DEVELOPMENT OF CLASSIFICATION ALGORITHMS OF HUMAN GAIT

KOH CHEE HONG

UNIVERSITI TUNKU ABDUL RAHMAN

DEVELOPMENT OF CLASSIFICATION ALGORITHMS OF HUMAN GAIT

KOH CHEE HONG

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Biomedical Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2022

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : Hong

Name : Koh Chee Hong

ID No. : 17UEB04189

Date : 25 April 2022

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DEVELOPMENT OF CLASSIFICATION ALGORITHMS OF HUMAN GAIT"** was prepared by **KOH CHEE HONG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Biomedical Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _______________________

Supervisor : _______________________

Date : _______________________

ACKNOWLEDGEMENTS

I would like to express my gratitude to my research supervisor, Dr. Goh Choon Hian for his invaluable advice, guidance and his enormous patience throughout the development of the research.

Next, I acknowledge the availability of open access gait data from Gait and Aging and Disease Database, Gait in Neurodegenerative Disease Database and Long-term Recordings of Gait Dynamics.

In addition, any attempt at any level can ‘t be satisfactorily completed without the support and guidance of my parents and friends. I am really thankful to them.

At last but not in least, I would like to thank everyone who helped and motivated us to work on this project.

ABSTRACT

Introduction: Gait analysis is essential for diagnosis, assessment, monitoring purpose, and prediction of gait disorder. However, the objective analysis method is less feasible in hospital environments for treatment purposes due to limited coverage of sources. Thus, this study aims to develop a classification algorithm that can effectively classify subjects with relatively simplified input data. ***Methods:*** This study employed several datasets acquired from PhysioNet containing subjects' gait data of three classes. The training dataset contains at total of 48318 instances of three target classes (young healthy adults, old healthy adults, and Parkinson's disease patients). Two classification algorithms were developed: Support Vector Machine (SVM) classification algorithm and Artifical Neural Network (ANN). Preprocessing was performed to the original dataset which includes data cleaning, data normalisation and new features generation. Next, fine-tuning on the manipulating hyperparameters was performed, and k-fold cross validation of k = 10 was used to obtain the average performance of the model. ***Results:*** The optimum confifuration of SVM model can generate an accuracy of 93.01% and F1 score of 92.58% with 43 minutes of computational time. On the contrary, the optimum configuration ANN classifier generates an accuracy of 90.56% and F1 score of 89.69% with 112 minutes computational time. ***Conclusion:*** In conclusion, comparing both of the proposed classification algorithms, the SVM classifier is more effectively than ANN classifier as overall for the gait dataset used in this study. In addtion, after compared with other state-of-the-arts of gait classification algorithms, our proposed classification algorithm produced comparable results with other state-of-arts using a smaller dataset with fewer training features.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

SVM	Support Vector Machine
ANN	Artificial Neural Network
SDG	Stochastic Gradient Descent

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Human gait is a series of alternating sinuous movements of different body segments in a rhythmic pattern that results in bipedal, biphasic forward propulsion of the centre of gravity of the human body with minimal energy expenditure (Hall, 2019). Therefore, evaluation and analysis are done by professionals to research and assess human gait is called gait analysis. Here is a brief introduction to the human gait cycle to further explain this topic. One complete gait cycle consists of many phases, but it can ultimately be split into two global phases: the stance and swing (Nigg and Herzog, 2007). The stance phase refers to the foot first touching the ground and ends when the same foot leaves the ground, this phase makes up approximately 60% of one gait cycle. For swing phase begins when the foot first leaves the ground and ends when the same foot touches the ground again, this phase makes up approximately 40% of one gait cycle. Both of these phases can be further narrowed down to more specific subphases. The stance phase consists of five subphases which are: (1) heel strike, (2) loading response, (3) mid-stance, (4) terminal stance and (5) pre-swing. Next, the swing phase consists of three subphases which are: (1) initial swing, (2) mid-swing and (3) terminal swing (Nigg and Herzog, 2007). Figure 1.1, as shown below, illustrates and visualises the human gait cycle.

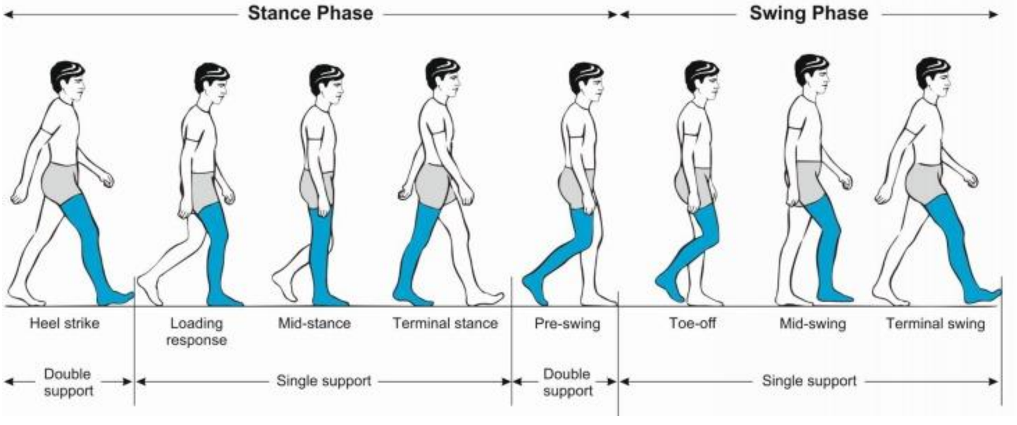


Figure 1.1: Phases in human gait cycle (Pirker and Katzenschlager, 2017).

Gait analysis is a study in the clinical field, especially in the field of rehabilitation. Richard A. Brand had proposed four leading reasons for performing gait analysis in the clinical area: (1) diagnosis between disease entities, (2) assessment of the severity, extent or nature of a disease or injury, (3) monitoring progress in the presence or absence of intervention, and (4) prediction of the outcome of intervention (Brand, 1989). These findings emphasise the importance of applying biomechanics in clinical areas in performing various clinical applications and activities such as patient monitoring, risk prevention, diagnosis of disease, etc. (Baker, 2006). Based on the development and implementation of biomechanics in the clinical field up to date, gait analysis can be break into wearable and non-wearable systems (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014). A wearable system uses an inertial sensor to be attached to the subject's body to acquire data.

On the other hand, a non-wearable system can be classified into two subgroups. Firstly, a motion-image capturing system requires controlled image capture facilities and image processing. Secondly, a floor-based system measures the data when the subject steps on the floor as they walk. Since the data acquired from all three techniques are very much different from each other, thus distinct analysing approaches are developed and implemented for each of the systems to carry out the analysing work. Therefore, choosing the suitable method will directly affect the accuracy and effectiveness of solving our problem.

Aforementioned, there are four main reasons for performing gait analysis in the clinical field. In this project, the scope focus on the diagnosis purpose of the gait cycle. Diagnosis of disease entities in gait analysis mainly implies the identification of gait disorder/abnormalities. Gait disorders are the abnormalities found in the gait pattern of the subject caused by problems in the nervous system and musculoskeletal system. Gait disorder can be generally classified into neurological gait disorder and non-neurological gait disorder. Examples of neurological disorders are sensory ataxia (18 %) and parkinsonian (16 %) gait disorders. Whereas non-neurological disorders such as musculoskeletal disorders (Pirker and Katzenschlager, 2017). Besides the common examples of diseases mentioned, many different gait disorder patterns can be analysed from human gait. Parkinson's disease is one of the most popular and well-known gait disorder diseases out of all these gait disorder patterns. Parkinson's disease is caused by the death or impairment of neurons in the brain that controls movement, which then causes the patient to be unable to control his muscle for movement. For this project, the scope is to develop a classification algorithm to classify patients with Parkinson's disease.

1.2 Importance of the Study

Gait disorder is a degradation of the neurological and musculoskeletal system in the human body, leading to a loss of personal freedom, falls and injuries, and depletion of quality of life. According to a study, the chances of getting gait disorders/abnormalities show a drastic increase from 10 % in people aged between 60 to 69 years to more than 60 % in people over 80 years (Pirker and Katzenschlager, 2017). This project highlights Parkinson's disease, a type of gait disorder caused by the death or impairment of brain neurons that control movement, giving rise to the patient's inability to maintain his muscle for movement. As compared to healthy individuals of the same age, Parkinson's disease patients are observed to have characteristics such as shorter stride length, higher cadence, an increase in double limb support phase, limbs imbalance increase in axial rigidity of hip, knee and ankle motions in their walking gait pattern (Zanardi et al., 2021). These characteristics in walking gait will cause the Parkinson's disease patients to experience inconvenience in movements and an increase in the prevalence of falling.

In recent years, four gait assessment methods have been commonly practised in this area of study. The first method is semi-subjective analysis techniques. Specialists analyse clinical conditions by observing and evaluating the subject's gait parameters through several pre-determined tests/courses (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014). The other three assessment methods are image processing, floor-based sensors and wearable sensor, which are considered objective analysis techniques.

These gait assessment methods are typically performed to analyse and extract meaningful information from the gait pattern of patients. The process requires more professional supervision and does not specify a specific disease or gait disorder. Besides, unlike the gait recognition method that observes the different gait features in patients, a gait classification algorithm can generate a simpler result using classification, which can target a specific problem. For example, in this project, the particular problem is targeted to Parkinson’s disease patients. Hence, a classification algorithm is useful in determining whether the test subject has Parkinson’s disease or is old or young. Therefore, a classification algorithm designed and developed, particularly for Parkinson’s disease, can make the process of gait assessment for Parkinson’s disease patients more intuitive and straightforward.

1.3 Problem Statement

The engineering problem is that the objective analysis method is less practical and feasible in hospital environments such as rehabilitation centres, clinics, and hospitals for treatment purposes. Therefore, a classification algorithm that combines the pros of both semi-objective and objective methods can effectively perform gait recognition, particularly for Parkinson's disease cases, using relatively simplified input data with lesser gait parameters. This approach allows quantitative analysis to classify Parkinson's disease patients more practical and effective in a hospital environment due to the simplified input data. In other words, those advanced and professional equipment is not required, and the data acquisition process can be practical and quickly done in the hospital environment.

In addition, many existing gait recognition algorithms only focus on classifying Parkinson’s and non-Parkinson’s disease patients. Therefore, developing a classification algorithm that can recognise more classes in subjects such as old healthy, young healthy and Parkinson’s disease can help improve the prevention and diagnosis process of Parkinson’s disease treatment.

1.4 Aim and Objectives

This project aims to develop a classification algorithm that can effectively identify young healthy, old healthy and Parkinson's disease subjects using relatively simplified input data with lesser gait parameters. The objectives to be done to achieve the goal include:

- To study and review various existing gait analysis methods.
- To develop and evaluate a Support Vector Machine (SVM) classification model and an Artificial Neural Network (ANN) classification model.
- To compare the performance of both classification models and with the current state-of-the-art.

1.5 Scope and Limitation of the Study

The scope of the study is to develop a classification algorithm that can effectively classify old healthy, young healthy and Parkinson's disease patients. First, the study of gait-affecting factors is important for the overall understanding of the project and helps justify the findings from the project results. Next, the study of gait analysis methods is useful for selecting and processing input data. In addition, the study of classification algorithms is the most important scope of the project because it directly affects the selection and development of classification algorithms.

Limitations include data collection was not conducted for this project due to the COVID-19 pandemic, and gait datasets from other researchers were used for developing the classification algorithm. Besides, the dataset consists of unbalanced examples for different classes due to the constraint that the Parkinson's disease subjects cannot walk for a long time compared to old and young subjects. Therefore, in order to overcome these limitations, other gait datasets should be used to validate the developed classification algorithm.

1.6 Contribution of the Study

This study's findings will redound to society's benefit, especially for the elderly groups, considering that gait analysis plays a vital role in gait research, rehabilitation and biomechanics today. Furthermore, as the knowledge and technology in the gait analysis field grow, the objective gait analysis method will become more important and superior to subjective gait analysis because objective approach is more accurate and reliable. Thus, findings from this study can make the objective gait analysis more practical and trustworthy to be done in an outpatient hospital environment without needing of advanced equipment. This can contribute to popularizing objective gait assessment to people in society. In addition, this study’s findings can contribute to the development of state-of-the-art gait classification on Parkinson’s disease subjects as the training dataset used focuses on gait data of Parkinson’s disease subjects.

1.7 Outline of the Report

Chapter 2 covers a comprehensive literature review of the recent progression of various gait-affecting factors, modern state-of-art gait analysis methods, and classification algorithms. This is followed by Chapter 3, which discusses the methodology in detail on the overall project's progression, which includes preprocessing of study datasets, development of classification algorithms, performance evaluation of classification algorithm, and work plan. Next, Chapter 4 presents and discusses the performance results of the classification algorithms and suggests the optimum configuration for the classification algorithm. Lastly, Chapter 5 covers the conclusion and the recommendations for future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Gait-affecting factors

The two direct gait-affecting factors to patients are physiological factor and psychological factor. Physiological factors are directly related to the human physical body, and psychological factors are related to human mental states. In addition, external factors such as environmental factors and sensor induced factors that affect gait patterns were discussed.

2.1.1 Physiological factors

2.1.1.1 Age

Gait disorder is one of the most dominant factors contributing to falling in the elderly. From recent studies, one-third of the population of elderly experiences falls at least once a year (Kojima et al., 2008). The chances of falling increase steadily with age. The statement is supported by a statistical study by Laurence Z. Rubenstein (2006), the prevalence of falling rise steadily with age, and the rate of loss in older-elderly (aged above 75 years) is shown to be twice of the younger-elderly population (aged 65-75 years) (Rubenstein, 2006). Thus, the falling incident is highly related to gait abnormalities caused by ageing since 70% of falls in the elderly occur during walking (Norton et al., 1997).

As the effects of ageing, humans will experience a decline in the nervous system and musculoskeletal system. The decrease in these body systems will cause a degradation in muscle strength during movement and difficulty controlling body muscles, leading to gait disorder. Therefore, a literature review on the parameters of gait patterns affected by ageing is done. Investigations of gait changes due to ageing are commonly done by comparing gait patterns between the elderly and the young. It is found that out of various gait parameters measured, not all of them are affected due to ageing. Compared to the young, the gait pattern of the elderly shows characteristics of a decrease in stride length, gait velocity and single-leg support time. As the body strength and control of muscles decrease, the elderly tend to decrease their gait speed by taking shorter steps, which increases double support time to maintain the dynamic balance (Ngo et al., 2014).

Furthermore, gait parameters such as swing time of contralateral leg and double support duration do not show significant age-related differences between the elderly and young. In addition, fallers offer a notable drop in stride length and step length and lead to slower walking speed (Auvinet et al., 1997). Thus, this further supports the statement that gait disorder due to ageing is the main factor of falling in the elderly because the risk factors that cause falling are highly overlapping with the gait characteristic of the elderly.

In a study carried out by Ngo et al. (2014), the impact of age on gait performance was discussed by applying the authentication performance method. The authentication performance method is evaluated using Detection Error Trade-off (DET) graph. This DET graph plots False Reject Rate (FRR) against False Accept Rate (FAR), which also can be interpreted as false-negative rate vs false positive rate (Precise Biometrics, 2021). The curve in the DET graph represents the trade-off between FRR and FAR in different authentication scenarios. In other words, as the y-axis represents the number of false-negative (match errors), the curve closest to the bottom of the plot has the best performance and vice versa. Figure 2.1 shows the DET graph gait authentication for different age groups from the research by Ngo et al., 2014.

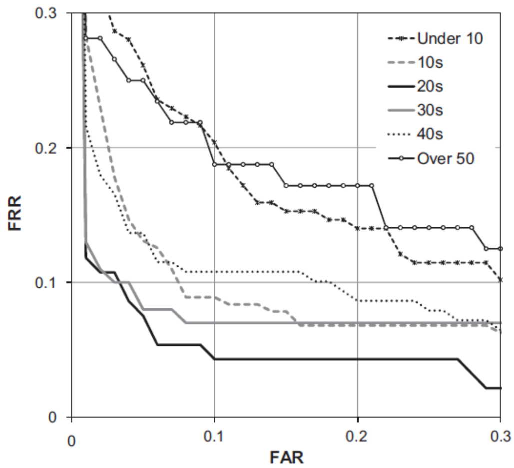


Figure 2.1: Detection Error Trade-off graph of gait authentication for each age interval (Ngo et al., 2014).

From Figure 2.1, the study was conducted on subjects every 10-year interval. The graph shows that the under-10 group and over-50 group authentication performance are the worst among all groups. The result for the over-50 group is understandable as the group is the oldest of them all, and subjects from this group may experience different levels of degradation in the nervous system and musculoskeletal system due to ageing, leading to weaker strength when walking. The reason for the under-10 group to show bad performance is mainly because of the immaturity of their walking skill (Ngo et al., 2014). Subjects from the under-10 group are still developing their walking style, which leads to an unstable gait pattern that contributes to lousy authentication performance. For the 40s group, the performance is better, but the fluctuation is more significant. This is because the body strength starts to degrade at this stage as age increases (Ngo et al., 2014). Finally, in the range between 10-30 years, the 10s and 30s group show good performance and the 20s group has the best performance. The reason is that the body strength and walking pattern are excellent and stable (Ngo et al., 2014).

As a result of ageing, most body functions will degrade. Three of the main factors closely related to gait in old age include lower limb muscle function, vision function and knee joint function. According to Demura et al. (2014), within the test subjects of older women with no visual acuity problem and knee joint pain, those with more muscular knee extension strength will have superior results over weaker knee extension strength (Demura et al., 2014). More muscular knee extension strength will help prevent a decrease in stride length and walking speed which are the main risk factors of falling and gait disorder. Next, researchers also report that age is not a direct factor that affects visual acuity and fall risk.

