Discussion, and Conclusion and Future Works.

Chapter 2 Literature Review presents an overview of basic concepts, reviews the fundamentals of a Human Activity Recognition system including the data-processing, feature extraction, classification, and optimization technique. Furthermore, a summary of comparisons between several previous research works is presented. Chapter 2 presents the reader with useful background knowledge about a Human Activity Recognition.

Chapter 3 Methodology formally defines the architecture of the proposed Opt-RKELM system. It presents studies that fulfill the research problems proposed in section(1.2), Starting with the data collection procedure, feature extraction methods and data preprocessing are presented. Then, the classification is present, finally, the effect of parameters is used QPSO to find best fit value.

Chapter 4 Experiments presents the implementation of the experiments, including the explanation of the of the available database and the tools used to implement the Opt-RKELM system.

Chapter 5 Result and Discussion presents and discusses the experiment results conducted through the experiments including the comparison with others classifier.

In the end, Conclusions and suggests some Future research directions of this work are presented in Chapter 6 Conclusion and Future Works.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

In the past few years, activity recognition has become an emerging field of research and one of the challenges for pervasive computing. Activity recognition researchers have explored different types of sensing technologies. In general, sensor technology can be broadly classified into three categories: (1) vision-based methods(Jalal, Uddin, & Kim, 2012); (2) ambient sensor-based methods(Ning, Shi, Zhu, Li, & Chen, 2019); and (3) wearable sensor-based methods(A. Wang, Chen, Yang, Zhao, & Chang, 2016).

The vision-based methods primarily use image or video to monitor and identify different activities. In well-controlled environment, the correct rate of vision-based activity recognition is relatively high, but in many special environments, such technologies have serious privacy problems, and this method is often limited by lighting changes, environmental occlusion, and background changes (Chaquet, Carmona, & Fernández-Caballero, 2013; Aggarwal & Xia, 2014; Hernández, Cabido, Montemayor, & Pantrigo, 2014; Yin, Tian, Feng, & Li, 2014).

Based on ambient sensor-based methods to determine activity by capturing the interaction between human and object. For example, if a sensor embedded in a chair is triggered, it is inferred that the monitored person may be sitting in a chair; If it is detected that the sensor placed on the bed is active for a long time, it is inferred that the monitored person may be in a sleep state. This method does identify human activities in daily life, such as washing, eating, sleeping, and watching TV, but this method is usually expensive and limited to indoor scenes (L. Chen & Khalil, 2011, 2011). In addition, for the entire system to work properly, the establishment and maintenance of the system are relatively complex, such as how to effectively deploy these sensors and devices in appropriate places without inconvenience to users.

Compared to the above two methods, the wearable sensor-based method is very flexible. Wearable sensors can be adapted to many parts of the body, such as the head, arms, wrists, legs, ankles, or even pockets. They are suitable for both indoor and outdoor environments, and can be worn with multiple devices at the same time. Among these wearable devices, smartphone are particularly powerful, with most of the additional sensing components available to users, and with powerful computing and communication capabilities. However, there are still many difficulties to be solved in how to learn a large number of samples, let the computer learn faster, and finally accurately identify at high speed. Therefore, smartphone sensor activity recognition is a valuable and challenging area of research. Figure 2.1 shows an smartphone and lists some of its features.

Figure 2.1: Example of a commercial smartphone and some of its features.

Among the development of micro-electrical mechanical systems (MEMS), the sensors including accelerometers ,gyroscope ,and magnetometer are miniaturized that can be embedded into smartphone. This is why activity recognition has become an increasingly popular topic in recent years. Many studies focus on activity recognition software built for mobile applications (Brezmes, Gorricho, & Cotrina, 2009; L. Sun, Zhang, Li, Guo, & Li, 2010; Anjum & Ilyas, 2013).

Mobile accelerometer-based activity recognition(Taylor, 2009) has received much attention for its wide range of applications in healthcare, personalized recommendations, advertising services, etc.(Cambria & Hussain, 2012; Mital et al., 2011; Wöllmer, Eyben, Graves, Schuller, & Rigoll, 2010) In the work of Ravi, Dandekar, Mysore, and Littman (2005) and Ward, Lukowicz, and Gellersen (2011), the authors used multiple accelerometers to classify different activities.

Y. Chen, Qi, Sun, and Ning (2010)used smartphones to detect six activities in order to find a state switching point. These models can achieve high recognition accuracy because their test and training samples are from the same batch of samples and follow the same distribution.

Kwapisz, Weiss, and Moore (2011) collected and labeled daily activities, including walking, jogging, stairs, standing, sitting in 29 subjects. They discussed features of each activity and compared performances of using J48, Logistic Regression, Multilayer Perceptron and Straw Man with 10-fold validation. The features they selected are mean, standard deviation, average absolute difference, average resultant acceleration, the time between peaks and binned distribution. It must be wise to choose features, because the features quality can affect performance.

Cheng et al. (2013) proposed a novel active data preprocessing method in their work. The collected accelerometer readings are first converted to a fixed Earth coordinate system using the gravity vector collected from the sensors. A 10Hz low pass filter is then applied to eliminate noise and improve the quality of the signal collected by the phone that is loosely attached to the body.

Murad and Pyun (2017) Adopting deep learning methods for human activity recognition, propose the use of deep recurrent neural networks (DRNNs) for building recognition models. In experimental, compared with ELM (Kumar, Bharadwaj, Sumukha, & George, 2016), CNN (Jiang & Yin, 2015), SVM (Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2013), random forest(Murad & Pyun, 2017), HMMs (Zappi et al., 2008), DBNS (Murad & Pyun, 2017), and k-nearest neighbors (KNN)(Hammerla, Kirkham, Andras, & Ploetz, 2013) with several databases include HARUS, achieved 96.7% of accuracy.

Traditional neural network learning algorithms, such as the most representative Back-Propagation Neural Network (BPNN) algorithm in the Single Hidden Layer Feedforward Neural Network (SLFN), In the learning process, a large number of network training parameters need to be set to continuously adjust the weight and threshold of the network, so that the square error of the network is minimized, and the training time is often several hours or even several days. In this regard, in 2004, Professor Huang GuangBin developed an ELM algorithm for solving single hidden layer neural networks (Huang, Zhu, & Siew, 2006). The so-called “Extreme” refers to the limitations of transcending traditional artificial learning methods, breaking the barrier between traditional artificial learning and biological learning mechanisms, and learning from higher-level brains (Huang et al., 2012). The ELM algorithm is based on neural network generalization theory, control theory, matrix theory and linear system theory, and represents a set of machine learning theories that do not need to adjust hidden layer nodes.

The presentation of ELM is of epoch-making significance. Because for decades, there has been a controversial issue in the fields of neural networks, machine learning, and neuroscience: Whether the hidden layer nodes need to be adjusted during the learning process, ELM theory proves that for most neural networks and learning algorithms, it is not necessary to iteratively adjust the hidden layer nodes. In addition, unlike traditional neural networks, ELM only needs to set the number of hidden layer nodes of the neural network throughout the learning process, without adjusting the input weight of the network and the bias of the hidden layer nodes, in order to ensure high learning accuracy, The speed of learning has been greatly improved.

In the past few years, ELM has demonstrated excellent learning accuracy and speed in many applications due to its outstanding structure, fast training, strong generalization ability and strong general classification ability. Such as face recognition (Mohammed, Minhas, Wu, & Sid-Ahmed, 2011), image segmentation (Pan, Park, Yang, & Yoo, 2012), human activity recognition (Minhas, Baradarani, Seifzadeh, & Wu, 2010). Different from the traditional neural network model, ELM only requires the infinite order of the activation function. The input weight of the hidden layer node and the hidden layer "bias" in the ELM network are randomly generated, and it is not necessary to iteratively adjust these parameters during the execution of the algorithm. The output weight of the network can be solved by a single calculation.

ELM, due to its efficacy, has drawn a significant amount of interest from researchers in various fields, such as face recognition (FR) (Mohammed et al., 2011; Zong & Huang, 2011), handwritten character recognition (Chacko, Krishnan, Raju, & Anto, 2012), action recognition (Iosifidis, Tefas, & Pitas, 2013), and activity recognition (W.-Y. Deng et al., 2014). Different versions of improved ELM have been proposed. Inspired by Mercer condition, a kernel ELM (KELM) was proposed for robust classification (Huang et al., 2012). (L. Zhang & Zhang, 2017) compared different versions of ELM perform classification, KELM shows an obvious superiority with 5% improvement in recognition compared with the conventional ELM.

2.2 Classification Methods

Feedforward Neural Network (FNN), especially BPNN, have been widely used in recent years. However, traditional BPNN has a slow convergence rate and convergence to local algorithms. In essence, the parameter is an optimization problem in order gradient method, It has a slow convergence speed and convergence to a local minimum problem. Even improved FNN have faster training speeds and better generalization capabilities than BPNN, but they still have not to get the global optimal solution.

In order to solve the above problems, Huang, Zhu, and Siew (2004) proposed the ELM method to train the SLFN in 2004. This chapter mainly introduces the main ideas and develop algorithms of Extreme Learning Machine, KELM and RKELM.

In ELM, hidden nodes are initialized randomly and are not adjusted during the entire training process. The only parameter that needs to be learned is the weight between the hidden layer and the output layer. In this way, ELM can be attributed to a parametric linear model that solves linear classification problems. Compared with the traditional FNN, ELM is more efficient and tends to get the best solution in the whole situation.

2.2.1 Basic Algorithm of Extreme Learning Machine Algorithm

Although the neural network has been widely used in different fields, the traditional feedforward network also has its own advantages, but it also has the following disadvantages: It takes a long time to repair the weights and thresholds in network training. The use of gradient descent has the drawback of falling into local minimum and thus does not reach the global minimum.

The ELM algorithm greatly improves the learning speed of the SLFN and is widely used in various fields.

In recent years, extreme learning machines have gained more and more scholars' research by virtue of their fast learning speed and good learning effect. Extreme Learning

Machine, a fast learning algorithm for SLFN, originally proposed by Huang et al. (2004), ELM randomly generates hidden node parameters, then, analyzes and determines output weight instead of iterative adjustment. As a result, ELMs run fast, are easy to implement, and appear to outperform other classifiers. ELM diagram shown in Figure 2.2, red curly brackets can apply Feature learning, Clustering, Regression, and Classification. The current research mainly focuses on the improvement and development of ELM's own algorithm theory and the expansion of ELM practical application field. The practical application range of ELM has been expanding in recent years and has been widely used in time series prediction, sales forecasting(Z.-L. Sun, Choi, Au, & Yu, 2008), face recognition(Mohammed et al., 2011), power system, control engineering, fault diagnosis and prediction, data analysis, large-scale data processing, and image processing. The achievements in the theoretical research of ELM algorithm are mainly based on the defects of the model and different fields and problems. ELM algorithms are proposed, which can be summarized as the following:

For N arbitrary different samples (x_i, t_i), where $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^n$, $t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in R^m$, Then the output of a FNN with L hidden nodes and active function $g(x)$ can be expressed as:

$$\sum_{i=1}^{L} \beta_i g_i(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j, \quad j = 1, \cdots, N, \tag{2.1}$$

In Equation (2.1), $w_i = [w_{i1}, w_{i2}, \cdots, w_{in}]^T$ is the weight vector connecting the ith hidden neuron and the input neurons, $w_i \cdot x_j$ denotes the inner product of w_i and x_j. Where $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the weight vector connecting the ith hidden node and the output node, b_i is threshold bias of the ith hidden node and o_j is the output vector of the input sample x_j. Activation function $g(x)$ can choose the "Sigmoid function", as well as the "radial basis", "sine", "cosine", "exponential", and many other nonregular

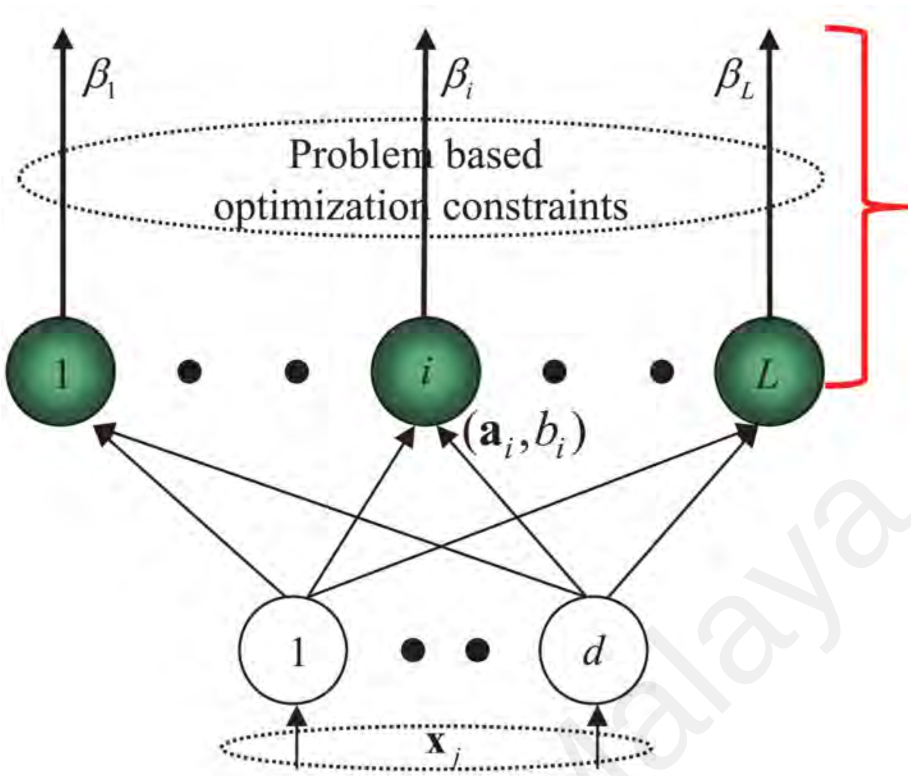

Figure 2.2: ELM Architecture. L Random Hidden Neurons (which need not be algebraic sum based) or other ELM feature mappings. Different type of output functions could be used in different neurons, d are Input Nodes

functions(Huang & Babri, 1998) .

