

School of Computing

FACULTY OF ENGINEERING



UNIVERSITY OF LEEDS

Analysing Wearable Sensor Data for Human Activity Recognition

Shanrui Li

**Submitted in accordance with the requirements for the degree of
MSc Advanced Computer Science**

2018/2019

The candidate confirms that the following have been submitted:

| Items | Format | Recipient(s) and Date |
|----------------|---|--|
| Deliverables 1 | Report | SSO (04/09/19) |
| Deliverable 2 | https://github.com/yoursherry/sc18sl.git | Vania Dimitrova, Isolde Adler (04/09/19) |

Type of Project: Empirical Investigation

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _____

Summary

With the development of electronic technology and sensing technology, increasingly intelligent electronic devices are integrated with micro inertial sensors. This makes wearable inertial sensor-based human activity recognition research of great application value, such as medical health, human-computer interaction, sports and games and other fields. Activity recognition based on wearable devices is an emerging research direction in the field of pattern recognition, which is essentially a process of human motion data acquisition, feature extraction and classification recognition. Inertial motion information includes acceleration and angular velocity information, which are ubiquitous in daily life. Compared with activity recognition based on visual information, inertial activity information can more directly reflect the meaning of motion, and is less influenced by the environment.

HuGaDB (Human Gait Database), an existing dataset, was used for this project. The dataset adopts MPU9250 inertial sensor module. By knowing the principle of inertial sensor data acquisition, help to understand the value of the dataset. The zero-score normalization method is used to scale the numerical range of sensor data. Feature extraction is to find the feature quantity that can best distinguish different activities. In this project, sliding window technique is used to extract several time-domain characteristic values. Six different classifiers were selected as data classification algorithms, and the performance of the classifier was evaluated by the 10-fold cross-validation method on WEKA platform. By comparing the recognition results of walking, running, cycling, standing and sitting, the experiment shows that the recognition precision reaches more than 95%.The recognition performance of RandomForest is the best.

Acknowledgements

First of all, I would like to express my deep gratitude to my supervisor Dr Vania Dimitrova who has been providing help and support to my MSc Project. The constructive guidance and valuable advice over the past three months have been of great help to my project. Besides, I would like to thanks my assessor Dr Isolde Adler for giving me useful feedback and suggestions.

Secondly, I would like to thanks all my teachers in School of Computing who have helped me directly and indirectly in my studies during the postgraduate years. I have benefited a lot.

Furthermore, I am greatly indebted to my parents who have been supporting, assisting and caring for me and always believed in me throughout my life.

Finally, special thanks also extend to my friends and classmates who have given me the warm help and precious time to work out my problems during the writing of this report. Thanks to their company, make this year much more fun.

Table of Contents

| | |
|---|-------------|
| Summary..... | iii |
| Acknowledgements..... | iv |
| Table of Contents | v |
| List of Figures | vii |
| List of Tables | viii |
| Chapter 1 Introduction..... | 1 |
| 1.1 Problem Statement..... | 1 |
| 1.2 Aim and Objectives | 2 |
| 1.3 Deliverables | 3 |
| 1.4 Methodology..... | 3 |
| 1.5 Project Schedule | 4 |
| 1.6 Degree Relevance..... | 4 |
| 1.7 Report Outline | 5 |
| Chapter 2 Background Research..... | 6 |
| 2.1 Human Activity Recognition | 6 |
| 2.2 Human Activity Recognition – Human Gait | 7 |
| 2.3 Wearable Sensors..... | 8 |
| 2.3.1 Inertial Sensor | 9 |
| 2.3.2 Surface Electromyography | 10 |
| 2.4 Recognition Methods..... | 11 |
| 2.4.1 Feature Engineering..... | 11 |
| 2.4.1.1 Time-Domain Features..... | 12 |
| 2.4.1.2 Frequency-Domain Features..... | 12 |
| 2.4.1.3 Sliding Window Technique | 12 |
| 2.4.1.4 Features Selection | 13 |
| 2.4.2 Machine Learning Algorithms..... | 13 |
| 2.4.2.1 Evaluation Metrics | 18 |
| Chapter 3 Experimental Setup | 20 |
| 3.1 Dataset Selection | 20 |
| 3.2 Dataset Description | 20 |
| 3.2.1 Data Collection..... | 21 |
| 3.2.2 Data Format | 22 |
| 3.3 Software tools | 23 |

| | |
|--|-----------|
| 3.3.1 Programming Language - Python..... | 23 |
| 3.3.2 Data Analysis Tool - WEKA..... | 24 |
| Chapter 4 Implementation | 26 |
| 4.1 Data Pre-processing..... | 26 |
| 4.1.1 Data Cleaning | 26 |
| 4.1.2 Data Normalization..... | 27 |
| 4.1.3 Data Normalization..... | 28 |
| 4.1.4 Pre-processing in WEKA..... | 28 |
| 4. 2 Implementation of Features..... | 29 |
| 4. 3 Model Building..... | 30 |
| Chapter 5 Evaluation..... | 31 |
| 5.1 Cross-validation | 31 |
| 5.2 Classification Results | 31 |
| 5.3 Summary..... | 33 |
| 5.4 Factors that improve accuracy | 34 |
| Chapter 6 Conclusion | 36 |
| 6.1 Project Summary..... | 36 |
| 6.2 Achievement of Objectives..... | 36 |
| 6.3 Recommendation of Future work | 37 |
| List of References | 40 |
| Appendix A HuGaDB Dataset..... | 45 |
| Appendix B Ethical Issues Addressed | 46 |
| Appendix C WEKA | 47 |
| Appendix D Project Schedule | 53 |

List of Figures

| | |
|--|----|
| Figure 1: Agile methodology | 3 |
| Figure 2: Project Schedule | 4 |
| Figure 3: MPU9250 Sensor Module..... | 9 |
| Figure 4: Axial direction of triaxial gyroscope and triaxial acceleration sensor (Anon, 2019)..... | 10 |
| Figure 5: Surface Electromyography, used from (ComLab- Course on Biological measurements. Pef.uni-lj.si. [Online])..... | 11 |
| Figure 6: Classification Principle of KNN algorithm | 14 |
| Figure 7: Support vector machine data mapping | 17 |
| Figure 8: Location of sensors, adapt from (Chereshnev, R. and Kertész- Farkas, A., 2017)..... | 22 |
| Figure 9: Data file sample | 23 |
| Figure 10: WEKA main interface and “ArffViewer” tool..... | 24 |
| Figure 11: The interface of Explore | 25 |
| Figure 12: “sitting in car” Activity File Sample..... | 26 |
| Figure 13: Classify interface | 30 |
| Figure 14: The result of IBK (k=2) | 34 |
| Figure 15: Data files samples | 45 |
| Figure 16: Dataset format | 45 |
| Figure 17: RandomForest..... | 47 |
| Figure 18: J48 | 48 |
| Figure 19: NaïveBayes | 49 |
| Figure 20: IBK..... | 50 |
| Figure 21: SMO | 51 |
| Figure 22: ClassificationViaRegression | 52 |
| Figure 23: Original Project Schedule..... | 53 |
| Figure 24: Revised Project Schedule | 53 |

List of Tables

| | |
|--|----|
| Table 1: Characteristics of HuGaDB, used from (Chereshnev, R. and Kertész-Farkas, A., 2017)..... | 21 |
| Table 2: Data pre-processing in python..... | 23 |
| Table 3: Dataset classes..... | 27 |
| Table 4: Dataset classes after combination..... | 28 |
| Table 5: Definition of some of the time-domain features adopted in the proposed research..... | 29 |
| Table 6: Result of cycling..... | 31 |
| Table 7: Result of standing | 32 |
| Table 8: Result of running..... | 32 |
| Table 9: Result of sitting..... | 32 |
| Table 10: Result of walking | 32 |
| Table 11: Time of model building..... | 33 |
| Table 12: The Information of the Participants | 38 |

Chapter 1

Introduction

In recent years, the development of sensor technology gradually makes people's Daily life inseparable from sensor data, and sensor network becomes ubiquitous (Chereshnev, R. and Kertész-Farkas, A., 2018) (Aggarwal, C.C. ed., 2013). Sensor based on the characteristics of small size, low price has been widely used in many fields. Among the most popular are smart watches, smart glasses and other trendy wearable devices, which seem to be high-tech products attracting people of all ages. It is because of the increasing popularity of wearable devices that people are attracted to collect sensor data on the human body with sensors and conduct research on it in various aspects. The research results derived from the combination of wearable sensors and human bodies may have a significant impact on the environment of humans and machines. Applications such as telemedicine monitoring and helper robots will have a huge impact and change in the field of human-computer interaction. This project will be based on wearable sensors, combined with human activities, and try to use machine learning method to study the identification content of human activities related to sensor data.

1.1 Problem Statement

Motivation. Born from the large fields of pervasive computing[4], identifying human activities is an important technology in this field because it can be applied to many real-life, human-centric problems in many areas (Anderson, P., 2016), especially for medical, military, and security. For instance, in health-care system, this can be very useful for monitoring patients' postoperative recovery, fall detection or diagnosis of Parkinson's disease and other conditions[5]. These motivations have prompted more research into human activities recognition.

Problem. The recognition of human activities has been approached in two different ways, namely using external and wearable sensors (Perez, A.J., Labrador, M.A. and Barbeau, S.J., 2010). This project focuses on the latter, by using the one of the state-of-the-art (SOTA) datasets and the most comprehensive to date is the publicly-available Human Gait Database (HuGaDB)[9]. The data was gathered from 18 healthy, young, adult participants with accelerometers, gyroscopes, and Electromyography (EMG) sensors on their thigh, shin, and foot of the right and left legs. The dataset has covered twelve actions which are diverse in the sense that they include both static and dynamic activities. By adopting diverse machine learning techniques for human activity recognition based on this dataset.

Solution. Machine learning, as an efficient tool for pattern discovery, have become a standard tool in such a problem. This project focuses on the analysis and comparison of different machine learning algorithm for the human activity recognition results, and also discusses the factors in the aspect of enhancing the recognition accuracy. The objective is to identify and extract possible features from the dataset that can be used to algorithmically recognise the human activity. The following are two technical approaches for analysing the dataset:

- **Feature Extraction.** Classification and feature selection techniques are used to study the effects of different sensor types and positions on accuracy. Different features can be extracted from the dataset, such as mean, variance, standard deviation, minimum, maximum and so on. Also the method of automatic feature acquisition in deep learning is considered.
- **Machine Learning Algorithm.** Supervised and unsupervised methods are investigated. Features are used as input to identify appropriate algorithms to run experiments and to compare and evaluate their performance.

1.2 Aim and Objectives

The prime goal of this project is to use machine leaning algorithm to recognize human activities from wearable sensor data, investigate and evaluate classification results of different algorithms, and analyse the factors that may affecting the classification performance. In order to achieve this goal, data pre-processing and several experiments will be designed and implemented using Python programming language and the WEKA tools. The results will then be analysed and interpreted in order to identify the factors that have a significant impact on the performance.

Key objectives:

- 1) Understand the background of wearable sensors and human activity recognition, and identify problems and objectives.
- 2) Collect and familiarise with the HuGaDB dataset.
- 3) Data pre-processing and feature extraction.
- 4) Identify appropriate tools to build models using machine learning algorithms.
- 5) Apply analysis by running experiments with different models.
- 6) Evaluate the performance of selected algorithms on the provided dataset.

1.3 Deliverables

Report with the data analysis and a comparative study of Machine Learning algorithms.

1.4 Methodology

In order to study human gait recognition based on machine learning algorithm and sensor data, an appropriate sensor dataset is needed. Extensive research works in this field have been done to construct and label a number of publicly available **datasets** that are sufficient to learn from. By understanding the project objectives and requirements, it needs to be creative with some keywords search to find the right source. Once the dataset is found, the next step is to **understand the dataset**. For that reason, comprehensive reading is needed to understand the annotations and formats, to identify data quality problems, to discover the initial level of insights into the data. Understanding the dataset is crucial to conduct the implementation and experiments because the following selected methods, algorithms and results for the analysis are strongly dependent on the dataset. Then, we have to **pre-process the dataset** and **design experiments**. Ensuring data quality by cleaning data and prepare data based on feature extraction methods. In the stage of experiment design, the goal of the project must be considered, and use machine learning algorithms to build an analytical model. After that, the selection of appropriate evaluation metrics need to be applied on the built models. The **evaluation** phase covers all activities that enable the evaluation and verification of the model in accordance to demonstrate the quality of the results. and classification results collected in the experiment need to be evaluated to draw reliable conclusions and recommendations.

The methodology to conduct this project is based on the **Agile methodology**. Agile is a process that can manage the project by breaking it up into several sections. Each section has its own timeline with objectives to be met and deliverables to be completed. Also, agile process is characterized by continuous iteration, allowing people to evaluate and review

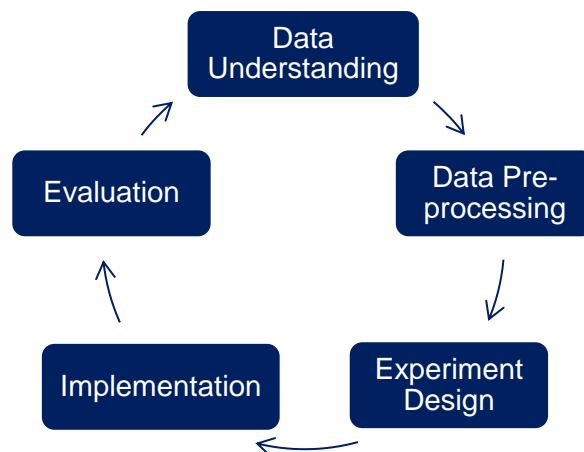


Figure 1: Agile methodology

each milestone and consider the continuation of the next milestone until the project is complete. The advantage of agile methodology is that it offers flexibility and enables people to respond quickly and efficiently to any changes in individual section in order to improve the project overall quality. The iterative pattern of the agile methodology is shown in the follow Figure 1.

1.5 Project Schedule

| Tasks | March | | | | April | | | | May | | | | June | | | | July | | | | August | | | | Sep. | | | |
|------------------------------|---------|---|---|---|-----------------------|---|---|---|-----|---|---|---|-------------|---|---|---|------|---|---|---|--------|---|---|---|------|---|--|--|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | | |
| Identify and Collect Data | ■ | ▶ | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Aims and Requirements | | | ■ | ■ | | | | | | | | | | | | | | | | | | | | | | | | |
| Project outline | | | | ▶ | | | | | | | | | | | | | | | | | | | | | | | | |
| Background Research | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | | | | | | | | | |
| Project Scoping and Planning | | | | | | | | | ■ | ■ | ▶ | | | | | | | | | | | | | | | | | |
| Data Preparation | | | | | | | | | | | | | | | | ■ | ■ | ■ | | | | | | | | | | |
| Modelling | | | | | | | | | | | | | | | | | ■ | ■ | ■ | ■ | ■ | | | | | | | |
| Progress Review Meeting | | | | | | | | | | | | | | | | | | | | | ▶ | | | | | | | |
| Evaluation of Results | | | | | | | | | | | | | | | | | | | | | ■ | ■ | ■ | ■ | | | | |
| Final Report Complete | | | | | | | | | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Project | | | | Courseworks and exams | | | | | | | | ▶ Milestone | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 2: Project Schedule

Initial preparatory work has been started early in March 2019 but was paused due to exams and study commitments. Three months of full-time effort spent on this project, from June to September. Due to uncertainties about the project direction and the lack of background concerning what techniques are used at the start time of the project, some items were scheduled unrealistically. The initial plan was also unrealistic in the length of time assigned to some tasks; therefore, the timescale was adjusted appropriately. Figure 2 shows the Gantt chart of the revised plan, and the initial plan can be found in Appendix D.