In addition, fall risk increases when visual acuity decreases and this relationship does not affect by age (Demura et al., 2014). Furthermore, it is reported that patients with knee joint pain due to knee joint disease such as osteoarthritis will show characteristics of gait abnormalities such as a decrease in stride length, step length and walking speed, and an increase in stance time and step width. These gait patterns are caused by the unconscious gait strategy of the elderly to relieve the pain in their knee when walking.

2.1.1.2 Gender

According to findings, women will need to take more steps than men to walk at a similar walking speed. In other words, females have shorter stride lengths, and females usually walk slower than males (Ko et al., 2011). This difference is not correlated with height. One of the gait-affecting features that differ by gender is the hip and ankle range of motion (ROM). It is found that women tend to have a more extended second knee flexion period and longer second ankle plantar flexion compared to men. These two characteristics might be one of the reasons for shorter stride length in women because the early onset of knee flexion causes the foot to touch the ground earlier than in men. Therefore, men rely more on hip angular motion during walking motion, whereas women rely more on ankle angular motion (Ko et al., 2011). Due to the dependence of ankle angular motion in women, women tend to have more energy absorbed in the knee and joint, explaining the higher chances of knee osteoarthritis or other joint-related illness in women (Hunt et al., 2006).

In the study of Kobayashi, Kakihana and Kimura (2014), the findings show that there is a significant effect of gender on the gait symmetry during walking. Researchers also point out that previous studies in earlier stages have reported that gender to gait symmetry and stability are insignificant (Auvinet et al., 1997). In addition, it is found that older women have higher gait variability in stride time and double support time (Ngo et al., 2014). Gait control in older women is more unstable and inefficient than in older men; thus, they will unconsciously increase the double support time to stabilise their gait when walking. In my opinion, gender differences will have a significant effect on gait. However, the result will only become notable for the elderly. Although for the average adult population (not the elderly), gender is not a key factor affecting gait, age has more impact on gait than gender difference.

2.1.2 Psychological factors

In a study conducted by Nagano et al. (2019), the researchers studied the relationship between mental health and gait asymmetry. The study is conducted in three steps: mental health assessment, gait assessment and correlation analysis (Nagano et al., 2019). The participants are residents of

Konosu City (Japan) aged over 50 years, and each of them undergoes a General Health Questionnaire 12 (GHQ-12). The questionnaire consists of 12 questions, and the final score will range from 0 to 12; as such lower score indicates good mental health and an oppositely higher score reflects less positive mental health (Nagano et al., 2019). The gait analysis is conducted using an image capture system; and the correlation analysis is done using multiple regression analysis (SPSS, Inc., Chicago, IL, USA) to determine the correlation of GHQ-12 score with parameters of variables, step length, step width, double support time, and foot-ground clearance (MFC). The result is presented in Table 2.1. The meaningful findings that can be interpreted are that higher scores (poorer mental health) subjects tend to have higher gait asymmetry and walk with broader steps and lower foot-ground clearance (Nagano et al., 2019).

Table 2.1: Result table of correlation of GFH-12 score with gait parameters.

Correlations with GHQ-12	r	P value
Step length (SI)	0.366	<0.001
Step length SD (SI)	0.401	<0.001
Step width (SI)	0.545	<0.001
Step width SD (SI)	0.537	<0.001
Double support (SI)	0.436	<0.001
Double support SD (SI)	0.480	<0.001
MFC (SI)	0.379	<0.001
MFC (SI)	0.545	<0.001

Examples of psychological factors that may affect the gait pattern are depression and fear of falling, commonly found in the elderly. Particularly for fear of falling, the elderly tend to slow down their walking speed and takes smaller steps to increase double stance time to maintain their balance. These changes in gait are done unconsciously, and it will impact gait pattern and gait variability. Another literature from William R. Young, Toby J. Ellmers, Noel P. Kinrade, John Cossar, and Adam J. Cocks (2020) demonstrates that psychological factors have a notable impact on gait patterns. The approach in this study tries to prove the statement from a different perspective. Researchers try to prove that a gait prediction model built using data from questionnaires regarding psychological factors can effectively predict the changes in gait (Young et al., 2020). The questionnaire model is called Gait-specific Attention Profile (GSAP). The questionnaire consists of four types of psychological factors, which is shown in Table 2.2.

The data is collected from 224 older adults with a mean age of 76.53 and a standard deviation of 8.85. The information is then processed through conscious movement processing (CMP) to score out the data. After that, the same participants will complete a walk on a 6-meter automated GAITRite walkway to collect the gait parameters. As the result of prediction, the GSAP^{CMP} model can significantly predict the changes such as slower velocity, shorter step length and longer double-limb support time in participants' gait (Young et al., 2020). P-value is used to determine the level of confidence for the prediction. The definition of P-value in the statistic is the level of significance in which the null hypothesis would be rejected. In this case, the null hypothesis is the correct prediction of gait parameters. In other words, the smaller the P-value, the more significant the result. Thus, the P-value for velocity, step length, and double-limb support is 0.033, 0.032 and 0.015. Theoretically, P-value less than 0.1 would consider highly significant, and thus the prediction of the GSAP^{CMP} is deemed to be substantial, which proves that psychological factors impact gait patterns.

Table 2.2: Factor names and associated Gait-Specific Attentional Profile.

Factor Number	Factor Name	Item
Factor 1	Anxiety	• Feeling of strained • Feeling of concerned about what people think of my movements • Tense
Factor 2	Conscious Movement Processing	• Always think about how I walk or move. • Attempt to control my movements • Always check the way I walk/move

Factor 3	Fall-related Ruminations	• Always imagine falling experience • Always think about what would happen if I fell • Always worried about falling run through my mind
Factor 4	Processing efficiency	• Confusion and making not logical decision • Cannot concentrate on two things at once

In summary, the psychological factors do have a beneficial impact on gait pattern, although the consequences are rather indirect and unscientific compared to physiological factors such as age and gender.

2.1.3 Environmental factors

Environment factors are the manipulating factors that act on the subject when walking. Some environmental factors discussed are shoe types, condition of the surface to walk on, and ground slope.

A study has shown that shoe types have a significant impact on gait patterns. It turns out that heavy shoes will have negative effects on gait performance. Gafurov et al. (2011) reported that subjects wearing heavier shoes would have more unsatisfactory gait recognition performance, reflecting changes in gait pattern due to heavy shoes (Gafurov, Bours and Snekkenes, 2011). This is mainly because the weight footwear requires more energy to walk; the bigger and bulkier footwear can be uncomfortable to walk on. The next factor is the type of surface. According to Sprager and Zazula (2011), the effect of surface conditions on gait is limited and can be neglected as the difference is not significant (Sprager and Zazula, 2011). In this study, the gait performances of the subject walking on four different surfaces: ground, gravel, stone plate and grass are similar. However, the study does not address surfaces with more extreme conditions such as wet surfaces, surfaces covered with holes and surfaces with water puddles. Although walking on a surface with more extreme conditions will change the gait pattern, the changes are relatively less reliable as the change in gait is to avoid the obstacles on the walkway.

Furthermore, the ground slope to walking gait may not be as significant as people and researchers think. For example, a study by Arnold et al., 2015 demonstrated that gait performances of subjects walking on incline slope (+5% gradient) and decline slope (-5% gradient) are similar to the performance of walking on a flat surface (Arnold, LaRose and Agu, 2015). Therefore, the inference that can be made is that as long as the change in surface slope does not change the movement mode, the gait pattern will not experience significant changes as the body muscle can adapt to the change in environment.

2.1.4 Sensor induced factors

In practice, change in gait pattern may not only cause by the factors that arose from subjects and environment; sensor-induced factors also need to be controlled to yield good gait analysis data. Sensor induced factors typically occur in the wearable system as the sensors are attached to different body parts of the subject. Wearable systems can utilise many different types of sensors such as capacitive pressure sensors, accelerometers, and gyroscopes to record body gait data in multiple aspects consisting of specific force, angular rate, and 3D inertial data. The data collected are generally one-dimensional for other gait analysis methods such as image processing and force plate. The placement is fixed as a fixed-motion capture camera system in the image processing method and fixed force plate. The sensor induced factors can be classified into position/location factor and orientation factor.

The position of sensors is vital in wearable system gait analysis. This is because each inertial sensor can only measure single-point motion trajectories (Sprager and Zazula, 2011). Therefore, the result will have a notable difference in gait pattern even though there is a slight variation in the position of sensors. For instance, the placement of sensors at the foot ankle can measure more intensive and valuable gait data than the sensor placed at body parts that are more rigid while performing walking gait, such as the centre back. Thus, consideration of kinetic and kinematic factors of interest body parts needs to be done carefully to select placement points of sensors.

Ngo et al. (2014) also reported that the gait performance for the sensor placement at the front and the back of the waist is quite similar (Ngo et al., 2014). This finding further supports that the more rigid part of the body yields less valuable data, and focussing too much on collecting data from these parts will not be a good decision. In addition, the gait data collected will also be affected by the way the sensors are attached to the body. In practice, the assumption is that all measurements are performed with the sensors placed precisely at the desired spot for every repetition. The assumption is valid for professional analysis by specialists as specialists will ensure that the sensors are attached as firmly as possible at the desired spot. However, in some commercialised gait monitoring systems that subjects themselves can do, the result would be more likely to be less discriminative (Sprager and Zazula, 2011). The reason for this is that users prefer sensor positions that is more common such as inside trouser pocket, inside purse or wallet, and even holding them in hand during gait measurement.

Problem orientation invariance in gait analysis of wearable inertial sensors has become an issue throughout wearable gait analysis systems development. Early-stage approaches focus on implementing acceleration signals from an accelerometer, which is invariant and insensitive to measure orientation. Therefore, the problem arises as the computation only utilises the orientation invariance signal. As a result, there will be a significant loss of information and a decrease in the distinction of gait patterns (Sprager and Zazula, 2011). Another study introduced a system utilising a triaxial accelerometer and gyroscope with calibration with transformation logic in recognition procedure to resolve sensor orientation inconsistency (Sun, Wang and Banda, 2014). According to Ngo et al. (2014), four dissimilarity measure models are used to identify the accuracy between accelerometer and gyroscope, as shown in Figure 2.2. Accelerometer produces better authentication performance in all dissimilarity measures except for TANIMOTO model (Ngo et al., 2014). These results indicate that including an orientation-variant signal from a gyroscope can measure more informative and multidimensional data. Still, the drawback is the complexity will reduce the performance of the recognition model. Over the years, more advanced and effective approaches were introduced to resolve orientation inconsistency in gait analysis. For example, Zhong et al. (2014) proposed an invariant gait representation model that implemented 2D representation gait dynamic images capable of capturing consistent motion dynamics over time from data acquired from accelerometer and gyroscope (Zhong and Deng, 2014).

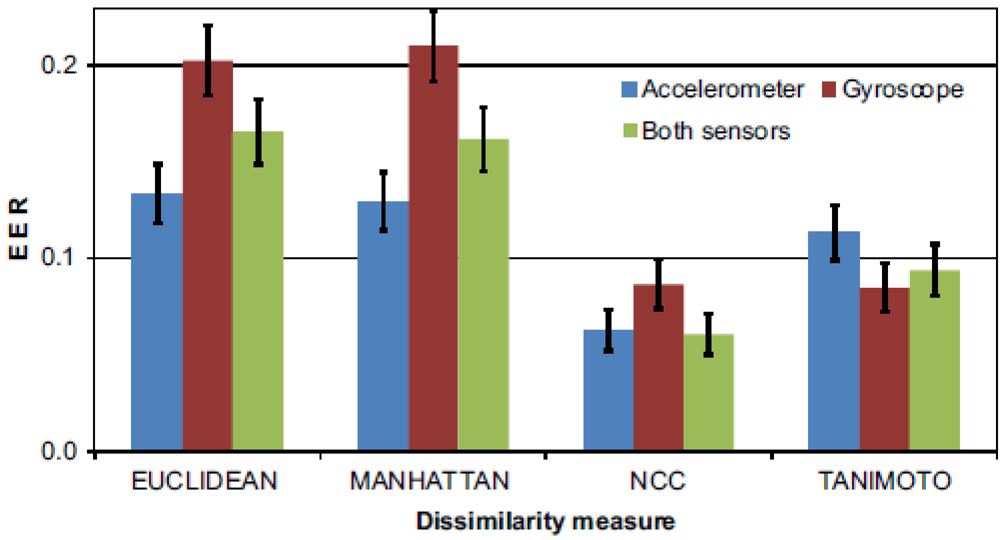


Figure 2.2: Equal error rate for dissimilarity measures versus sensor types (Ngo et al., 2014).

2.2 Gait analysis methods

A semi-subjective analysis is an analysis method widely spread in clinical fields, especially in rehabilitation centres, clinics and hospitals for clinical treatment purposes. One of the advantages of this method is that the result is more on a qualitative result. Thus, the result is relatively easy to understand by patients themselves and easier to record and evaluate by a specialist. However, although this approach is informative in certain situations, the results from these observations are usually considered restricted and less accurate due to the qualitative nature of the study and the reliance on the experience of the specialist who performed the assessment (Muñoz Ospina et al., 2019).

On the other hand, objective gait analysis methods are often applied for research and experiment purposes. Compared to the semi-qualitative analysis, objective analysis is more on quantitative methods. Thus, the results data collected are more accurate and informative. Furthermore, the data can be processed and analysed by a specialist using statistical tools or analysed by classification/recognition algorithm to identify better gait disorder of the subject (Djurić-Jovicić et al., 2014). The limitation of objective gait analysis is that advanced and professional equipment and analytic tools are required, limiting their application in rehabilitation centres and hospitals.

2.2.1 Semi-objective analysis

Entirely subjective based gait analysis is uncommon now as analysis nowadays can only be more reliable and convincing enough to people when there is data to support it. Therefore, a semi-objective approach is introduced and commonly used as a primary assessment tool for treatment purposes by specialists in the hospital environment. Furthermore, semi-objective is helpful in direct clinical practice. This method can overview patient gait abilities through a faster and easier procedure without or with minimum equipment compared to the objective gait analysis approach (Moissenet and Armand, 2015).

2.2.1.1 Questionnaire-based scales

The aim is to give an overall score that helps evaluate a patient's gait ability by answering questionnaires. Questionnaires that measure for Parkinson's Disease include the SF-12 health survey, fall status, Short Falls Efficacy Scale-International (Short FES-I), the visual analogue scale (VAS) for pain, UPDRS (parts I, II, and III), and H&Y staging (disease stage) (Toosizadeh et al., 2015). SF-12 health survey focuses on physical and mental components in evaluating the generic health status of patients. FES-I is used to estimate the level of fear of falling to the falling rate among older people. Toosizadeh et al. (2015) suggested that a shorter validated version of the SF-12 health survey (12 instead of 35 items) and FES-I (7 instead of 16 items) can be adapted to evaluate the patients when the assessment procedure has to be done in one visit (Toosizadeh et al., 2015). Next, VAS assessment assesses pain attributes as it usually comes with Parkinson's disease. Finally, UPDRS consists of three parts which correspond to rate patient's disability on behaviour and mood (Part 1), activities of daily living (Part 2), and motor examination (Part 3). To ensure the validity and understanding of the questionnaire, it is crucial to make sure the questionnaires are adapted according to the target age or disorder; and valid with correct translation (Moissenet and Armand, 2015).

2.2.1.2 Observation-based scales

Observation-based scales aim to evaluate patients' gait patterns and gait performance through specialists' direct or indirect (video recordings) approach. There are generally four aspects to conduct during qualitative observation assessment, and the steps are elaborated in Table 2.3 (Malani, 2008).

Table 2.3: Different aspects to observe during qualitative assessment.

Aspects	Description
Preliminary evaluation	• Perform a neurologic test on cranial (controls visual fields and acuity), cerebellar (heel to the shin, Rhomberg) and peripheral nervous systems. • Check for foot sensation and proprioception. • Check for musculoskeletal abnormalities and deformities.
Standing and balance	• Observe how the patient rises and stands from a sitting position. • Observe the performance of the patient turn 360 degrees with eyes closed.
Walking	• Observe whether the patient has difficulty when walking, such as hesitancy or multiple attempts. • Observe the step height of both feet and the symmetry of the left and right sides.
Endurance	• Observe the sign of fatigue during walking.

Many classification systems can help researchers and specialists to conduct qualitative gait assessments. One of the most well-known classification systems is Gross Motor Function Classification System (GMFCS) (Moissenet and Armand, 2015).

Table 2.4: Detailed description of Gross Motor Function Classification System.

Level	Description
Level 1	Children take walks at home, school, outside, and in the neighbourhood. They are able to climb stairs without using a guardrail. Children perform gross motor abilities such as running and jumping, but their speed, balance, and coordination are limited.
Level 2	Children walk and ascend stairs while holding onto a railing in most situations. Due to uneven terrain, inclines, or cramped quarters, they may have trouble walking long distances and balance in busy settings or limited spaces. Over extended distances, children can walk with physical help, a hand-held mobility device, or wheeled mobility. Gross motor skills such as running and jumping are limited in children.
Level 3	In most indoor situations, children walk with a hand-held mobility device. They can ascend steps with supervision or support while holding on to a handrail. When travelling lengthy distances, children use wheeled mobility and may self-propel for lesser distances.
Level 4	Children adopt means of movement that need physical help or powered mobility in most situations. They can walk short distances at home with physical help or use powered mobility or a body support walker after they are properly positioned. At school, outside, and in the community, children either use a manual wheelchair or motorised mobility.
Level 5	In all contexts, children are conveyed in a manual wheelchair. As a result, maintaining antigravity head and trunk positions and controlling leg and arm motions are difficult for children.