If this L hidden nodes with $g(x)$ activation function SLFN can approximate those N samples with no error, will have:

$$\sum_{j=1}^{L} \left\| \mathbf{o_j} - \mathbf{t_j} \right\| = 0 \tag{2.2}$$

For input sample x_j, t_j is the sample class label vector. consequently, there exist β_i, w_i and b_i such that:

$$\sum_{j=1}^{L} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, \quad j = 1, \cdots, N, \tag{2.3}$$

This can be written as:

$$\begin{pmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{pmatrix}_{N\times L} \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{pmatrix}_{L\times m} = \begin{pmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_L^T \end{pmatrix}_{N\times m} \tag{2.4}$$

The above equations (2.4) can be written compactly as

$$\mathbf{H}\beta = \mathbf{T}, \tag{2.5}$$

Since $\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{pmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{pmatrix}$ is hidden layer output matrix, the input weight vectors w_i and the hidden biases b_i are randomly chosen, To train an SLFN is simply equivalent to finding a least-squares solution $\hat{\beta}$ of the linear system $\mathbf{H}\beta = \mathbf{T}$:

$$\| \mathbf{H}\hat{\beta} - \mathbf{T} \| \quad = \min_{\beta} \| \mathbf{H}\beta - \mathbf{T} \| . \tag{2.6}$$

The smallest equivalent to finding norm least-square solution is:

$$\hat{\beta} = \mathbf{H}^{+}\mathbf{T}, \tag{2.7}$$

The $\mathbf{H}^{+}$ is inverse of the matrix H generalized by using Moore-Penrose, Matrix H is hidden layer output matrix.

2.2.2 Kernel Extreme Learning Machine Algorithm

ELM not only has computational efficiency, but also tends to achieve similar or better generalization performance than SVM (Huang, Ding, & Zhou, 2010). However, because of the randomly assigned input weight and bias, ELM can produce a large variation in classification accuracy with the same number of hidden nodes. On 2010 and 2012, Huang et al. (2012) combined with the learning principle of support vector machine, and proposed KELM. the Kernel Extreme Learning Machine (KELM), which replaces the ELM hidden layer with kernel function, is proposed to solve this problem. It is worth noting that the kernel function used in KELM does not need to satisfy Mercer's theorem and KELM provides a unified solution to multiclass classification problems. KELM improved the generalization ability of ELM, by replacing the random map with a kernel function.

Talk about stability, the hidden layer output matrix of KELM calculated by training samples through a kernel mapping is not associated with L, so the recognition rate of KELM is unchanged. We see that the proposed method is more robust to the variation of model parameter and hidden neurons.

The methods to calculate Moore-Penrose generalized inverse of a matrix is the orthogonal projection method:

$$\mathbf{H}^{\dagger} = \mathbf{H}^{T}(\mathbf{H}\mathbf{H}^{T})^{-1} \tag{2.8}$$

From (2.8), Huang et al. (2012) suggested adding a positive value I/C (regularization coefficient) to help calculate the output weights. According to the ridge regression theory, where C is used to fix the over-fitting problem:

$$\beta = \mathbf{H}^{T}(\frac{I}{C} + \mathbf{H}\mathbf{H}^{T})^{-1}\mathbf{T} \tag{2.9}$$

Since the output function for the ELM is:

$$f(\mathbf{x}_i) = \mathbf{h}(\mathbf{x}_i)\beta \tag{2.10}$$

$h(x_i)$ is the output of the hidden nodes, **H** is hidden layer output matrix. Put Equation

(2.9) and Equation (2.10), the output function defined as follows:

$$f(\mathbf{x}_i) = \begin{bmatrix} \mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_1)^T \\ \vdots \\ \mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_N)^T \end{bmatrix} \left(\frac{I}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{T} \tag{2.11}$$

Define a kernel function k as:

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_1) = \mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_1)^T = \mathbf{H}\mathbf{H}^T \tag{2.12}$$

Thus, the output function of KELM can be written as:

$$f(\mathbf{x}_i) = \mathbf{h}(\mathbf{x}_i)\mathbf{H}^T \left(\frac{I}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{T} = \begin{pmatrix} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_1) \\ \vdots \\ \mathbf{K}(\mathbf{x}_i, \mathbf{x}_N) \end{pmatrix}^T \left(\frac{I}{C} + \mathbf{K}\right)^{-1} \mathbf{T} \tag{2.13}$$

2.3 Pre-processing

Data pre-processing is one of the most important steps in the data mining process. In activity recognition system, Pre-processing usually can be separate to feature selection and data sample selection, to improve efficiency and accuracy to further achieve the ultimate goal of improving overall performance.

2.3.1 Feature Selection

Feature selection consists of selecting a subset of relevant features from the original feature set (Guyon & Elisseeff, 2003). To differentiate between samples, classification algorithms need representative features. Using inappropriate or redundant features may deteriorate the performance of a classification algorithm.

Activity recognition problem is type of classification problem, both using classifier separate different human states. Emotion recognition is very similar to activity recognition, As we know, the ReliefF algorithm is a widely used feature selection method. J. Zhang et al. (2016) propose a ReliefF-based selection methods with SVM classifier, The results show the effectiveness of ReliefF as a feature selection algorithm. Similar results have also been reported in other literatures (Schmidt & Trainor, 2001; W.-L. Zheng & Lu, 2015), which prove the significance of ReliefF weight to some extent.

Kira and Rendell (Gupta, 2014) came up with a Relief algorithm in 1992 for a general problem with a high number of features. In the same year, (Gupta & Dallas, 2014) used ReliefF and sequential forward floating search (SFFS) to solve an activity recognition problem, combine with Naive Bayes and k-nearest neighbor (k-NN), results shown ReliefF can select relevant and robust features. Kononenko (Kononenko, 1994) improved the basic Relief algorithm, into ReliefF, by improving noise immunity and introducing support for multi-class problems. The Relief and ReliefF algorithms use a statistical approach rather than heuristic search for finding the feature subset. ReliefF provides a relevance weight to each of the potential feature and the ones above a set relevance threshold are selected.

Capela, Lemaire, and Baddour (2015) selected features using three filter-based, classifier-independent, feature selection methods (ReliefF, Correlation-based Feature Selection(CFS), Fast Correlation Based Filter(FCBF)). ReliefF has been reported to be useful in cases with strong interdependencies between fields (Liu & Motoda, 2007), the accuracy achieved using the ten highest ranked features was within 5% of the maximum accuracy achieved. Thus, subsets of the top 10 ranked features were used to compare populations. the results have shown ReliefF is a good feature selection method.

2.3.2 Clustering Method

Clustering or cluster analysis is the task of assigning a set of objects into clusters, so that the objects in the same cluster are more similar to each other than to those in other clusters. In data mining, clustering is always attracting the attention of researchers, many cluster models include hierarchical clustering (Sibson, 1973; Defays, 1977), density based clustering (Ester, Kriegel, Sander, Xu, et al., 1996; Roy & Bhattacharyya, 2005), centroid-based clustering(Lloyd, 1982), and clustering models related to statistics (Moon, 1996). In order to take advantage of feature mapping technology, kernel method has been used in clustering (L. Zhang, Zhou, & Jiao, 2002; Girolami, 2002; Camastra & Verri, 2005), and better results have been obtained. As an explicit feature mapping technique, ELM feature mapping (Huang & Chen, 2007, 2008) is more convenient than the kernel-based method. Therefore, it is meaningful to use clustering technology to replace random select for RKELM. compare to select center point randomly in RKELM, by using a clustering method to select center point will improve the performance of the classifier. We believe that replacing the RKELM kernel with the K-Means calculated center point kernel can obtain more satisfactory classification and regression results.

2.4 Optimization Method

Different studies use evolutionary algorithms to search the parameters of the classifier, which are used to build classification models with high prediction accuracy and stability. For example, in SVM, many optimization algorithms are used to optimize SVM parameters, such as penalty parameters and kernel parameters that control the complexity and accuracy of the prediction model(Subasi, 2013; Tharwat & Hassanien, 2018).

2.4.1 Particle Swarm Optimization Algorithm

The particle swarm optimization algorithm is a new evolutionary algorithm. It starts from the random solution and finds the individual optimal and global optimal solutions through continuous iteration without passing through the "crossover" and "mutation" processes, simplifies the operation of the algorithm. Suppose there is a group of birds in the space foraging. Each bird in the flock is called a "particle", and the "food" that the final flock is looking for is equivalent to the optimal solution of the problem. In the optimization process, all particles have a fitness value, which is determined by the optimized function. At the same time, each particle has two properties: position and velocity. The velocity of a particle determines the direction of flight of the particle, and its velocity is determined by the experience of the particle itself and the location of the surrounding good particles.

Tharwat, Mahdi, Elhoseny, and Hassanien (2018) introduced a model uses k-Nearest Neighbor (k-NN) classifier, collect data from sensor units on the chest, legs and arms. k-NN has only one parameter, k, to determine the number of selected nearest neighbors of the test or unknown sample to predict the category label of the unknown sample. Especially for high-dimensional data, it is difficult to search the k value which has a great impact on the classification performance. Particle Swarm Optimization (PSO) algorithm is used to search the optimal value of k parameter in k-NN classifier. Three experiments are carried out to prove how PSO in this algorithm searches for the optimal value of k parameters in order to reduce the false classification rate of k-NN classifier. results of PSO-kNN algorithm are compared with two well-known algorithms: Genetic Algorithm (GA) and Ant Colony Optimization (ABCO) ant. The results also show that the error rate is lower than that of GA and ABCO algorithms.

The analysis of the trajectory of a single particle reveals the mechanism of PSO (Clerc & Kennedy, 2002; Ozcan & Mohan, 1999). As far as classical mechanics is concerned, a particle is described by its position vector $\vec{x}$ and velocity vector $\vec{v}$, which determines the trajectory of the particle. Particles move along a definite trajectory in Newtonian mechanics, but not in quantum mechanics. In the quantum world, the term trajectory is meaningless, because according to the principle of uncertainty, the $\vec{x}$ and $\vec{v}$ of particles cannot be determined at the same time. Therefore, if a single particle in the PSO system has quantum behavior, the PSO algorithm must work in a different way.

In J. Sun, Feng, and Xu (2004), The quantum theory is introduced into PSO and the quantum behavior PSO (QPSO) algorithm is proposed. the experimental results show that QPSO is better than the standard PSO in several benchmark dataset, and it is a promising algorithm. It happens that there is a similar case. (Peng et al., 2016) did similar study, proposal a novel multi-class classification method termed quantum-behaved particle swarm optimization-based kernel extreme learning machine (QPSO-KELM), solve also time and frequency domain problems like human activity. KELM is compared with five existing classification methods: Linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), extreme learning machine (ELM), k-nearest neighbor (KNN) and support vector machine (SVM). Meanwhile, three traditional optimization methods including particle swarm optimization algorithm (PSO), genetic algorithm (GA) and grid search algorithm (GS). Where QPSO-KELM obtains 95%, PSO-KELM, GA-KELM and GS-KELM can only achieve 88.75%, 87.5% and 86.25% classification rates, respectively.Therefore, In this Thesis we will implement QPSO for parameter selection and optimization.

2.5 Research Gaps

As can be seen from the previous literature, HAR has a large number of classification methods after a long period of research. For example, the use of large-scale deep learning, and the use of new methods such as SVM, ELM, KELM and other methods in recent years, the use of offline and online learning methods, the use of feature selection or other various optimization methods. There are more literature review are further summarized in Table 2.5. Through the study of the literature, the lack of reasonable feature selection, in many algorithms leads to a lot of unrelated feature substitution operations, which is time-consuming and computing-consuming; in recent years, ELM methods are faster and more efficient than deep learning or SVM algorithms, so that KELM or Reduced KELM algorithms are also be introduced. But they either have a lot of computation, or they use a simple random similar to ELM, and we decided to combine the Clustering algorithm to solve the above problems; finally, a good optimization affects the entire algorithm. the previous algorithm still has problems in selecting parameter. We will also propose our approach on the selected optimization parameter selection. We will present our model by analyzing the following aspects: Choosing the appropriate Pre-processing aspect to reduce the noise and speed up the operation, the main classifier will optimize the ordinary ELM and KELM to propose our OPT method, and what kind of The method of parameter selection is used.