1.6 Degree Relevance

Among the number of various modules studied for this degree, the following two are most relevant for conducting this project:

The module Machine Learning (COMP3611) provided me with an essential understanding of machine learning algorithms, further practical experience which is relevant for this project. In the module Data Mining and Text Analytics (COMP5840M), the data analytics approach CRISP-DM as best practise approach to conduct analytics projects was introduced. CRISP-DM is followed in this work. And also the knowledge of the data mining software WEKA to explore various algorithms and analyse the results.

1.7 Report Outline

The report of this project studies data processing, feature extraction and data classification based on human motion recognition of wearable sensors. The paper is divided into six chapters. The first chapter is the introduction, which mainly introduces the research background, significance and research status of the movement. Through the analysis of the research status and problems in the first chapter, the research direction and problems to be solved in this paper are determined, and then the research plan is put forward.

In the second chapter of the report, background research on motion recognition is presented. This chapter first defines the type of action to be identified, and then introduces the working principle of the inertial sensor used in the project and the most commonly used data classification algorithm. This chapter is the basic content of this paper and provides data support for research work.

The third chapter discusses the experimental setup. Firstly, the data set is introduced in detail, including data set collection, sensor location, data format and data analysis tools. Secondly, data processing and feature extraction are carried out.

Chapter 4 describes the experimental steps in detail. Firstly, the data set is pre-processed and feature extracted, and the powerful Weka tool is used to model different classifiers and get classification results.

Chapter five analyses and optimizes the results of chapter four. Firstly, the results of several different algorithms are presented and their optimal performance is compared. The factors that may affect the recognition rate and the methods to improve the recognition rate are analysed.

Finally, chapter six summarizes the research content and results of this project, and then summarizes the deficiencies and possible future research directions of this paper.

Chapter 2

Background Research

In this chapter, the main concepts and technologies regarding human gait recognition are reviewed and comprehensively discussed. The structure of this chapter is as follows: firstly, the overview of human activity recognition is reviewed, with emphasis on the research and application of human gait. Then the main concept of wearable sensors and the review of three types of technologies are also discussed. After that, feature engineering and several machine learning algorithms will be introduced as the basic methods and tools for activity classification, as well as some comparisons. The aim of this chapter is to provide a general knowledge and a solid foundation to conduct the required experiments for this project.

2.1 Human Activity Recognition

Human movement is completed under the control of the human brain and nervous systems, which is complex, diverse and meticulous, with flexibility and changeability that can't be achieved by any instrument. Human activity recognition is a challenging research topic, which is widely used in many fields such as national defence security and medical treatment. Many products and applications have made significant contributions to national construction and people's lives. With the continuous deepening of research on inertial sensors, human motion recognition based on wearable sensors is of great significance in the following aspects:

- Game animation and film production

Collect relevant data of human body movements, and then load these data into computer animation through relevant software, to realise animation production and relevant game development. At the same time, in terms of film production, some complicated movements in many films can be made by computer technology according to human movement data, which has a good effect.

- Professional motion analysis

Human motion recognition is also widely used in sports related fields. Through studying the movements in various sports and scientifically analysing and utilising the collected data, it can not only promote the athletes' training water bottle and mechanical movements but also effectively reduce sports injuries.

- Medical application

In the medical field, there are many applications of gait recognition, mainly for the recognition of normal and abnormal gait. The abnormal gait here usually refers to pathological abnormal gait, such as Parkinson's gait, stroke gait, and does not include non-

pathological abnormal gait, such as drunken gait. In the existing research of clinical gait data collection, researchers usually adopt wearable sensor system to collect gait data. The wearable sensor system has the characteristics of a reasonable price and suitable for laboratory environment analysis.

- Virtual reality

Virtual reality is a virtual space where experiencers interact to simulate real life. It USES modern technology such as computer science to create realistic visual, auditory and tactile scenes. In this environment, to provide technical support for in-depth interaction and multi-mode interaction between human and environment, motion capture and motion recognition technologies are needed.

(Khachay, M.Y., etc, 2014.) Inertial sensors are usually installed in the joints and muscles of the body in applications such as the legs, chest and wrists. The first research on human activity identification dates back to the late 1990s (Basterretxea, K., Echanobe, J. and del Campo, I., 2014) (Sant'Anna, A., Salarian, A. and Wickstrom, N., 2011), and there are still many unanswered questions that are driving the development of new technologies.

Combined with the most recent machine learning technologies (Varkey, J.P., Pompili, D. and Walls, T.A., 2012), it can model a wide range of human activities. At present, there is a lot of literature that has witnessed the progress in different aspects of the field, such as improving the accuracy of identification. Different factors to improve the accuracy of identification are also discussed in the report.

2.2 Human Activity Recognition – Human Gait

Gait is branch of human activity system. This project will analyse different human activities mainly on gait movements by analysing the data obtained from different sensors installed on the legs. The data usually focus on walking-related activities, such as walking, running, jumping, sitting down, going up or down the stairs, and so on.

The analysis of gait data has also led to many different applications. Gait can represent the motion characteristic of individuals (Kale, A., Rajagopalan, A.N., Cuntoor, N. and Kruger, V., 2002). On this basis, the recognition of human can be realized by using gait information. For example, existing studies have estimated human age from gait signatures (Lu, J. and Tan, Y.P., 2010), identified gender based on gait information (Yu, S., Tan, etc, 2019). It can be seen from the above three applications that gait analysis can play a great role in many fields in the future. This report will do a brief analysis of the selected database in this area.

2.3 Wearable Sensors

Traditional human activity recognition research is realised through the analysis of video or image sequence. Firstly, the motion of a human body is detected, the general outline of the motion of the human body is proposed, then the motion target detected is tracked, and finally, the motion of moving the human body is judged by analysing the information data. However, the movement of information acquisition method has some problems: due to the motion image taken in dynamic scene, the scene illumination change,, keep out the influence of various factors such as the object of human movement characteristics in the process of extraction and segmentation, vulnerable to the interference of background noise, which result in the movement of the human body contour segmentation result is not accurate. Get human motion data parameters, on the other hand, must be based on a different point of view of the human body movement when streaming video or image sequence, combined with the related test equipment calibration and specific calculation model, indirect human movement parameter (including the exercise of acceleration, displacement and the rotation Angle, etc.), its drawback is obvious: first of all, can't direct access to the kinematics and dynamics of information movement, resulting in human movement data obtained relies too heavily on the accuracy of the location of capture device, Angle and conversion accuracy required by the model. Secondly, the acquisition equipment based on high-speed camera has disadvantages such as fixed Angle of view and complex acquisition process, which lead to high cost, poor portability and low accuracy of the acquisition equipment. Therefore, this paper proposes human motion recognition based on wearable devices.

Wearable sensors, just as the name implies, are portable devices that can be worn directly or worn on the body. Most research in this field, recent focus has shifted to wearable sensing platforms, exploiting stretchable and flexible electronics. After Google released the prototype of smart glasses attracted global attention in 2012 (Google.co.uk. Google Glass UK), Apple, Samsung, SONY and other terminal manufacturers also launched their own wearable devices. Wearable sensing technology has rapidly moved from largely a vision of science fiction to a wide array of established consumer and products. This explosion of wearable sensors can be attributed to several factors, such as the advance in miniaturized electronics, and the popularization of smart-phones and connected devices. Wearable devices use sensors to collect raw data from measurements which are stored and used for the continuous monitoring of health, exercise activity, assessing performance, and others (Kamišalić, A., Fister, I., Turkanović, M. and Karakatič, S., 2018).

Since the late 1970s, with the development of Microelectromechanical systems (MEMS), all kinds of sensors are developing towards miniaturization and economy, to meet the market demand better and are widely used in various fields, such as smartphone and motion-

sensing game equipment. In the rapidly developing MEMS, the accelerometer is a device capable of measuring acceleration force, sensing acceleration and converting it into usable output signals. The gyroscope can measure the rotation motion of the device itself and calculate the actual direction through the deviation from the initial direction. Its main principle is that the spin axis of the internal gyroscope will not change the direction due to the gyroscope effect. At present, Paul's three-axis gyroscope sensor and three-axis acceleration sensor, which are used for motion signal acquisition, are relatively mature and can well realize human motion capture. On this basis, inertial sensor-based human motion analysis and recognition is a new research field of pattern recognition, which overcomes many shortcomings and limitations of traditional video-based motion recognition and has higher operability and practicability.

2.3.1 Inertial Sensor

This section describes the sensors used to collect motion data. The inertial sensor is a commonly used sensor, which mainly monitors and measures acceleration, tilt, impact, vibration, rotation and multi-degree of freedom motion, and is a vital component to solve navigation, positioning and motion carrier control. Inertial sensors include acceleration sensors and gyroscopes and their uniaxial, biaxial and triaxial combinations. Most activity classification systems use inertial sensors. Data from inertial sensors are processed to provide many different types of motion, position, and direction. Compared with the activity recognition based on visual stream, sensor-based recognition can reflect the meaning of motion more directly and has lower requirements on the use environment.

The inertial sensors consisted of a triaxial accelerometer and a triaxial gyroscope integrated into a single chip. In this project, the MPU9250 inertial sensor (see Figure 3: MPU9250 Sensor Module) is used to collect the data.

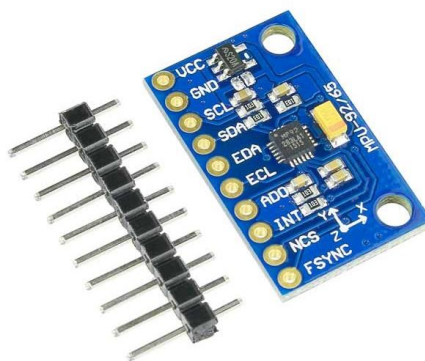


Figure 3: MPU9250 Sensor Module (MPU-9250 9-DOF 3-Axis Accelerometer, 2019)

It is composed of two parts, one is the three-axis accelerometer and three-axis gyroscope, and the other is the AKM company's AK8963 three-axis magnetometer. Inertial sensors can be complemented by a magnetic compass or magnetometer (Parkka et al 2006), which can

enable more accurate orientation measurement about the vertical axis (Sabatini 2006) and by GPS (Murakami and Makikawa 1997) to enable location tracking. Usually, the gyroscope, accelerometer and magnetometer are combined in the same device, since each sensor has its own strong sides. However, this paper only studies the data collected by the former. Figure 4 describes the axial directions of the three-axis acceleration sensor and the three-axis gyroscope. Values of the gyroscopes and the accelerometers encoded by int_16 datatype. Values of the EMGs encoded by uint_8 datatype.

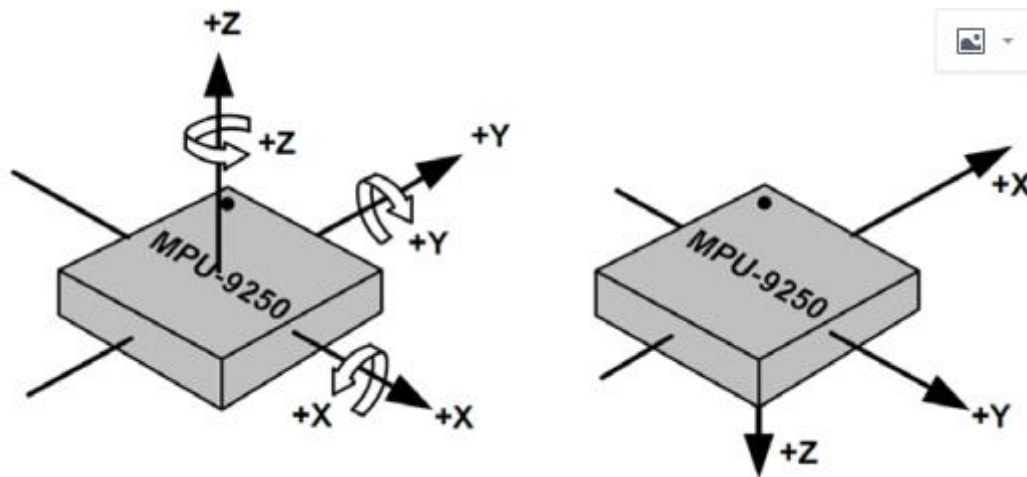


Figure 4: Axial direction of triaxial gyroscope and triaxial acceleration sensor (Anon, 2019)

An accelerometer, also known as a gravity sensor, can sense acceleration in any direction, including gravity acceleration, by measuring the force on a component in an axial direction to get the result, expressed as the magnitude and direction of the axial acceleration. The three-axis accelerometer works based on the basic principle of acceleration. Acceleration is a space vector. On the other hand, when the direction of motion of the object is not known in advance, only the triaxial acceleration sensor is applied to detect the acceleration signal. A gyroscope is conceptually a rotating wheel that measures the rotational motion of the device itself (Sarcevic, P., Kincses, Z. and Pletl, S., 2019). According to the conservation of angular momentum, the spin axis inside the gyroscope does not change direction as the gyroscope tilts or rotates. According to this principle, the gyroscope can measure the bearing and the rate of change. The three-axis gyroscope can simultaneously measure the position, trajectory and acceleration in six directions. A single-axis can only measure things in two directions.

2.3.2 Surface Electromyography

Surface Electromyography (sEMG) signal is a continuous transfer cell of human body central nervous nerve impulses to the nerve endings, causing muscle fiber membrane action potential of continuous formation sequence of action potentials in the skin surface and become a kind of weak signal, can reflect the corresponding skeletal muscle movement

state, and the instructions of the nervous system information. Therefore, the sEMG signal can be used to realize the classification, recognition and processing of different body movements, to provide adequate help for the analysis of body movements and state characteristics, which is widely used in the fields of sign language recognition, manipulator control, hand rehabilitation and other fields (Matsubara, T. and Morimoto, J., 2013) (Zhang, D., Zhao, X., Han, J. and Zhao, Y., 2014).