2.2.1.3 Techniques to measure the performance of gait

In semi-objective analysis, tests and techniques will be applied to patients to collect quantitative data on patients' gait. These tests require minimum demand of equipment and are easy to perform. The parameters measured in these tests may not be as advanced and detailed as those measured in objective analysis. However, these tests help specialists quickly overview the patient's gait performance, which is very useful to complete the assessment. Some of the most common and popular techniques used in clinical practice are discussed in Table 2.5 (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014).

Table 2.5: Techniques used to measure the gait performance of patients during semi-objective analysis.

Techniques	Description
Timed 25-Foot Walk (T25-FW)	In this test, a specialist will record the time taken for the subject to walk in a 7.5m long straight line. This test is also used in the Multiple Sclerosis Functional Composite (MSFC) as the first assessment part. MSFC is a standardised quantitative evaluation instrumentation module that uses clinical studies, especially in multiple sclerosis (Cutter et al., 1999).
Tinetti Performance-Oriented Mobility Assessment (POMA)	In this test, the patient must stand up from the chair and walk forward for at least 3 meters, turn 180 degrees, and walk back to the chair in a quicker manner. This test can effectively evaluate balance and gait disorders in the elderly.
Timed Get up and Go (TUG)	In this test, time is recorded for the patient to complete a standardised series of movements. For example, starting from a sitting position, standing up, performing a short distance walk

	forward, turning around, returning to the chair, and sitting down.
Multiple Sclerosis Walking Scale (MSWS-12)	This test assesses the gait performance of the subject through 12 parameters. The 12 parameters are concluded by the researchers based on the result of interviews with 30 patients, expert advice and feedback, and literature review regarding the effect of multiple sclerosis on a patient's gait (Hobart et al., 2003).
Extra-Laboratory Gait Assessment Method (ELGAM)	This method is gait evaluation, used commonly in-home or in the community. In this method, parameters in gait are collected and analyse the gait performance. Examples of the gait parameters include step length, step speed, gait style, and ability to turn the head while walking and balance in a static position.

2.2.2 Objective analysis

In contrast to semi-objective gait analysis, objective gait analysis makes use of different devices to capture and measure various gait parameters that can represent the gait pattern of patients. There are two types of objective gait analysis methods, wearable system and non- wearable system.

2.2.2.1 Wearable system

A wearable gait analysis system is done by attaching wearable sensors to the subject's body to measure different gait parameters and gait characteristics. Some commonly used sensors to conduct the wearable gait analysis include force sensors, pressure sensors, accelerometers, gyroscopes, etc. In practice, there are three main achievements of the application of wearable sensors: kinematics, kinetic and electromyogram (EMG) (Tao et al., 2012). Therefore, in this part of the literature review, the methods and sensors used are reviewed based on these three research areas.

Kinematics of the human gait describes the movements of body parts in gait, including the major joints and lower extremity's components, without considering forces (di Biase et al., 2020). An accelerometer is used to measure the acceleration motion of the body part it is attached to. Three types of accelerometers are usually used in practice: piezoelectric, piezoresistive, and capacitive accelerometers. Piezoresistive and capacitive accelerometers can give dual acceleration components and are more stable (Tao et al., 2012). In the early stage of the study of kinematics application of wearable accelerometers, Mathie et al. (2004) discussed the use of accelerometer-based systems in human movement, including monitoring a range of motion (ROM), measuring physical activity levels, and identifying and classifying movements performed by subjects (Mathie et al., 2004). They also discussed a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring (Karantonis et al., 2006). As accelerometer alone can only collect a limited amount of information, other sensors such as gyroscopes and magnetometers are combined with accelerometers to form the inertial sensor. Gyroscopes can measure the angular rate and the angle of various joints on the lower extremities; a magnetometer can measure relative results for body orientation (Tao et al., 2012). Inertial Measurement Units (IMUs) are one of the most popular types of inertial sensor used by researcher in gait analysis. In some circumstances, goniometers are used to measure the angles for ankles, knees, hips and metatarsals.

Secondly, gait kinetic studies forces and moments that contribute to the movement of body segments and gait motion (di Biase et al., 2020). Feature of kinetic generally focuses on the Ground Reaction Force (GRF) measurement between foot and joints such as the ankle, knee, hip, and pelvis with the ground. Earlier methods use stationary systems to perform kinetics studies on human gaits, such as force plates and instrumented treadmill devices. However, wearable pressure and force sensors are slowly replacing conventional methods due to the limitation of the force plate and treadmill, such as low portability, availability, cost, etc. In addition, pressure and force sensors come in much smaller sizes and are cost-effective. For example, some of the commonly used sensors in practice include resistive sensors, in which their resistance decreases when the force applied to them increases;

piezoelectric sensors, in which the deformation of gel is measured as pressure is applied to them; and capacitive sensors, in which the condenser capacity changes according to different parameters. Liu, Inoue and Shibata (2010) designed a wearable GRF system prototype, as shown in Figure 2.3. Five small triaxial force sensors were used to build the prototype (Liu, Inoue and Shibata, 2010). GRF and CoP can be easily calculated using the triaxial forces Fxi, Fyi, and Fzi measured from the five small triaxial force sensors. Liu, Inoue and Shibata, 2010 reported that wearable sensor systems could be used in clinical applications, namely clinical decisions, and assist in medical diagnosis as the measured data is suitable to analyse the kinetics of the ankle, knee, and hip joints (Liu, Inoue and Shibata, 2010).

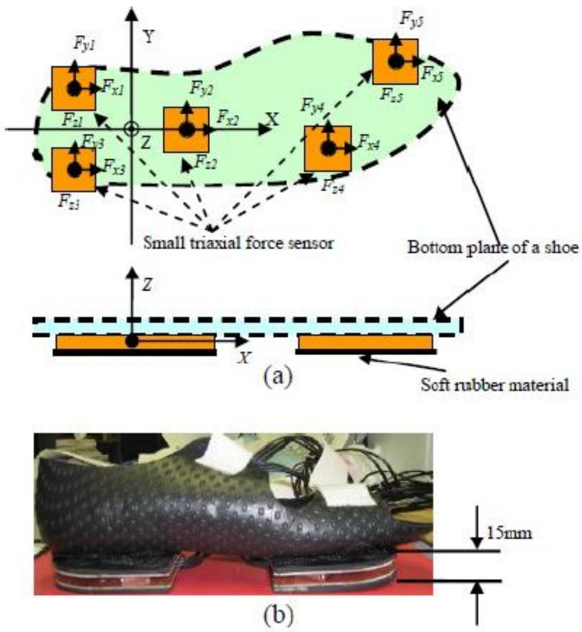


Figure 2.3: A wearable Gound Reaction Force sensor system prototype built using five small triaxial force sensors. (a) The ordinate of sensors and sensor mechanism; (b) Prototype of an instrumented shoe for the right foot (Liu, Inoue and Shibata, 2010).

An electromyogram is an electrical manifestation of muscle contraction (either a voluntary or involuntary muscle contraction) by measuring the small electric current produced during muscle contraction (Tao et al., 2012). EMG technique can be used to observe muscle activity during human walking gait. The measuring equipment for EMG can be surface electrodes (non-invasive) or using wire or needle electrodes (invasive) (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014). EMG data of walking gait is a piece of precious information in clinical gait analysis. With the development of wireless technology in EMG sensor applications, EMG has become a mature and reliable wearable measurement tool for gait analysis.

Furthermore, surface EMG is a valid non-invasive assessment for pathophysiological mechanisms that may affect the gait function, such as changes in passive muscle-tendon properties, paresis, spasticity, and loss of selectivity of motor output in functionally antagonist muscles (Frigo and Crenna, 2009). Besides, EMG analysis can recognise the neural injury or compression, denervated muscles, or primary pathological processes of the specific localised muscle group. This application is beneficial, particularly in studying exercise physiology, athletic training, ergonomics, physical therapy, and physical medicines (Dietz et al., 1995; Dietz, Leenders and Colombo, 1997; Whittle, 1996). A more recent study by Wentink et al. (2014) reported that EMG data measured from the prosthetic leg could predict the initial movement in gait initiation with a latency up to 138ms in advance (Wentink et al., 2014).

Analysis techniques for gait recognition and identification of gait characteristics can generally be divided into functional analysis and machine learning approach (Rueterbories et al., 2010). First, functional analysis is a mathematical approach to extract features by curve sketching and analysis, and this method is commonly used to indicate certain gait phases or events. Next, the machine learning approach is more to gait recognition. Common algorithms used for wearable systems include neural networks, mutual information classifiers, fuzzy computational algorithms, and support vector machines (Rueterbories et al., 2010). In addition, in the context of kinetic analysis, the inverse dynamics method is another commonly used approach due to its simplicity and ease of application (Tao et al., 2012).

2.2.2.2 Non-wearable system

2.2.2.2.1 Floor-based system

A floor-based system is a stationary gait measurement system that implements a sensor or device along the floor for the subject to walk on. A floor-based system is suitable for measuring GRF with exact accuracy. In addition, it can compute more gait information such as length, time and speed parameters for step and stride by measuring the pressure over time data. Floor-based instrumentations commonly utilise a combination of pressure and force sensors to acquire the desired data, such as GRF in 3-axis, the centre of gravity, and shear and moment components of applied forces (Robertson et al., 2013). There are three types of floor-based systems that have been researched in a practical situation in gait analysis.

Firstly, a force plate device is a complex sensor matrix consisting of pressure and force sensors. Force plates can measure GRF in multiple axes and the centre of pressure very accurately on the area of the plates. However, several studies have reported that force plate is not suitable for gait analysis due to its constraint. Force plates can provide accurate measurement within the plate area, but the size/area for subjects to stand on restricts the measurement for more than one stride. Thus, to perform gait analysis for continuous walking across certain distances, a complicated system comprising many force plates and a data fusion method must be constructed (Rabuffetti and Frigo, 2001; Chen et al., 2010). Therefore, force plates are more suitable for stationary applications such as balance training, development of balancing medical devices and sports analysis like vertical jump analysis.

Secondly, it is the floor sensor mat. The concept of this approach is to place the force sensors embedded in a mat and the mat placed on the floor to form a walkway for subjects to walk on. A prototype of floor sensors mat from Middleton et al. (2015) is depicted in Figure 2.4. The prototype utilises capacitors as pressure sensors. The mat consists of 16 by 96 sensors, and the sensors are arranged into grids with the assumption that the subjects' footsteps do not overlap along the line of their forward motion to avoid the ghosting effect (Middleton et al., 2005). As a result, the prototype can collect foot profile and gait pressure data over time, as illustrated in Figure 2.5 and Figure 2.6. The advantage of this approach is that the construction method and process are simple and cheap, and it can measure continuous walking behaviour (Middleton et al., 2005; Leusmann et al., 2011).

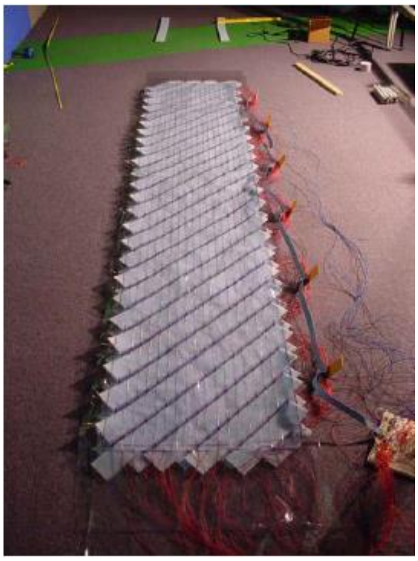


Figure 2.4: Prototype of sensor mat on the floor (Middleton et al., 2005).

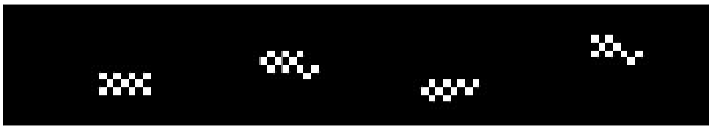


Figure 2.5: Footsteps profile (Middleton et al., 2005).

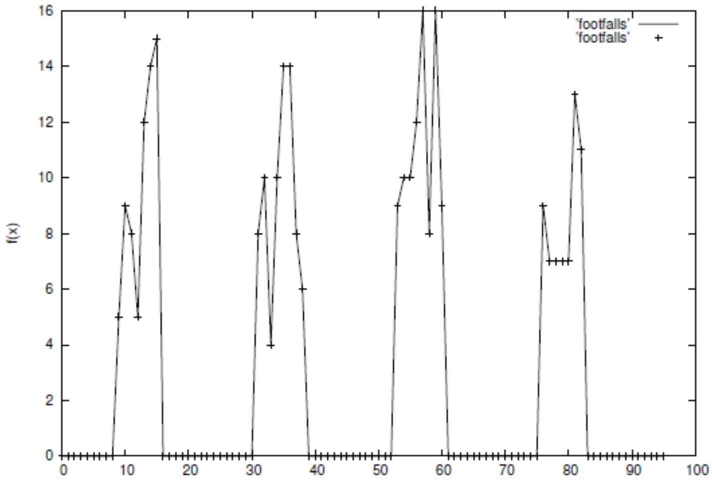


Figure 2.6: The pressure of four footsteps on the sensor mat (Middleton et al., 2005)

Thirdly, instrumented treadmills have all of the functionality of ordinary treadmills but with pressure or force sensors to measure kinetics characteristics of gait. Instrumented treadmills provide spatiotemporal gait information and GRF in an almost instantaneous manner. The data are reported to have a high-reliability level, with coefficients of variation (CVs) of less than 10% for data collected using this system (Kesar et al., 2011). Reed et al. (2013) reported that newer instrumented treadmill technology that incorporates pressure transducers demonstrates high reliability and accuracy compared to the conventional technology, which implements force sensors (Reed, Urry and Wearing, 2013). The system up to date has been used in research areas related to gait analysis, for example, fundamental control mechanisms in gait, disturbances associated with neurological disorders, including Parkinson's disease and cerebellar ataxia; and as an assessment tool to evaluate the effectiveness of various clinical, and neurorehabilitation trials; and for monitoring the progression of ergonomic training programs (Reed, Urry and Wearing, 2013). In the study conducted by Donath et al., 2016, an instrumented treadmill compromising a high-quality capacitive force sensors matrix and analysis software is compared to a portable gait analysis system, RehaGait to evaluate the validity and reliability of gait data collected (Donath et al., 2016). The result shows that the RehaGait system can obtain a similar result to the instrumented treadmill most of the time. However, the RehaGait system is observed to overestimate the data of stride length (+2.7 %) and stride time (+0.8 %) and underestimate the cadence (−1.5 %) (Donath et al., 2016). Therefore, the instrumented treadmill can provide data with superior accuracy, and the versatility to adjust the slope and speed still makes it the essential instrumentation in gait analysis. However, constraints still exist as the assessment on the treadmill will always be along a straight line, and the subjects cannot experience turning or direction changes that reflect the performance during real-life activities.

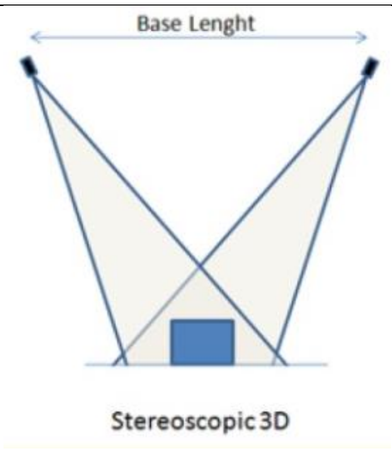
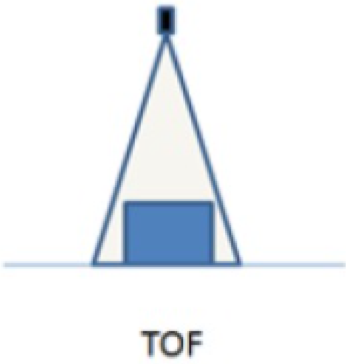
2.2.2.2.2 Image processing system

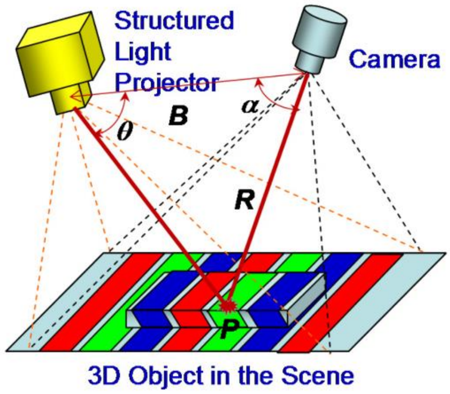
An image processing method is a non-invasive human gait analysis as there are no sensors attached to the human body. This method uses several digital or analogue cameras with lens that can be used to gather gait-related information. Compared to a motion capture system that requires markers to be attached to the human body, the process to deliver this analysis is simplified by a lot as the preparation time is reduced. Besides, there is no more restriction to the human body when assessing, allowing accurate and efficient motion assessment in more applications.

In the image processing analysis, one parameter is very important to control in the analysis: depth measurement (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014). The depth information data can increase the dimensionality of the image to a 2.5D depth image. A 2.5D image is technically 2D but with depth information. Therefore, less work needs to be done to process this type of image as the 2.5D dimension simplified the task for foreground/background subtraction. Four techniques can be applied during image capture phase to capture 2.5D depth images. The techniques are organised and discussed in Table 2.6.

Table 2.6: Different image capturing techniques in the image processing method.

Techniques	Description
Stereoscopic Vision	A stereo camera system is used to capture the image of the subject. The stereo images will undergo stereo matching process to identify the depth measurement in the image. In the stereo matching process, the first step is to identify the corresponding feature points in a pair of stereo image. Next, the disparity of all of the matching points is calculated. Then, using the disparity, the known calibrating parameter and known 2D coordinate to recover into 3D coordinate and form 3D image (Liu, Cao and Wang, 2010).