2.6 Summary

In this chapter, we first review some basic concepts that are fundamental for the rest of the thesis.machine learning methods and algorithms are reviewed. Then a comprehensive literature review is given, from pre-processing methods, classification methods, and optimization methods. especially reviewed various ELM methods. The background and literature review provide a good foundation for us to develop our own HAR applications based on both supervised and unsupervised learning.

Author/Year	Methods	Dataset	Accuracy	Findings
(Y. Chen et al., 2012)	PCA-ELM	Self Collection	79.68%	First paper implement using ELM
(Seera, Loo, & Lim, 2014)	FMM-CART	HARUS	96.52%	The extracted rules offer an insight into human activity tasks
(Al Jeroudi, Ali, Latief, & Akmeliawati, 2015)	OS-ELM	HARUS	82.05%	First ELM based online method, It is proved that ELM algorithm plays an important role in human body recognition.
(Y. Zheng, 2015)	Least Squares Support Vector Machine (LS-SVM) and Naive Bayes (NB)	USC-HAD	95.6%	compared Relief-F, Correlation-based Feature Selection (CFS), and Fast Correlation Based Filter (FCBF)
(HanYing et al., 2018)	OS-KELM	HARUS	91.89%	state-of-art result for online learning
(C. Wang, Xu, Liang, Huang, & Zhang, 2018)	weighted observation hidden Markov model (WOHMM) + SVM	HARUS	92.5%	Proves that a good feature selection method is necessary.
(Saha et al., 2018)	SVM	HARUS	96%	SVM method best result
(Saha et al., 2018)	SVM	HAPT	94.2%	SVM method best result, bring fall detection idea to future study

CHAPTER 3: METHODOLOGY

3.1 Introduction

In the previous two chapters, we described the state of art for solving HAR problems, and also presented some limitations. In this chapter, we will describe the methods and processes of solving the problems highlighted in the previous chapter. Human activity recognition system using Opt-RKELM algorithm is proposed in detail. The algorithm is comprised of three steps: We will describe the Data Preprocessing, Classification Algorithms, and Optimization parameter selection. A detailed description of these steps follows:

- Step (1): In the data pre-processing stage, the readings of three axes are combined into magnitude sequence to eliminate the directional dependence. Statistic and frequency C domain features are extracted from the magnitude series of resultant acceleration. Finally using ReliefF to reduce feature dimension, K-Means to reduce data dimension.
- Step (2): The classification model construction stage. For the classification model construction, with the characters of fast learning speed, stability, and good generalization capability, Optimized Reduced Kernel Extreme Learning Machine is used to build the classification model.
- Step (3): Optimization stage, Quantum-behaved Particle Swarm Optimization QPSO is used to optimize the parameters of KELM kernel.

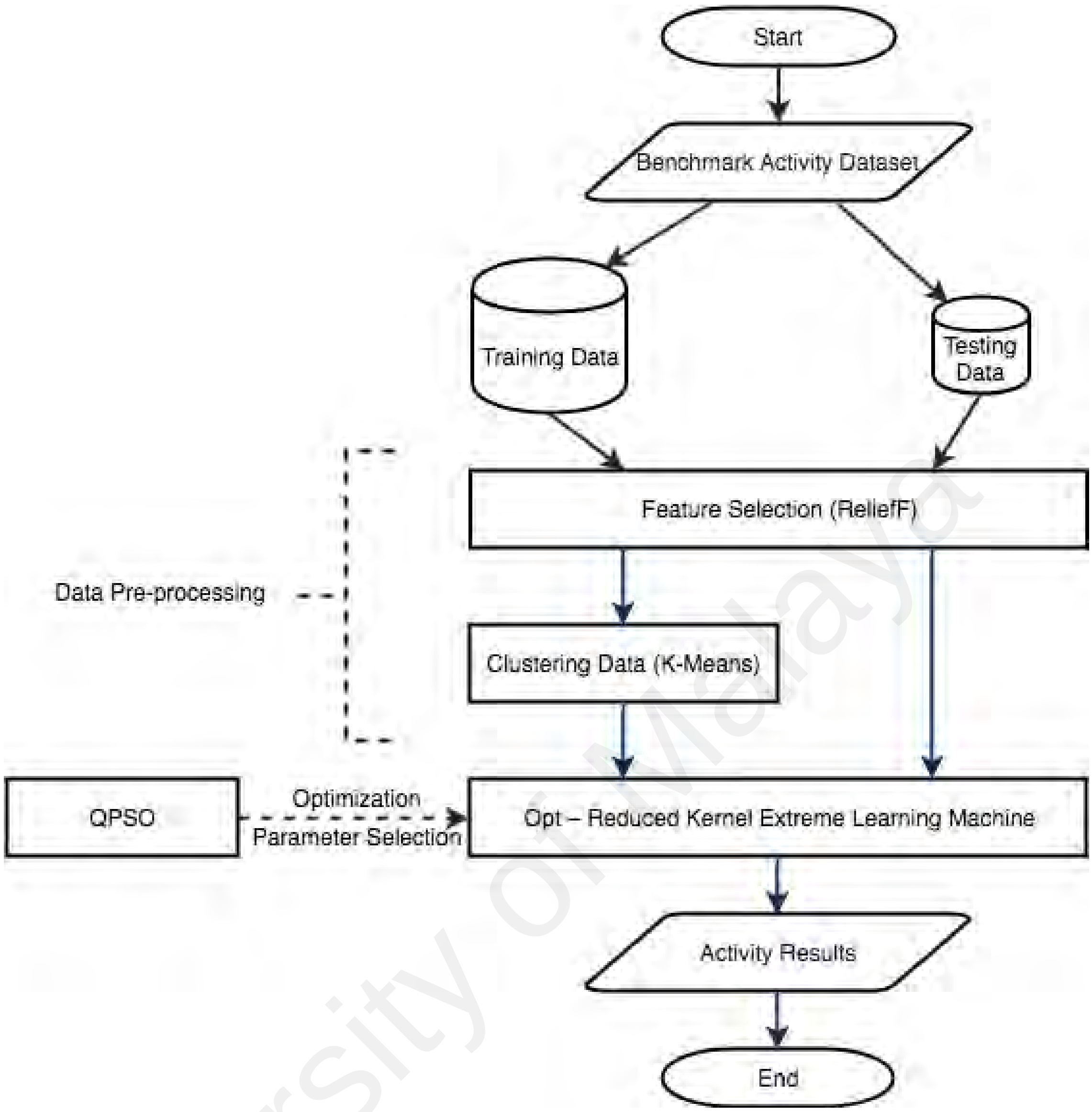

Figure 3.1: Overview flow of Opt-RKELM

3.2 System Overview

As illustrated in Figure 3.1, Human Activity benchmark Datasets send to feature reduction step by using ReliefF (ReliefF) to ranking the importance of features, This is to improve the speed, accuracy, and also data noise. After select features, give K-Means (K-Means) do clustering for data, find 10% data prepare for Kernel Matrix calculation, This is based on the article, W.-Y. Deng et al. (2014) $\widetilde{n}$ is always smaller than n, too large, will make the calculation slower, we choose close to 10% $\widetilde{n}$ in order to mention the speed of operation (W.-Y. Deng et al., 2014), also because normal RKELM choose 10%, As an extreme case when n = 100%, it is the same as the KELM algorithm. After the pre-processing stage, our Optimized Reduced Kernel Extreme Learning Machine will be used as the main classifier, by optimization parameter selection, we use QPSO find best suitable parameter.

3.3 Preprocessing Methods

There are several pre-processing steps needed before we can use the sensor signal data as input for our classification algorithms. We first reduce the number of noise features in the data, then we use K-means find key point help calculate Kernel matrix for later classifier use.

3.3.1 Feature Selection - ReliefF Algorithm

To refine our feature set, we applied the ReliefF filter method (Kononenko, 1994). The ReliefF filter method is an improvement from the Relief method, ReliefF algorithm evaluates the ability of each feature to distinguish instances from different classes. ReliefF not only considers the correlation between feature values and target classes but also the distances between the instances, so the key features defined by ReliefF can collect similar instances and separate different instances, thus effectively promoting accurate classification. In addition, because the feature selection algorithm can reduce calculations, it can save you plenty of time.

The ReliefF algorithm measures the features data to determine a ranking of feature importance. Figure 3.2 and Figure 3.3 shows the results of 561 features from Human Activity Recognition Using Smartphones (HARUS) and Smartphone-Based Recognition of Human Activities and Postural Transitions (HAPT) activity benchmark datasets, ordered by the predicted parameter importance. The higher the bar, the more important this feature is. We select all positive features in this experiments.

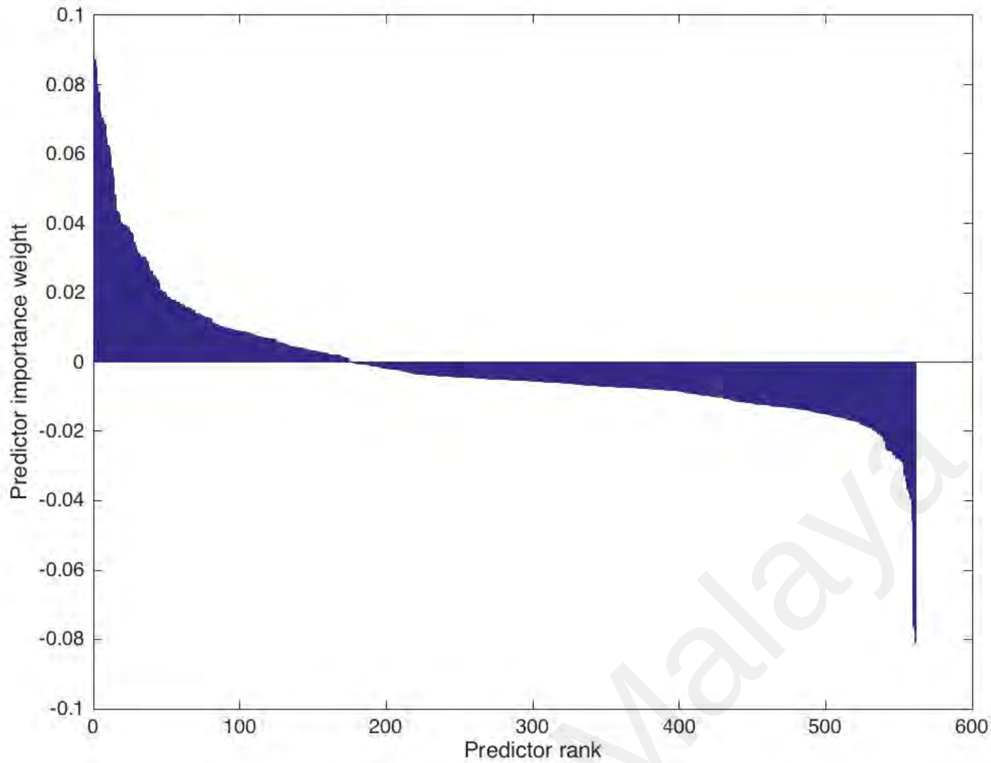

Figure 3.2: Human Activity Recognition Using Smartphones Dataset from the ReliefF algorithm sorted by relevance. The higher features are more important.

This is because, in human activities, there are a lot of noises been collected that affect the classification of the algorithm. These features not only increase the amount of calculation but also affect the performance. Therefore, the feature can be filtered by ReliefF. As shown in the two figures: Figure 3.2 and Figure 3.3, we take the features separately. 179 and 427 feature out of 561 are finally selected.