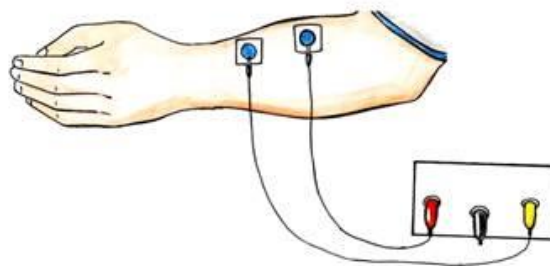


Figure 5: Surface Electromyography, used from (ComLab- Course on Biological measurements. Pef.uni-lj.si. [Online])

2.4 Recognition Methods

Human activity recognition based on the inertial sensor is a process of data collection and classification, the core part of which is data classification. The motion information of the human body is collected by inertial sensors fixed in specific parts of the human body, and then the data is pre-processed, feature extracted, and motion classified. This section will briefly introduce the common feature extraction methods and different classification algorithms.

2.4.1 Feature Engineering

A complete data analysis system usually has the following three steps: first, obtain the original data, and then use data processing technology to process and extract essential features and attributes from the data. Finally, these features are modelled using technology such as machine learning. As the core of artificial intelligence, machine learning algorithms can use data to generate knowledge and provide insights. The algorithm itself is very general but does not work well on ordinary raw data. When doing data analysis, data is the basis of all problems which affect the whole process of the project. Feature engineering is an indispensable step to human activity recognition, which has a direct impact on subsequent recognition results. Therefore, we need to extract essential features from the original data so that we can understand and use the data. However, for the recognition of different human activities, the selection of features is not specific. In simple terms, features can be classified into time-domain, frequency domain and time-frequency domain. Previous human activity recognition studies have used a variety of methods to generate features. These features are

then used as input to classification schemes. Retrieving useful information from sensor data is a challenge of activity and motion recognition. Conventional feature extraction methods can be divided into three categories: time-domain feature, frequency-domain feature and time-frequency domain feature.

2.4.1.1 Time-Domain Features

Time-domain features refer to the time-related features in the process of time-varying sequence. They are usually extracted directly from a window of sensor data, which are statistics of the original measurement. Common time-domain features used in activity recognition include mean, standard deviation, maximum, minimum, median, variance, skewness, kurtosis and correlation.

2.4.1.2 Frequency-Domain Features

The method of frequency domain analysis is used to analyse the motion data in the window to extract features. Fast Fourier transform (FFT) is one of the most popular techniques used in the transition from time domain to frequency domain. The typical frequency domain characteristics are energy, FFT coefficient, energy spectrum density, and so on. In reference 42, good recognition results are obtained through the classification and recognition of the motion data through the frequency domain feature extraction. However, compared with the time domain features, the recognition rate is not effectively improved. FFT operation is often required to extract the frequency domain features, which consumes a large amount of system computing resources. Therefore, the frequency domain features extraction is no longer carried out in this project.

2.4.1.3 Sliding Window Technique

In human gait recognition, because the data collected by inertial sensors are human movement information in a period, the data length is relatively long, and it is not easy to extract and classify features. Window segmentation is the most widely used segmentation technique in human activity recognition. Sensor signals can be divided into smaller periods, each time segment is an observation window, and features can be extracted from each window. The sliding window is a window technology with fixed window length, which divides action data into equal-length data segments. This method does not need to do other processing to the action data. (Huynh, t. and Schiele, B., 2005) studied the influence of different window lengths on the accuracy of the classification algorithm. Existing studies on human activity recognition generally adopt the scheme that adjacent windows have specific overlap rate. (Khan, A.M., Lee, Y.K., Lee, S.Y. and Kim, T.S., 2010) applied a 50% overlap rate to identify human movements. The disadvantage of sliding window technology is that when the movement changes, it is difficult to accurately identify the transformation between movements because a window contains two movements.

2.4.1.4 Features Selection

There are two reasons for not doing feature selection in this project: first, you don't want to reduce the feature and size of the data set. Secondly, known from (Baldominos, A., Saez, Y. and Isasi, P., 2015) that the influence of feature selection on different classifiers is various. Therefore, the adoption of a feature selection method may have a positive or negative impact on some classifiers.

2.4.2 Machine Learning Algorithms

Machine learning is the most common method for constructing classifiers. According to the training type of the model, it is mainly divided into supervised learning and non-supervised learning. Supervised learning algorithm needs to give training set and corresponding category attribute, and form classifier through training data to classify test set. Unsupervised learning does not need training data, but only divides similar data into one kind through clustering, to achieve the purpose of data classification. Supervised learning algorithms include K-Nearest Neighbours (KNN), Naive Bayes, Decision Tree, support vector machine algorithm, etc. Unsupervised learning algorithms include k-means clustering and hidden Markov model. The classification algorithm used in this project is described below.

- K-Nearest Neighbours (IBK):

IBK classifier in Weka uses the K-Nearest Neighbours (KNN) algorithm which is the simplest classification algorithm in machine learning. Its idea is: there are K sample points in a certain range centred on the test point. When the data of a certain type of sample points account for the majority, the test point also belongs to this category. In the k-proximity algorithm, the neighbours of the measured points are all training samples whose categories are known. This method only determines the category of the class to be classified according to the category of the nearest one or several data sets. Because the classification results of the KNN algorithm are only related to the most recent training sets of the test set, this algorithm has a better classification effect when there is a lot of overlap among different classes. The following figure shows the classification principle of KNN algorithm:

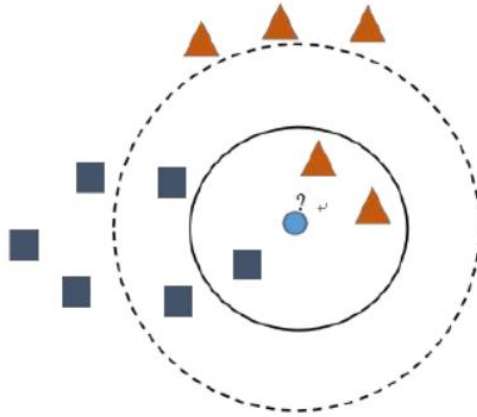


Figure 6: Classification Principle of KNN algorithm

Dot in the figure to be classified test points, red and blue are known, it can be seen that when the K value is 3, dot will be identified to account for about two-thirds of the total red category, and when the K value is 5, dot will be identified as 3/5 of the total number of blue category, which illustrates the value of K has a larger effect on classification results. When the selection of K is too small and the recognition results are only related to the recent training samples of test data points, the problem of overfitting will occur. When K value is selected too large, training samples far away from the test point will have a greater impact on classification results, and classification errors will also occur. In practical applications, K value needs to be optimized according to different data characteristics, and cross-validation method is used to find the most suitable K value.

In the KNN algorithm, the distance between objects is calculated as an indicator of whether objects are similar or not, to avoid complex similar matching operations between objects. Examples between objects are generally expressed as Euclidean distance and Manhattan distance, and the expressions are as follows:

- Euclidean distance: $d_1(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
- Manhattan distance: $d_2(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|}$

Although good at overlapping cross KNN more classification, but it belongs to the lazy algorithm, to test sample classification need to compute the object under test from training set of all points, too large amount of calculation, and with the increase of feature dimension, ability to distinguish between variation of Euclidean distance, when calculating the Euclidean distance and different characteristics to the influence of different distance, so you need to first to normalization processing characteristics.

- Naïve Bayes:

Bayesian classification, based on Bayes' theorem, is the general name of a class of classification algorithms. Bayes' theorem solves the problem of how to get the probability of

two-time exchanges given a certain conditional probability. For example, how do I get $P(BA)$ if I have $P(A|B)$? First of all, the meaning of conditional probability is explained here. $P(A|B)$ represents the probability of event A occurring in the case of event B, the formula is as follows:

$$P(A|B) = \frac{P(AB)}{P(B)}$$

In general, $P(A|B)$ can be easily obtained directly, while $P(B|A)$ is difficult to be obtained directly. However, sometimes we are more concerned about $P(B|A)$. At this time, $P(B|a)$ can be obtained through Bayesian reasoning $P(A|B)$, Bayes' theorem has the following expression:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Naïve Bayes classification algorithm is a simple data classification algorithm. Its principle is as follows: for the given category to be classified, the probability of each category to be classified, which is the largest under the condition of the occurrence of this item, shall be considered as belonging to which category. Because of its simple classification idea, it is named Naïve Bayes classification, and its algorithm process is as follows:

- a) Let $x = (a_1, a_2, \dots, a_m)$ is a category to be classified, where a_i is the characteristic attribute of x .
- b) Set the category set as $C = (y_1, y_2, \dots, y_n)$
- c) Calculate $P(y_1|x), P(y_2|x) \dots P(y_n|x)$
- d) If $P(y_k|x) = \text{Max} \{P(y_1|x), P(y_2|x) \dots P(y_n|x)\}$, so x is of class y_k

Among the above steps, the most important one is the solution of each conditional probability in step 3. The conditional probability can be obtained by the following steps:

- a) Use a sample set of known classifications as the training set
- b) Conditional probability estimation of each characteristic attribute of statistical training set under each category. That is:

$$P(a_1|y_1), P(a_2|y_1) \dots P(a_m|y_1), P(a_1|y_2), P(a_2|y_2) \dots P(a_m|y_2), \dots, P(a_1|y_n), P(a_2|y_n) \dots P(a_m|y_n),$$

- c) When each characteristic attribute is conditionally independent, the root element Bayes theorem can be deduced as follows: $P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$. The denominator in the

expression is a constant for all categories, so we just need to maximize the numerator. It

is known that each characteristic attribute is conditionally independent, therefore:

$$P(x|y_i)P(y_i) = P(y_i) \prod_{j=1}^m P(a_j|y_i), \text{ which turns the problem into a conditional probability}$$

estimation problem for the data to be measured.

The biggest advantages of Naïve Bayes classification are simple to use, fast calculation speed, clear and easy rules, and fewer data in the training set. However, it also has obvious disadvantages, such as higher requirements on training samples and the need to maintain the independence between features, so it is necessary to carefully analyze the correlation of variables.

- Decision tree (J48):

Decision tree algorithm is a classical machine learning algorithm. We can regard the decision tree as a tree structure prediction model, which is a tree structure composed of oriented edges and nodes. The tree contains the root, inner and leaf nodes. The decision tree regards the set of all training data as the root node, and each internal node of tree species can be seen as a splitting problem: the sample reaching this node will be divided by judging an attribute. Each leaf node in the tree is a data set with classification attribute, that is, the classification to which the data belongs.

How to choose a decision order to achieve the best classification effect is the key to the decision tree algorithm. Many algorithms can realize the optimal decision, such as ID3, C4.5, CART and so on. These algorithms adopted by the greedy algorithm is top-down, the decision tree classification of each internal node will be able to achieve the best effect of the tree as a classification criterion, the data set is divided into two or more child nodes, iterative this process directly to all of the decision tree classification of training data correctly, or all the tree are used.

Decision tree algorithm has the following advantages:

- a) Easy to understand and implement. After a simple explanation, people can understand the practical meaning of the decision tree.
- b) The required data format is simple, and other classification algorithms often need to generalize the data first, such as removing redundant attributes.
- c) High efficiency, the decision tree only needs to be constructed once and used repeatedly, and the maximum calculation times of each prediction should not exceed the depth of the decision tree.

- RandomForest:

Random forests is a classification algorithm based on decision tree algorithm, it through the self-help method heavy sampling technology, from the original training focus back to repeat random sample n generated new training sample set a decision tree, and then according to the above steps to generate n decision tree of random forests, the classification of the new data the result according to the classification tree to vote how many scores. In essence, it is

an improvement of the decision tree algorithm by merging multiple decision trees together, and the establishment of each tree depends on independent samples.

- Support vector machine (SMO) :

Support vector machine is referred to as SVM. From the perspective of classification principle, it is a model for data-binary classification and is defined as a linear classifier that maximizes classification interval in feature space. Before the advent of deep learning algorithms, SVM was considered to be the most successful and best-performing algorithm in machine learning in recent decades.

SVM is a supervised machine learning algorithm, based on structural risk minimum principle, the confidence limits and to minimize the risk, so as to improve its generalization ability, even in the case of the training sample is not much, also can test the better classification effect, the core of the algorithm is how to find a sample can be divided into two kinds of optimal hyperplane, the main idea is to make the plane linear separable problem to high-dimensional classification problem.

In support vector machines (SVM), low-dimensional linear inseparability problems are mapped to high-dimensional linear separability problems for classification. In order to avoid the dimension disaster caused by too high dimension and reduce the computation, SVM introduces kernel function method for high-dimensional discrimination. The training sample points closest to the optimal hyperplane are called support vectors, and the optimal hyperplane is only related to these points. Therefore, after the classifier training is completed, the SVM computation is relatively small when classifying test data, which is one of the advantages of the SVM algorithm. The larger the distance between the support vector and the optimal hyperplane means that the plane has the strongest ability to distinguish the two categories and the higher the classification accuracy.

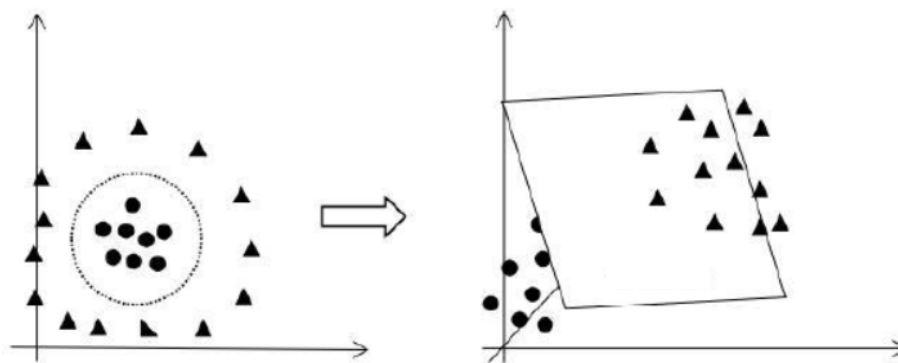


Figure 7: Support vector machine data mapping

Can be seen from the diagram above, for the given data, within the intrinsic dimension can't linearly separable, and looking for a no regular curve is unrealistic, and by increasing the

data dimension, can find a separating hyperplane in the high latitudes differentiate between data, the classification of plane can be described by simple mathematical equations.

- Classification via Regression

The linear regression approach is used for classification in this classifier. When classifying, each generated regression model is configured for each value of the class.

2.4.2.1 Evaluation Metrics

Assessment is a key step in making real progress with data classification. What kind of classifier is adopted to solve a specific problem needs to be evaluated or systematically compared and evaluated among different classifiers. When comparing different classifiers, the key performance indicators often need to be referred to are:

- **True Positives (TP):** “The number of positive instances that were classified as positive.”
- **True Negatives (TN):** “The number of negative instances that were classified as negative.”
- **False Positives (FP):** “The number of negative instances that were classified as positive.”
- **False Negatives (FN):** “The number of positive instances that were classified as negative.” (KEÇECİ, A., YILDIRAK, A., ÖZYAZICI, K., AYLUÇTARHAN, G., AĞBULUT, O. and ZİNCİR, İ.,2018)
- The **accuracy** is the most standard metric to summarize the overall classification performance for all classes and it is defined as follows:

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

There are many performance evaluation criteria for classifiers, but for a long time, as the training set and test set examples only provide class labels, the accuracy or error rate of prediction naturally becomes the main evaluation criteria for the classifier's prediction performance.