	
Time-of-Flight Systems (ToF)	ToF image capturing system uses cameras with signal modulation to measure distances and translate to images based on the phase-shift principle. 
Structured Light	A special light projector modulated by a spatial light modulator is used to project onto the desired scene. A device that is commonly used for this purpose is the Kinect sensor. An active illumination pattern is captured. The pattern varies depending on the designed 2D structured illumination with varying intensity patterns projected from the source. Then, 3D information can be computed by analysing the deformation of the projection of the pattern onto the scene with respect to the original projected pattern (Geng, 2011).

	
Infrared Thermography	Infrared thermography refers to the process of forming differential images that are visualised with colours based on the object's surface temperature. This technology can apply to the human body because the body skin's emissivity is 0.98 ± 0.01, which is free of limitations of visualising thermography images such as pigmentation, absorptivity, reflectivity, and transmissivity. This technique was applied in the research by Xue et al. in 2010, and the result for gait recognition is quite promising, which is 78%–91% (Xue et al., 2010).

One of the newer methods that provide promising results and is used frequently in recent studies is the ToF system. Advantages of Tof system such as only requiring a single camera and not requiring manual depth parameter calculations, make it stand out compared to methods. In the research done by Samson et al., the use of ToF system in higher resolution calculation of pressure for dynamic footprint analysis is demonstrated (Samson et al., 2012). Compared to all of the methods discussed, the ToF and Infrared Thermography system requires a more expensive data acquisition instrument (Derawi, Ali and Alaya Cheikh, 2011). The stereoscopic method does not require a special camera for capturing data, but high computational cost is needed for computing stereoscopic data to determine the distance and position of the subject (Liu, Cao and Wang, 2010). A structured light method is known for its higher versatility and availability of sensors compared to other methods (Gabel et al., 2012).

Image data used for gait recognition are mostly in the form of a silhouette. According to Benabdelkader, Cutler and Davis, 2004, the first step of pre-processing for raw colour image data is motion segmentation. Motion segmentation can be computed using a nonparametric background modelling/subtraction model, which is robust and good at handling issues such as lighting changes, camera jitter, and the presence of shadows. Motion segmentation is an important step as it removes the non-interest background and outlines the subject's segment of motion. Once the segmentation is detected, the motion of the subject is tracked in the subsequence frames for N consecutive frames. Then, a sequence of N successive frames silhouette template is created, corresponding to the gait motion of interest (Benabdelkader, Cutler and Davis, 2004). Since the size of the subject segment may vary as the distance of the subject from the camera system can change during the recording of the motion, the scaling and aligning process is required to compute a sequence of silhouette templates that is more robust and more accurate. Lastly, threshold filtering which converts the image to black and white is performed depending on the needs of the study (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014).

Three outcomes can be obtained after pre-processing of images depending on the needs and nature of the study: (1) original/greyscale image, (2) foreground segment image, and (3) binary image (Benabdelkader, Cutler and Davis, 2004). The three types of images are illustrated in Figure 2.7. These three types of templates have their own trade-off when computing for gait recognition. The original image is more robust to segmentation errors; the binary image is more robust for clothing and background errors; and the foreground image is the middle option of these two, robust to background errors but sensitive to segmentation and clothing variations.

Figure 2.7: Silhouette images corresponding to the original, foreground, or binary images (from left to right). Retrieved from (Benabdelkader, Cutler and Davis, 2004).

Before training the gait recognition model, feature extraction is normally taken to extract the significant information/features from the images (Pratheepan, Condell and Prasad, 2009). Since the dimensionality of the dataset of images generally is very big. For example, the resolution for one coloured image taken for analysis is 128 x 128 x 3 = 49152 pixels (two-dimensional image times three for three of the red, green and blue colour contours). This number will increase with the resolution of the image taken and the example of image in dataset. Feature extraction techniques that are commonly used in this type of application are principal component analysis (PCA) and linear discriminant analysis (LDA). After feature extraction, the classification algorithm used to train and test the model is Nearest Neighbours. The feature extraction techniques and classification algorithm were implemented by Benabdelkader, Cutler and Davis (2004) with the promising performance of their gait recognition is very promising with 94% and 100% accuracy for PCA and LDA, respectively with binary image dataset; and 94% accuracy for both PCA and LDA with foreground image dataset (Benabdelkader, Cutler and Davis, 2004). A detailed review of the feature extraction techniques and classification algorithms used in gait recognition will be elaborated in the next chapter of the literature review.

Out of so many feature points that can be extracted and identified in the silhouette images, some are significant, and some are irrelevant, such as background and clothing. Those significant can further be categorised into static and dynamic components. The static components refer to the parts of the body that stay rigid during gaits, such as head and body trunk, whereas dynamic components refers to the swings of arms and legs (Veres et al., 2004). Identifying which component contributes more to gait recognition is important so that adjustments and fine-tuning can be made to improve the performance of the model. Veres et al. (2004) reported that static component information would impact gait recognition more than dynamic component information. In their study, training silhouette images are processed into average silhouettes and differential silhouettes (Veres et al., 2004). The average silhouette is the average appearance of each recorded sequence formed by averaging the sum of all silhouettes representing a full cycle; the differential silhouette is obtained by applying a differential equation on silhouettes representing a full cycle (Veres et al., 2004). The gait recognition performance result shows that the static components consistently demonstrate a superior impact over the dynamic components in either dataset.

In addition, similar findings have also been reported by Pratheepan, Condell and Prasad (2009). Three datasets that are extracted from the same source dataset can be biased to different feature points which are Dynamic Static Silhouette Template (DSST), Dynamic Silhouette Template (DST) and Static Silhouette Template (SST) (Pratheepan, Condell and Prasad, 2009). DSST consists of dynamic and static components, DST consists of dynamic components and SST consists of static components. The gait recognition result is demonstrated in Table 2.7 (Pratheepan, Condell and Prasad, 2009). Besides that, a study introduced Cumulative Match Score (CMS) evaluation method to evaluate the performance (Pratheepan, Condell and Prasad, 2009). The rank parameter in CMS evaluation indicates the accuracy of the N number of top matches for the particular test sample. For instance, the Rank 1 result for DSST is 77.8% suggests the classification model can predict 77.8% of N=1 top matches. Therefore, the higher the rank, the less value its result delivers. Overall, DSST gives the best result among the three, and SST has a better performance than DST, which indicates that static component information in silhouette contributes more to gait recognition (Pratheepan, Condell and Prasad, 2009).

Table 2.7: Performance result for each dataset.

	Rank 1 (%)	Rank 2 (%)	Rank 3 (%)
GEI	55.6	66.7	77.8
SST	61.1	77.8	83.3
DST	61.1	72.2	88.9
DSST	77.8	83.3	88.9

2.2.2.3 Comparison between wearable and non-wearable systems

The non-wearable system (NWS) method is implemented more in professional applications because it requires laboratory or controlled conditions where gait data capturing equipment and devices such as cameras, laser depth sensor, force plate, pressure mat and instrumented treadmill are set up. The major advantage of these systems is that they can exclude the external factors to subjects' body that may restrict and cause discomfort during the assessment. Thus, under the controlled conditions, the repeatability and reproducibility level of the assessment becomes higher, and a more ideal and natural walking gait can be observed from subjects (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014). The greatest disadvantage of NWS systems is that evaluation cannot be done during subjects' daily activities, and short and repetitive study does not 100% reflect the patient's condition in real life. For example, the image processing system needs the patient to walk within a certain range of cameras and sensors; and floor-based system needs the patient to walk on a sensor mat or instrumented treadmill that has limitations in direction and distance.

On the contrary, for wearable systems (WS), the main advantage addresses the disadvantage of NWS. Through the development of miniaturised sensors and wireless communication systems such as Bluetooth or Zigbee, and the simplicity of the sensor system, WS can deliver gait evaluation for patients during daily activities outside of the laboratory (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014). Besides, by utilising accelerometer, gyroscopes, and pressure sensors, WS can provide a cheaper gait analysis approach with no limitation of location either inside or outside the laboratory. On the other hand, disadvantages of WS include amplification of the measurement error which occur when estimating speed and distance travelled using integration method; high level of complexity for computing result; and uncomfortable or intrusive for patients during gait analysis which may affect the performance (Horak and Mancini, 2013).

The summary comparison between NWS and WS is illustrated in Table 2.8. In short, every gait analysis method has its pros and cons and is practical for general analytical purposes. Thus, choosing a suitable and adequate method is important to increase accuracy and performance. We can all agree that the approach that can provide simultaneous, in-depth analysis with a higher number of parameters is the combination of NWS methods and WS methods in the laboratory environment. However, WS methods deliver their value as a cost-effective and non-intrusive alternative method for gait analysis.

Table 2.8: Summary comparison between non-wearable system and wearable system.

System	Advantages	Disadvantages
Non-wearable System (NWS)	• High repeatability, reproducibility and less external factor interference due to controlled environment. • Non-intrusive. • Analysis process is controlled in real-time by the specialist in a controlled environment.	• Subject's gait may be altered due to the restriction of walking space. • Expensive equipment, test and environment. • Hard to monitor gait performance outside of laboratory.
Wearable System (WS)	• Lower cost. • No limitations on location for gait analysis. • Transparent analysis • Can used to monitor subject's gait during daily activities.	• Complex algorithm to compute results. • Limits to a smaller amount of gait parameters. • Noise and interference from external forces, especially outside the laboratory where there is no supervision from specialists.

2.2.3 Comparison between semi-objective and objective approach

In many clinical scenarios, semi-objective gait analysis is often performed to rate the gait performance of patients as the necessity to implement objective analysis is low. The advantage of this approach is that advanced and professional equipment is not required to carry out the test. As long as there is a trained specialist, they can obtain an overview of the gait performance of the patient in a much easier and faster manner than objective analysis. In most clinical practice, based on the observation from experienced specialists, they can provide suitable treatment and prescriptions to the patient. However, due to the subjective nature of the semi-objective approach, the accuracy, exactitude, repeatability and reproducibility of the measurements are highly questionable (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014).

Semi-objective analysis suffers from several severe limitations. First, this method highly relies on the experience and skill of the observer. Thus, it is challenging to avoid assessor bias because there is no guarantee that each assessment's condition and observer are the same. Therefore, comparative and objective results are challenging to obtain. Toosizadeh et al. (2015) reported that the in-clinic evaluation of motor impairment in Parkinson's disease generates better results than in-home assessment. In-home assessment happens when the patient's caregiver is taught how to perform an assessment at home to monitor the condition and improvement of the patient (Toosizadeh et al., 2015). Because specialists do not supervise in-home evaluation, there may be errors in performing the assessment, such as not using standardised chairs for the TUG test and completing the assessment during the medication period. Next, due to the limitation of human eyes, forces and high-speed movement are impossible to measure and observe. Although, a video recording approach has been introduced, which can help resolve the limitations of eyes in some circumstances. However, the subjective nature of this method is still the same, so it does not resolve the accuracy and reliability problems of this method.

In comparison, objective analysis, which implements advanced and professional data capturing technologies and analysing tools, can better quantify different gait characteristics of the subject. Therefore, this method can produce a more accurate and informative analysis of human gait (Muro-de-la-Herran, Garcia-Zapirain and Mendez-Zorrilla, 2014). Furthermore, these advanced sensor technologies and analysing tools can better capture different gait parameters in human gait and more likely discover helpful information that cannot be provided in semi-objective analysis by simply watching a patient walk.

In short, the objective analysis should be put in higher priority than semi-objective analysis when performing gait assessment. Although semi-objective analysis has the advantage of being fast and straightforward, the serious limitations will significantly decrease the accuracy and reliability of the evaluation. In recent years, objective analysis shows a trend to replace semi-objective analysis in clinical practice as technology development has reduced the difficulty of implementing objective analysis.

2.3 Gait classification

2.3.1 Data preprocessing

The vast majority of real-world raw data sources are imperfect and cannot deploy directly to human or manual applications. Thus, data preprocessing is an essential step to minimise or to get rid of the imperfections in the data source. This process becomes more crucial when dealing with Big Data scenario in which the volume, velocity and variety of data are higher. There are four main data preprocessing tasks: instance reduction, missing value imputation, data normalisation, and noise reduction.

2.3.1.1 Instance reduction

Instance reduction techniques are popular in minimising the effect of a very large data set in data mining algorithm. The method can decrease the number of instances/samples in the data set without sacrificing the quality of its interpretation. There are two steps involved in instance reduction: instance selection and instance generation.

Instance selection is perceived as the essential step in instance reduction. The purpose of instance selection is to identify suitable examples from many instances and remove the unwanted, not valuable data, including duplicate data, irrelevant data, and outliers (García et al., 2016). Duplicate data has a high chance to happen, especially after combining data from multiple places or parties, and it is the main problem to address in instance selection. Irrelevant data refers to the data that do not fit with the problem statement of machine learning. Outliers are those observations out of the range of data trends that are likely to be bad data points (Tableau Software, 2021). The filter and wrapper approaches are the two approaches to performing instance selection. The filter approach does not consider activities when performing data reduction; the wrapper approach emphasises machine learning and uses certain machine learning algorithms to reduce data (Kotsiantis, Kanellopoulos and Pintelas, 2006).

Unlike instance selection, it is not a must for instance reduction. For instance generation methods, besides selecting data, can generate and replace the original data with new artificial data. Instance generation techniques can be used to fill regions in the problem domain that lacks representative and valuable samples (García et al., 2016).

2.3.1.2 Missing value imputation

Missing values treatment is very important in data preprocessing. Most of the data mining and machine learning algorithms do not accept missing values because results will not be optimal and valid. The presence of missing values in data sets can cause deficiency and strong biases in extracting information and data training. Therefore, missing values treatment has to be handled carefully as inappropriate approaches may cause poor knowledge extraction and wrong conclusion (García et al., 2016). The first option of treatment is to drop the instances that contain missing values. This option is simple, but it is rarely beneficial because it may produce a bias in the learning process and cause loss of information. The second option is to fill in the missing values based on observation. Once again, the approach must be handled carefully as there is a chance to the loss of integrity of data since it is operated based on assumptions. There are several methods to perform the second option (Lakshminarayan, Harp and Samad, 1999):

a) **Most Common Feature Value:** The highest occurrence value is chosen to fill in the missing value of the feature.

b) **Concept Most Common Feature Value:** The value of the feature that has the most occurrence within the same class is chosen to fill in the missing values of the feature.

c) **Mean substitution:** The mean value of the feature is determined to fill the missing values of the feature.

d) **Hot deck imputation:** Identify the most similar instance to the instance with missing values and substitute the missing value with the respective value from the most similar instance.

e) **Treating Missing Feature Values as Special Values:** The missing values are treated as "unknown" as a new value.

2.3.1.3 Data normalisation

Data normalisation is a "scaling down" transformation to restrict the range of features. There might be a great difference within a feature between the minimum and maximum values, for example, 0.01 and 1000. After normalisation, the value magnitude will be scaled to appreciable low values (Kotsiantis, Kanellopoulos and Pintelas, 2006). Data normalisation is important in normalising the scale of different features with different ranges (e.g., Age: 20-70 and Salary: 3000-10000) in the data set to compare those different features on common grounds. Two common methods to perform data normalisation are:

Min-max normalisation:

$$v' = \frac{v - \min(v)}{\max(v) - \min(v)}\left(new_{\max(v)} - new_{\min(v)}\right) + new_{\min(v)} \tag{2.1}$$

Z-score normalisation:

$$v' = \frac{v - mean_A}{stand_{dev_A}} \tag{2.2}$$

2.3.1.4 Noise reduction

As seen in the previous section, raw data sources are rarely in perfect condition, defections and corruption are often found. Noise in data refers to the parts in data that are meaningless and cannot be interpreted and understood by machines. In supervised approach, noise can affect the quality of input and output features (García et al., 2016). Noise in input features is called attribute noise, whereas noise in output attributes that increase the bias is called class noise. The first method to deal with noisy data is by using data polishing methods, especially effective when related to the labelling problem of instances. However, this method is very complex, therefore, usually apply to a small amount of noise (Zhu and Wu, 2004). Next, the second method is by using noise filter. Noise filter can identify and take out the noise components in data and it does not affect the data mining and machine learning algorithm.

2.3.2 Feature selection

Dataset with high dimensional data consists of many features that can be irrelevant, misleading, or redundant, which increase search space size. That condition will result in difficulty to process data further and thus affect the performance of machine learning. Therefore, feature selection technique is applied to remove irrelevant, misleading and redundant features to prevent overfitting and difficulty in training. Feature selection methods can be categorised into wrapper, filter, and embedded/hybrid (Kotsiantis, 2007).

Wrapper methods use cross-validation on the induction algorithm to predict and evaluate the meaningfulness of adding or removing a feature from the feature subset. Wrapper methods can perform better than the filter method because the feature selection process can choose the best features and optimise the classifier's usage. However, wrapper methods are more expensive than other methods because of the computational cost when dealing with a large dimensional dataset. Recursive Feature Elimination (SVM-RFE) is an example of the wrapper method, which performs backward elimination. The weight vector is used as a ranking criterion in SVM-REF to determine the m features leading to the greatest class separation margin (Guyon et al., 2002).

Filter methods select features based on information content: interclass distance or statistic dependence. Compared to wrapper approaches, filter methods have a lower computational cost and are faster, but they have lower classification reliability and are better suited to high-dimensional data sets. The subset search algorithm is one of the filter methods. There are three steps in the characterisation of subset search algorithm, namely (1) search organisation, (2) generation of successor and (3) evaluation measure (Ladha, 2011). First, search organisations search through the dataset by three types of search: exponential, sequential and random. Next is the generation of successor (subsets) through five operator options: Forward, Backward, Compound, Weighted, and Random. Lastly, the successors (features) will be evaluated through evaluation functions: Divergence, Dependence, Interclass Distance, Information or Consistency Evaluation.