A fundamental idea of the ReliefF algorithm, presented in Figure 3.4 is to estimate the quality of attributes according to how well their values distinguish between instances that are near to each other. For that purpose, ReliefF randomly selects a feature R_i, then searches for k of its nearest neighbors from the same class, named nearest hits H_j, after that also searches k nearest neighbors from each of the different classes, named nearest misses $M_j(C)$. It updates the quality estimation $W[A]$ for all attributes A depending on their

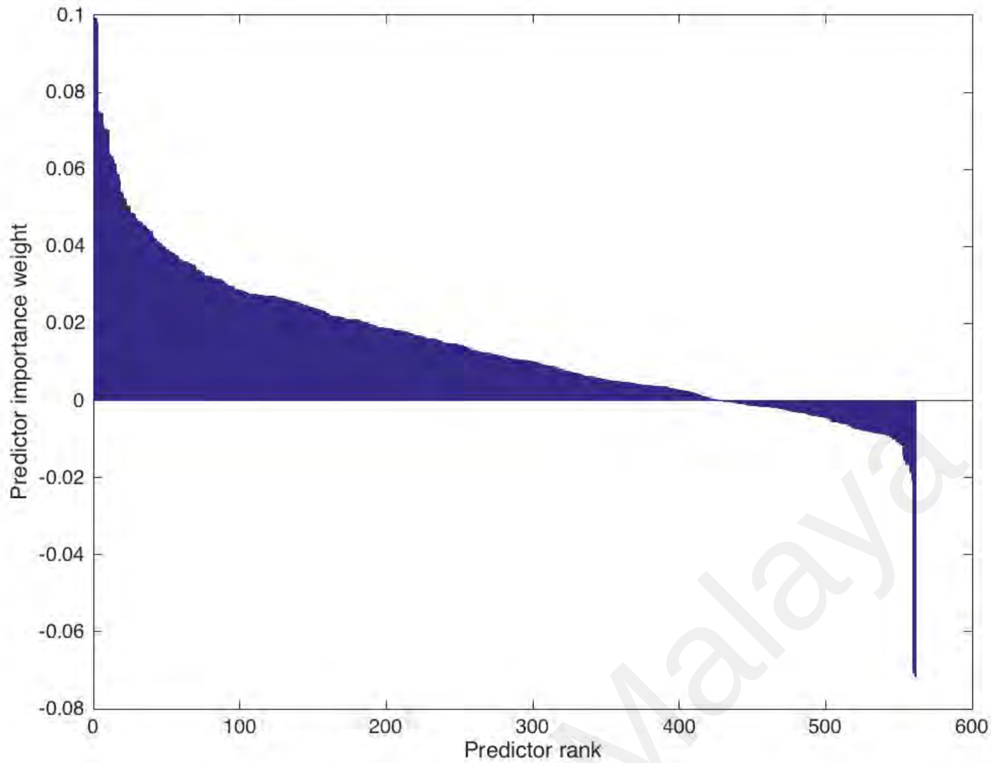

Figure 3.3: Human Activities and Postural Transitions Dataset from ReliefF algorithm sorted by relevance. The higher features are more important.

values for R_i, hits H_j and misses $M_j(C)$. The update formula is similar to that of Relief, except in reliefF we need average the contribution of all the hits and all the misses. The contribution for each class of the misses is weighted with the prior probability of that class $P(C)$. Contributions of hits and misses in each step will be in [0, 1] and also symmetric (we explain reasons for that below) we have to ensure that misses' probability weights sum to 1. As the class of hits is missing in the sum we have to divide each probability weight with factor $1 - P(class(R_i))$ (which represents the sum of probabilities for the misses' classes). The process is repeated for m times.

Selection of k hits and misses is the basic difference to Relief and ensures greater robustness of the algorithm concerning noise. User-defined parameter k controls the locality of the estimates. For most purposes. ReliefF originally used constant influence of k nearest neighbors with k set to small number like 10 (Kononenko, 1994).

Algorithm ReliefF
Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes

set all weights $W[A] := 0.0$;
for i := 1 **to** m **do begin**
    randomly select an instance R_i;
    find k nearest hits H_j;
    for each class $C \neq class(R_i)$ **do**
        from class C find k nearest misses $M_j(C)$;
for A := 1 **to** a **do**
$W[A] := W[A] - \sum_{j=1}^{k} \text{diff}(A, R_i, H_j)/(m \cdot k)+$
$\sum_{C \neq class(R_i)} [\frac{P(C)}{1-P(class(R_i))} \sum_{j=1}^{k} \text{diff}(A, R_i, M_j(C))]/(m \cdot k)$;
end;

Figure 3.4: Pseudo code of ReliefF algorithm.

3.3.2 Clustering Method - K-Means Algorithm

In human activity recognition, there are usually a large amount of experimental data. These data structures have high complexity and high complexity, and contain a lot of redundant information. The interference caused by the classifier detection process is relatively large, which increases the time complexity of the intrusion detection system. In our proposed method, K-means is a classical clustering method (Fu & Sun, 2011), which solves the optimal clustering center and optimal classification by learning. It has high learning efficiency and can process large-scale data (JIA, DING, & SHI, 2015). In this thesis, in order to stably reduce the computational complexity of the Kernel algorithm in RKELM, the K-means algorithm is used to cluster the data to achieve stable and high accuracy.

The K-Means algorithm has many advantages such as small computational complexity, high efficiency for large datasets, and high linearity of time complexity (Zhao, Li, & Cao, 2018). According to W.-Y. Deng et al. (2014), "Of course, just like the work in Wang, Kong, and Zheng (2010) where clustering technique is applied to optimize the algorithm, if optimization technology is used for samples selection in RKELM, the performance of RKELM will be improved further."

K-means is an unsupervised learning clustering technology, we used in replace original RKELM's random select for calculate kernel matrix. The K-Means clustering algorithm is one of the most popular clustering algorithms at present. It uses the Euclidean distance metric as the standard similarity analysis method, and divides the whole into k classes with high similarity by the number of randomly generated clusters. Although the random generation of K values is still random, but it is still better than the simple random calculation of the kernel matrix, and has sufficient generalization ability. The experimental results also show that RKELM using K-means is significantly better than ordinary KELM or RKELM.

As shown in Figure 3.5. K-means is based on some optimization measures, partitions the data set into a given number of clusters.

The clustering problem that the K-means algorithm aims to solve can be expressed as follows: Given the representation of N patterns, find K clusters based on the similarity measure, so that the patterns within the cluster are more similar to each other (higher than intra-cluster similarity) Instead of patterns belonging to different clusters (low inter-cluster similarity).

Let $X = \{\mathbf{x}_i, i = 1, \ldots, N\}$ be the set of N patterns to be clustered into a set of K clusters, $C = \{c_k\}\{k = 1, \ldots, K\}$. Typically each pattern is a vector of dimension $d(\mathbf{x}_i \in R^d)$ and $K \ll N$. The K-means algorithm finds a partition that minimizes the square of the Euclidean distance between the cluster center and the modes in the cluster. Let μ_k be the average of the cluster c_k, defined as:

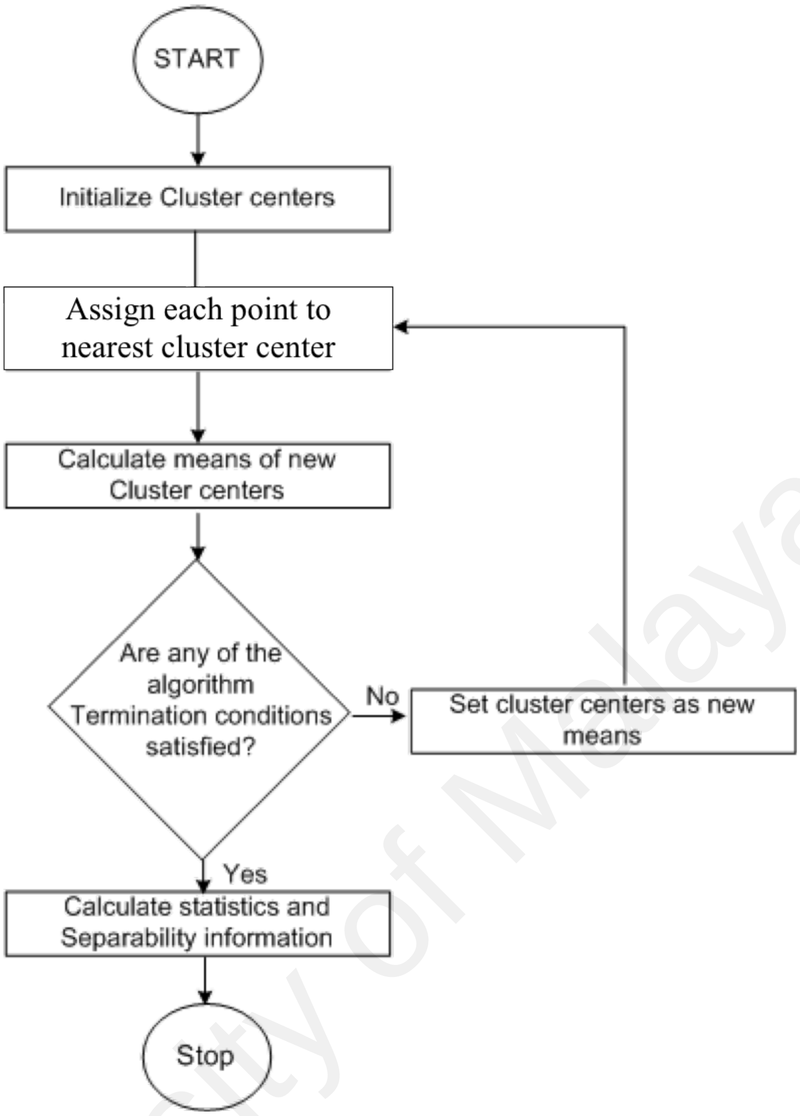

Figure 3.5: Flowchart of K-Means

$$\mu_k = \frac{1}{N_k} \sum_{\mathbf{x_i} \in \mathbf{c_k}} \mathbf{x}_i \tag{3.1}$$

N_k is the number of patterns in cluster c_k

$$J(c_k) = \sum_{\mathbf{x_i} \in \mathbf{c_k}} \| \mathbf{x}_i - \mu_k \|^2 \tag{3.2}$$

The main goal of the K-means algorithm is to minimize the sum of the squared errors on all K clusters.

$$J(C) = \sum_{k=1}^{K} \sum_{\mathbf{x_i} \in \mathbf{c_k}} \| \mathbf{x}_i - \mu_k \|^2 \tag{3.3}$$

The human activity benchmark dataset N sample data, and given the number K of clusters, based on reference paper from W.-Y. Deng et al. (2014), we select $K = 10\%$ whole dataset. Firstly, randomly select K samples as the cluster center of the initial partition, and then use iterative according to the similarity measure function. Method, calculating the distance of the undivided sample data to each cluster center point, and dividing the sample data into the cluster class in which the cluster center is closest, and calculating each cluster class by the calculation The average of all the data in the cluster class continuously moves the cluster center, and the cluster is re-divided until the sum of squared errors in the class is minimal and there is no change.

The main steps of the K-means algorithm are as follows:

Algorithm 1 K-Means algorithm

1: Base on feature selection dataset from ReliefF
2: Initialize K cluster centers randomly. (K = 10% whole dataset)
3: Allocate each pattern to its closest cluster.
4: Compute new cluster centers using Equation (3.1).
5: Repeat steps 2 and 3 until there is no change for each cluster.
6: Out put the 10% as center point pass to classifier.

3.4 Classification Method

The Kernel Extreme Learning Machine developed by Huang et al. (2010). has been widely adopted in the machine learning community as a classification technique with high prediction performance. its good generalization even better than SVM (Support Vector Machine) (Huang et al., 2012). But it does not suitable apply to human activity recognition applications in mobile devices because the kernel matrix $K(X, X)$ need to be built based on the entire data set, Which is very expensive n terms of storage and computation.

3.4.1 Extreme Learning Machine with Reduced Kernel Algorithm

In more detail, although KELM has good generalization and stability, because KELM introduces Kernel calculation, when the amount of data is relatively large, the entire data and Kernel calculation operation will take up a lot of computing resources, and the training time is greatly increased, which reduces KELM efficiency. In order to reduce the computational resources required for Kernel computation and shorten the training time of KELM, W. Deng, Zheng, and Zhang (2013) proposed the RKELM algorithm. The RKELM algorithm uses the idea of random sampling to randomly select a subset of samples from the entire data sample, replacing the entire data set for the Kernel calculation. However, the random use leads to the instability of RKELM. The random selection subset also affects the accuracy. According to wei Wang, Kong, and ying Zheng (2010) research, Clustering technology is applied to optimize the algorithm, we use K-Means to perform pre-processing to select the same size subset solved the above problem and maintains a very short training time on the premise of maintaining accuracy, name it Opt-RKELM.

Given $\aleph$ arbitrary distinct sample $\left\{(\mathbf{x}_i, \mathbf{t}_i) \middle| \mathbf{x}_i \in \mathbf{R}^d, \mathbf{t}_i \in \mathbf{R}^m\right\}_{i=1}^{N}$, based on K-Means results, $\widetilde{X} = \{x_i\}_{i=1}^{\widetilde{n}}$ was selected from original $X = \{x_i\}_{i=1}^{n}$ data set, and $\widetilde{X} \ll X$, and use $\widetilde{K}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}})$ in place of $K(\mathbf{X}, \mathbf{X})$ to reduced problem data size of Huang's KELM (Huang et al., 2012).