- **The ROC curve**

Receiver Operating Characteristic, or ROC, is a term used in signal detection to reflect the balance between the hit rate of noisy channels and the error warning. In theory, compared with the accuracy assessment method, ROC curve analysis method has the following advantages:

1. Probability value of full utilization prediction

2. Different distribution conditions of different classes are given, that is, when unbalanced data is presented, different data distribution will yield different classification results, while accuracy evaluation defaults to all data sets being balanced data sets.
3. Different types of error classification costs are considered, and the accuracy evaluation defaults that all error costs are the same. This is impractical in real life.
4. The evaluation results of classifiers can be more intuitively displayed in the two-dimensional space in the form of curves.

- **Precision**, often referred to as positive predictive value, is the ratio of correctly classified positive instances to the total number of instances classified as positive:

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

- **Recall**, also called true positive rate, is the ratio of correctly classified positive instances to the total number of positive instances:

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

- **F-measure** combines precision and recall in a single value:

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Although defined for binary classification, these metrics can be generalized for a problem with n classes. In such case, an instance could be positive or negative according to a particular class, e.g., positives might be all instances of running while negatives would be all instances other than running.

- **Speed**. This involves the time spent producing and using the model.

Chapter 3

Experimental Setup

In this chapter, all necessary steps to set up the experiments are presented to implement. With respect to the CRISP-DM approach, this chapter focuses on the data understanding and data preparation phases to relate the selected dataset. Along with a justification for choosing the dataset, major problems of the dataset are addressed. Further, the software tools for analysis are explained.

3.1 Dataset Selection

Extensive research works in the field of using sensors to identify human activity have been done to construct and label a number of publicly available datasets that are sufficient to learn from. The performance of subsequent models depends largely on the quality of the dataset. Therefore, the challenge is to choose an appropriate wearable sensor data set among many options. By understanding the project objectives and requirements, it needs to be creative with some keywords search to find the right source. And the data set needs to be reliable and available. The size of data set is an important criterion for selecting data set. If the selected data set is small, the model cannot obtain enough discriminating features for generalization. Such a model will overfit the data, resulting in low training errors but high test errors.

A search on GitHub for the keyword 'wearable sensor data' turned up nearly 100 datasets. HuGaDB was chosen from a few of the highest-rated datasets. See the next section for a detailed description of this dataset.

3.2 Dataset Description

The Human Gait Database (HuGaDB) [4] is one of the most recent datasets in the existing literature. This dataset is unique in the sense that it is the first to provide human gait data in great detail mainly from inertial sensors and contains segmented annotations [9]. The data is comprised of twelve different activities, and each has an ID number (walking, running, going up, going down, sitting, sitting down, standing up, standing, bicycling, up by elevator, down by elevator, sitting in the car). Activities are described in Table 1. Data was collected from six wearable inertial sensors worn on the body, consisting of: accelerometer, gyroscope, and **EMG****Error! Reference source not found.**, which were placed on the right and left thighs, shins, and feet respectively. The data was acquired from 18 healthy, young people. The gender profile of test subjects is four females and fourteen males.

Table 1: Characteristics of HuGaDB, used from (Chereshnev, R. and Kertész-Farkas, A., 2017)

| ID | Activity | Description |
|----|------------------|--|
| 1 | Walking | Walking and turning at various speeds on a flat surface |
| 2 | Running | Running at various paces |
| 3 | Going up | Taking stairs up at various speeds |
| 4 | Going down | Taking the stairs down at various speeds and steps |
| 5 | Sitting | Sitting on a chair; sitting on the floor not included |
| 6 | Sitting down | Sitting on a chair; sitting down on the floor not included |
| 7 | Standing up | Standing up from a chair |
| 8 | Standing | Static standing on a solid surface |
| 9 | Bicycling | Typical bicycling |
| 10 | Up by elevator | Standing in an elevator while moving up |
| 11 | Down by elevator | Standing in an elevator while moving down |
| 12 | Sitting in car | Sitting while travelling by car as a passenger |

3.2.1 Data Collection

In terms of data collection, the MPU9250 inertial sensors (see **Error! Reference source not found.**) and electromyography (EMG) sensors (see Figure 5: Surface Electromyography) are adopted. The inertial sensors consists of a triaxial accelerometer (see **Error! Reference source not found.**) and a triaxial gyroscope integrated into a single chip. Three pairs of inertial sensors and one pair of EMG sensors were installed(mounted) symmetrically on the left and right legs with elastic bands. The three pairs of inertial sensors were installed on the rectus femoris muscle 5 centimetres above the knee, the middle of the shinbone at the level where the calf ends, and the metatarsal bone on the foot. (Badawi, A.A., Al-Kabbany, A. and Shaban, H., 2018) Two EMG sensors were placed on the vastus lateralis and connected to the skin with three electrodes to measure muscle activity. The locations of all sensors are shown in Figure 8. The annotations in the figure represent right foot (RF), right shin (RS), right thigh (RT), left foot (LT), left shin (LS), and left thigh (LT). EMG sensor are shown as circles while boxes represent inertial sensors. In total, 38 signals were collected, including 36 from the inertial sensors and 2 from the EMG sensors.

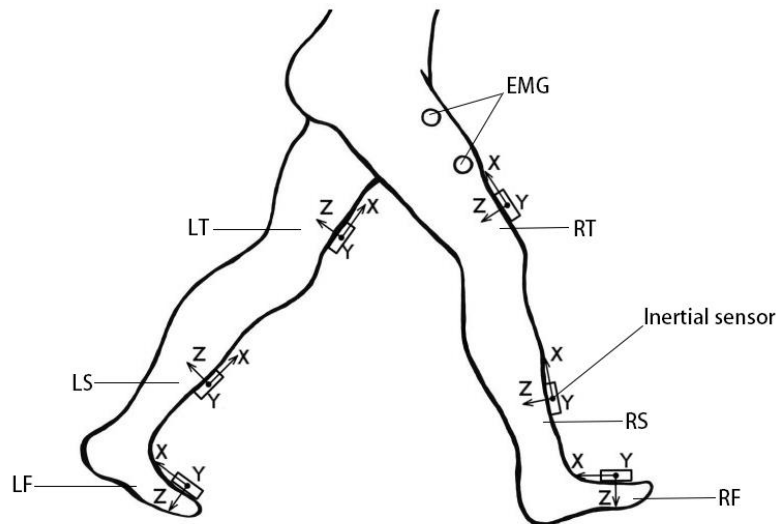


Figure 8: Location of sensors, adapt from (Chereshnev, R. and Kertész-Farkas, A., 2017)

The participants performed a combination of activities, and the experimenter developed their own data collector program, using a laptop to continuously record data and annotate the data with the activities performed. At the end, 2,111,962 samples performed up to 10 hours were collected. Provides a long, continuous sequence of segmented data annotated with activities.

3.2.2 Data Format

Data obtained from inertial sensors are stored in text files because they have one of the most universal formats and can be preprocessed easily on every system in any programming languages. The dataset contains 635 files from 18 participants, one of which contains one recording. There are two types of data files, one for a single activity and one for a series of activities (for example, walking and running). The main body of each file contains three rows of metadata and 39 columns. Each column corresponds to a sensor, and each row other than metadata corresponds to a sample. The first 36 columns correspond to the inertial sensors, the next 2 columns correspond to the EMG sensors, and the last column contains the activity ID. Each inertial sensor produces three acceleration data on x, y, z axes and three gyroscope data on x, y, z axes. For instance, the column named 'acc_rf_x' contains data acquired from the x-axis of accelerometer located on the right foot. Data samples are shown in the figure below.

| | | | | | | | |
|-------------|-------------|-----------|-----------|-----------|-----------|----------|----------|
| #Activity | cycling | | | | | | |
| #ActivityI[| 9 | | | | | | |
| #Date | 10-10-12-53 | | | | | | |
| acc_rf_x | acc_rf_y | acc_rf_z | gyro_rf_x | gyro_rf_y | gyro_rf_z | acc_rs_x | acc_rs_y |
| -14392 | -96 | 10628 | 138 | -388 | 492 | -13980 | -1408 |
| -11248 | 164 | 7744 | 169 | 699 | 429 | -14044 | -2608 |
| -15892 | -1952 | 9324 | -217 | 709 | 360 | -13336 | -3564 |
| -13020 | 892 | 8604 | -297 | -62 | 170 | -13832 | -3220 |
| | | | | | | | |
| acc_lt_y | acc_lt_z | gyro_lt_x | gyro_lt_y | gyro_lt_z | EMG_r | EMG_l | act |
| 2228 | 9120 | 261 | 277 | 522 | 130 | 125 | 9 |
| 3292 | 8916 | 132 | 86 | 581 | 136 | 124 | 9 |
| 2608 | 8796 | 174 | -96 | 641 | 127 | 127 | 9 |
| 4236 | 8312 | 88 | -63 | 553 | 122 | 127 | 9 |

Figure 9: Data file sample

3.3 Software tools

This section describes all the tools for experimentation, including the programming language Python for data processing, and the data analysis tool - WEKA.

3.3.1 Programming Language - Python

Python is an object-oriented programming language with a simple structure and easy to learn. One of its biggest advantages is its wide range of libraries and good compatibility. Therefore, it is often used as a tool for data processing, which can handle data from KB to T, with high development efficiency and maintainability.

The following table summarizes the data pre-processing in Python:

Table 2: Data pre-processing in python

| Data pre-processing work | Description |
|--------------------------|--|
| Data transformation | Convert TXT format to CSV format by replacing the space with commas in data files and then change the suffix name. |
| Data cleaning | Handle outliers in the dataset: <ul style="list-style-type: none"> • Modify the wrong string. • Replace the infinite value with NAN. |
| Merge files | Merge 635 data files into one file. |

| | |
|------------------------|---|
| Modify data attributes | Replace the activity ID with strings for easy viewing of the results. |
| Features extraction | Valid features are extracted. |

3.3.2 Data Analysis Tool - WEKA

WEKA (Waikato Environment for Knowledge Analysis) is a free, open source machine learning and data mining software based on the JAVA environment. As an open data mining platform, WEKA integrates a large number of machine learning algorithms that can undertake data mining tasks, including data pre-processing, classification, regression, clustering, association rules and visualization on the new interactive interface. The software version used for this project is 3.8.3. WEKA supports many file formats, including ARFF, XRFF and CSV. ARFF (Attribute-Relation File Format) is the most common format. File, which is an ASCII text File. In this project, WEKA was used to convert the processed CSV file into an ARFF file, due to ARFF is the best file format WEKA supports. For the purpose of this work, Weka provide a handy tool called "ArffViewer" to load a CSV file and save it as an ARFF file. The main interface of Weka is the Weka GUI selector, which provides four main applications for users to choose through the buttons on the right. The main interface and the "ArffViewer" block is shown below:

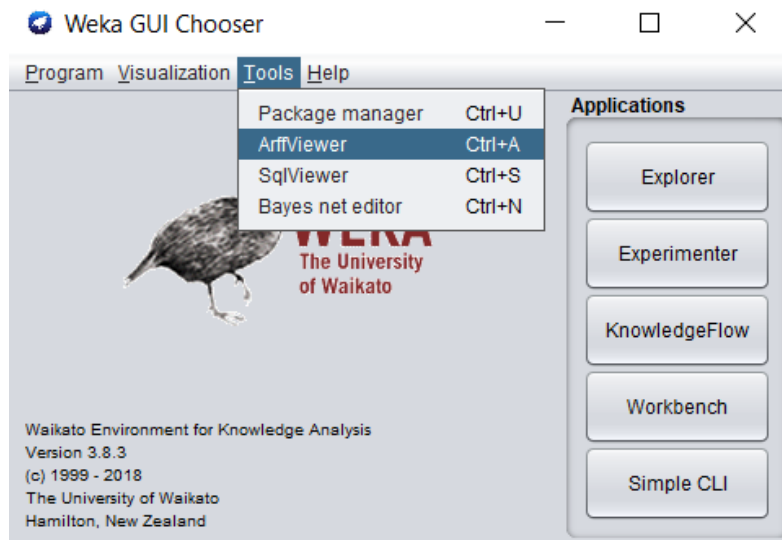


Figure 10: WEKA main interface and "ArffViewer" tool

The following is a brief introduction of each part of the function:

- **Explorer**, the system provides the easiest to use image user interface. All of the Weka's functions can be invoked by selecting menus and filling out forms. Generally, almost all functions in Weka can be completed only by using part E, and only the main functions of

this part have been used in the research of this project. See figure 1 for the detailed interface.

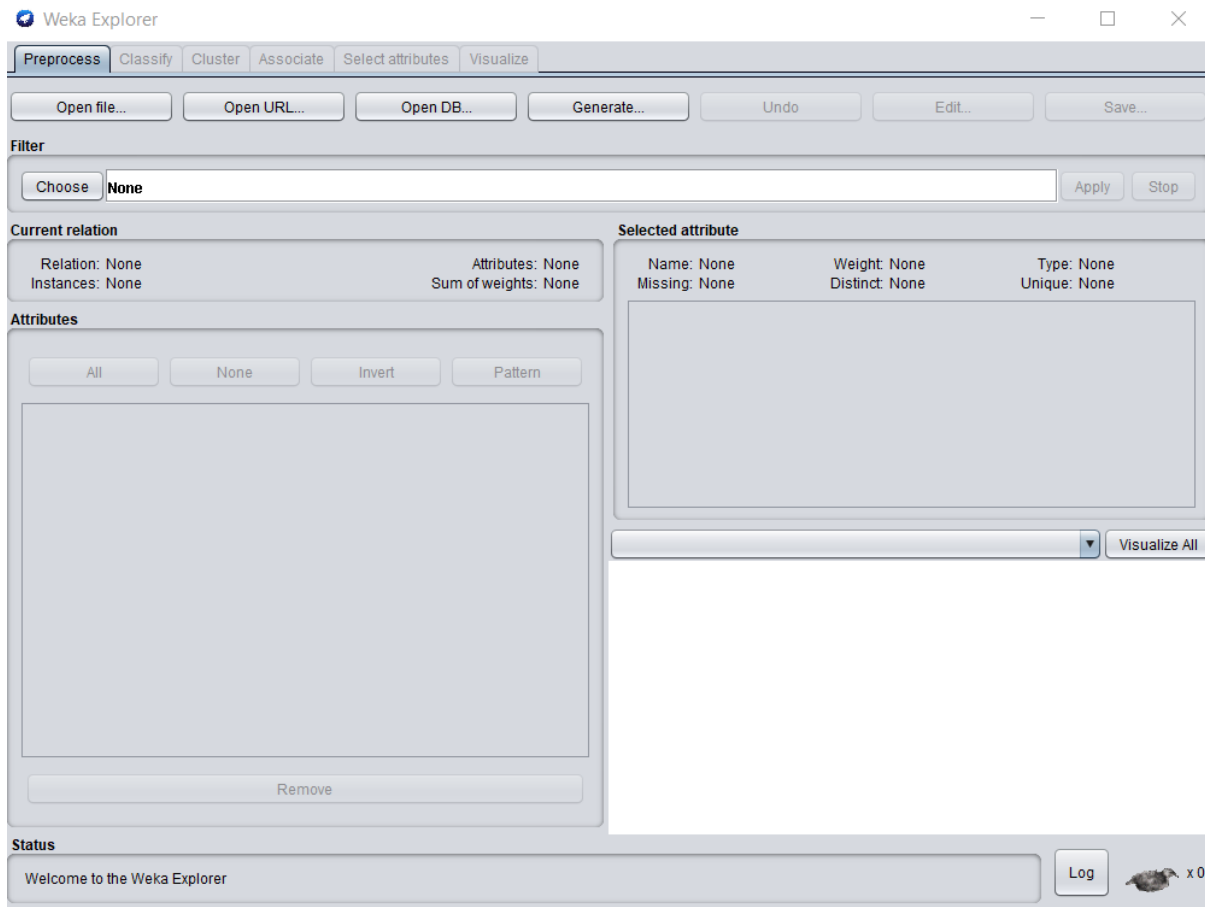


Figure 11: The interface of Explore

- **Experimenter**, this interface makes it easier to set up classifiers and filters with different parameters, make them run in a set of data sets, collect performance statistics, and realize important test experiments.
- **KnowledgeFlow**, the functionality is similar to Explorer, but operates in a drag-and-drop manner and supports incremental learning.
- **Workbench** provides a unified operation interface for other interfaces.
- **Simple CLI** provides a simple command-line interface to call all Weka classes.