Then lastly, the embedded/hybrid method is a feature selection method developed more recently, utilising the advantages of both filter and wrapper methods. When selecting meaningful features, the embedded approach implements both an independent test and performance evaluation function to the feature subset. (Veerabhadrappa and Rangarajan, 2010).

2.3.3 Feature extraction

The feature extraction technique generates new subsets of more significant features by performing a transformation algorithm on the original features. The feature extraction aims to reduce the complexity of feature sets by representing each variable in feature space as a linear combination of the original input variable (Khalid, Khalil and Nasreen, 2014).

The most popular feature extraction method is Principal Component Analysis (PCA), introduced by Karl Pearson in 1901. This method is a simple non-parametric linear transformation to extract the most useful information from a data set by minimising the redundancy and noise in data and maximising the information (Khalid, Khalil and Nasreen, 2014). The PCA method extracts new features projecting the input data into a lower dimensional subspace and analyses multivariate statistical instances' covariance structure (Engel, Hüttenberger and Hamann, 2012). Many variants of PCA have been proposed to improve the limitations of original PCA (Storcheus, Dmitry; Rostamizadeh, Afshin; Kumar, 2015). For example, Probabilistic PCA (PPCA) resolve the constraints of lack of probabilistic model structure in modelling by giving an isotropic structure to noise component; the Kernel PCA (KPCA) can perform non-linear dimensionality reduction using kernel function; and Probabilistic Kernel PCA (PKPCA) is the combination of PPCA and KPCA (Zhou, 2003). The computation of the PCA transformation matrix S is given as (Elavarasan and Mani, 2015):

$$S = \left(\sum_{i=1}^{n} (Y_i - m)(Y_i - m)^{\tau} \right) \tag{2.3}$$

Where,
n is the number of instances
Y_i is the i-th instance
m is the mean vector of the input data

Another technique is Linear Discriminant Analysis (LDA) proposed by Hastie and Tibshirani (Hastie and Tibshirani, 1996). LDA apply the notion of distance metric learning through parametrising the class of metric functions. This technique projects the high-dimensional data into lower-dimensional feature space. As a result, the between-class distance is maximised, and the within-class distance is minimised in the low dimensionality space (Elavarasan and Mani, 2015). The computation for LDA is given as:

$$f(X) = trace\big((S^T S_w X)^{-1} (S^T S_b X)\big) \tag{2.4}$$

Where,
S_b is the between-class matrix
S_w is the within-class matrix
X_i is the index set of i-th class
C_i is the mean vector of i-th class.

2.3.4 Classification algorithms

There are generally two types of approaches in machine learning: supervised learning and unsupervised learning. The main distinction between the two approaches is that supervised learning uses labelled datasets while unsupervised does not. This project aims to classify young healthy, old healthy, and Parkinson's disease subjects. Thus, classification algorithms under supervised learning are studied in this literature review.

2.3.4.1 Artificial neural network

An artificial neural network (ANN) is a multi-layer map consisting of many units (neurons) linked together in a patterned connection, as illustrated in Figure 2.8. A neural network can be segregated into three part: input layer, which input the data to be processed/trained; output layer, which output the result of processing/training; and hidden layers, layers in between the input and output layer. When designing a neural network, the size of hidden layers and hidden units is the most crucial part to consider. An insufficient number of neurons can cause poor performance in generalisation and approximation. In contrast, an excessive number of nodes can lead to overfitting and increase the difficulty of searching for a global optimum (Kotsiantis, 2007).

Three aspects built up an ANN: input and activation functions of the unit, network architecture, and each input connection's weight (F.Y et al., 2017). The first two aspects are predefined before the initiation of ANN and fixed throughout the process; thus, the behaviour and performance of ANN depend on the current weights values. When starting the training of ANN, weights of the net are set to random values. Next, input data in deployment through the ANN and output results are acquired. The backpropagation algorithm is used to estimate the difference of output value of ANN with the desired output of this instance. Then, the weights in the net are adjusted in the direction that would generate output closer to the desired result according to the value determined using the Backpropagation algorithm in the next repetition. The process is repeated until the output value of ANN is converged, and this process is called Gradient Descent.

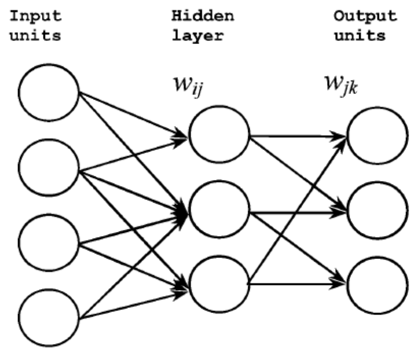


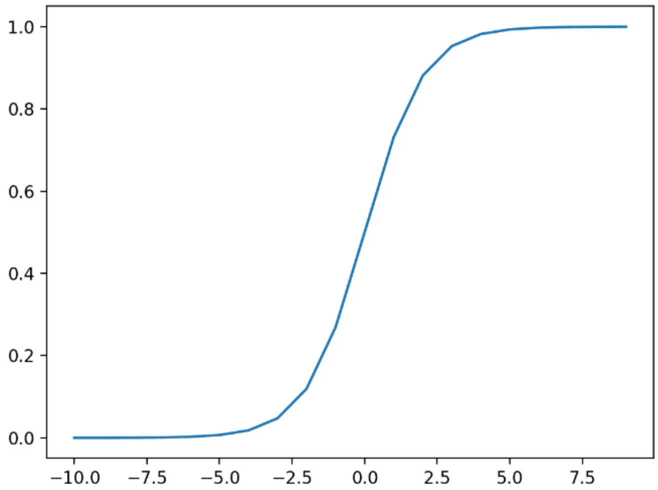
Figure 2.8: Artificial Neural Network structure (Kotsiantis, 2007).

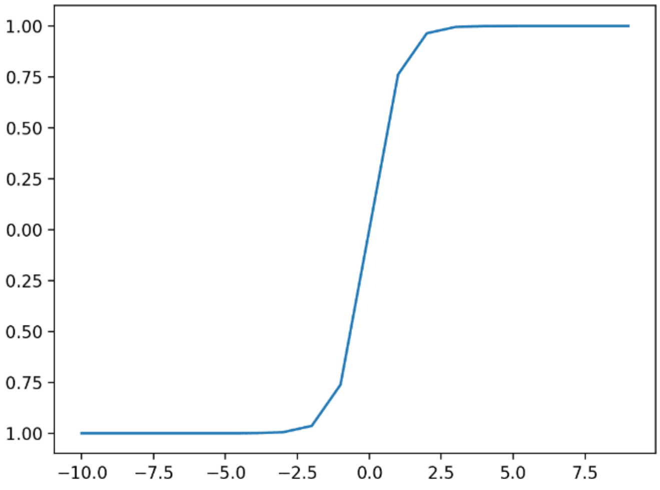
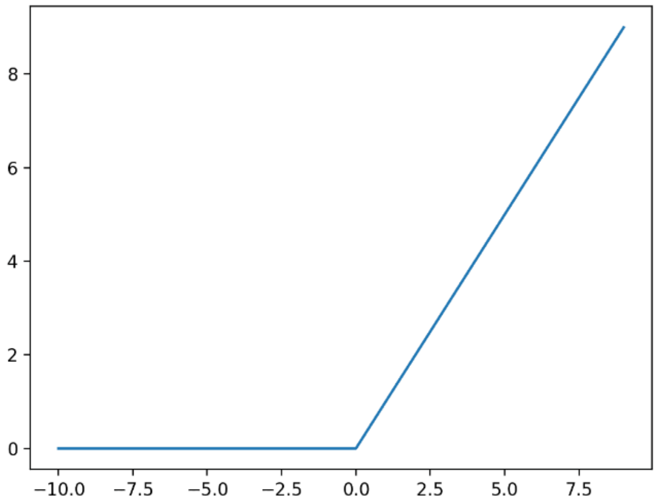
The development of ANN is done by implementing the *Keras* library that can be imported from *TensorFlow* library. *Keras* is an open-source library that allowed users to develop ANN in Python environment. The Keras library provided an easy developing process for ANN. Moreover, it allowed high flexibility configurations of different parameters in ANN, such as number of hidden layers, number of neurons in layers, activation function, loss function, etc.

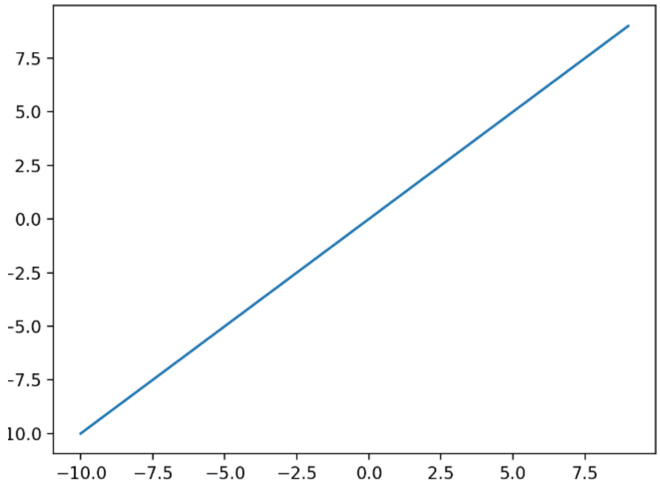
2.3.4.1.1 Sequential model

A sequential model is a plain stack of layers in which every layer will have one input tensor and one output tensor, and the data will go through every layer and eventually output a prediction. The sequential model allows developers to configure the number of hidden layers, number of neurons and the type of activation function used in every layer. Generally, a neural network consists of three layers: input layers that take raw untrained data from the domain, hidden layers that take input from the previous layer and perform training using activation function and then output to the next layer, and an output layer that output a prediction. Activation functions are commonly used in hidden layers and output layers. Activation functions are used to process the data from previous layer to help the neural network learn complex patterns of data and outputs the processed data to the next layer. Table 2.9 shows the summary of activation functions used for hidden layer and output layer.

Table 2.9: Summary of activation functions used for hidden layer and output layer. (Reynolds, Woods and Baker, 2006)

	Activation functions	Explanation
Hidden layer	Logistic function (Sigmoid)	• The default activation function in the 1990s. • Sigmoid function turns input data into output values ranging from 0 to 1. • Susceptible to vanishing gradient problem. • Output closer to 1 represents greater input, and output closer to 0 represents smaller input. • The figure below shows the example plot of input against output for Sigmoid output.  Plot of Inputs vs. Outputs for the Sigmoid Activation Function.
	Hyperbolic tangent function (Tanh)	• The default activation function in the 1990s to 2010s. • Tanh function turns input data into output values in the range -1 to 1. • Susceptible to vanishing gradient problem. • Similar to the Sigmoid function, an output value closer to 1 represents larger input and an output value closer to -1 represents smaller input. • The figure below shows the example of input against output for Tanh output.

		 Plot of Inputs vs. Outputs for the Tanh Activation Function.
	Rectified linear activation function (ReLU)	• The most common activation function nowadays. • The function is simple to implement, memory efficient and less susceptible to vanishing gradient problems. • Outputs value 0 if the input value is negative and returns the input value if the input value is positive. • The figure below shows the example plot of input against output for ReLU output.  Plot of Inputs vs. Outputs for the ReLU Activation Function.
Output layer	Linear function	• The method is also called “no activation”. • The function returns the input value directly.

		• Commonly used for regression problems. • The figure below shows the example plot of input against output for linear function. 
	Logistic regression (Sigmoid)	• The characteristics are same as in the previous section. • Commonly used for binary classification and multilabel classification.
	Softmax function	• The function computes a vector of values that represents the probabilities of each class as output. • Commonly used for multiclass classification.

2.3.4.1.2 Loss function

The loss function calculates the difference between model-predicted output and the true test data of every example. The loss function can be configured using compile method in Keras, and it is one of the components for the back-propagation. Keras library provides various loss functions, and choosing the suitable loss function for different kinds of problems is essential. Generally, there are three types of problems: regression, binary, and multi-class classification (Brownlee, 2019).

A regression problem refers to problems that requires model to predict the real-value quantity as output. Next, a binary classification problem is where examples are assigned with one of two labels and need model to predict the label of the examples. A multiclass classification problem is different from a binary classification problem because the examples are assigned with one or more than one label of multiple labels. The suggested loss functions suitable for various problems are summarized in Table 2.10 below.

Table 2.10: Summary of loss functions for various problems. (Brownlee, 2019)

Type of problem	Loss function	Description
Regression	Mean Squared Error Loss	This function calculates the average value of the squared differences between predicted and actual values. This is the default function for regression problems.
	Mean Squared Logarithmic Error Loss	The function is similar to Mean Squared Error Loss, except it computes the logarithm of each predicted and actual value then computes for the mean squared error. This function is commonly used when dealing with widely spread data and making large value predictions. The reason is that for these kinds of problems, the computational load using Mean Squared Logarithmic Error Loss is lesser.
	Mean Absolute Error Loss	This function computes the average value of the absolute difference between predicted and actual values. The loss function is helpful when dealing with data with Gaussian distribution and contains a certain

		amount of outliers.
Binary classification	Binary cross-entropy loss	This function is the default function for binary classification with a binary target value (0 or 1). The function computes the cross-entropy loss between actual and predicted values.
	Hinge loss	Hinge loss function is commonly used for binary classification problems with a target value of -1 or 1. The hinge loss function also calculates the average difference between actual and predicted values, but it will assign more errors when the sign of actual and predicted values are different.
	Squared hinge loss	The squared hinge loss is one of the extensions of the hinge loss function. Similar to hinge loss, this function works for binary classification with a target value of -1 or 1, and it calculates the squared hinge loss between actual and predicted values.
Multi-class classification	Multi-Class Cross-Entropy Loss	The default function for multi-class classification problems is that the target variable has multiple classes. The function computes the average difference of probability distribution of all classes between actual and predicted output, and the target value needs to be in one-hot encoding representation.
	Sparse Multiclass Cross-Entropy Loss	The function is identical to the multi-class cross-entropy loss function, except that this function does not

		require the target variable to be in one-hot encoding. This property is useful when the number of labels is large, and the target variable in one-hot encoding may cause difficulty in training.
	Kullback Leibler Divergence (KL-divergence) Loss	The KL-divergence loss calculates the probability distribution of predicted output with baseline distribution. Although the behaviour of KL-divergence is similar to the cross-entropy function, KL-divergence loss is more suitable for models that try to predict more complex classification problems.

2.3.4.1.3 Optimiser

The optimiser is the second argument in the compile method in Keras and it is also one of the components in the back-propagation in ANN sequential model. The purpose of optimiser is to tune and update the hyperparameters (weights and biases) of neural network in the next iteration to minimize the cost function. There is a slight difference between a lost function and a cost function. A lost function calculates the difference in distance between the predicted and actual output of every example, whereas a cost function is basically the average of the lost functions (Brownlee, 2016). The commonly used optimisers are summarized in Table 2.11.

Table 2.11: Various popular optimisers in Keras library. (Ketkar, 2017)

Type of optimiser	Description
Stochastic gradient descent (SDG)	• A gradient descent optimizer with momentum. This type of optimizer requires learning rate tuning to produce good results and is not

	suitable for sparse data. • The characteristic of SDG is that the updates of hyperparameters are frequent, and the variance is high. • This characteristic allows the prediction function to have higher chance to jump to better local minima, but this can also contribute to convergence in specific local minima.
Adagrad (Duchi, Hazan and Singer, 2011)	• An adaptive optimizer. This type of optimizer does not require learning rate tuning and is good with sparse data. • Adagrad optimizer updates the hyperparameters depending on the occurrence of the features. It will perform small updates for updated parameters and big updates for parameters with less frequent updates. • Because Adagrad optimizer adjusts the learning rate of hyperparameters depending on the past gradients, vanishing gradient problem may happen as the learning rate becomes very small after a high number of iteration in training.
Adadelta (Zeiler, 2012)	• Adadelta optimizer is the improvement of Adagrad as it combine the advantage of Adagrad and SDG optimizers. • Adadelta can overcome the vanishing gradient problem present in Adagrad by considering a fixed number of past gradients during model training. It does not require manual selection of learning rate.
RMSprop (Hinton, Srivastava and Swersky, 2012)	• An unpublished adaptive optimizer proposed by Geoff Hinton in 2012. • RMSprop is very similar to Adadelta except for the way they handle the past gradients.

	RMSprop computes the learning rate by dividing to the average of exponentially decaying square of past gradients.
Adam (Kingma and Ba, 2014)	• An optimizer that adds up the advantages of Adadelta and RMSprop. • High computational efficiency, little memory demand, works well with large data and invariant to diagonal rescaling gradients

2.3.4.2 Support vector machine (SVM)

Support vector machine (SVM) is a statistical learning theory-based machine learning method. In the case of linearly separable data, the basic idea is illustrated in Figure 2.9. Once the optimum hyperplane that separates the two classes is found, the instances that lie on the hyperplane margin are known as support vectors. Support vectors are the data points that lie closest to the decision hyperplane. SVM implies that only support vectors are important for classification whereas other instances are ignorable. In addition, the maximal margin classifier is the basic SVM method that helps determine the classification problem in linear separable training data with binary classification (Wu et al., 2008).