We can define kernel matrix as follow:

$$\widetilde{\Omega}_{ELM} = h(\mathbf{x}_i) \cdot h(\widetilde{\mathbf{x}}_j) = \widetilde{K}(\mathbf{x}_i, \widetilde{\mathbf{x}}_j) = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_{\widetilde{n}}) \\ \vdots & \vdots & \vdots \\ K(\widetilde{x}_n, x_1) & \cdots & K(\widetilde{x}_n, x_{\widetilde{n}}) \end{bmatrix}_{\widetilde{n} \times \widetilde{n}} \tag{3.4}$$

and the output weights without regulator can be:

$$\beta = \left(\widetilde{K}^T \widetilde{K}\right)^{-1} \widetilde{K}^T T \tag{3.5}$$

According to the ridge regression theory (Hoerl & Kennard, 1970), adding a positive value I/C (C is regularization coefficient) to diagonal of $\widetilde{K}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}})^T \widetilde{K}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}})$, can have better generalization performance(Huang et al., 2012; W. Deng, Zheng, & Chen, 2009). so the output weight β finally as follows:

$$\beta = \left(\frac{I}{C} + \widetilde{K}^T \widetilde{K}\right)^{-1} \widetilde{K}^T T \tag{3.6}$$

Where C is the regularization parameter, handle over-fitting problem, T is output matrix, the size of $\widetilde{K}$: $\widetilde{n}$ is always smaller then n, if $\widetilde{n} = n$, Opt-RKELM will become KELM, in this thesis, we will set $\widetilde{n}$ nearly 10% of n for compare with Deng's model (W.-Y. Deng et al., 2014).

There are some common kernel function, linear kernel function, polynomial kernel function, Gaussian kernel function, wavelet kernel function: (radial basis function)

$$\text{RBF Kernel: } K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{3.7}$$

$$\text{Linear Kernel: } K(x_i, x_j) = x_i^T x_j \tag{3.8}$$

$$\text{Polynomial Kernel: } K(x_i, x_j) = (x_i^T x_j + r)^d \tag{3.9}$$

$$\text{Sigmoid Kernel: } \mathcal{K}\left(x_i, x_j\right) = \tanh\left(\rho\langle x_i, x_j\rangle + r\right) \tag{3.10}$$

$$\text{Wavelet Kernel: } K(x_i, x_j) = cos(w\frac{\|x_i - x_j\|}{b})e^{(-(\|x_i - x_j\|^2/f))} \tag{3.11}$$

d, f, b, w and σ are real constant parameters, It has been found that RBF Kernel(Gaussian kernel) is giving the best result.

Thus, the output function of Opt-RKELM can be written as:

$$f(\mathbf{x}) = \begin{pmatrix} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_1) \\ \vdots \\ \mathbf{K}(\widetilde{\mathbf{x}}_i, \widetilde{\mathbf{x}}_N) \end{pmatrix}^T \left(\frac{I}{C} + \widetilde{K}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}})^T \widetilde{K}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}})\right)^{-1} \widetilde{K}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}})^T \mathbf{T} \qquad (3.12)$$

Kernel parameter of kernel function, together with regularization coefficient C in Equation (3.6) will be optimized by QPSO.

3.5 Optimization - Quantum-behaved Particle Swarm Optimization Algorithm

It is well known that the performances can be affect by parameters in algorithms. Therefore, Optimization need use to optimize the value of parameters of classifier.

Opt-RKELM is built by ELM combined with Kernel function instead of the hidden node. Then, due to the existence of kernel function, Opt-RKELM is very sensitive to Kernel parameters setting. Therefore, it is very important to use QPSO to optimize the parameters of Opt-RKELM. In this thesis, it is used QPSO combination method is used to improve the accuracy of classification. The results show that the method has excellent classification performance in motion recognition. The flowchart of this QPSO is illustrated in Figure 3.6

The quantum particle swarm optimization algorithm is developed on the basis of particle swarm optimization. QPSO uses the basic idea of quantum mechanics to optimize the particle swarm, which overcomes the shortcomings of PSO convergence rate and easy to fall into local optimum. The search can be performed in the entire solution space, and the evolution equation has few parameters, simple form, strong global search ability and robustness. In QPSO, the position equation of a particle is:

$$X(t+1) = P \pm \beta \cdot |P_{mbest} - X(t)| \cdot \ln(1/u) \qquad (3.13)$$

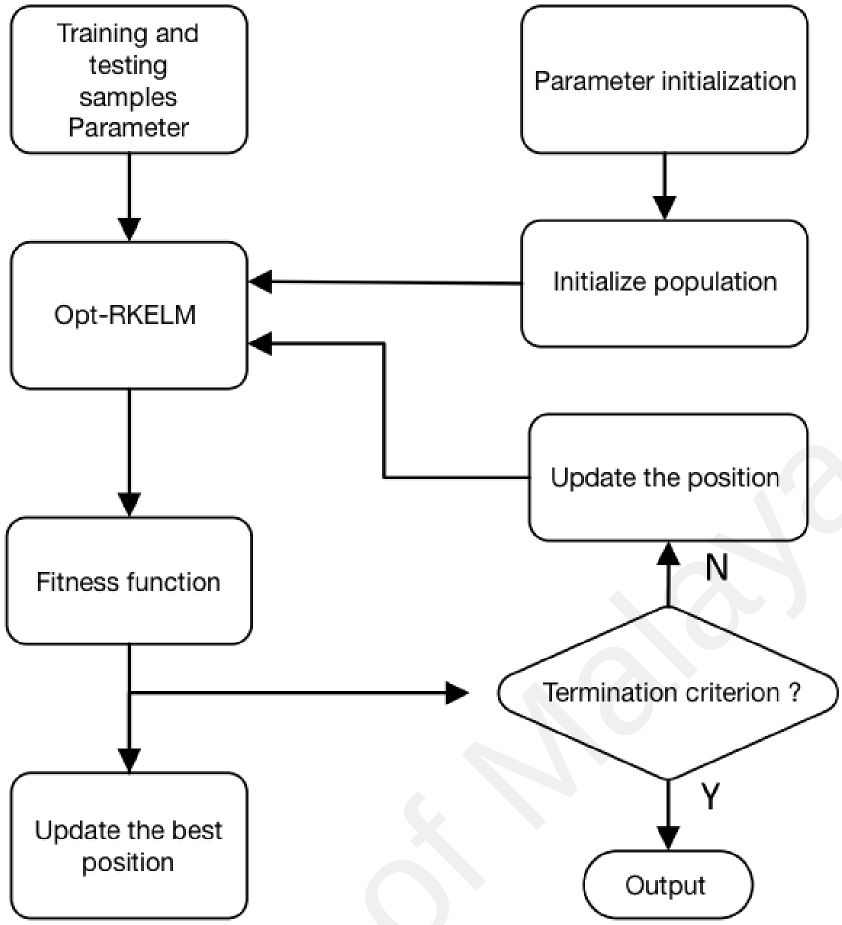

Figure 3.6: Flowchart of QPSO.

Where t represents the number of iterations, $X(t)$ represents t is the current position of the particle group, P is the best individual location, β is the contraction expansion coefficient (the convergence speed of the control algorithm), P_{mbest} is the optimal position for particle swarm evaluation: $U \in (0, 1)$. Suppose M is the population size, N is the dimension of the particle, and $P = [p_{i1}, p_{i2}, \cdots, p_{in}]$ is the optimal position of the ith particle. then:

$$P_{mbest} = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \ldots \frac{1}{M}\sum_{i=1}^{M} P_{in}\right) \tag{3.14}$$

And each particle converges to its own random point P:

$$P = \alpha P_i + (1 - \alpha) P_g \tag{3.15}$$

Among them, P_i represents the best position of the individual history of the particle, p_g represents the global optimal position, $\alpha \in (0, 1)$.

It is well known that parameters in an algorithm can affect performance. Therefore, QPSO(J. Sun et al., 2004) is used to optimize the parameters of the kernel functions. The dimension of the searching space corresponds to the number of parameters of the RKELM with different kernel functions, and the position of each particle represents the parameter value of the kernel function. Because QPSO can help find the best kernel parameter for RKELM, we use QPSO to find the corresponding parameters. Figure 3.7 shows the error rate reduce and become stable after 50 iterations using QPSO for HARUS human activity dataset.

Algorithm 2 QPSO algorithm

```
Initial on Opt-RKELM classifier kernel parameter;
Uniformly randomly generatean initial population with positions;
Initial pbest, gbest, mbest, and the fitness values;
repeat
Update the coefficient;
Calculate the mean best position mbest ;
for i = 1 to N do
Update pbest_i and gbest ;
Generate the random number φ and u;
Calculate local attractor p_ij;
for j = 1toDdo
Update the position of particle i;
end for
Update the fitness value of particle i;
end for
until termination criterion is met;
Based on best position update parameter to Opt-RKELM.
```

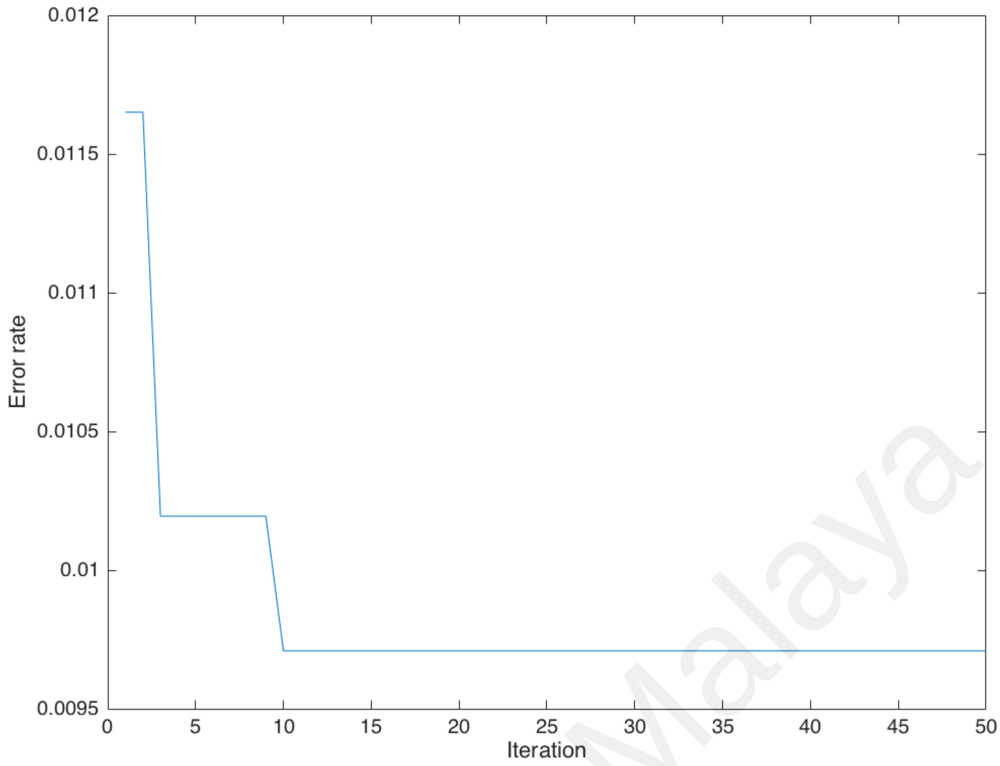

Figure 3.7: Using QPSO to optimize error rate, after 10 iterations until 50 iterations stabled at lower than 0.01 error rate

3.6 Summary

In this chapter we have described our proposed method, the methodology of human activity recognition system, named Opt-RKELM, The proposed method can deal with raw phone sensor data and recognize the activity. In section 3.2 we gave an overall overview of how the system works. We explained detail in the rest of the chapter. We first give pre-processing in section 3.3, and split up to ReliefF and K-Means for both feature and data dimensions in section 3.3.1 and section 3.3.2. For main classifier Opt-RKELM we explained how we better than Deng's RKELM in section 3.4.1. Optimization by using QPSO explained in section 3.5. In the next Chapters, we will explain Experiments and Implementation and results.

CHAPTER 4: EXPERIMENTS

4.1 Introduction

In this chapter, experiments are conducted. we discuss all the details that are not mentioned in the methodology, Such as, how our research build, the environment we set up, how to prepare the database, what parameter we used, how to verify that our method is good enough.

4.2 Datasets

We used two Human activity recognition benchmark Datasets to test our method. They are download from the UCI Machine Learning Repository, they are widely used in various HAR system.

Figure 4.1: Overview of Activities

4.2.1 Human Activity Recognition Using Smartphones Dataset

The first database we used was named Human Activity Recognition Using Smartphones (HARUS), Created by Anguita et al. (2013). The database collected a group of 30 volunteers, aged 19-48 years old, each of them was asked to do 6 activities, Walking, Walking upstairs, Walking downstairs, Sitting, Standing, Laying, showing in figure 4.1, They are carrying a Samsung Galaxy series smartphone while conducting the experiments.

In this smartphone, built-in accelerometer and gyroscope sensors, authors captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. They are recorded as video clips and label the data manually. Data collected are randomly divided into 2 sets: the Training set selects 70% and the remaining 30% is used for Testing.