Chapter 4 Implementation

This chapter shows the implementation process of this project, using the experimental method introduced in the previous chapter, including data pre-processing, data cleaning, data normalization feature extraction and modelling.

4.1 Data Pre-processing

Data pre-processing is an essential phase before performing classification. This section describes the different actions that take place in the pre-processing stage and returns a new, transformed dataset.

The HuGaDB data format used in this project is a text file, as shown in section 3.2.2. In the data pre-processing phase, first, a python script is used to convert the delimiter of all text files into a comma and convert them into a CSV file format, the multiple CSV files are then merged into one file. Finally, a series of data cleaning operations are performed on this CSV file.

4.1.1 Data Cleaning

Data cleaning is the process of re-examining and validating data to remove duplicate information, correct existing errors, and provide data consistency. Mainly involves deleting irrelevant data and duplicate data in the original data, smoothing noise data, filtering out data irrelevant to mining topics, and handling missing values and outliers. For example, the easiest way to handle invalid and missing values is to replace them with the mean, median, or mode value.

Errors. All activity names in data files are delimited by underscores, for example “going_up” and “standing_up”. However, in 15 of the 635 data files, the activity name in the “sitting in the car” activity file are separated by spaces (see Figure10). As mentioned earlier, use commas to replace space for formatting conversion. When the transformation ends, the activity name of "sitting in car" in the data file becomes "sitting,in,car". Use the replacement function of Python to change the 15 misspelled activity names into a underscore delimited format.

| | A | B | C | D | E | F |
|---|-------------|----------------|----------|-----------|-----------|-----------|
| 1 | #Activity | sitting in car | | | | |
| 2 | #ActivityID | 12 | | | | |
| 3 | #Date | 09-19-16-08 | | | | |
| 4 | acc_rf_x | acc_rf_y | acc_rf_z | gyro_rf_x | gyro_rf_y | gyro_rf_z |
| 5 | -7972 | -3564 | 13596 | 47 | -5 | 5 |
| 6 | -7884 | -5172 | 13276 | -6 | -14 | -43 |
| 7 | -8360 | -4696 | 12468 | 6 | 17 | -25 |

Figure 12: “sitting in car” Activity File Sample

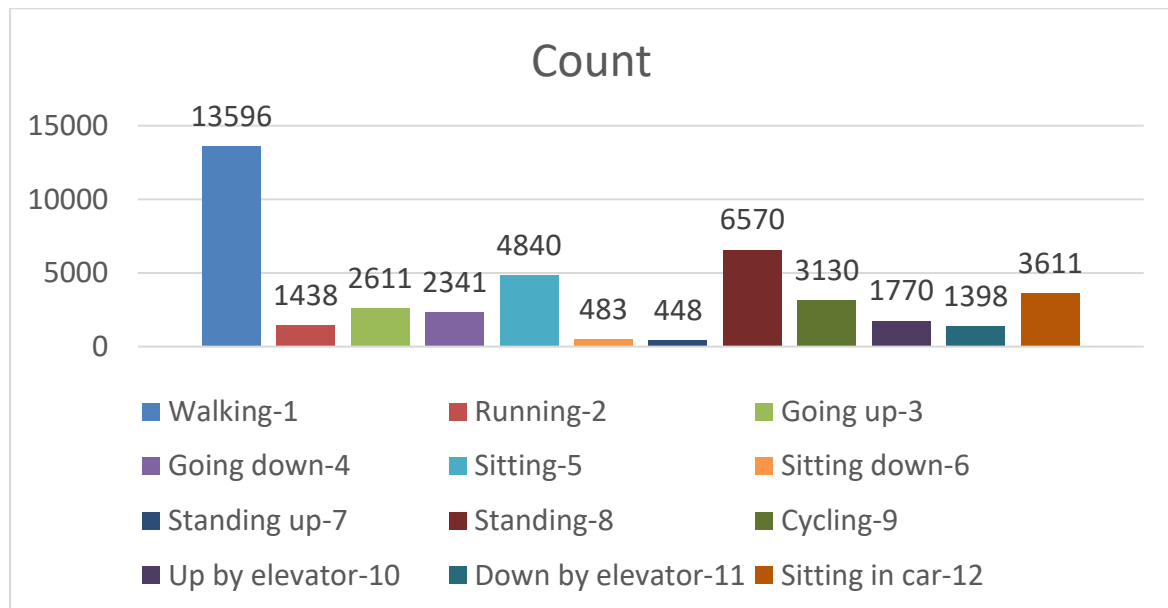
Modify attributes. The activity ID and activity name provided by the dataset are corresponding to each other. For details, see Table 1. When merging files, replace the activity ID in the last column “act” with the corresponding activity name, as the classes attribute of the training set. For example, transfer “12” to “sitting_in_car”.

Irrelevant data. The first three lines of each file are metadata so that the data file needs to be modified structure for file merging. The metadata including activity ID, activity name and date time. In this metadata, the activity ID has been replaced with the corresponding activity name in the main body of the data in the previous step. The data collected by the sensor is already chronological, and time has no effect on activity recognition and classification, so the metadata can be deleted before data merging.

4.1.2 Data Normalization

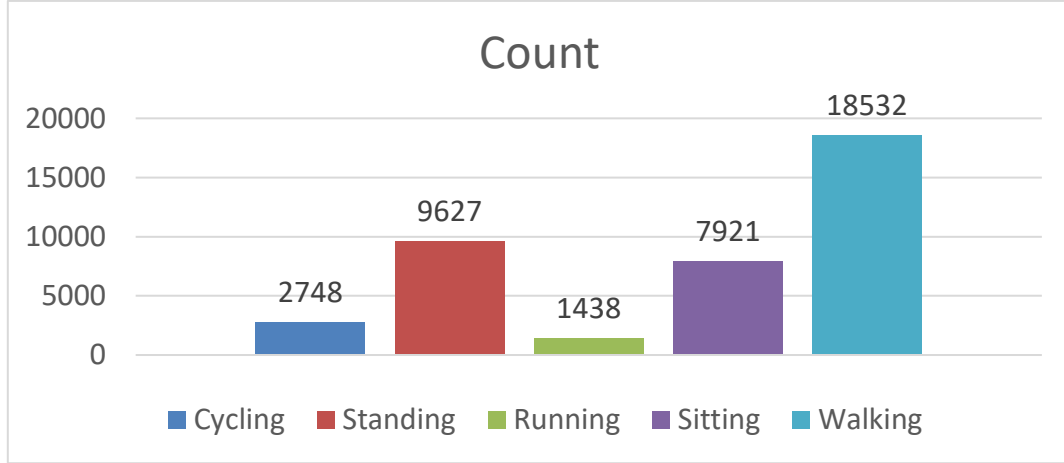
In the data file, there are totally 42236 instances, the number of each activity is shown in the figure below. We can see that there is a lot more kinds of movement than static one. In particular, "walking" accounts for about 32% of the total dataset. And "sitting down" and "standing up" accounted for only around 1%.

Table 3: Dataset classes



Therefore, remove activity categories 6 and 7, and merge 10 and 11 into 8, and 12 into 5, 3 and 4 into 1. So there are five categories, namely cycling, standing, running, sitting and walking. (see Table 5)

Table 4: Dataset classes after combination



4.1.3 Data Normalization

Before analysing the data, data transformation is required. It mainly refers to the normalization of data and the conversion of data into appropriate forms to meet the needs of classification tasks and algorithms. In other words, normalization of data is to scale the data into a specified interval. After the normalization of the raw data, the unit limit of the data is removed, and all variables are in the same order of magnitude, which is suitable for comprehensive comparative evaluation. The commonly used data normalization methods include: Min-max normalization, z-score normalization and fuzzy quantization. Following previous work (Hughes, J. and Iida, F., 2018), this project adopts the z-score normalization method. The basic idea is to subtract the mean from the original value and divide by its standard deviation to get a standard normal distribution with a mean of 0 and a standard deviation of 1.

$$X_{norm} = \frac{X_{raw} - \mu}{\sigma}$$

where μ and σ are the mean and the standard deviation respectively.

For some machine learning algorithms, such as KNN or SVM, normalization is particularly important because they need to calculate the distance.

4.1.4 Pre-processing in WEKA

Weka's data pre-processing, also called data filtering, can be found in `weka.filters`. According to the nature of the filtering algorithm, it can be divided into supervised and unsupervised. For the former, the filter needs to set a class attribute and consider the class attribute and its distribution in the data set to determine the number and size of the best container; Properties of the latter class may not exist. Meanwhile, these filtering algorithms can be reduced to attribute-based and instance-based algorithms. The attribute-based approach is primarily used to work with columns and the instance-based approach is used to process rows.

- **Missing value processing:** *weka.filters.unsupervised.attribute.ReplaceMissingValues*. For numeric attributes, replace missing values with mean value. For nominal properties, use its mode value to replace the missing value.
- **Data Normalization:** *weka.filters.unsupervised.attribute.Standardize*. Normalize the values of all numeric attributes in a given data set to a normal distribution of 0 means and unit variances.

4. 2 Implementation of Features

Features extraction adopted a fixed-width sliding window with 100 samples and 50% overlap (S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, 2009). These features have proven the most effective in the literature of the work in hand [6]. Those features include statistical, time series, and frequency features to extract the information we need from the signal. Table 5 below shows the definitions or formula of some of the adopted features. The nine features namely mean, standard deviation, maximum, minimum, variance, skewness, kurtosis, correlation and covariance are computed for the signals acquired from the x, y and z axes.

Table 5: Definition of some of the time-domain features adopted in the proposed research

| Type | Features | Formula (n = window size, i = row) |
|-------------|--------------------|---|
| Time-Domain | Mean | $mean = \frac{1}{n} \sum_{i=1}^n x_i$ |
| | Standard Deviation | $std = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - mean)^2}$ |
| | Maximum | $max = \max(x_i), i \in \{1, 2, \dots, n\}$ |
| | Minimum | $min = \min(x_i), i \in \{1, 2, \dots, n\}$ |
| | Skewness | $skew = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - mean}{std} \right)^3$ |
| | Kurtosis | $kurt = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - mean}{std} \right)^4 - 3$ |

4.3 Model Building

In this section, six machine learning algorithms introduced in chapter 2 are used to establish classification models for data sets. They are IBK, KNN, NaïveBayes, J48, RandomForest, SMO and ClassificationViaRegression. Following the previous steps, we have completed data pre-processing, with the results as input to the model. The interface for data classification is shown in Figure 13. The specific steps are:

1. Open the training set "Classes. arff" in "Explorer" interface and check out if it is handled as required.
2. Switch to the Classify tab and click the "Choose" button to see that many classification or regression algorithms are grouped in a tree box.
3. After selecting the corresponding classification algorithm, click the text box on the right of "Choose", and a new window pops up to set various parameters for the algorithm.
4. To see the Capabilities, click "More". Here leave the parameters as default.
5. Now look at the "Test Option" module. In order to ensure the accuracy of the generated model without overfitting, 10-fold cross validation was adopted to select and evaluate the model. That is, Weka randomly divided the training set into 10 parts, used 9 parts for training and 1 part for test. A total of 10 experiments will be conducted, and the corresponding results will be obtained for each experiment. The average value of the results of 10 experiments will serve as the final result of the algorithm.
6. Click the "Start" button to Start the algorithm generating the decision tree model.

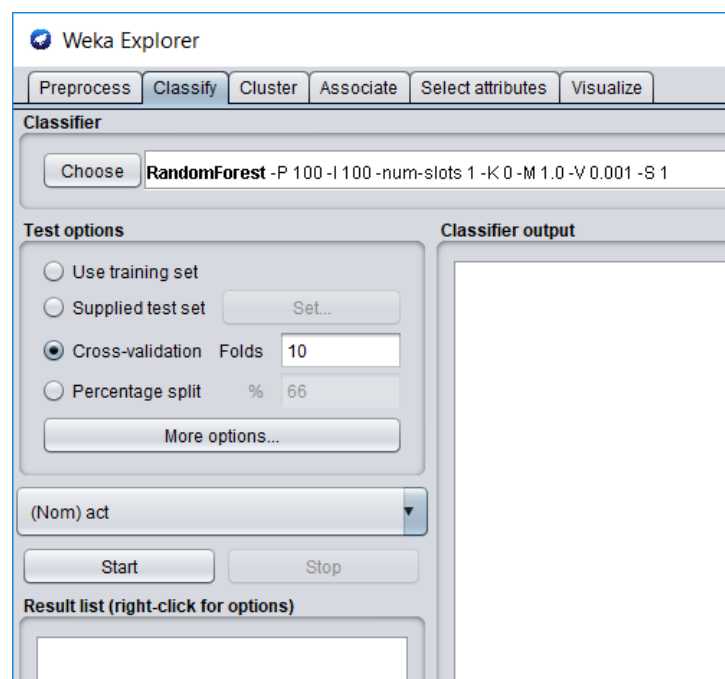


Figure 13: Classify interface

Chapter 5 Evaluation

This chapter presents the methods and experimental results of evaluating classifiers. By comparing the performance of the selected classifiers, the classifiers with the highest accuracy and the lowest accuracy are analysed. Compare and figure out the best way to identify human activity. Finally, this chapter discusses several factors that may affect the recognition accuracy.

5.1 Cross-validation

K-fold cross-validation on training sets has become the main evaluation method in Weka.