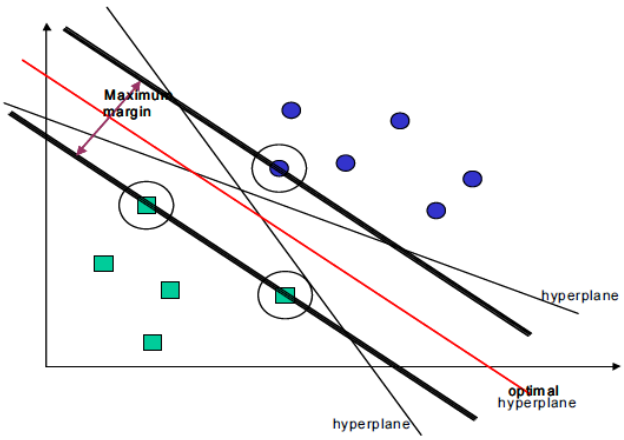


Figure 2.9: Optimal separating surface (Kotsiantis, 2007).

However, most of the real-world cases will consist of non-linear separable data, and the classification solution will not be as simple as mentioned previously. One solution for problems with data that cannot be linearly separated is to map the data points onto a higher dimensional space (feature space) and determine the classification hyperplane there. However, the computational cost for mapping the data points to higher-dimensional space is expensive, especially when dealing with high-dimensionality problems. Therefore, a "kernel trick" or "kernel function" is introduced to reduce the computational cost. A "kernel function" is a special class of function that can calculate inner products directly in feature space without performing the process of mapping data points to higher-dimensional space (Scholkopf, Burges and Smola, 1998).

The main advantage of SVM is that it can work for many kinds of classification problems, including problems with high dimensional and non-linear separable data sets. In contrast, the major limitation of SVM is the tuning for the key parameters is important to attain good classification results (Soofi and Awan, 2017).

2.3.4.3 Decision tree

Decision tree provides easy to control technique for classification and simplified modelling process. The data will be split into features that can best divide the training data through numerous techniques like Gini Index, Chi-square, information gain, and reduction variance. Murthy, 1998 reported that there is no best method; comparison between individual methods is important to choose the most suitable method for a particular data set (Murthy, 1998). The feature that can best split the data will be the root of decision tree. The decision tree branches downward and the step is repeated for each partition of splitting node, creating branches and sub-tree until the training data is divided into the same class category (Kotsiantis, 2007).

The decision tree approach is very likely for the model to overfit training data. Overfit happens when the desired hypothesis (h) has a smaller error than other hypotheses (h') in training set, but has a larger error than h' when tested in the entire data set. Generally, two approaches will be considered to deal with overfitting problem: (1) Stop the training model before the training data gets fit perfectly in the training dataset, (2) Pruning of decision tree. Most algorithms adapt the pruning method. Pruning is a data compression technique in machine learning that reduces the size of decision trees by removing the not meaningful and redundant parts (leaves) of the tree to help prevent overfitting of training data (Elomaa, 1999).

2.3.4.4 Bayesian network

The structure of the Bayesian network is a directed acyclic graph (DAG) with only one parent (unobserved node) and possibly several children (observed nodes). The nodes in S have strong independence among the child nodes to have a one-to-one correspondence with their parent node (Kotsiantis, 2007). For example, node X is considered the parent node of node Y when an arrow connects the pair of nodes from node X to node Y. Moreover, a node is conditionally independent from another non-descendent node given its parents (Soofi and Awan, 2017). Thus, two subtasks build a complete Bayesian network: learning the network structure and parameter determination.

One of the exciting characteristics of the Bayesian network over other algorithms such as decision trees or neural network is the possibility to consider the structural relationships among its features of given problems. Other advantages of Bayesian network include small influence on the working system when minor changes are made in the network; flexible adaptation of the same Bayesian model for regression and classification model problems; and suitable for handling missing data (Uusitalo, 2007).

A drawback of Bayesian network classification is that it is unsuitable for high dimensionality (many features) datasets as an extensive network is not time and space-effective in Bayesian network method. Besides, the training attributes have to be class features; therefore, continuous attributes need to be discretised. During the discretisation of continuous data, loss of information and consciousness may occur (Wang, Gao and Wang, 2016).

2.3.4.5 K-nearest neighbours

K-nearest neighbour (KNN) algorithm is one of the most popular algorithms for instance-based learning. The characteristics of instance-based learning algorithms are less computation time during the training phase but require more computation time for the classification process. In the classification process, an instance is placed into an n-dimensional instance space where each n-dimensions correspond to the n-features. The class for the example is determined based on the k number of nearest neighbours in the instance space. Users can control the k number during the training phase, in which the suitable k number that produces the best classification will vary depending on different datasets. The relative distance of the instance in the space is more critical than the absolute distance when deciding on the nearest neighbour. Examples of distance metrics to calculate relative distance are Minkowski Distance, Manhattan Distance, Euclidean Distance, Cosine Distance, and Hamming Distance (Witten and Frank, 2002).

Advantages of the KNN technique include high effectiveness for large training data and can work robustly on noisy data (Teknomo, 2017). In contrast, the limitations of KNN are as such: (1) requirement for large storage space, (2) the output is sensitive to the similarity function used to compare instances, (3) lack of standardised procedure to choose k value, and (4) long classification time (Viswanath and Sarma, 2011; Kotsiantis, 2007). One example of a straightforward technique that the researcher practices is filtering out the useful input features as training data. This method can enhance classification accuracy and reduce the classification time (Lopez de Mantaras and Armengol, 1998).

2.4 Summary of findings

To conclude this chapter, literature reviews on several important topics useful in developing the classification algorithm for this project are discussed. Firstly, the different gait-affecting factors are reviewed. The purpose of reviewing the various factors is to understand how gait-affecting factors can influence the gait pattern in humans, particularly Parkinson's disease patients. Understanding their relationship with gait patterns can help in the design and tuning process of the classification algorithm and aid in explaining the result of my classification algorithm. Next, reviewing different gait analysis methods is important for understanding the overall general concept of gait analysis in practice. Thus, having a better comprehension of how the data acquisition process works for different gait analysis methods and direction in choosing the suitable classification algorithm based on the method used.

Furthermore, data preprocessing techniques and classification algorithms are reviewed. Feature selection and feature extraction are highlighted in this subtopic. Feature extraction is essential when processing the input data because this technique can extract useful new features from the original features. In comparison, feature selection is not compulsory as it will remove the less valuable features from the dataset, which can be helpful in large datasets that have many irrelevant, misleading and redundant and insignificant features. However, applying feature selection to a small dataset would be a bad decision as reducing features may cause loss of information in the dataset. The training algorithm cannot perform well when the information and knowledge in the training dataset are less.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This project aimed to develop a classification algorithm that can effectively recognise young healthy, old healthy and Parkinson's disease subjects using relatively simplified input data. The project is quantitative research because the performance was evaluated according to the accuracy of classification of the developed model, and the input data for training and testing are quantitative data. In addition, due to the COVID-19 pandemic, practical data collection is not feasible, so a dataset by other researchers was used to develop the classification algorithm. The general methodological approach was summarised in Figure 3.1. Details of the methodology will be explained in-depth in the subsequent subsections.

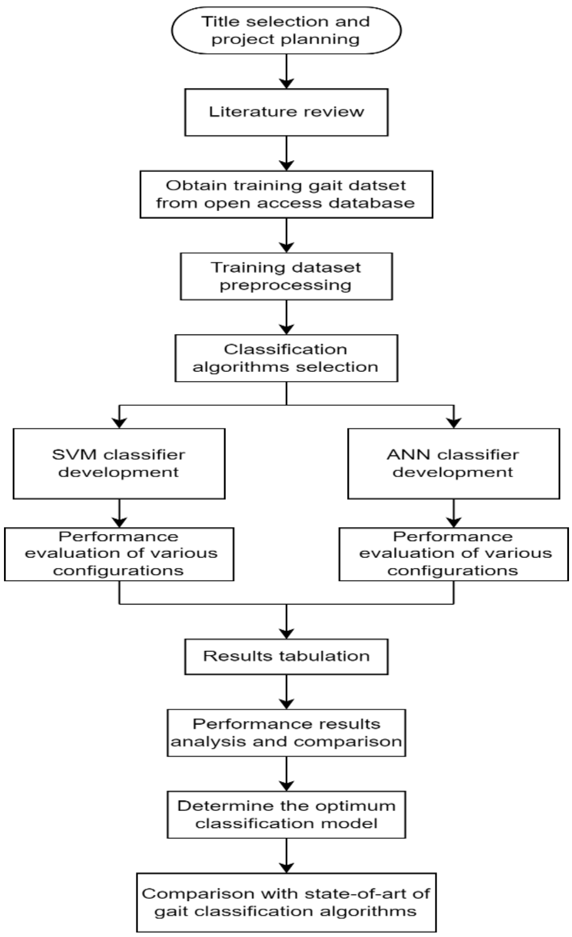


Figure 3.1: Overview of methodology.

3.2 Requirement/ Specification

3.2.1 Software

Equipment involved in this project only consisted of software equipment which included:

I. Google Colab (Google Corporations, 2021)
II. Microsoft 365 Excel (Microsoft 365, 2021)

3.2.2 Training dataset

Due to the COVID-19 pandemic and the nature of our project, practical gait data collection was not feasible and unnecessary. Therefore, in FYP Part I, one dataset from other researchers was chosen as the input dataset for this project. The dataset (denoted as D1) chosen was obtained from PhysioNet website. The name of the dataset was Gait in Aging and Disease Database (Massachusetts Institute of Technology, 1999).

The dataset consists of data from 15 subjects which includes five healthy young adults (between 23 to 29 years old), five healthy old adults (between 71 to 77 years old), and five Parkinson's disease old adults (between 60 to 77 years old). For each subject, there are two columns of feature data. The first column is the foot strike time (seconds), and the second column is the stride interval (seconds). The total size of the raw dataset is 9129 rows x 2 columns which refer to 9129 instances and two features in the dataset.

The reason for selecting this particular dataset was because of the simplicity of the data, as the data only contains two features: time of foot strike and stride interval. The data in the dataset is collected using wearable sensors which is the cheapest and most straightforward gait analysis method to perform compared to the other two gait analysis methods: image processing system and floor-based system. Besides, unlike the other two methods, the wearable sensors in this experiment only measured the time of foot strike and stride interval, which do not require inertial sensors and other advanced equipment. This characteristic granted the simplicity of the data and data collection process. In addition, datasets acquired using image processing and floor-based systems were not considered because of the requirement of advanced and professional equipment and the higher complexity of data to process. These two factors did not comply with my aim to develop an effective classification algorithm using simple gait data that can be easily acquired in the hospital environment.

In FYP Part II, a new dataset denoted D2 was introduced to this project. The reason for adding another dataset was to increase the volume of the dataset used for training and testing of classification algorithms because the D1 only contained 9129 instances. The D2 dataset was a processed dataset that extracted useful data from two open-source databases of research by Goldberger et al. (2000) and Hausdorff et al. (2000). First, the dataset from Goldberger et al. (2000) study measured the stride fluctuation of ten healthy young men aged 18-29 years (Goldberger et al., 2000). Next, the dataset from Hausdorff et al. (2000) study contained gait data of subjects with neuro-degenerative disease, including Parkinson's disease, Hungtinton's disease, and amyotrophic lateral sclerosis (Goldberger et al., 2000). Because D2 needs to be used for training and testing the classification model, hence D2 need to have similar features and label classes with D1. Therefore, these two datasets were chosen. The stride time data for healthy young and Parkinson's disease class were extracted, which form 7376 instances of healthy young label and 31813 instances of Parkinson's disease label, in a total of 48318 additional instances.

3.2.3 Dataset preprocessing

Generally, raw data collected contains imperfection factors, so direct implementations are not recommended. Data preprocessing minimised the imperfections in the raw data so that the training algorithm can better recognise the useful information and knowledge. Three preprocessing procedures were done in this project: data cleaning, data normalisation, and derivation of new features.

For data cleaning, tasks like missing values imputation and outlier removal were performed. After scanning through the dataset of each of the subjects using Microsoft Excel 365, there were no missing values found. Thus, missing values treatment was not required. Next, outlier removal was performed. The outliers in the dataset were those instances that contained zero value in features. The outliers were generally the first instances in each dataset because the time of foot strike feature and stride speed feature was zero value. The outlier instances can be directly removed from the dataset because the number was small (one outlier in each subject's dataset) and the effect of outlier removal was negligible.

Next, data normalisation was performed to normalise the scale of different features with different ranges in the dataset to become comparable and relevant. Data normalisation was applied using a python code called *StandardScaler* imported from *sklearn* library. The normalisation calculation of the *StandardScaler* function was z-score normalisation shown in Equation 2.2 in Chapter 2. Another importance of data normalisation was because the classification algorithm implemented for this project was SVM algorithm, and the algorithm was not scale-invariant (Scikit-learn, 2021).

Thirdly, the derivation of new features was done. The raw dataset of D1 only contained two features, time of foot strike and stride interval, and two features alone were not capable of effective algorithm training. Thus, new features were derived from the original ones: stride count and stride speed. The stride count feature was an incremental number count started from the count of one. Stride speed was computed by dividing the stride count by the accumulated time of foot strike with the unit of stride/seconds. As for the D2 dataset, the raw dataset only consisted of one column of stride time features and one column of class label. Therefore, the feature augmentation was performed on D2 as well. In the end, both of the datasets will consist of four columns of features: stride time, accumulate time, stride count and stride speed; and one column of class label.

In short, preprocessing datasets was essential in developing a classification algorithm that could not be skipped. In addition, data preprocessing was useful for improving the dataset's quality so that the algorithm's training process could be more effective and efficient. However, in this project, dimensionality reduction was not implemented. This was because the dimension and the size of this dataset were small. Therefore, dimensionality reduction was not needed to prevent the loss of information.

3.3 Classification algorithms

This sub-section will discuss the methodology of designing the classification algorithms. Two classification algorithms, Support Vector Machine (SVM) and Artificial Neural Network (ANN) were chosen to be developed in the project.

3.3.1 Support Vector Machine

SVMs were chosen because of key characteristics such as memory efficiency, high versatility as various parameters can be tuned, and effectiveness for low and high dimensional spaces (Scikit-learn, 2021). There were three classes in SVMs, namely SVC, NuSVC and LinearSVC. NuSVC was not suitable because it only works with data in the range of -1 to 1; LinearSVC lacks versatility because it only uses linear kernel as default. Therefore, SVC was chosen as the method to build the classification algorithm.

For the SVC method, kernel selection was important because it can directly affect the classification process's performance. Kernels referred to the decision boundary for classification, as shown in Figure 3.2. There are three types of kernel: linear, RBF and polynomial. The selection of kernel was highly dependent on the input dataset. The input dataset used consisted of three subjects (old healthy, young healthy, and Parkinson's disease subjects. The instances for old healthy and young healthy subjects were more than Parkinson's disease subjects. Therefore, the RBF kernel was the most suitable option because it was suitable for multi-class classification and datasets with imbalanced classes. In contrast, the linear kernel had high chances to perform poorly for multi-class classification problems and imbalanced datasets, and alternately, the polynomial kernel was suitable for high dimensionality datasets.

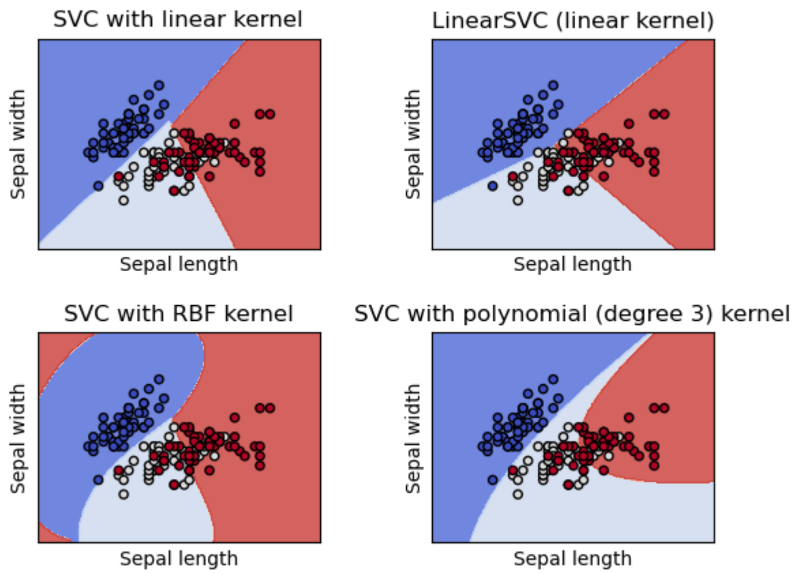


Figure 3.2: Illustration of classification results using different kernels (Scikit-learn, 2021).

There were two parameters that can be adjusted to get the best result using RBF kernel: c and gamma. The parameter C is the regularisation parameter in SVM classification algorithm. A lower C makes the decision function smoother and simpler, resulting in a larger decision margin. Therefore, higher misclassification of training examples, while a higher C aims to classify all training examples correctly by making the decision margin smaller. On the other hand, the gamma parameter defines the degree of curvature of the decision boundary, and the larger gamma, the larger the curvature.

3.3.2 Artificial Neural Network

3.3.2.1 ANN baseline model

For the application of this project, a three layers neural network is shown in Figure 3.3 was implemented, consisting of one input layer, two hidden layers, and one output layer. According to the notational convention, the input layer is known as layer zero; hence the notation of a neural network does not include the input layer as an official layer.

```python
def baseline_model(X_train, X_test, Y_train, Y_test):
    # create ANN baseline model
    model = Sequential()
    model.add(Dense(10, input_dim=4, activation='relu'))  # first hidden layer
    model.add(Dense(10, activation='relu'))               # second hidden layer
    model.add(Dense(3, activation='softmax'))             # output layer
```

Figure 3.3: Screenshot of baseline model configuration.