The accelerometer and gyroscope sensor signals were pre-processed by applying a noise filter and then sampled in a fixed width sliding window (128 readings/window) of 2.56 seconds and 50% overlap. The sensor acceleration signal has a gravity and body motion component that is separated into body acceleration and gravity using a Butterworth low pass filter. It is assumed that gravity has only a low-frequency component, so a filter having a cutoff frequency of 0.3 Hz is used. From each window, the feature vector is obtained by computing variables from the time domain and the frequency domain.

Furthermore, in each window frame, a vector of 17 features is obtained by calculating variables from the accelerometer and gyroscope signals in the time and frequency domain. These features include such as mean, standard deviation, energy, mean-crossing rate and so on ..., finally, to create a total 561 feature vector. Consequently, each record in the dataset includes:

- Triaxial acceleration from accelerometer (total acceleration) and estimated body acceleration.
- Triaxial angular velocity from the gyroscope.

Figure 4.2: Example of the performed basic activities during the collection of experimental data. From left to right and top to bottom: standing, sitting, lying down, walking, walking downstairs and walking upstairs

- A feature vector of 561 values.
- The corresponding label for each action.
- The ID of the subject who performed the experiment.

Table 4.1 shows the number of samples obtained for each activity. There are total of 10299 samples, 7352 records collected for training and 2947 records for testing. Each record contains 128 signals and 9 data fields, where 3 fields to represent a total acceleration from the accelerometer signals, 3 fields to represent body acceleration signals, and the other 3 fields to represent gyroscope signals).

Table 4.1: HARUS datasets information.

Activity Name	Label	Number of samples
Walking	1	1722
Walking Upstairs	2	1544
Walking Downstair	3	1406
Sitting	4	1777
Standing	5	1906
Laying	6	1944
Total Samples		10299

4.2.2 Human Activities and Postural Transitions Dataset

The postural transitions is a short-lived motion that describes a change in state from one static pose to another. In several HAR systems, these transitions cannot be ignored due to their significant incidence in the duration of other basic activities. Posture transitions are events with limited duration. They are characterized by start and end times, usually slightly different from one person to another. Moreover, these events are bounded between the other two activities and correspond to the transition period between them. Instead, basic activities, such as standing and walking, can be extended indefinitely. The data collection for these two types of activities is also different: the pose transformation needs to be performed repeatedly to obtain separate samples; on the contrary, the basic activities are continuous, thus allowing many (window) samples to be taken from a single test, limited only by their time range.

The second dataset we used is called Human Activities and Postural Transitions DataSet (HAPT), This dataset also creates from Jorge Luis team Reyes-Ortiz, Oneto, Samà, Parra, and Anguita (2016). They performed a protocol of activities composed of total twelve activities, six basic activities are: three static postures (standing, sitting, lying) and three dynamic activities (walking, walking downstairs and walking upstairs), also included six postural transitions activities that occurred between the static postures. These are stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand. They used the same calculation for obtaining features, also have 561 features in initial. The postural transitions activities are shown in Figure 4.3

Figure 4.3: Postural Transitions activities: stand to sit, sit to stand, stand to lie, lie to sit, sit to lie, lie to stand.

4.3 Environment

All experiments are conducted on Macbook Pro 13' laptop 2016 version, 16GB RAM, simulations have been done on Matlab 2015b environment. Some of the results of the operation are using the Python environment, because there are a lot of excellent packages to generate intuitive and easy-to-understand results.

4.4 Process with Dataset

In the section, we explain data normalization, how we split datasets, and explain the features related to recognizing activity.

4.4.1 Data Normalization

The datasets were normalized to the range of [0-1] by applying the following formula to all features:

$$X' = \frac{X - X\min}{X\max - X\min} \tag{4.1}$$

Where X represents the feature values, $X\max$ is the sample maximum and $X\min$ is the sample minimum.

4.4.2 Data Split

Each dataset was split into a training set containing 70% of the observations and a test set containing the remaining 30%. The training set was used for training and tuning of the classifier, whilst the test set was used to evaluate the performance of the final system model.

4.4.3 Features

Table 4.2 shows how we calculate the signal through mathematical formulas. Our dataset has a total of 561 features. Most of them are common features. In the first table, there are some unusual feature calculation formulas. We are also Listed below. Through the ReliefF method in the previous chapter Methodology, We set parameter k-nearest neighbors per class (we used $k = 10$) and outputs a weighted ranking for the features, each candidate feature is assigned an importance weight ranging in [-1, 1] where a positive weight value indicates the feature can distinguish the instances, while a negative weight value means the feature overlaps the instances. Therefore, we use the features with positive weights as critical features in our experiments. The two databases selected 179 and 427 valid features for Opt-RKELM to process, which greatly reduced the calculation speed, but did not affect the accuracy too much. , the schematic shows Figure 3.2 and Figure 3.3.

4.5 Performance Measures

To evaluate our proposed method is better than comparative methods, we compare them in terms of the speed of training, the performance of prediction. a confusion matrix that contains actual and predicted outputs is applied (A. Wang, An, Chen, Li, & Alterovitz, 2015). the following four measures will show the performance, which are precision, recall, F1 and accuracy.

In order to evaluate our proposed method over the comparison method, we compare the

Table 4.2: List of measures for computing feature vectors.

Function	Description	Formulation
mean $(\boldsymbol{s})$	Arithmetic mean	$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i$
std $(\boldsymbol{s})$	Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2}$
mad $(\boldsymbol{s})$	Median absolute deviation	$\text{median}_i\left(\lvert s_i - \text{median}_j(s_j)\rvert\right)$
max $(\boldsymbol{s})$	Largest values in array	$\max_i(s_i)$
min $(\boldsymbol{s})$	Smallest value in array	$\min_i(s_i)$
skewness $(\boldsymbol{s})$	Frequency signal Skewness	$\mathrm{E}\left[\left(\frac{\boldsymbol{s}-\bar{s}}{\sigma}\right)^3\right]$
kurtosis $(\boldsymbol{s})$	Frequency signal Kurtosis	$\frac{\mathrm{E}\left[(\boldsymbol{s}-\bar{s})^4\right]}{\mathrm{E}\left[(\boldsymbol{s}-\bar{s})^2\right]^2}$
maxFreqInd $(\boldsymbol{s})$	Largest frequency component	$\arg\max_i(s_i)$
energy $(\boldsymbol{s})$	Average sum of the squares	$\frac{1}{N}\sum_{i=1}^{N} s_i^2$
sma $(\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3)$	Signal magnitude area	$\frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{N}\lvert s_{i,j}\rvert$
entropy $(\boldsymbol{s})$	Signal Entropy	$\sum_{i=1}^{N}(c_i \log(c_i)),\ c_i = s_i / \sum_{j=1}^{N} s_j$
iqr $(\boldsymbol{s})$	Interquartile range	$\mathrm{Q3}(\boldsymbol{s}) - \mathrm{Q1}(\boldsymbol{s})$
autoregression $(\boldsymbol{s})$	4th order Burg Autoregression coefficients	$\boldsymbol{a} = \text{arburg}(\boldsymbol{s}, 4), \boldsymbol{a} \in \mathbb{R}^4$
correlation $(\boldsymbol{s}_1, \boldsymbol{s}_2)$	Pearson Correlation coefficient	$C_{1,2}/\sqrt{C_{1,1}C_{2,2}}, C = \text{cov}(\boldsymbol{s}_1, \boldsymbol{s}_2)$
meanFreq $(\boldsymbol{s})$	Frequency signal weighted average	$\sum_{i=1}^{N}(i s_i) / \sum_{j=1}^{N} s_j$
energyBand $(\boldsymbol{s}, a, b)$	Spectral energy of a frequency band $[a, b]$	$\frac{1}{a-b+1}\sum_{i=a}^{b} s_i^2$
angle $(\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, v)$	Angle between triaxial signal mean and vector	$\tan^{-1}\left(\lVert[\bar{s}_1, \bar{s}_2, \bar{s}_3] \times \boldsymbol{v}\rVert, [\bar{s}_1, \bar{s}_2, \bar{s}_3] \cdot \boldsymbol{v}\right)$

speed of training, the speed of testing, predictive performance, and stability in results. We also applied the most common method for evaluation, which is the confusion matrix. The confusion matrix allows algorithm performance to be represented by clearly identifying the type of error (false positives and negatives) and the samples that are correctly predicted on the test data. From this, you can also extract various indicators such as model accuracy, accuracy, and F1-Score(Bulling, Blanke, & Schiele, 2014; Lara, Labrador, et al., 2013). In addition, other comparative qualitative indicators, such as the number of available activities, are also presented in the results.

We can visualize four different helpful values used to estimate the various statistical measures regarding the system performance. These are visible in the simplified confusion matrix of two classes (a and b) from Table 4.3. We take these values assuming class a is the class of interest or positive condition.

Table 4.3: The four fundamental numbers for estimating statistical performance measures of a classifier

		Predicted Class	
		a	b
Actual Class	a	True Positives	False Negatives
	b	False Positives	True Negatives

- True Positives (TP): actual samples of class a correctly predicted as class a
- True Negatives (TN): actual samples of class b correctly predicted as class b
- False Positives (FP): actual samples of class b incorrectly predicted as class a
- False Negatives (FN): actual samples of class a incorrectly predicted as class b

Base on Table 4.3, can calculate Precision, Recall, F1 Score, and accuracy:

- Precision represents the weighted average of the scores of the inferred activity tags that are correctly predicted for each activity class.
- Recall is the weighted average of the scores of the real active tags that are correctly categorized for each activity class.
- F1 Score provides a way to combine precision and recall into a single metric.

4.5.1 Accuracy

The generalization performance of the algorithms on classification datasets is measured by classification accuracy defined as:

$$Accuracy = \frac{N_{Accurate}}{N_{Total}} \times 100\% \tag{4.2}$$

where $N_{Accurate}$ is the number of correctly classified samples while N_{Total} is the number of total testing samples.

4.6 Summary

In this chapter, we described the background required to contextualize the problem of HAR. We first covered the two benchmark datasets we used, Then, we explored the environment we set up, how to use the dataset, and talk about features from this datasets. In the end, we discussed the evaluation methods, the Confusion Matrix, Classification report, Precision, Recall, F1 Score, and Accuracy will use to measure our system. In the next chapter, we will analyze the results and discuss the advantages of our method.

CHAPTER 5: RESULT AND DISCUSSION

5.1 Introduction

This chapter discusses the experimental results of the study. Our results section will focus on how to validate our method. By verifying both two sets of databases, we had improved the accuracy, but the calculation time is very less, even if it is faster than RKELM, We also can prove the validity of ReliefF by comparing the presence or absence of ReliefF method. By displaying the QPSO curve to display the appropriate parameter, the accuracy of the entire algorithm can be changed. Finally, by comparing ELM, KELM, RKELM and other methods, the Accuracy of our method is The highest, while the speed is amazingly fast.

5.2 Performance Comparison

In this section we will compare the performance of ELM(Huang et al., 2006), Huang's KELM(Huang et al., 2012) and Deng's RKELM(W. Deng et al., 2013) methods in activity recognition. All methods are implemented in the Matlab environment, ELM uses the sigmoid activation function $h(x, a, b) = 1/\left(1 + e^{-\gamma(a \cdot x + b)}\right)$; the Gaussian kernel

Table 5.1: Results for HARUS datasets.

	SVM	ELM	KELM	RKELM	Opt-RKELM
Training time (s)	-	4.911	8.6619	1.9967	**1.9598**
Testing time (s)	-	0.21	1.0924	0.33	**0.1999**
Training accuracy (%)	-	97.89	99.73	98.62	**99.51**
Testing accuracy (%)	96.00	96.45	99.13	98.05	**99.17**
Testing accuracy Std	-	0.0041	0	0.037	**0**

Table 5.2: Results for HAPT datasets.

	SVM	ELM	KELM	RKELM	Opt-RKELM
Training time (s)	-	4.484	8.4484	1.4331	**1.2957**
Testing time (s)	-	0.186	1.053	0.124	**0.1334**
Training accuracy (%)	-	97.36	99.71	97.68	**98.76**
Testing accuracy (%)	94.20	95.13	97.90	95.24	**97.85**
Testing accuracy Std	-	0.0034	0	0.053	**0**

$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ are used in KELM, RKELM and our Opt-RKELM methods.

Regularized parameter C We use QPSO to find the best parameters. The parameters of the other methods used the parameters given by this paper(W.-Y. Deng et al., 2014). Each experiment was averaged 100 times to ensure stability. Table 5.1 lists the training time (seconds), test time (seconds), training accuracy and test accuracy.