There are K subsets, each of which is a test set, and the rest is a training set. Cross validation was repeated for k times, each time a subset was selected as the test set, and the average recognition accuracy of cross validation for k times was taken as the result. The advantage of k-fold cross-validation: all samples are used as training set and test set, and each sample is verified once. Weka defaults to 10-fold. Divide the dataset into ten parts, take turns to train nine parts and verify one point. As an estimation of the accuracy of the algorithm, the mean value of the results of ten times is generally required to conduct multiple 10-fold cross-validation of the mean value.

5.2 Classification Results

The following table 6 to 10 respectively show the classification result of five classes under different algorithms. The bold and underlined font is the maximum value of the column, and the highlighted data is the minimum value of the column. And the speed of modelling is shown in table 11.

Table 6: Result of cycling

| Algorithms | TP | Recall | ROC | Precision |
|-----------------------------|---------------------|---------------------|---------------------|---------------------|
| RandomForest | 0.992 | 0.992 | <u>1.000</u> | <u>1.000</u> |
| J48 | 0.986 | 0.986 | 0.992 | 0.989 |
| NaïveBayes | 0.874 | 0.874 | 0.984 | 0.909 |
| IBK (k=1) | 0.997 | 0.997 | 0.998 | 0.991 |
| SMO | <u>0.999</u> | <u>0.999</u> | <u>1.000</u> | 0.996 |
| ClassificationViaRegression | 0.989 | 0.989 | <u>1.000</u> | 0.997 |

Table 7: Result of standing

| Algorithms | TP | Recall | ROC | Precision |
|-----------------------------|---------------------|---------------------|---------------------|---------------------|
| RandomForest | 0.977 | <u>0.997</u> | <u>0.999</u> | <u>0.988</u> |
| J48 | 0.973 | 0.973 | 0.992 | 0.978 |
| NaïveBayes | 0.959 | 0.959 | 0.978 | 0.953 |
| IBK (k=1) | 0.971 | 0.971 | 0.983 | 0.987 |
| SMO | 0.981 | 0.981 | 0.991 | <u>0.988</u> |
| ClassificationViaRegression | <u>0.979</u> | 0.979 | 0.998 | 0.987 |

Table 8: Result of running

| Algorithms | TP | Recall | ROC | Precision |
|-----------------------------|---------------------|---------------------|---------------------|---------------------|
| RandomForest | 0.951 | 0.951 | <u>1.000</u> | <u>0.959</u> |
| J48 | 0.928 | 0.928 | 0.971 | 0.939 |
| NaïveBayes | <u>0.969</u> | <u>0.969</u> | 0.991 | 0.819 |
| IBK (k=1) | 0.954 | 0.954 | 0.977 | 0.956 |
| SMO | 0.962 | 0.962 | 0.996 | 0.963 |
| ClassificationViaRegression | 0.943 | 0.943 | 0.997 | <u>0.959</u> |

Table 9: Result of sitting

| Algorithms | TP | Recall | ROC | Precision |
|-----------------------------|---------------------|---------------------|---------------------|---------------------|
| RandomForest | 0.996 | 0.996 | <u>1.000</u> | <u>0.997</u> |
| J48 | <u>0.997</u> | <u>0.997</u> | 0.998 | <u>0.997</u> |
| NaïveBayes | 0.975 | 0.977 | 0.988 | 0.989 |
| IBK (k=1) | 0.995 | 0.995 | 0.997 | 0.995 |
| SMO | 0.996 | 0.996 | 0.999 | <u>0.997</u> |
| ClassificationViaRegression | 0.996 | 0.996 | 0.999 | <u>0.997</u> |

Table 10: Result of walking

| Algorithms | TP | Recall | ROC | Precision |
|--------------|---------------------|---------------------|---------------------|-----------|
| RandomForest | <u>0.993</u> | <u>0.993</u> | <u>0.999</u> | 0.985 |

| | | | | |
|-----------------------------|---------------------|---------------------|-------|---------------------|
| J48 | 0.986 | 0.986 | 0.985 | 0.982 |
| NaïveBayes | 0.975 | 0.975 | 0.986 | 0.981 |
| IBK (k=1) | 0.991 | 0.991 | 0.989 | 0.984 |
| SMO | <u>0.993</u> | <u>0.993</u> | 0.994 | <u>0.989</u> |
| ClassificationViaRegression | 0.992 | 0.992 | 0.998 | 0.985 |

Table 11: Time of model building

| Algorithms | Speed (second) |
|-----------------------------|-----------------------|
| RandomForest | 64.9 |
| J48 | 101.97 |
| NaïveBayes | 2.87 |
| IBK (k=1) | 0.03 |
| SMO | 108.71 |
| ClassificationViaRegression | <u>172.08</u> |

5.3 Summary

In general, RandomForest has the highest accuracy and recall value among the six algorithms. SMO is the second best and NaïveBayes is the worst. Except for one NaïveBayes whose accuracy is 81.9%, the accuracy of the other six classifiers can be maintained at more than 95%, with high recognition accuracy.

RandomForest algorithm is one of the most popular machine learning algorithms at present. Compared with decision tree, it has higher prediction accuracy, good tolerance to outliers and noises, and it is not very easy to overfit. It is proved that the random forest algorithm has a high accuracy in the analysis of recognizing human activities.

IBK (k=1) algorithm uses one neighbour to judge categories, which leads to more misjudgements. In this case, IBK can be an alternative. Selecting the appropriate k value can improve the classification accuracy and recall rate. Generally speaking, the selection of k value is judged by experience. Later, k=2, 3, 4 was used for experiments, and it was found that this data set had the highest accuracy when k=2. See Figure 14.

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      39679          98.5422 %
Incorrectly Classified Instances    587           1.4578 %
Kappa statistic                    0.9788
Mean absolute error                0.0059
Root mean squared error            0.0678
Relative absolute error            2.1408 %
Root relative squared error        18.2889 %
Total Number of Instances         40266

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0.998   0.001   0.981    0.998   0.990     0.989   0.999    0.995    cycling
          0.976   0.006   0.980    0.976   0.978     0.972   0.988    0.981    standing
          0.967   0.003   0.932    0.967   0.949     0.947   0.983    0.942    running
          0.994   0.000   0.998    0.994   0.996     0.995   0.998    0.996    sitting
          0.986   0.011   0.988    0.986   0.987     0.976   0.992    0.985    walking
Weighted Avg.  0.985   0.007   0.986    0.985   0.985     0.978   0.993    0.985