The two main factors that need to be considered when designing the ANN sequential model are the number of hidden layers and the number of neurons/nodes in the layer. As mentioned above, the size of the ANN model implemented in the project is a three layers ANN model, which consists of two hidden layers. Theoretically, Keras library allowed developers to add as many hidden layers as possible and form a “deep” neural network. However, there is no theoretical reason that the performance of the model is directly proportional to the number of hidden layers and neurons (Heaton, 2008). In fact, most practical ANN solutions do not use more than two layers of hidden layer in their model. Therefore, the guidelines shown in Table 3.1 were considered before implementing two hidden layers in the ANN for this project.

The next factor is the number of neurons in each hidden layer. The number of neurons was the manipulating variable/parameter in this study, unlike other parameters in the Keras sequential model. The number of neurons has a big influence on the architecture and performance of the neural network model. Ultimately, there is no standard number of neurons for every problem, so it always comes down to trial and error. However, several rule-of-thumb methods were considered when deciding the starting point for this study (Heaton, 2008):

I. Choose the number of neurons in between the size of the input and output layer.

II. Take 2/3 of the input layer's size plus with the output layer's size.

III. The number of neurons should not exceed twice the output layer’s size.

Table 3.1: Relationship between the number of hidden layers with the desired results. (Heaton, 2008)

Number of hidden layers	Results
0	Can make predictions on linearly separable data or function.
1	Can make predictions on datasets that contain continuous mapping across multiple finite spaces.
2	Can make predictions based on random decision boundary using rational activation functions.

Furthermore, the activation function for every layer was something to choose wisely to implement a robust and effective model. For hidden layer, there are generally three most popular activation functions to choose from, which are logistic regression (Sigmoid), Hyperbolic Tangent (Tanh) and Rectified Linear Activation (ReLU). Sigmoid and Tanh activation functions are default activation functions in the 1900s to 2010s. However, in modern ANN classification solutions, the default activation function for the hidden layer is the ReLU function as Sigmoid and Tanh functions are considered "outdated" for ANN classification problems (Goodfellow, Bengio and Courville, 2016). This is because the ReLU function can overcome the main limitation in the Sigmoid and Tanh function, which is the vanishing gradient problem that will cause the model not to be appropriately trained. Therefore, the activation function used for the hidden layers of the baseline model was ReLU function.

There are three popular options for the output layer: regression/linear, Sigmoid, and Softmax. Because this project is a classification problem, the regression activation function that predicts a numerical value was removed from consideration. After that, for classification problems, there were generally three cases: (1) binary classification, which uses a Sigmoid function, (2) multi-class classification that uses a Softmax function and (3) multilabel classification that uses a Sigmoid function. Because the target data for our dataset was not a binary type data, and every instance is assigned only one class. Therefore, this project was a multi-class classification problem, and the activation function used for the output layer was the Softmax function.

3.3.2.2 ANN model training configurations

After designing the sequential model of the ANN classification model, the model was configured for training using Keras's model training application program interface (API): *model.compile()* and *model.fit()*. Three parameters in the compile method were configured: loss function, optimiser, and metrics. Firstly, the loss function can be generally divided into three categories: regression, binary classification, and multi-class classification. In addition, there were several suggested loss functions for different prediction purposes as shown in Table 3.2 (Brownlee, 2019).

```
# Model training configurations
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])  # compile the model
model.fit(X_train, Y_train, epochs=200, batch_size=16, verbose=0)                     # training the model
```

Figure 3.4: Screenshot of the compilation of model.

In this project, the multi-class classification was implemented; hence the three loss functions illustrated in Table 3.2 were considered. The multi-class cross-entropy loss function and the sparse multi-class cross-entropy loss function are identical except that they are suited for different label data types. The multi-class cross-entropy suits the label in one-hot encoding representation, whereas the sparse multi-class cross-entropy suits label in integer representation. As for the Kullback Leibler divergence (KL divergence) loss function is more suitable for a higher complexity problem. Therefore, the multi-class cross-entropy function was used for the classification model because the label data of our dataset was preprocessed into one-hot encoding format. The pretesting results in Figure 4.9 showed that using KL divergence function does not necessarily produce better results.

Table 3.2: Keras loss function. (Brownlee, 2019)

Loss function type	Suitable loss functions
Regression	• Mean squared error function • Mean squared logarithmic error function • Mean absolute error function
Binary classification	• Binary cross-entropy function • Hinge loss function • Squared hinge loss function
Multi-class classification	• Multi-class cross-entropy loss function • Sparse multi-class cross-entropy loss function • Kullback Leibler divergence (KL divergence)

Next, the choice of optimiser was the Adam optimiser. The main reason was that Adam optimiser is an adaptive optimiser and adaptive optimiser has advantages over the gradient descent optimiser in terms of good when dealing with sparse data and no need for learning rate fine-tuning. In addition, Adam optimiser has the characteristics that include high computational efficiency, little memory demand, works well with large data and is invariant to diagonal rescaling gradients. These characteristics make Adam optimiser the best in general compared to the other three adaptive optimisers. Lastly, the fit method was used to configure the training of the classification model. The fit method required training input/feature data and training target/label data as input. The training parameters: epochs and batch size were set to 200 and 16, respectively. This means that the model will take in 16 training examples for each iteration to run through the ANN sequential model (forward-propagation) and then perform loss calculation and update the hyperparameters (back-propagation). The method was repeated for 200 iterations, and the final prediction was made.

3.4 Performance evaluation of classification algorithms

3.4.1 K-fold cross-validation

The k-fold cross-validation can be considered the improved method of the standard train-test split method. The k-fold cross-validation was a very effective way to prevent the classification model from being overfitted to training data because it can evaluate the performance of the classification algorithm when making predictions on data that were not used during the training of the model. Another reason k-fold cross-validation is essential in this project is that the SVM classification model could not directly provide probability estimates of the classification performance. Figure 3.5 demonstrates the overview of k-fold cross-validation.

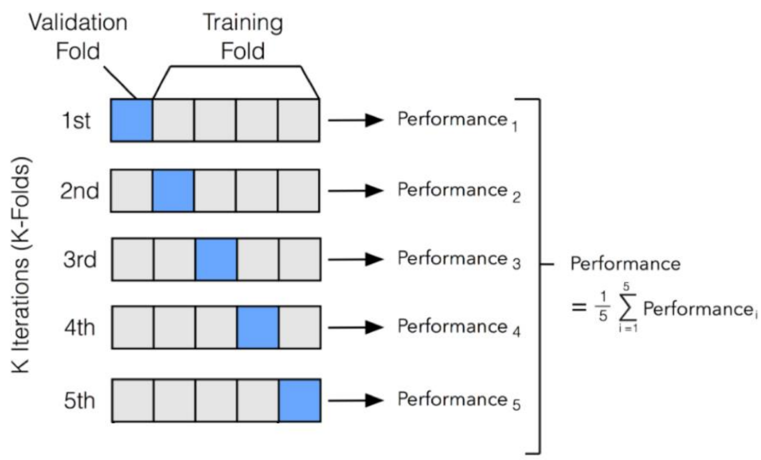


Figure 3.5: Illustration of k-fold cross-validation (Goyal, 2021).

The general concept of k-fold cross-validation is explained in this paragraph. Firstly, the dataset was shuffled randomly and split into k groups. The first group was taken as the test dataset for the first iteration, and the remaining groups were the training dataset. Next, the model was trained using the training dataset and evaluated using the test dataset. Then, the evaluation score was recorded. The steps were repeated for k iterations, and for each consecutive iteration, the next group was taken as a test dataset. After finishing all the k iterations, the average score was determined for all the k iterations.

3.4.2 Training and testing datasets of the classification algorithm

There were two X and Y datasets for both training and testing datasets. X dataset contained all the features for each instance, and Y dataset was the class of the instances. For example, in our input datasets, X should contain four features: foot strike time, stride interval, stride count and stride speed, whereas Y contained one class column with numbers 1, 2, and 3 corresponded to class old healthy, young healthy and Parkinson's disease.

Therefore, the methodology used in this project is summarized in this paragraph. Firstly, the k-fold splitting process was performed separately on three datasets: old healthy, young healthy, and Parkinson's disease. In this project, k =10 was used, corresponding to 10 iterations so that each array will contain ten subsets from index 0 to 9.

For example, in the class old healthy datasets, four arrays were created: X_train_df1, X_test_df1, Y_train_df1 and Y_test_df1, to store k-fold datasets. Then, the old subject X dataset was split into ten subsets. Subset 0 was the testing set stored as X_train_df1[0] for the first round, and subsets 1-9 were the training sets stored as X_test_df1[0]. For the second round, subset 1 was the testing dataset stored as X_train_df1[1], and subset 0, 2-9 were the training sets stored as X_test_df1[1]. The process was repeated for the rest of the iterations. The method was similar for X_test_df1, Y_train_df1 and Y_test_df1, and the steps were performed repeatedly for young and Parkinson's disease datasets. Then, four arrays of X_train, X_test, Y_train and Y_test were created to store the real dataset for training and testing the model. Take X_train as an example, a for loop was used to append the data from X_train_df1, X_train_df2 and X_train_df3 to X_train array according to the respective index. For example, data of X_train_df1[0], X_train_df2[0] and X_train_df3[0] were appended and stored as X_train[0]. Similarly, the process was repeated for X_test, Y_train and Y_test. As a result, X_train, X_test, Y_train and Y_test each contained ten subsets corresponding to 10-fold cross-validation. Therefore, for each iteration, 9/10 of instances for each class in the training dataset and 1/10 instances in the testing dataset. The evenly weighted exampled from each class in the training and testing dataset allowed better and more robust model training.

3.4.3 Performance metrics for the classification algorithms

For SVM classification model, the manipulating hyperparameters were the c and gamma values. On the other hand, the manipulating hyperparameters for the ANN classification model were the number of neurons in hidden layers. The classification models were trained and tested with different datasets using different configurations manipulating hyperparameters. After that, the performance of the classification models was evaluated using performance metrics.

The performance metrics used in this study were accuracy and F1 score. The accuracy metric is the most intuitive and typical metric used to determine the accuracy of the classifier's prediction. As for the F1 score, although it is not as intuitive as accuracy, the F1 score was usually more informative than accuracy. This is because it took both false positives and false negatives into account. F1 score is beneficial for datasets with uneven classes distribution where the cost for false positive and false negative has a greater difference. These metrics were imported from *sklearn* library, and the details of the measurements were summarized in Table 3.3. The 10-folds cross-validation was applied when evaluating the performance of the classification models. The accuracy and F1 score for each iteration were calculated, and the average values were computed.

Table 3.3: Summary of performance metrics.

Measurements	Description
Accuracy	Accuracy is the percentage of the correct predicted cases out of the total cases.
F1 score	F1 score is the weighted average of precision and recall. Precision is the ratio of correctly predicted cases to total predicted cases. Recall is the ratio of correctly predicted cases to total cases in the actual class. The equation for the F1 score is: $$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

3.5 Work plan

The project planning for FYP 1 and FYP 2 was done using Gantt chart as showed in Figure 3.6 and Figure 3.7. The allocated tasks were followed strictly according to the Gantt chart and timeline, and all the tasks were accomplished by the end of the course UEGE4118 Project as shown in Table 3.4 and Table 3.5.

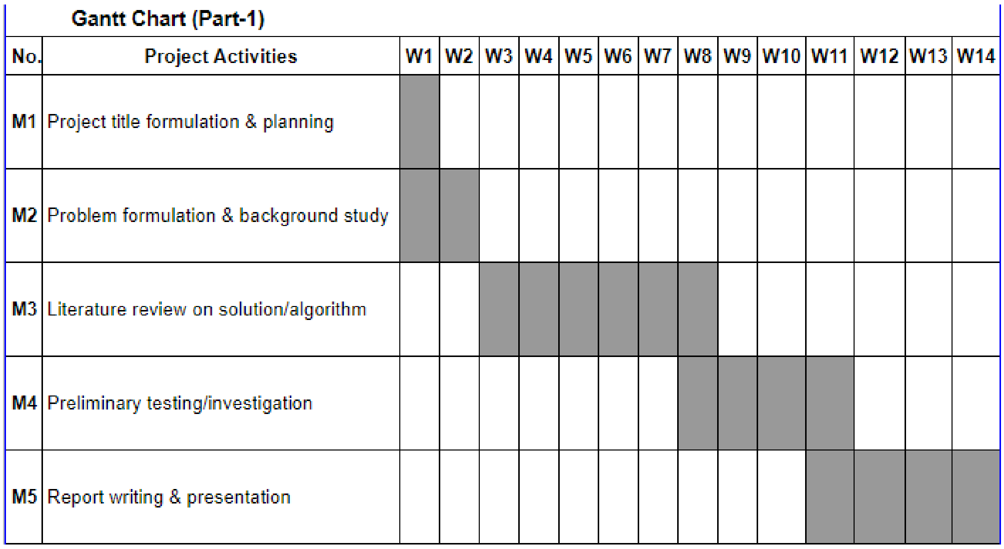


Figure 3.6: Gantt chart of Final Year Project 1.

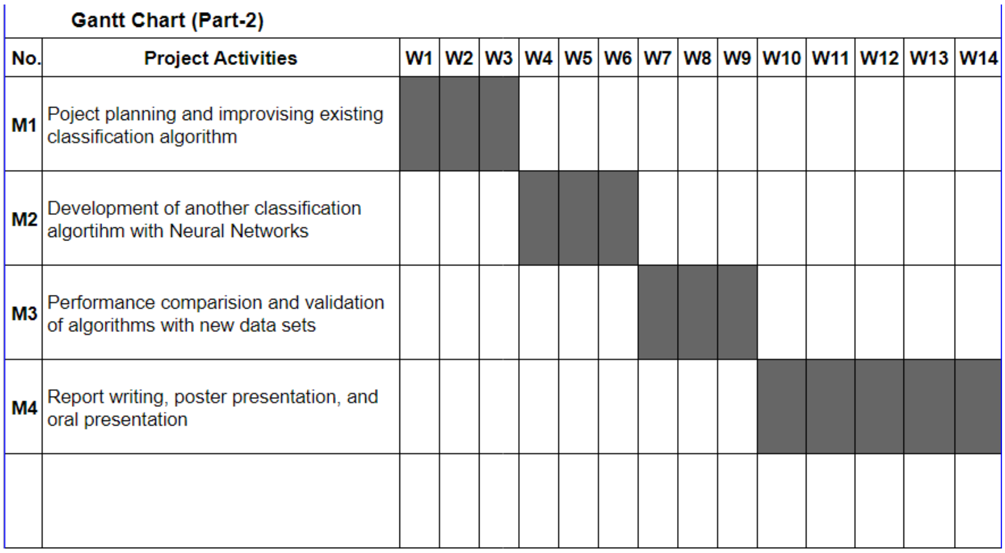


Figure 3.7: Gantt chart of Final Year Project 2.

Table 3.4: Final Year Project 1 milestones.

Milestones	Planned tasks	Completion date	
		Planned	Is it achieved before the deadline?
Project title formulation and planning	1.1 Project title selection	11 June	Yes
	1.2 Preparation of work schedule and Gantt chart		Yes
	1.3 Determine outputs		Yes

	and tasks		
Problem formulation and background study	2.1 Formulation of problem statement, aim and objectives	18 June	Yes
	2.2 Background study to understand the project title		Yes
Literature review	3.1 Review on various gait-affecting factors	30 July	Yes
	3.2 Review on various gait analysis methods		Yes
	3.3 Review on classification algorithms		Yes
Preliminary investigation	4.1 Dataset preprocessing	20 Aug	Yes
	4.2 Algorithm development and testing		Yes
Report writing and presentation	5.1 Writing progress report	3 Sept	Yes
	5.2 Prepare presentation		Yes

Table 3.5: Final Year Project 2 milestones.

Milestones	Planned tasks	Completion date	
		Planned	Is it achieved before the deadline?
Poject planning and improvising existing classification algorithm	1.1 Validate the performance of previous model using new dataset.	11 Feb	Yes
	1.2 Fine-tuning the hyperparameters of SVM model for better results.		Yes
	1.3 Plan the tasks for this semester.		Yes

Development of another classification algortihm with Neural Networks	2.1 Train and test different dataset using different combinations of hyperparameters.	4 Mar	Yes
	2.2 Tabulate the results for comparison.		Yes
	2.3 ANN model development		
Performance comparision and validation of algorithms with new data sets	3.1 Tabulate the results of performance metrics and computatinal tiime for both SVM and ANN model.	25 Mar	Yes
	3.2 Generate comparison graphs for SVM and ANN.		Yes
	3.3 Tabulate the results for comparison.		Yes
Report writing, poster presentation, and oral presentation	4.1 Report writing	24 April	Yes
	4.2 Prepare FYP poster		Yes
	4.3 Prepare presentation		

3.6 Summary

In summary, the methodology and work plan of the project was discussed in detail in this chapter. The overview flowchart of methodology and work plan for the project was introduced. Next, the project's requirements and specifications were discussed, including the software used, training dataset, and data preprocessing. Furthermore, detailed methodology for developing gait classification algorithms was addressed. In addition, the methodology to evaluate the performance of gait classification algorithms was explained in detail.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents and discusses the performance results of different configurations of both the SVM and ANN models. Besides, the optimum practical configuration of both SVM and ANN models was justified and compared. Three datasets were used to validate the performance of classification models. D1 is the original dataset which contained 9129 instances of three classes; D2 is the second dataset implemented in FYP Part 2, which included 39189 examples of two classes; and D3 is the merged dataset of D1 and D2, which contained a total of 48318 instances of three classes.

4.2 Support Vector Machine classification model

In this section, the performance results of SVM model with different configurations and using three different datasets were illustrated. The two manipulating hyperparameters studied were the c value and gamma value.

4.2.1 Performance results

4.2.1.1 Tabulation of performance results

Table 4.1 shows the performance results of SVM classifier with different configurations on D1, D2 and D3 datasets. For all three datasets, the configuration of c = 1000 and gamma = 10 will produce the best performance score. The results imply that higher c and gamma value can generate better performance.