It can be seen from Table 5.1 that Opt-RKELM is the most accurate and fast, its accuracy is 99.17% only take 1.9598 s to train. compare to Deng's RKELM, we are still faster in training and testing time and better testing accuracy. SVM results are getting from Saha et al. (2018), only provide accuracy to compare, but from other articles W.-Y. Deng et al. (2014) and comparative experience, SVM is slower than ELM in both training and testing. In addition, Opt-RKELM's training time and testing time is far less than ELM and Huang's KELM, while Huang's KELM and ELM consume 8.6619 seconds and 4.911 seconds respectively. The second best result is KELM, but our approach training and testing faster by approximately 5x and 3x, respectively, with slightly improved accuracy. Compare to Deng's RKELM, our model still has better performance, because we used the method used in preprocessing, we achieved the highest accuracy, but the time is shorter. This indicates that Opt-RKELM is much better than ELM, Huang's KELM and Deng's RKELM. In order to test the stability of different models, 100 simulation tests were performed. For each trial, the ratio of training and testing sets was determined based on 70%: 30%, but the order of the data was randomly shuffled.

From Table 5.2 can see similar results, Opt-RKELM improved more than 2% accuracy then Deng's RKELM, but using less time in training and testing, compare to Huang's KELM similar result in Testing accuracy, but faster in training and testing stage.

The advantages of Opt-RKELM are also manifested in stability, calculated by the standard deviation (Std), displayed in Table 5.1 and Table 5.2, shown OPT-RKELM has

Table 5.3: Classification Report for HARUS datasets.

Class ID	precision	recall	f1-score	support
A1	1.00	1.00	1.00	344
A2	1.00	1.00	1.00	306
A3	1.00	1.00	1.00	287
A4	0.96	0.96	0.96	358
A5	0.97	0.97	0.97	379
A6	1.00	1.00	1.00	385
avg / total	0.99	0.99	0.99	2059

Table 5.4: Class ID from A1 -A6 are: walking, walking upstairs, walking downstairs, sitting, standing, laying.

Table 5.5: Classification Report for HAPT datasets.

class ID	precision	recall	f1-score	support
A1	1.00	1.00	1.00	327
A2	1.00	1.00	1.00	313
A3	1.00	1.00	1.00	273
A4	0.97	0.96	0.97	365
A5	0.96	0.98	0.97	396
A6	1.00	1.00	1.00	397
A7	0.92	0.85	0.88	13
A8	1.00	1.00	1.00	4
A9	0.80	0.83	0.82	24
A10	0.84	0.70	0.76	23
A11	0.87	0.79	0.83	33
A12	0.65	0.76	0.70	17
avg / total	0.98	0.98	0.98	2185

Table 5.6: Class ID from A1 -A12 are: walking, walking upstairs, walking downstairs, sitting, standing, laying, stand to sit, sit to stand, sit to lie, lie to sit, stand to lie, lie to stand.

the same stability as KELM, through 100 times standard deviation calculations, in contrast, Deng's RKELM and Original ELM are unstable.

5.3 Classification Report

The classification report of HARUS Datasets and HAPT Datasets are shown in Table 5.3 and Table 5.5, precision, recall and F1-score are shown, as can observe, Opt-RKELM achieved very good results no matter which activity, For two benchmark dataset more than 4000 testing data, total 18 activities.

Table 5.7: Confusion Matrix for HARUS datasets.

	A1	A2	A3	A4	A5	A6
A1	343	1	0	0	0	0
A2	0	306	0	0	0	0
A3	0	1	287	0	0	0
A4	0	1	0	345	13	0
A5	0	0	0	13	366	0
A6	0	0	0	0	0	385

Table 5.8: Class ID from A1 -A6 are: walking, walking upstairs, walking downstairs, sitting, standing, laying.

5.4 Confusion Matrix

A common method to visualize the performance of a machine learning algorithm is through the Confusion Matrix. The confusion matrices for the HARUS dataset are shown in Table 5.7, For HAPT dataset, shown in Table 5.9. For more intuitive, we also created a heat map in Figure 5.1 and Figure 5.2.

From Table 5.7 and Figure 5.1, As can be observed, Opt-RKELM achieved very high accuracy, Only A4 and A5 which are sitting and standing are prone to confusion, because both actions are kept the body no move, difficulty in recognition, only if but the basic recognition rate of other activity can reach almost 100%.

For Table 5.9 and Figure 5.2, also A4 and A5, sitting and standing easy to confuse machine, also A9 and A11: sit to lie and stand to lie; A10 and A12:lie to sit and lie to stand, Which are also because of standing and sitting postural need more attention in future work. From both Classification Report and Confusion Matrix can prove, Optimized Reduced Kernel Extreme Learning Machine model are suitable for human activity recognition.

5.5 Review Contribution

As can be seen from the previous chapters, HAR has a large number of classification methods after a long period of research. For example, the use of large-scale deep learning, and the use of new methods such as SVM, ELM, KELM and other methods in recent

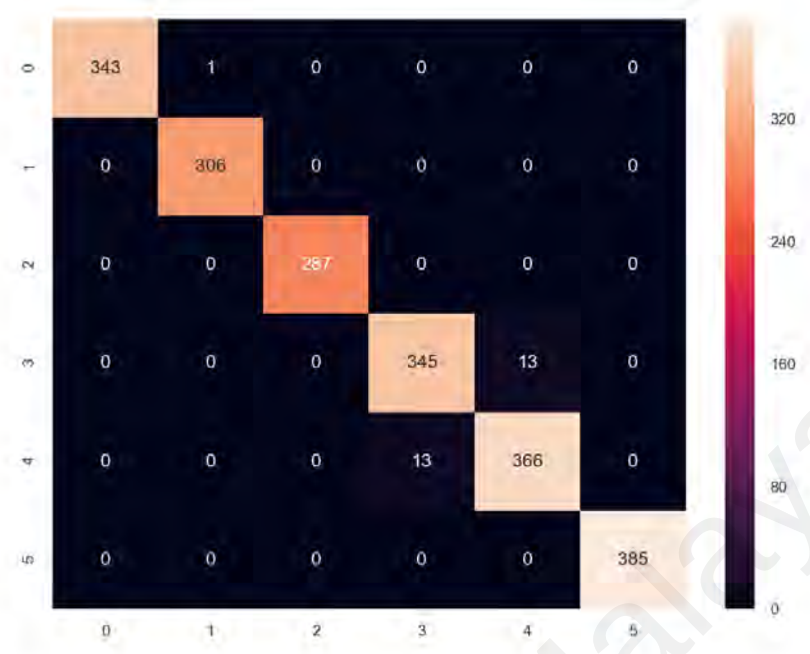

Figure 5.1: Confusion Matrix Heatmap for HARUS Dataset

years, the use of offline and online learning methods, the use of feature selection or other various optimization methods. We have also found some ways to further improve. With the excellent feature selection method, ReliefF can quickly find good features, while maintaining the accuracy of the scene while reducing the amount of computation; preprocessing the dataset by K-Means to reduce the amount of computation at the classifier level; by using Reduced KELM To achieve the best results, fast, accurate and stable. Also, use QPSO to select a good parameter to eliminate the impact of the parameter.

5.6 Summary

In this chapter, we have presented the results of our experiments by using training time and testing time, accuracy and showing in classification report and confusion matrix. In results showing compare to Deng's RKELM our method Opt-RKELM are stable and faster, can high accuracy for both two benchmark datasets even than KELM.

Table 5.9: Confusion Matrix for HAPT datasets.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	27	0	0	0	0	0	0	0	0	0	0	0
A2	0	313	0	0	0	0	0	0	0	0	0	0
A3	1	0	272	0	0	0	0	0	0	0	0	0
A4	0	0	0	351	14	0	0	0	0	0	0	0
A5	0	0	0	8	388	0	0	0	0	0	0	0
A6	0	0	0	0	0	397	0	0	0	0	0	0
A7	0	0	0	0	2	0	11	0	0	0	0	0
A8	0	0	0	0	0	0	0	4	0	0	0	0
A9	0	0	0	0	0	0	0	0	20	0	4	0
A10	0	0	0	0	0	0	0	0	0	16	0	7
A11	0	0	0	1	0	1	1	0	4	0	26	0
A12	0	0	0	0	0	0	0	0	1	3	0	13

Table 5.10: Class ID from A1 -A12 are: walking, walking upstairs, walking downstairs, sitting, standing, laying, stand to sit, sit to stand, sit to lie, lie to sit, stand to lie, lie to stand.

provide

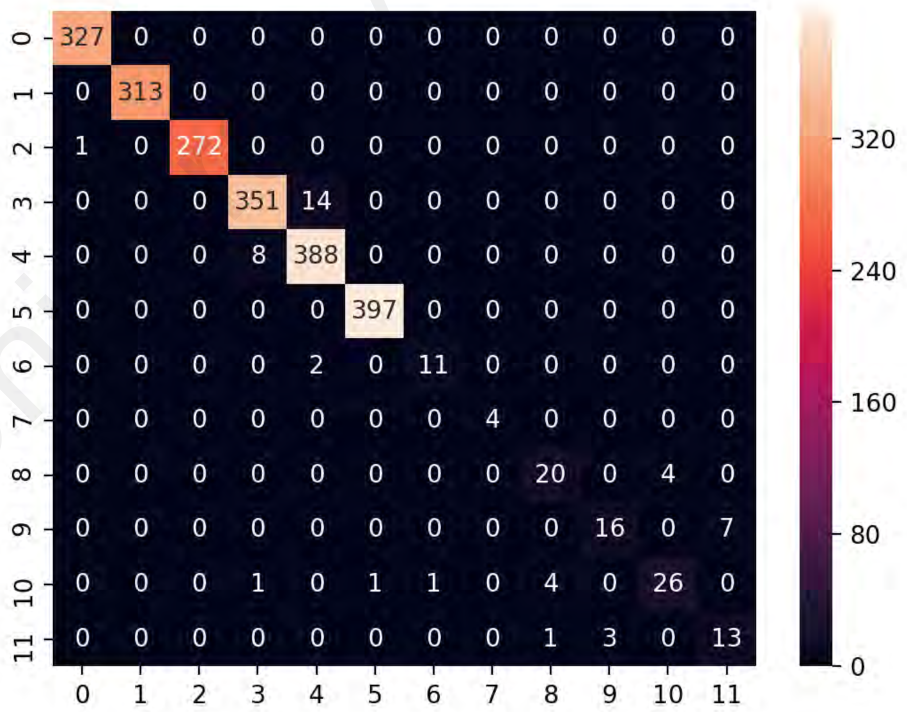

Figure 5.2: Confusion Matrix Heatmap for HAPT Dataset

CHAPTER 6: CONCLUSION AND FUTURE WORKS

In this chapter, the comprehensive summary of this thesis is presented, furthermore, the contribution of this thesis are discussed. Lastly, future works are discussed from the viewpoint of the limitations of the models in this thesis.

6.1 Conclusions

The rapid development of smartphone technology and the growing maturity of sensor technology have made smartphones with embedded sensors an ideal platform for activity recognition and activity monitoring. In order to recognize the activity, machine learning and artificial intelligence is the key to make it work. ELM is a popular research topic in the field of machine learning and artificial intelligence because of its simple structure, fast training speed, strong generalization ability, and strong general classification ability.

In this Thesis, an improved Opt-RKELM method for human activity recognition from raw inertial sensors signal data (accelerometers and gyroscopes) system were proposed. Through the study of the recent methods, in order to further improve the accuracy and stability of activity recognition and shorten the training time of the activity recognition model, the proposed Opt-RKELM method did some contributions to the activity recognition field:

- By using the ReliefF method, the number of effective features is successfully reduced, the calculation efficiency is improved, and the computing resources are saved.
- By using the K-Means clustering method, the core 10% of the entire database in were calculated in advance, so the random selection for kernel matrix is replaced with the calculated result, which greatly improves the accuracy and is stable.

- By using QPSO to eliminate the effects of parameter inaccuracies, the classifier is guaranteed the best answer.

We test our methods by used Two benchmark database were tested in the study which has 18 human activities, The experiment results have shown that the proposed method is the best in performance among others ELM family models, which it proved that Opt-RKELM system is effective to recognize human activity.

6.2 Future Works

There are several possible future research directions that can be explored: There may be differences between different individuals, and how to identify different system for each person to improve the way to identify more complex movements.

Real-Time Implementation of Opt-RKELM for human activity recognition will be a great contribution. Since Opt-RKELM algorithm very light can be either embedded in to a smartphone application into other microprocessors of wearable devices

Complex and coherent movements have more learnable features, how to extract and use these features. Is it possible to identify human movements through a more powerful environment, such as combining image, video with the body sensor or environment sensor, and even identifying the activities through Wi-Fi, how to deal with these challenges?

Finally, how human motion recognition can be extended to more precise uses, such as the prediction of falls for the elderly, or how to make VR glasses, how to optimize through the sensor. We will conduct more research to think about and solve these problems in the future.

REFERENCES

Aggarwal, J. K., & Xia, L. (2014). Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, *48*, 70–80.

Al Jeroudi, Y., Ali, M., Latief, M., & Akmeliawati, R. (2015). Online sequential extreme learning machine algorithm based human activity recognition using inertial data. In *2015 10th asian control conference (ascc)* (pp. 1–6).

Altun, K., Barshan, B., & Tunçel, O. (2010). Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, *43*(10), 3605–3620.