=== Confusion Matrix ===

  a    b    c    d    e  <-- classified as
2743   5    0    0    0 |  a = cycling
  41 9399   3   13   171 |  b = standing
   0    0 1390   0   48 |  c = running
   1   35   1  7874  10 |  d = sitting
  10  148   98   3 18273 |  e = walking

```

Figure 14: The result of IBK (k=2)

Although the accuracy of IBK is not the best, the time spent is only 0.03 seconds, which is very fast. Linear regression classification takes the most time, followed by support vector machine algorithm.

NaïveBayes classifier has the worst performance in the experiment. Although it is simple and efficient, Naive Bayes classifier has a strong assumption of conditional independence. In fact, most data sets are difficult to meet this feature, so the classification effect on some data sets is not good. Therefore, some researchers want to retain the simple and funny characteristics of Naive Bayes classifier, and appropriately relax the restriction of conditional independence hypothesis. A semi-naive Bayesian classifier is proposed and some of its algorithms have been applied in practice.

All in all, the experimental results show that the recognition accuracy is the highest by using random forest classifier.

5.4 Factors that improve accuracy

- **The location of inertial sensors.**

When the body is moving, the muscles in different parts of the body move in different ways. Therefore, it is necessary to discuss the location of sensor installation. (daily) studied the

classification accuracy of different sensor positions, and the results showed that the x axis of the accelerometer was on the sitting leg, while the y axis of the gyroscope was on the sitting leg with the highest identification accuracy. Also, (multimodal) concluded that the best position for the accelerometer and gyroscope was in the left thigh. Since the sensor position of this data set is fixed, it is impossible to study the recognition accuracy of other positions on the leg. But sensor location is an important aspect that affects the accuracy of recognition.

- **Dynamic window**

(Chen, Y. and Shen, C., 2017.) used a sliding window of duration 1 second and 50% overlap in the research of human activity recognition based on smartphone-sensor, and classified different movements according to the peak point. (Xiao, D., Yu, Z., Yi, F., Wang, L., Tan, C.C. and Guo, B., 2016) used the sliding window technology with a window size of 2 seconds and a sliding length of 0.5 seconds to obtain sensor data in swimming posture identification. The sliding window sizes used in the above researches are all fixed, but in practice, their signal characteristics are different for various actions. The fixed size window may not be well classified for all actions. Therefore, (Noor, M.H.M., Salcic, Z., Kevin, I. and Wang, K., 2017.) proposed a dynamic sliding window approach with variable window size to adopt to signal characteristics of different actions. The basic idea of this method is to first use a window of fixed size, and constantly determine whether the window size needs to be adjusted through the probability density function in the process of partition, so as to finally get the best window size. The experiment results show that the dynamic window method effectively improves the accuracy of activity identification.

Chapter 6

Conclusion

In this chapter, the overall project is summarized by summarizing and evaluating the challenges faced and the results achieved. In view of the shortcomings of the experiment and the unachieved goals, the future research direction is proposed.

6.1 Project Summary

This paper mainly studies and verifies the human motion recognition method based on wearable inertial sensor. The research process is as follows: firstly, the data are obtained from the inertial sensor and normalized by scaling the sensor data to the same scale. Moving data are segmented and feature values are extracted by sliding window technique. Finally, various classifier algorithms are used to classify and recognize the feature set on Weka. Different algorithms are evaluated through experimental results and factors that may affect accuracy are analysed. In addition, seven of the 18 participants with specific characteristics (such as height and weight) were selected to test whether different individuals could be identified from the sensor data.

The main work of this paper includes the following points:

1. Find a suitable public data set. Data sets related to wearable inertial sensors and human motion recognition are found through keyword search on the Internet.
2. A data normalization method suitable for motion recognition is proposed. In this project, the zero mean standard method is adopted to scale the measurement range of sensor data to a specific form, maximizing the difference between axes of different actions. By comparing the results of unnormalized and normalized, we can know that. Data normalization has higher recognition rate.
3. Implement an effective action data segmentation method -- sliding window technology.
4. Time domain feature extraction of data sets. Nine most conventional and effective time domain features are extracted.
5. Adopt multiple classifiers to model on Weka platform, evaluate the performance of different classifiers for action recognition, and make comparison.

6.2 Achievement of Objectives

1. Understand the background of wearable sensors and human activity recognition, and identify problems and objectives.

This report is based on the recognition of human activity by wearable sensor device data. Through background investigation and literature analysis in chapter 2, the research background of human activity recognition and characteristics of wearable inertial sensors are understood.

2. Collect and familiarise with the HuGaDB dataset.

This project adopts a publicly available wearable inertial sensor gait database with a large number of labels. The detailed database description is shown in section 3.2.

3. Data pre-processing and feature extraction.

Different tools and methods are used for data format conversion, data cleaning, data normalization and feature extraction of the original data. Detailed steps are shown in chapter 4.

4. Identify appropriate tools to build models using machine learning algorithms.

This is achieved and chapter 3 explains the rationale for validating python and Weka as basic tools for working with data sets.

5. Apply analysis by running experiments with different models.

In the experiment in chapter 4, six different classifiers such as Naive Bayes, random forest and support vector machine are used to train the data set and get the results. In addition, additional experiments were conducted to identify individuals with gait characteristics.

6. Evaluate the performance of selected algorithms on the provided dataset.

The results are interpreted, analyzed and evaluated by means of evaluation.

6.3 Recommendation of Future work

With the wide application of inertial sensors in electronic devices, the research on motion recognition and human-computer interaction based on inertial sensors is of great practical value. This paper studies the recognition methods of five kinds of gait activities in daily life. Although the recognition accuracy is very high, there are still some shortcomings. For example, the recognition action is relatively simple, the original feature set is not aligned, the sensor is fixed in the wearing position, and the test data sample is not wide enough. Based on the above problems, the future research direction can be as follows:

Increase the variety of recognition actions. The data set acquisition device adopted in this project is located in the legs of people, unable to capture the motion characteristics of the hand or chest part, so the recognition of hand raising and other movements cannot be

realized. There are still insufficient researches on how to reduce the interference between different movements and improve the recognition rate.

Research on the characteristic analysis of motion data. As for the extraction of motion features, there are several common feature extraction methods for data at present, such as mean value, maximum value and minimum value. This unified feature extraction method cannot directly explain the relationship between it and the specific action to be recognized, and even through subset selection in the later stage, it cannot fundamentally maximize the recognition rate. Therefore, how to improve the original characteristics of chicken related research is also the next research focus.

Research on classification algorithm. Action recognition is essentially a data classification process, so classification algorithm has the most direct impact on the final recognition effect. The six algorithms selected in this project, some of which are binary classification algorithms, are in essence intelligent to achieve the distinction of two categories. If there are five actions to be identified in this project, additional multi-classification algorithms need to be designed, which increases the complexity of the system to some extent. If there are multi-classification algorithms that can get results through one-time data classification, the system complexity will be effectively reduced. Better data classification algorithms should be the focus of further research.

Using sensor data to identify individuals. Due to the lack of in-depth research on the use of sensor data to identify individuals, the following table contains the personal information of 18 participants. Four samples, the heaviest, the tallest, or the lightest and the shortest, can be selected for identification (the four samples underlined). This would also be an interesting research aspect.

Table 12: The Information of the Participants

| ID | Weight (kg) | Height (cm) | Age | Sex |
|----|-------------|-------------|-----|--------|
| 1 | 75 | 177 | 24 | Male |
| 2 | 80 | 183 | 22 | Male |
| 3 | 65 | 183 | 23 | Male |
| 4 | 93 | 189 | 24 | Male |
| 5 | <u>63</u> | 183 | 35 | Male |
| 6 | 54 | 168 | 25 | Female |
| 7 | 52 | <u>161</u> | 22 | Female |
| 8 | 80 | 176 | 23 | Male |

| | | | | |
|-----------|-------------------|-------------------|----|--------|
| 9 | <u>65</u> | <u>175</u> | 24 | Female |
| 10 | <u>118</u> | 183 | 27 | Male |
| 11 | 85 | <u>203</u> | 24 | Male |
| 12 | 85 | 192 | 23 | Male |
| 13 | 64 | <u>174</u> | 18 | Male |
| 14 | 68 | 175 | 19 | Male |
| 15 | 72 | 178 | 23 | Male |
| 16 | <u>48</u> | 164 | 26 | Female |
| 17 | 85 | 179 | 25 | Male |
| 18 | 70 | 180 | 19 | Male |

List of References

- Chereshnev, R. and Kertész-Farkas, A., 2018. RapidHARe: A computationally inexpensive method for real-time human activity recognition from wearable sensors. *Journal of Ambient Intelligence and Smart Environments*, 10(5), pp.377-391.
- Aggarwal, C.C. ed., 2013. *Managing and mining sensor data*. Springer Science & Business Media.
- Zubair, M., Song, K. and Yoon, C., 2016, October. Human activity recognition using wearable accelerometer sensors. In *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)* (pp. 1-5). IEEE.
- Basterretxea, K., Echanobe, J. and del Campo, I., 2014, October. A wearable human activity recognition system on a chip. In *Proceedings of the 2014 Conference on Design and Architectures for Signal and Image Processing* (pp. 1-8). IEEE.
- Sant'Anna, A., Salarian, A. and Wickstrom, N., 2011. A new measure of movement symmetry in early Parkinson's disease patients using symbolic processing of inertial sensor data. *IEEE Transactions on biomedical engineering*, 58(7), pp.2127-2135.
- Badawi, A.A., Al-Kabbany, A. and Shaban, H., 2018, December. Daily Activity Recognition using Wearable Sensors via Machine Learning and Feature Selection. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)* (pp. 75-79). IEEE.
- Perez, A.J., Labrador, M.A. and Barbeau, S.J., 2010. G-sense: a scalable architecture for global sensing and monitoring. *IEEE Network*, 24(4), pp.57-64.
- Anderson, P., 2016. *Web 2.0 and beyond: Principles and technologies*. Chapman and Hall/CRC..
- Chereshnev, R. and Kertész-Farkas, A., 2017, July. Hugadb: Human gait database for activity recognition from wearable inertial sensor networks. In *International Conference on Analysis of Images, Social Networks and Texts* (pp. 131-141). Springer, Cham.
- Heikenfeld, J., Jajack, A., Rogers, J., Gutruf, P., Tian, L., Pan, T., Li, R., Khine, M., Kim, J. and Wang, J., 2018. Wearable sensors: modalities, challenges, and prospects. *Lab on a Chip*, 18(2), pp.217-248.
- Google.co.uk. Google Glass UK. [online] Available at: <https://www.google.co.uk/intl/en/glass/start/>
- Kamišalić, A., Fister, I., Turkanović, M. and Karakatič, S., 2018. Sensors and functionalities of non-invasive wrist-wearable devices: A review. *Sensors*, 18(6), p.1714.

Swan, M., 2012. Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. *Journal of Sensor and Actuator networks*, 1(3), pp.217-253.

Khan, Y., Ostfeld, A.E., Lochner, C.M., Pierre, A. and Arias, A.C., 2016. Monitoring of vital signs with flexible and wearable medical devices. *Advanced Materials*, 28(22), pp.4373-4395.

Kim, J., Campbell, A.S. and Wang, J., 2018. Wearable non-invasive epidermal glucose sensors: A review. *Talanta*, 177, pp.163-170.

McGrath, M.J. and Scanail, C.N., 2013. Wellness, fitness, and lifestyle sensing applications. In *Sensor Technologies* (pp. 217-248). Apress, Berkeley, CA.

Sarcevic, P., Kincses, Z. and Pletl, S., 2019. Online human movement classification using wrist-worn wireless sensors. *Journal of Ambient Intelligence and Humanized Computing*, 10(1), pp.89-106.

Liu, X., Liu, L., Simske, S.J. and Liu, J., 2016, October. Human daily activity recognition for healthcare using wearable and visual sensing data. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 24-31). IEEE.

Awolusi, I., Marks, E. and Hallowell, M., 2018. Wearable technology for personalized construction safety monitoring and trending: Review of applicable devices. *Automation in construction*, 85, pp.96-106.

Varkey, J.P., Pompili, D. and Walls, T.A., 2012. Human motion recognition using a wireless sensor-based wearable system. *Personal and Ubiquitous Computing*, 16(7), pp.897-910.

Yamada, T., Hayamizu, Y., Yamamoto, Y., Yomogida, Y., Izadi-Najafabadi, A., Futaba, D.N. and Hata, K., 2011. A stretchable carbon nanotube strain sensor for human-motion detection. *Nature nanotechnology*, 6(5), p.296.

Kale, A., Rajagopalan, A.N., Cuntoor, N. and Kruger, V., 2002, May. Gait-based recognition of humans using continuous HMMs. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition* (pp. 336-341). IEEE.

Lu, J. and Tan, Y.P., 2010. Gait-based human age estimation. *IEEE Transactions on Information Forensics and Security*, 5(4), pp.761-770.

Yu, S., Tan, T., Huang, K., Jia, K. and Wu, X., 2009. A study on gait-based gender classification. *IEEE Transactions on image processing*, 18(8), pp.1905-1910.

Arra, A., Bianchini, A., Chavez, J., Ciravolo, P., Nebiu, F., Olivelli, M., Scoma, G., Tavoletta, S., Zagaglia, M. and Vecchio, A., 2019, June. Personalized Gait-based Authentication Using

UWB Wearable Devices. In Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization (pp. 206-210). ACM.

Lara, O.D. and Labrador, M.A., 2012. A survey on human activity recognition using wearable sensors. IEEE communications surveys & tutorials, 15(3), pp.1192-1209.

Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J. and Korhonen, I., 2006. Activity classification using realistic data from wearable sensors. IEEE Transactions on information technology in biomedicine, 10(1), pp.119-128.

Maurer, U., Smailagic, A., Siewiorek, D.P. and Deisher, M., 2006. Activity recognition and monitoring using multiple sensors on different body positions. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

Yin, J., Yang, Q. and Pan, J.J., 2008. Sensor-based abnormal human-activity detection. IEEE Transactions on Knowledge and Data Engineering, 20(8), pp.1082-1090.

NicoleLeaper, V. (2009). a visual guide to CRISP-DM methodology. [online] EXDE. Available at: <https://exde.wordpress.com/2009/03/13/a-visual-guide-to-crisp-dm-methodology/>

Mukhopadhyay, S.C., 2014. Wearable sensors for human activity monitoring: A review. IEEE sensors journal, 15(3), pp.1321-1330.

Castillejo, P., Martinez, J.F., Rodriguez-Molina, J. and Cuerva, A., 2013. Integration of wearable devices in a wireless sensor network for an E-health application. IEEE Wireless Communications, 20(4), pp.38-49.

Diniz, V.B., Borges, M.R., Gomes, J.O. and Canós, J.H., 2008. Decision making support in emergency response. In Encyclopedia of decision making and decision support technologies (pp. 184-191). IGI Global.

www.dictionary.com. (n.d.). Definition of gait | Dictionary.com. [online] Available at: <https://www.dictionary.com/browse/gait>.

Ngo, T.T., Makiyara, Y., Nagahara, H., Mukaigawa, Y. and Yagi, Y., 2014. The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication. Pattern Recognition, 47(1), pp.228-237.

S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensorsa review of classification techniques," Physiological Measurement, vol. 30, no. 4, p. R1, 2009. [Online]. Available: <http://stacks.iop.org/0967-3334/30/i=4/a=R01>

Hughes, J. and Iida, F., 2018. Multi-Functional Soft Strain Sensors for Wearable Physiological Monitoring. Sensors, 18(11), p.3822.

Tao, W., Liu, T., Zheng, R. and Feng, H., 2012. Gait analysis using wearable sensors. *Sensors*, 12(2), pp.2255-2283.

Watanabe, K.; Hokari, M. Kinematical analysis and measurement of sports form. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* 2006, 36, 549–557.

Kimmeskamp, S.; Hennig, E.M. Heel to toe motion characteristics in Parkinson patients during free walking. *Clin. Biomech.* 2001, 16, 806–812.

Anon 2013. Triaxial Vibration Collection in Route-Based Machinery Monitoring. Emerson Automation Experts. [Online]. Available from:
<https://www.emersonautomationexperts.com/2013/asset-optimization/triaxial-vibration-collection-in-route-based-machinery-monitoring/>.

MPU-9250 9-DOF 3-Axis Accelerometer, &. Addicore MPU9250 9-DOF 9-Axis Accel/Gyro/Magnetometer. www.addicore.com. [Online]. Available from:
<https://www.addicore.com/mpu-9250-p/ad280.htm>.

ComLab- Course on Biological measurements. Pef.uni-lj.si. [Online]. Available from:
<http://www.pef.uni-lj.si/eprolab/comlab/sttop/sttop-bm/bm-elephy.htm>.

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

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

Xiao, D., Yu, Z., Yi, F., Wang, L., Tan, C.C. and Guo, B., 2016, May. Smartswim: An infrastructure-free swimmer localization system based on smartphone sensors. In *International Conference on Smart Homes and Health Telematics* (pp. 222-234). Springer, Cham.

Noor, M.H.M., Salcic, Z., Kevin, I. and Wang, K., 2017. Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. *Pervasive and Mobile Computing*, 38, pp.41-59.

Anon 2019. MPU9250 Introduction. [Blog.csdn.net](http://blog.csdn.net). [Online]. Available from:
https://blog.csdn.net/qg_41925420/article/details/88654368.

Zhang, D., Zhao, X., Han, J. and Zhao, Y., 2014, May. A comparative study on PCA and LDA based EMG pattern recognition for anthropomorphic robotic hand. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4850-4855). IEEE.

Matsubara, T. and Morimoto, J., 2013. Bilinear modeling of EMG signals to extract user-independent features for multiuser myoelectric interface. *IEEE Transactions on Biomedical Engineering*, 60(8), pp.2205-2213.

Huynh, T. and Schiele, B., 2005, October. Analyzing features for activity recognition. In Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies (pp. 159-163). ACM.

Khan, A.M., Lee, Y.K., Lee, S.Y. and Kim, T.S., 2010. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. IEEE transactions on information technology in biomedicine, 14(5), pp.1166-1172.

Baldominos, A., Saez, Y. and Isasi, P., 2015, July. Feature set optimization for physical activity recognition using genetic algorithms. In Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation (pp. 1311-1318). ACM.

Khachay, M.Y., Panchenko, A., Konstantinova, N., Yavorsky, R.E. and Ignatov, D.I. eds., 2014. Analysis of Images, Social Networks and Texts. Springer.

KEÇECİ, A., YILDIRAK, A., ÖZYAZICI, K., AYLUÇTARHAN, G., AĞBULUT, O. and ZİNCİR, İ., Gait Recognition via Machine Learning.

MPU-9250 9-DOF 3-Axis Accelerometer, &. 2019. Addicore MPU9250 9-DOF 9-Axis Accel/Gyro/Magnetometer. www.addicore.com. [Online]. Available from: <https://www.addicore.com/mpu-9250-p/ad280.htm>.

Appendix A

HuGaDB Dataset

This project uses the publicly available Human Gait Database (HuGaDB), provided by Chereshev R., Kertész-Farkas A. (2018). Database is described in this paper:

https://link.springer.com/chapter/10.1007/978-3-319-73013-4_12 The GitHub link is:

<https://github.com/romanchereshev/HuGaDB>
















 HuGaDB_v1_bicycling_01_12.txt
 HuGaDB_v1_bicycling_01_13.txt
 HuGaDB_v1_bicycling_01_14.txt
 HuGaDB_v1_bicycling_01_15.txt
 HuGaDB_v1_bicycling_01_16.txt
 HuGaDB_v1_bicycling_01_17.txt
 HuGaDB_v1_bicycling_01_18.txt
 HuGaDB_v1_down_by_elevator_12_00.txt
 HuGaDB_v1_running_03_00.txt
 HuGaDB_v1_running_03_01.txt
 HuGaDB_v1_running_07_00.txt
 HuGaDB_v1_running_07_01.txt
 HuGaDB_v1_running_07_02.txt
 HuGaDB_v1_running_08_00.txt
 HuGaDB_v1_running_08_01.txt

Figure 15: Data files samples

| | | | | | | | |
|-------------|-------------|-----------|-----------|-----------|-----------|----------|----------|
| #Activity | cycling | | | | | | |
| #ActivityID | 9 | | | | | | |
| #Date | 10-10-12-53 | | | | | | |
| acc_rf_x | acc_rf_y | acc_rf_z | gyro_rf_x | gyro_rf_y | gyro_rf_z | acc_rs_x | acc_rs_y |
| -14392 | -96 | 10628 | 138 | -388 | 492 | -13980 | -1408 |
| -11248 | 164 | 7744 | 169 | 699 | 429 | -14044 | -2608 |
| -15892 | -1952 | 9324 | -217 | 709 | 360 | -13336 | -3564 |
| -13020 | 892 | 8604 | -297 | -62 | 170 | -13832 | -3220 |
| | | | | | | | |
| acc_lt_y | acc_lt_z | gyro_lt_x | gyro_lt_y | gyro_lt_z | EMG_r | EMG_l | act |
| 2228 | 9120 | 261 | 277 | 522 | 130 | 125 | 9 |
| 3292 | 8916 | 132 | 86 | 581 | 136 | 124 | 9 |
| 2608 | 8796 | 174 | -96 | 641 | 127 | 127 | 9 |
| 4236 | 8312 | 88 | -63 | 553 | 122 | 127 | 9 |

Figure 16: Dataset format

Appendix B

Ethical Issues Addressed

No ethical issues apply for this project as no person related data was processed.

Appendix C WEKA

Here are the results of six classifiers running on WEKA..

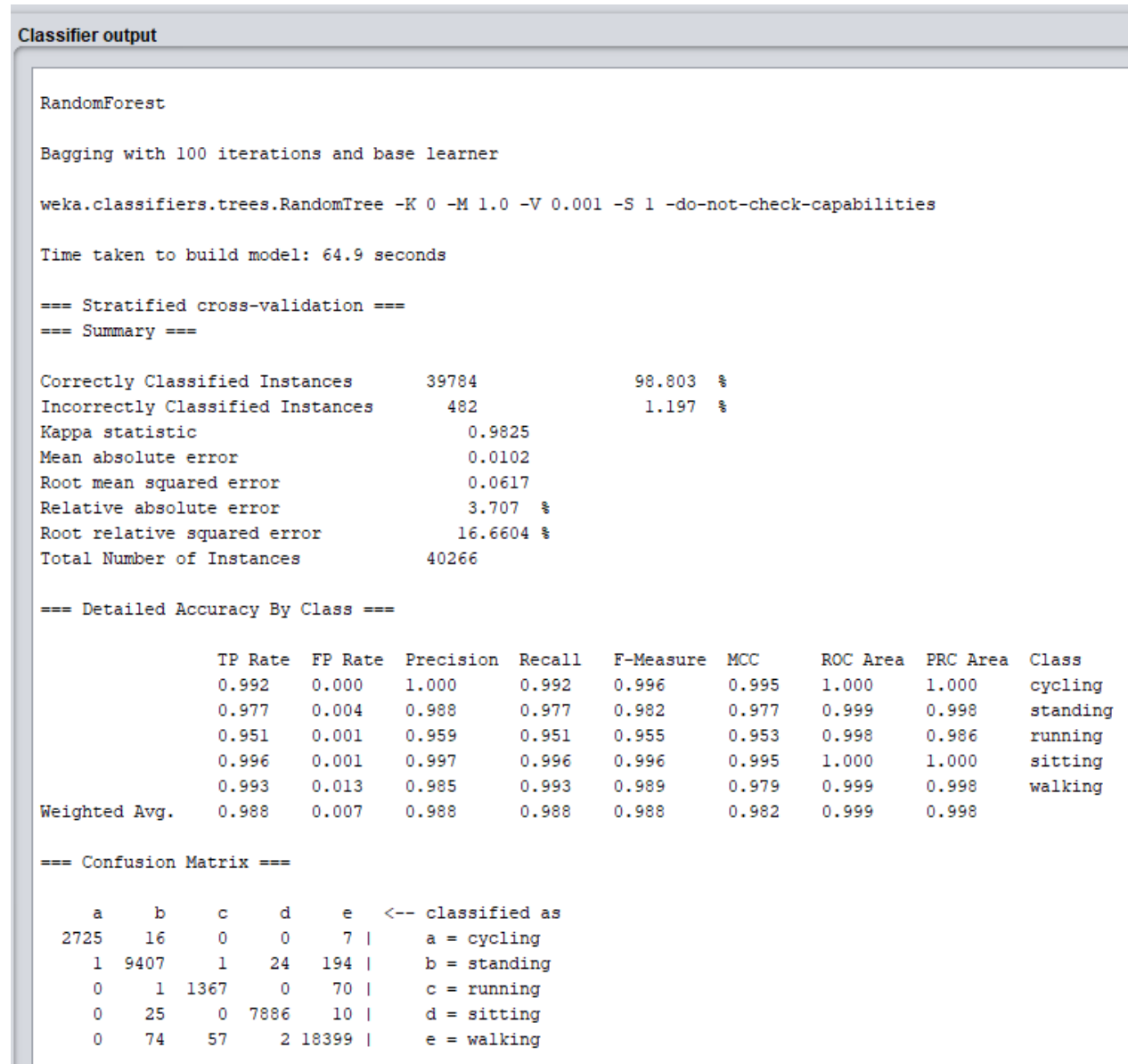


Figure 17: RandomForest

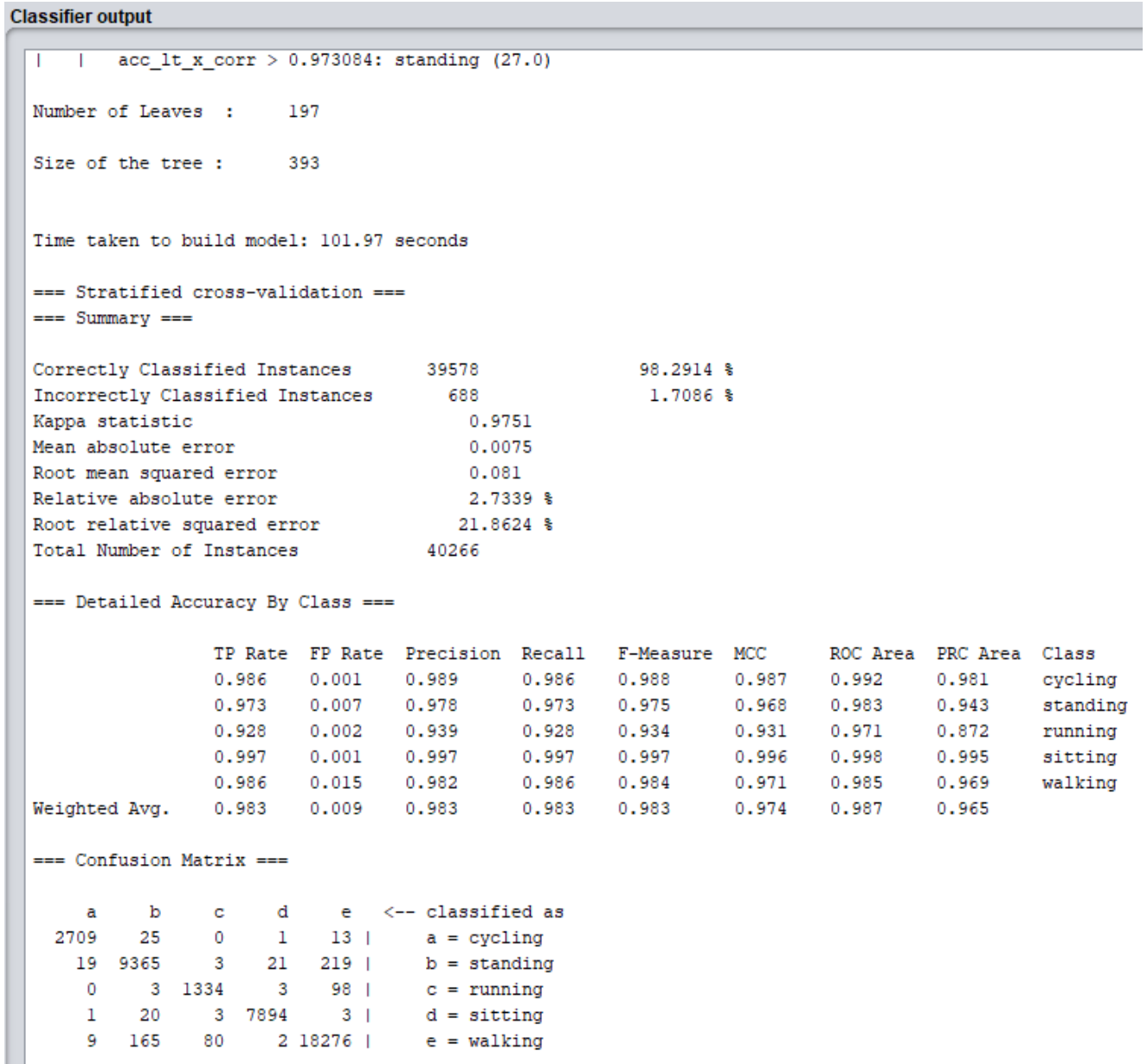


Figure 18: J48

Classifier output

| | | | | |
|------------|---------|---------|---------|--------|
| mean | -0.0773 | -0.0374 | -0.0871 | 0.2013 |
| std. dev. | 0.0649 | 0.2913 | 0.8548 | 0.3136 |
| weight sum | 2748 | 9627 | 1438 | 7921 |
| precision | 0.035 | 0.035 | 0.035 | 0.035 |

Time taken to build model: 2.87 seconds

=== Stratified cross-validation ===

=== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 38833 | 96.4412 % |
| Incorrectly Classified Instances | 1433 | 3.5588 % |
| Kappa statistic | 0.9483 | |
| Mean absolute error | 0.0142 | |
| Root mean squared error | 0.1192 | |
| Relative absolute error | 5.1852 % | |
| Root relative squared error | 32.1586 % | |
| Total Number of Instances | 40266 | |

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|----------|
| | 0.874 | 0.006 | 0.909 | 0.874 | 0.891 | 0.884 | 0.984 | 0.883 | cycling |
| | 0.959 | 0.015 | 0.953 | 0.959 | 0.956 | 0.942 | 0.978 | 0.935 | standing |
| | 0.969 | 0.008 | 0.819 | 0.969 | 0.888 | 0.887 | 0.991 | 0.826 | running |
| | 0.977 | 0.003 | 0.989 | 0.977 | 0.983 | 0.979 | 0.988 | 0.978 | sitting |
| | 0.975 | 0.016 | 0.981 | 0.975 | 0.978 | 0.959 | 0.986 | 0.976 | walking |
| Weighted Avg. | 0.964 | 0.012 | 0.965 | 0.964 | 0.965 | 0.951 | 0.985 | 0.955 | |

=== Confusion Matrix ===

| a | b | c | d | e | <-- classified as |
|------|------|------|------|-------|-------------------|
| 2402 | 254 | 0 | 51 | 41 | a = cycling |
| 105 | 9232 | 1 | 32 | 257 | b = standing |
| 0 | 0 | 1394 | 0 | 44 | c = running |
| 67 | 106 | 2 | 7739 | 7 | d = sitting |
| 68 | 93 | 305 | 0 | 18066 | e = walking |

Figure 19: NaïveBayes

Classifier output