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *Esann.*

Anjum, A., & Ilyas, M. U. (2013). Activity recognition using smartphone sensors. In *Consumer communications and networking conference (ccnc), 2013 ieee* (pp. 914–919).

Brezmes, T., Gorricho, J.-L., & Cotrina, J. (2009). Activity recognition from accelerometer data on a mobile phone. In *International work-conference on artificial neural networks* (pp. 796–799).

Bulling, A., Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, *46*(3), 33.

Camastra, F., & Verri, A. (2005). A novel kernel method for clustering. *IEEE transactions on pattern analysis and machine intelligence*, *27*(5), 801–805.

Cambria, E., & Hussain, A. (2012). *Sentic computing: Techniques, tools, and applications* (Vol. 2). Springer Science & Business Media.

Capela, N. A., Lemaire, E. D., & Baddour, N. (2015). Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients. *PloS one*, *10*(4), e0124414.

Chacko, B. P., Krishnan, V. V., Raju, G., & Anto, P. B. (2012). Handwritten character recognition using wavelet energy and extreme learning machine. *International Journal of Machine Learning and Cybernetics*, *3*(2), 149–161.

Chaquet, J. M., Carmona, E. J., & Fernández-Caballero, A. (2013). A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, *117*(6), 633–659.

Chen, L., & Khalil, I. (2011). Activity recognition: Approaches, practices and trends. In *Activity recognition in pervasive intelligent environments* (pp. 1–31). Springer.

Chen, Y., Qi, J., Sun, Z., & Ning, Q. (2010). Mining user goals for indoor location-based services with low energy and high qos. *Computational Intelligence*, *26*(3), 318–336.

Chen, Y., Zhao, Z., Wang, S., & Chen, Z. (2012). Extreme learning machine-based device displacement free activity recognition model. *Soft Computing*, *16*(9), 1617–1625.

Cheng, Q., Juen, J., Li, Y., Prieto-Centurion, V., Krishnan, J. A., & Schatz, B. R. (2013). Gaittrack: Health monitoring of body motion from spatio-temporal parameters of simple smart phones. In *Proceedings of the international conference on bioinformatics, computational biology and biomedical informatics* (p. 897).

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, *6*(1), 58–73.

Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal*, *20*(4), 364–366.

Deng, W., Zheng, Q., & Chen, L. (2009). Regularized extreme learning machine. In *Computational intelligence and data mining, 2009. cidm'09. ieee symposium on* (pp. 389–395).

Deng, W., Zheng, Q., & Zhang, K. (2013). Reduced kernel extreme learning machine. In *Proceedings of the 8th international conference on computer recognition systems cores 2013* (pp. 63–69).

Deng, W.-Y., Zheng, Q.-H., & Wang, Z.-M. (2014). Cross-person activity recognition using reduced kernel extreme learning machine. *Neural Networks*, *53*, 1–7.

Dijkstra, B., Kamsma, Y. P., & Zijlstra, W. (2010). Detection of gait and postures using a miniaturized triaxial accelerometer-based system: accuracy in patients with mild to moderate parkinson's disease. *Archives of physical medicine and rehabilitation*, *91*(8), 1272–1277.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).

Fu, T., & Sun, Y.-M. (2011). Pso-based k-means algorithm and its application in network intrusion detection system. *Computer Science*, *38*(5), 54.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, *13*(3), 780–784.

Gupta, P. (2014). *Algorithm development and implementation of activity recognition system utilizing wearable mems sensors* (Unpublished doctoral dissertation).

Gupta, P., & Dallas, T. (2014). Feature selection and activity recognition system using a single triaxial accelerometer. *IEEE Transactions on Biomedical Engineering*, *61*(6), 1780–1786.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157–1182.

Hammerla, N. Y., Kirkham, R., Andras, P., & Ploetz, T. (2013). On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proceedings of the 2013 international symposium on wearable computers* (pp. 65–68).

HanYing, Hao, Z., Jian, L., , C. L., , , & . (2018). Activity recognition based on online sequential kernel extreme learning machine elm . *Microelectronics & Computer*, *35*(1), 91–95.

Hernández, J., Cabido, R., Montemayor, A. S., & Pantrigo, J. J. (2014). Human activity recognition based on kinematic features. *Expert Systems*, *31*(4), 345–353.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Huang, G.-B., & Babri, H. A. (1998). Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks*, *9*(1), 224–229.

Huang, G.-B., & Chen, L. (2007). Convex incremental extreme learning machine. *Neurocomputing*, *70*(16-18), 3056–3062.

Huang, G.-B., & Chen, L. (2008). Enhanced random search based incremental extreme learning machine. *Neurocomputing*, *71*(16-18), 3460–3468.

Huang, G.-B., Ding, X., & Zhou, H. (2010). Optimization method based extreme learning machine for classification. *Neurocomputing*, *74*(1-3), 155–163.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(2), 513–529.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural networks, 2004. proceedings. 2004 ieee international joint conference on* (Vol. 2, pp. 985–990).

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*(1-3), 489–501.

Iosifidis, A., Tefas, A., & Pitas, I. (2013). Dynamic action recognition based on dynemes and extreme learning machine. *Pattern Recognition Letters*, *34*(15), 1890–1898.

Jalal, A., Uddin, M. Z., & Kim, T.-S. (2012). Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Transactions on Consumer Electronics*, *58*(3), 863–871.

JIA, H., DING, S., & SHI, Z. (2015). Approximate weighted kernel k-means for large-scale spectral cluste-ring. *Journal of Software*, *26*(11), 2836–2846.

Jiang, W., & Yin, Z. (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd acm international conference on multimedia* (pp. 1307–1310).

Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *European conference on machine learning* (pp. 171–182).

Kumar, R. C., Bharadwaj, S. S., Sumukha, B., & George, K. (2016). Human activity recognition in cognitive environments using sequential elm. In *2016 second international conference on cognitive computing and information processing (ccip)* (pp. 1–6).

Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, *12*(2), 74–82.

Lara, O. D., Labrador, M. A., et al. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, *15*(3), 1192–1209.

Liu, H., & Motoda, H. (2007). *Computational methods of feature selection*. CRC Press.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, *28*(2), 129–137.

Mannini, A., & Sabatini, A. M. (2010). Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, *10*(2), 1154–1175.

Minhas, R., Baradarani, A., Seifzadeh, S., & Wu, Q. J. (2010). Human action recognition using extreme learning machine based on visual vocabularies. *Neurocomputing*, *73*(10-12), 1906–1917.

Mital, P. K., Smith, T. J., Hill, R. L., & Henderson, J. M. (2011). Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, *3*(1), 5–24.

Mohammed, A. A., Minhas, R., Wu, Q. J., & Sid-Ahmed, M. A. (2011). Human face recognition based on multidimensional pca and extreme learning machine. *Pattern Recognition*, *44*(10-11), 2588–2597.

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, *13*(6), 47–60.

Murad, A., & Pyun, J.-Y. (2017). Deep recurrent neural networks for human activity recognition. *Sensors*, *17*(11), 2556.

Ning, H., Shi, F., Zhu, T., Li, Q., & Chen, L. (2019). A novel ontology consistent with acknowledged standards in smart homes. *Computer Networks*, *148*, 101–107.

Ozcan, E., & Mohan, C. K. (1999). Particle swarm optimization: surfing the waves. In *Proceedings of the 1999 congress on evolutionary computation-cec99 (cat. no. 99th8406)* (Vol. 3, pp. 1939–1944).

Pan, C., Park, D. S., Yang, Y., & Yoo, H. M. (2012). Leukocyte image segmentation by visual attention and extreme learning machine. *Neural Computing and Applications*, *21*(6), 1217–1227.

Peng, C., Yan, J., Duan, S., Wang, L., Jia, P., & Zhang, S. (2016). Enhancing electronic nose performance based on a novel qpso-kelm model. *Sensors*, *16*(4), 520.

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. (2005). Activity recognition from accelerometer data. In *Aaai* (Vol. 5, pp. 1541–1546).

Reyes-Ortiz, J.-L., Oneto, L., Samà, A., Parra, X., & Anguita, D. (2016). Transition-aware human activity recognition using smartphones. *Neurocomputing*, *171*, 754–767.

Roy, S., & Bhattacharyya, D. (2005). An approach to find embedded clusters using density based techniques. In *International conference on distributed computing and internet technology* (pp. 523–535).

Saha, S. S., Rahman, S., Rasna, M. J., Zahid, T. B., Islam, A. M., & Ahad, M. A. R. (2018). Feature extraction, performance analysis and system design using the du mobility dataset. *IEEE Access*, *6*, 44776–44786.

Schmidt, L. A., & Trainor, L. J. (2001). Frontal brain electrical activity (eeg) distinguishes valence and intensity of musical emotions. *Cognition & Emotion*, *15*(4), 487–500.

Seera, M., Loo, C. K., & Lim, C. P. (2014). A hybrid fmm-cart model for human activity recognition. In *2014 ieee international conference on systems, man, and cybernetics (smc)* (pp. 182–187).

Sibson, R. (1973). Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, *16*(1), 30–34.

Subasi, A. (2013). Classification of emg signals using pso optimized svm for diagnosis of neuromuscular disorders. *Computers in biology and medicine*, *43*(5), 576–586.

Sun, J., Feng, B., & Xu, W. (2004). Particle swarm optimization with particles having quantum behavior. In *Evolutionary computation, 2004. cec2004. congress on* (Vol. 1, pp. 325–331).

Sun, L., Zhang, D., Li, B., Guo, B., & Li, S. (2010). Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In *International conference on ubiquitous intelligence and computing* (pp. 548–562).

Sun, Z.-L., Choi, T.-M., Au, K.-F., & Yu, Y. (2008). Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, *46*(1), 411–419.

Taylor, J. G. (2009). Cognitive computation. *Cognitive Computation*, *1*(1), 4–16.

Tharwat, A., & Hassanien, A. E. (2018). Chaotic antlion algorithm for parameter optimization of support vector machine. *Applied Intelligence*, *48*(3), 670–686.

Tharwat, A., Mahdi, H., Elhoseny, M., & Hassanien, A. E. (2018). Recognizing human activity in mobile crowdsensing environment using optimized k-nn algorithm. *Expert Systems With Applications*, *107*, 32–44.

Wang, A., An, N., Chen, G., Li, L., & Alterovitz, G. (2015). Predicting hypertension without measurement: A non-invasive, questionnaire-based approach. *Expert Systems with Applications*, *42*(21), 7601–7609.

Wang, A., Chen, G., Yang, J., Zhao, S., & Chang, C.-Y. (2016). A comparative study on human activity recognition using inertial sensors in a smartphone. *IEEE Sensors Journal*, *16*(11), 4566–4578.

Wang, C., Xu, Y., Liang, H., Huang, W., & Zhang, L. (2018). Woody: A post-process method for smartphone-based activity recognition. *IEEE Access*, *6*, 49611–49625.

Ward, J. A., Lukowicz, P., & Gellersen, H. W. (2011). Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(1), 6.

wei Wang, H., Kong, B., & ying Zheng, X. (2010, November). An improved reduced support vector machine. In *Proc. computing and telecommunications 2010 ieee youth conf. information* (pp. 170–173). doi: 10.1109/YCICT.2010.5713072

Wöllmer, M., Eyben, F., Graves, A., Schuller, B., & Rigoll, G. (2010). Bidirectional lstm networks for context-sensitive keyword detection in a cognitive virtual agent framework. *Cognitive Computation*, *2*(3), 180–190.

Yin, J., Tian, G., Feng, Z., & Li, J. (2014). Human activity recognition based on multiple order temporal information. *Computers & Electrical Engineering*, *40*(5), 1538–1551.

Zappi, P., Lombriser, C., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., & Tröster, G. (2008). Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *European conference on wireless sensor networks* (pp. 17–33).

Zhang, J., Chen, M., Zhao, S., Hu, S., Shi, Z., & Cao, Y. (2016). Relieff-based eeg sensor selection methods for emotion recognition. *Sensors*, *16*(10), 1558.

Zhang, L., & Zhang, D. (2017). Evolutionary cost-sensitive extreme learning machine. *IEEE transactions on neural networks and learning systems*, *28*(12), 3045–3060.

Zhang, L., Zhou, W.-D., & Jiao, L. (2002). Kernel clustering algorithm. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-*, *25*(6), 587–590.

Zhao, S., Li, W., & Cao, J. (2018). A user-adaptive algorithm for activity recognition based on k-means clustering, local outlier factor, and multivariate gaussian distribution. *Sensors*, *18*(6), 1850.

Zheng, W.-L., & Lu, B.-L. (2015). Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, *7*(3), 162–175.

Zheng, Y. (2015). Human activity recognition based on the hierarchical feature selection and classification framework. *Journal of Electrical and Computer Engineering*, *2015*, 34.

Zong, W., & Huang, G.-B. (2011). Face recognition based on extreme learning machine. *Neurocomputing*, *74*(16), 2541–2551.