```

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      39720           98.644 %
Incorrectly Classified Instances     546           1.356 %
Kappa statistic                     0.9802
Mean absolute error                  0.0055
Root mean squared error              0.0736
Relative absolute error              1.9913 %
Root relative squared error          19.8759 %
Total Number of Instances           40266

=== Detailed Accuracy By Class ===

```

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|----------|
| | 0.997 | 0.001 | 0.991 | 0.997 | 0.994 | 0.994 | 0.998 | 0.987 | cycling |
| | 0.971 | 0.004 | 0.987 | 0.971 | 0.979 | 0.973 | 0.983 | 0.967 | standing |
| | 0.954 | 0.002 | 0.956 | 0.954 | 0.955 | 0.953 | 0.977 | 0.915 | running |
| | 0.995 | 0.001 | 0.995 | 0.995 | 0.995 | 0.994 | 0.997 | 0.992 | sitting |
| | 0.991 | 0.014 | 0.984 | 0.991 | 0.988 | 0.977 | 0.989 | 0.980 | walking |
| Weighted Avg. | 0.986 | 0.008 | 0.986 | 0.986 | 0.986 | 0.980 | 0.989 | 0.977 | |

```

=== Confusion Matrix ===

  a    b    c    d    e  <-- classified as
2740    7    0    0    1 |  a = cycling
  18 9351    1   32  225 |  b = standing
    0    0 1372    0   66 |  c = running
    1   24    0 7885   11 |  d = sitting
    5   89   62    4 18372 |  e = walking

```

Figure 20: IBK

Classifier output

```

+      5.5112

Number of kernel evaluations: 1521799 (63.09% cached)

Time taken to build model: 108.71 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      39861           98.9942 %
Incorrectly Classified Instances      405           1.0058 %
Kappa statistic                     0.9853
Mean absolute error                  0.2404
Root mean squared error              0.3168
Relative absolute error              87.5699 %
Root relative squared error          85.5043 %
Total Number of Instances           40266

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.999   0.000   0.996     0.999   0.998     0.997   1.000    0.996    cycling
                0.981   0.004   0.988     0.981   0.985     0.980   0.991    0.976    standing
                0.962   0.001   0.963     0.962   0.962     0.961   0.996    0.934    running
                0.996   0.001   0.997     0.996   0.997     0.996   0.999    0.997    sitting
                0.993   0.010   0.989     0.993   0.991     0.983   0.994    0.986    walking
Weighted Avg.   0.990   0.006   0.990     0.990   0.990     0.985   0.995    0.985

=== Confusion Matrix ===

  a    b    c    d    e  <-- classified as
2745    3    0    0    0 |  a = cycling
  10  9445    1   18  153 |  b = standing
    0    2  1383    0   53 |  c = running
    0   23    2  7891    5 |  d = sitting
    0   83   50    2 18397 |  e = walking

```

Figure 21: SMO

Classifier output

=== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 39773 | 98.7756 % |
| Incorrectly Classified Instances | 493 | 1.2244 % |
| Kappa statistic | 0.9821 | |
| Mean absolute error | 0.008 | |
| Root mean squared error | 0.0636 | |
| Relative absolute error | 2.9245 % | |
| Root relative squared error | 17.1742 % | |
| Total Number of Instances | 40266 | |

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|----------|
| | 0.989 | 0.000 | 0.997 | 0.989 | 0.993 | 0.992 | 1.000 | 0.999 | cycling |
| | 0.979 | 0.004 | 0.987 | 0.979 | 0.983 | 0.977 | 0.998 | 0.996 | standing |
| | 0.943 | 0.001 | 0.959 | 0.943 | 0.951 | 0.949 | 0.997 | 0.982 | running |
| | 0.996 | 0.001 | 0.997 | 0.996 | 0.997 | 0.996 | 0.999 | 0.999 | sitting |
| | 0.992 | 0.013 | 0.985 | 0.992 | 0.989 | 0.979 | 0.998 | 0.998 | walking |
| Weighted Avg. | 0.988 | 0.007 | 0.988 | 0.988 | 0.988 | 0.982 | 0.999 | 0.997 | |

=== Confusion Matrix ===

| a | b | c | d | e | <-- classified as |
|------|------|------|------|-------|-------------------|
| 2718 | 15 | 1 | 0 | 14 | a = cycling |
| 6 | 9424 | 0 | 21 | 176 | b = standing |
| 0 | 2 | 1356 | 1 | 79 | c = running |
| 0 | 23 | 2 | 7891 | 5 | d = sitting |
| 3 | 88 | 55 | 2 | 18384 | e = walking |

Figure 22: ClassificationViaRegression

Appendix D

Project Schedule

| Tasks | March | | | | April | | | | May | | | | June | | | | July | | | | August | | | | Sep. | | | |
|------------------------------|-------|---------|---|---|-------|---|-----------------------|---|-----|---|---|-------------|------|---|---|---|------|---|---|---|--------|---|---|---|------|---|--|--|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | | |
| Identify and Collect Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Aims and Requirements | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Background Research | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Project Scoping and Planning | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data Preparation | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Modelling | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evaluation of Results | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Final Report Complete | | | | | | | | | | | | | | | | | | | | | | | | | | * | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Project | | | | | Courseworks and exams | | | | | *Submission | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 23: Original Project Schedule

| Tasks | March | | | | April | | | | May | | | | June | | | | July | | | | August | | | | Sep. | | | |
|------------------------------|---------|---|---|---|-----------------------|---|---|---|-----|---|---|---|-------------|---|---|---|------|---|---|---|--------|---|---|---|------|---|--|--|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | | |
| Identify and Collect Data | ■ | ▶ | | | ■ | | | | | | | ■ | ■ | | | | | | | | | | | | | | | |
| Aims and Requirements | | | ■ | ■ | ■ | ■ | | | | | | ■ | ■ | ■ | | | | | | | | | | | | | | |
| Project outline | | | | ▶ | | | | | | | | ■ | ■ | ■ | | | | | | | | | | | | | | |
| Background Research | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | | ■ | ■ | ■ | ■ | | | | | | | | | |
| Project Scoping and Planning | | | | | ■ | ■ | | | ■ | ■ | ▶ | ■ | ■ | ■ | | | | | | | | | | | | | | |
| Data Preparation | | | | | ■ | ■ | | | | | | ■ | ■ | ■ | | | ■ | ■ | ■ | | | | | | | | | |
| Modelling | | | | | ■ | ■ | | | | | | ■ | ■ | ■ | | | | ■ | ■ | ■ | ■ | | | | | | | |
| Progress Review Meeting | | | | | ■ | ■ | | | | | | ■ | ■ | ■ | | | | | | | ▶ | | | | | | | |
| Evaluation of Results | | | | | ■ | ■ | | | | | | ■ | ■ | ■ | | | | | | ■ | ■ | ■ | ■ | | | | | |
| Final Report Complete | | | | | ■ | ■ | | | | | | ■ | ■ | ■ | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Project | | | | Courseworks and exams | | | | | | | | ▶ Milestone | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 24: Revised Project Schedule