Table 4.1: Performance results of Support Vector Machine classification model.

Dataset	Gamma value	Computational time (minutes)	Performance metrics	
			Accuracy (%)	F1 score (%)
c = 100				
D1	0.01	1.5	68.56	63.92
	0.1	0.5	94.61	94.19
	1	0.18	99.12	99.11
	10	0.7	99.04	99.04
D2	0.01	7	83.32	79.39
	0.1	27	86.46	83.92
	1	30	89.88	88.55
	10	28	92.61	92.12
D3	0.01	14	76.13	68.35
	0.1	15	82.61	78.74
	1	30	89.7	88.36
	10	43	93.01	92.58
c = 1000				
D1	0.01	94	85.39	84.08
	0.1	34	98.04	97.98
	1	13	99.5	99.5
	10	44	99.06	99.06
D2	0.01	10	84.12	80.83
	0.1	30	87.45	85.31
	1	90	91.22	90.26
	10	104	93.56	93.30
D3	0.01	22	80.23	74.92
	0.1	40	85.49	82.74
	1	150	90.7	89.6
	10	210	93.53	93.27

4.2.1.2 Graphs of performance results

Figure 4.1 to Figure 4.3 show the plot of SVM performance metrics results of D1, D2, D3 training datasets, respectively.

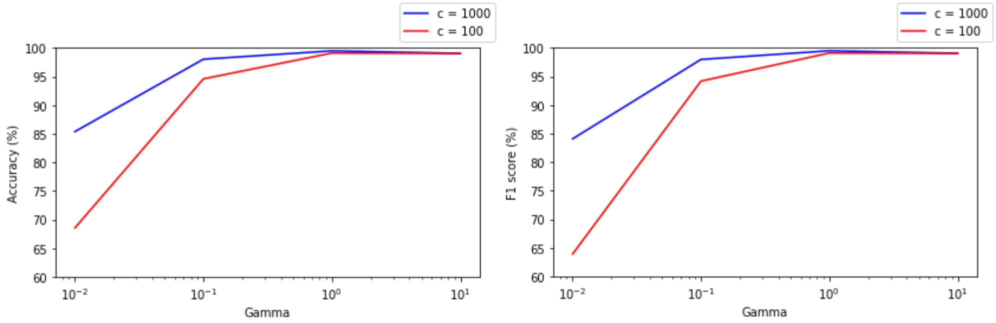


Figure 4.1: Support Vector Machine performance metrics of D1 dataset: Accuracy (left) and F1-score (right).

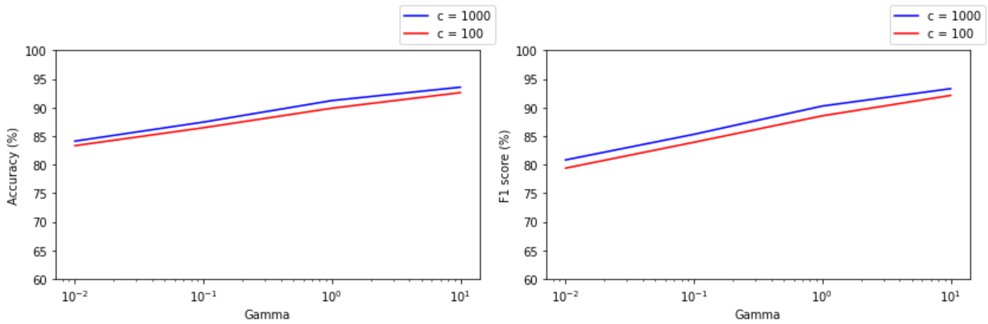


Figure 4.2: Support Vector Machine performance metrics of D2 dataset: Accuracy (left) and F1-score (right).

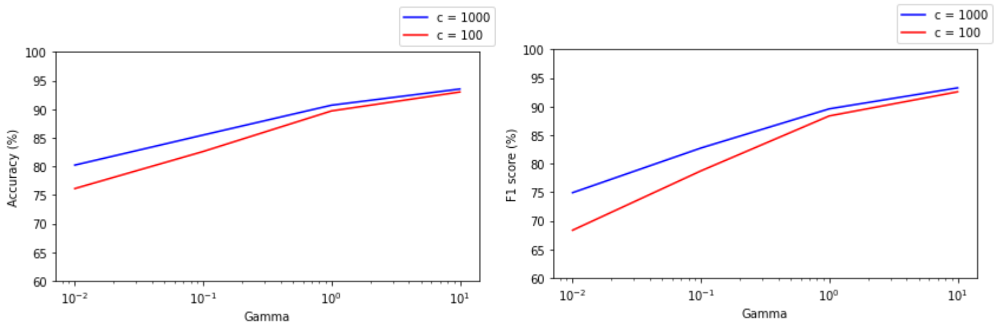


Figure 4.3: Support Vector Machine performance metrics of D3 dataset: Accuracy (left) and F1-score (right).

4.2.1.3 Graphs of computational time

Figure 4.4 to Figure 4.6 show the plot of SVM computational time of D1, D2, D3 training datasets, respectively

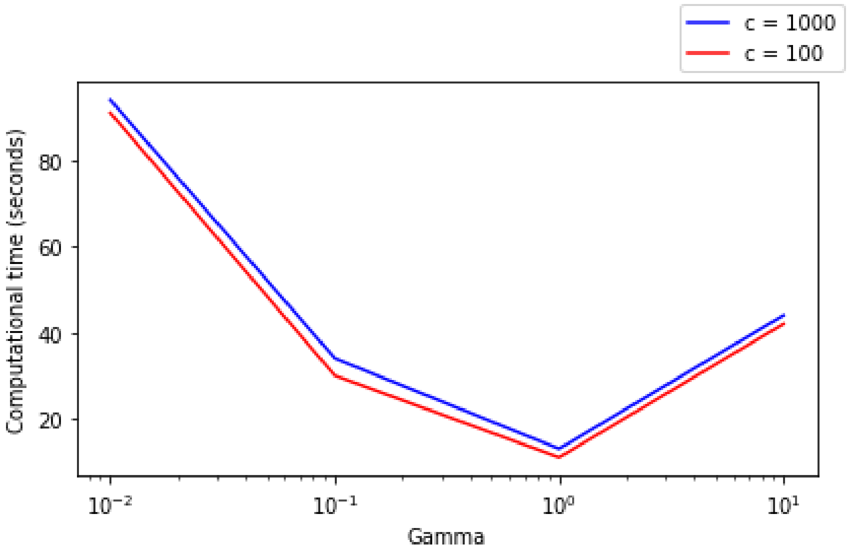


Figure 4.4: Support Vector Machine computational time of D1 dataset.

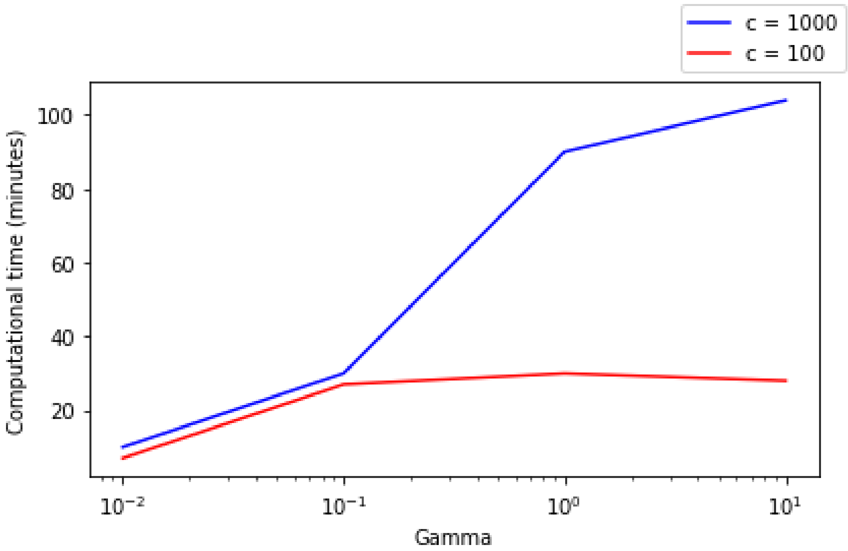


Figure 4.5: Support Vector Machine computational time of D2 dataset.

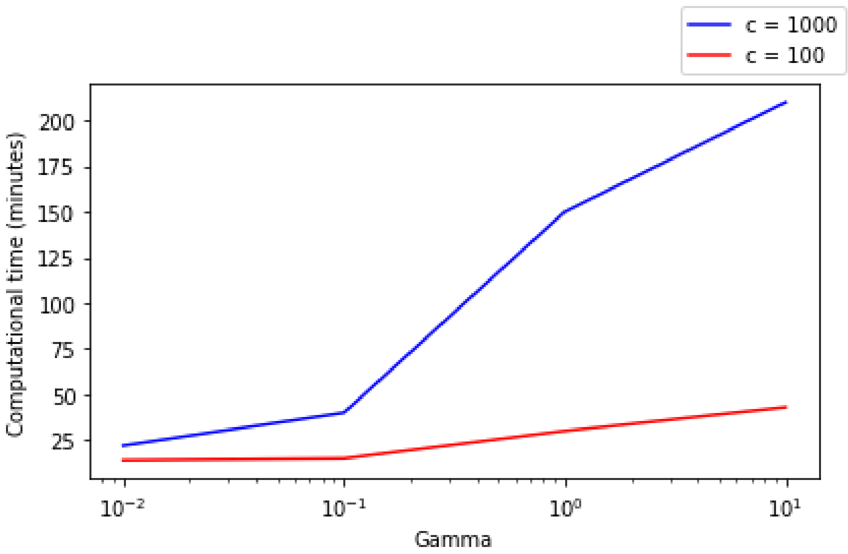


Figure 4.6: Support Vector Machine computational time of D3 dataset.

4.2.2 Results discussion

4.2.2.1 Performance metrics results

The presentation of performance results was illustrated in section 4.2.1. The section demonstrated the tabulation of detail values of performance metrics and the graphs of manipulating hyperparameters against the performance metrics and computational time. The graphs can give an overview of performance metrics results using different configurations of manipulating hyperparameters.

In this study, the two manipulating hyperparameters for configuring SVM were c and gamma values. As a recap, the c value controls the shape of the decision function and the size of the decision margin. A higher c value results in a more rigid decision function and smaller decision margin. As for the gamma value, this parameter defines the influence of points in the dataset to the hyperplane. In this study, the c value were 100 and 1000, and the gamma value ranged from 0.01 to 10 on an increment logarithmic scale of base 10. In addition, the hyperparameters that were kept constant were the number of cross-validation of 10, and kernel type of RBF kernel.

Observing the graphs in section 4.2.1.2, an increase in gamma value will increase the accuracy score and F1 score. A low gamma value will cause the decision boundary to have less curvature and fail to capture the shape of the dataset well. Conversely, higher gamma values will have more curvature in decision boundary, capture the shape of datasets well, and make more accurate predictions.

As shown in the graphs in section 4.2.1.2, the classification model with a c value of 1000 generally produced better results in three different datasets. When a low c value was applied, the penalty of misclassifying points was low, and the classifier could maximise the decision margin. Furthermore, the c value above 1000 was not included in the study because a large c value will cause overfitting problems in model training that leads to poor performance results. On the other hand, for large c values, the penalty of misclassification is high. Hence, the classifier tends to separate the data in the training dataset as perfect as possible, causing the decision boundary to be overfitted and cannot predict effectively for other testing datasets.

4.2.2.2 Computational time

As shown in the graphs in section 4.2.1.3, the computational time for different datasets and configurations was illustrated. Generally, three factors affect the computational time: the size of datasets, c value, and gamma value.

By observing the tabulation of results comparing the computational between different datasets of the same c value and gamma value, the D3 dataset has the highest computational time, and the D1 dataset has the shortest computational time. The highest computational time is 210 min (3 hours 30 minutes) for classifying D3 dataset with c value of 1000 and gamma value of 10. Compared to D3 dataset (48318 examples), D1 dataset (9129 examples) and D2 dataset (39189 examples) with the same hyperparameters took a significantly shorter computational time of 44 seconds and 104 minutes, respectively. Because when the size of the dataset increases, the number of example increase, which will cause the classification model to take a longer time to run through the whole dataset and generate the decision boundary and decision margin. Hence, the larger the dataset size, the longer the computational time.

Next, similar to performance metrics results, computational time increases for higher c and gamma values because the classification model needs to generate better decision boundaries for making predictions with higher accuracy. However, for the D1 dataset, it was interesting that the computational time was higher for gamma = 0.01 and 10, and shorter for gamma = 0.1 and 1. This may be due to the shape of D1 being more suitable for gamma in the range of 0.1 to 1. On the contrary, D2 and D3 followed the trend that increasing the gamma value would increase the computational time. As for c value, different c values have significant differences in computational time for D2 and D3 datasets, whereas for D1, the difference is less significant. The reason may be that D2 and D3 are larger in dataset size and have a more complex distribution of data points than the D1 dataset.

4.2.3 Optimum practical configuration

This section suggested and justified the optimum practical configuration for the SVM classifier.

The results of performance metrics and computational time were shown and discussed in detail in previous sections. It is clear that the higher the c and gamma value, the higher the performance metrics and the longer the computational time. Therefore, the resource and performance trade-offs were considered thoroughly when deciding the optimum configuration. The performance refers to the results of performance metrics in which we want to achieve as high as possible; and the resource refers to the computational time.

From the graphs in section 4.2.1.2, the increase in gamma value improves the performance metrics score. Thus, the optimum gamma value of 10 is suggested to achieve the highest possible performance metrics results. Next, to decide the optimum c value, the results of D3 dataset were used. The reason is that D3 dataset has the largest size of data example to provide the most comprehensive reference. Referring to Table 4.1, when gamma of 10 was applied to D3 dataset, the accuracy of c = 100 is 93.01 compared to c = 1000 is 93.53. As for computational time, using c = 100 is 43 minutes and for c = 1000 is 210 minutes. Therefore, using c = 100, the accuracy decreased by 0.5560% but can save 79.52% of the computational time compared to using c = 1000. Figure 4.7 below shows the graph to illustrate the relationship of accuracy and computational time of different configurations of SVM classifier.

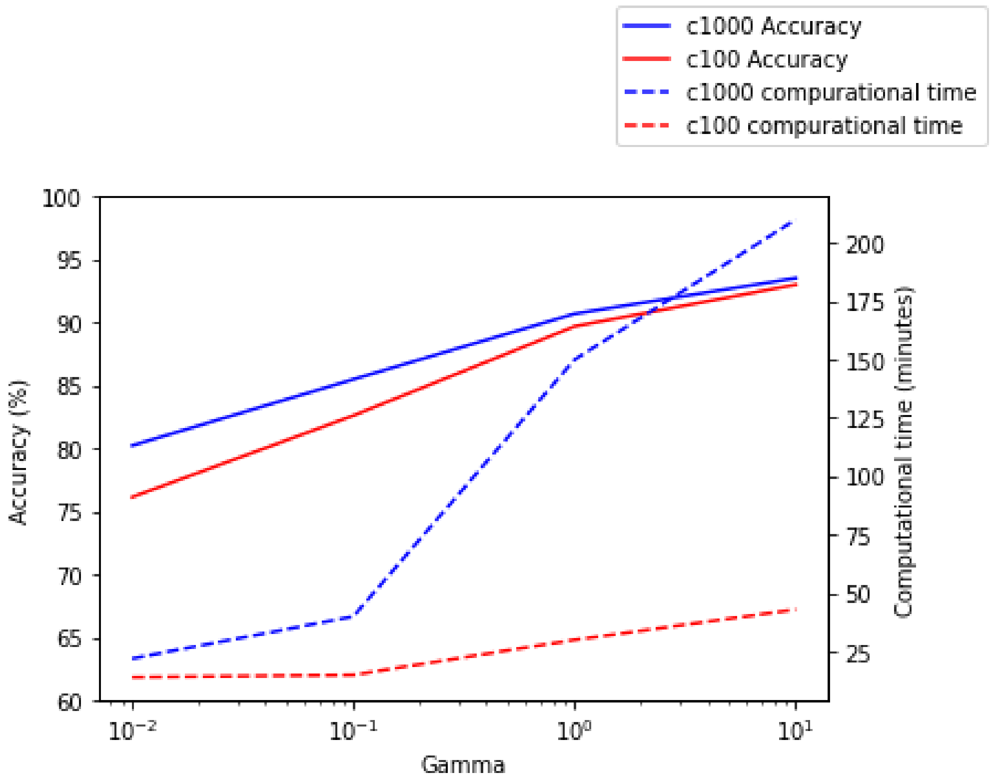


Figure 4.7: Graph of accuracy and computational time of different configurations of Support Vector Machine classifier.

In short, the optimum configuration suggested is using c value of 100 and gamma of 10, which can produce 93.01% accuracy with a computational time of 43 minutes. This configuration consumes reasonably low resources with the minimum trade-off of performance. It can still predict with high accuracy and F1 score but with much shorter computational time than other configurations.

Figure 4.8 below shows the confusion matrix of prediction results from optimum SVM configuration. The precision score for class old healthy, young healthy, and Parkinson's disease are 96.08%, 92.88% and 92.21% respectively, and the overall precision is 93.72%. Besides, the recall score for class old healthy, young healthy, and Parkinson's disease are 97.81%, 98.64% and 67.74% respectively, and the overall recall score is 88.06%. Interestingly, the recall score for class Parkinson's disease is significantly lower than the other two classes. This shows that the trained SVM model faces some problems identifying the input that is class Parkinson's because only 67.74% out of all the actual Parkinson's disease examples were predicted correctly.

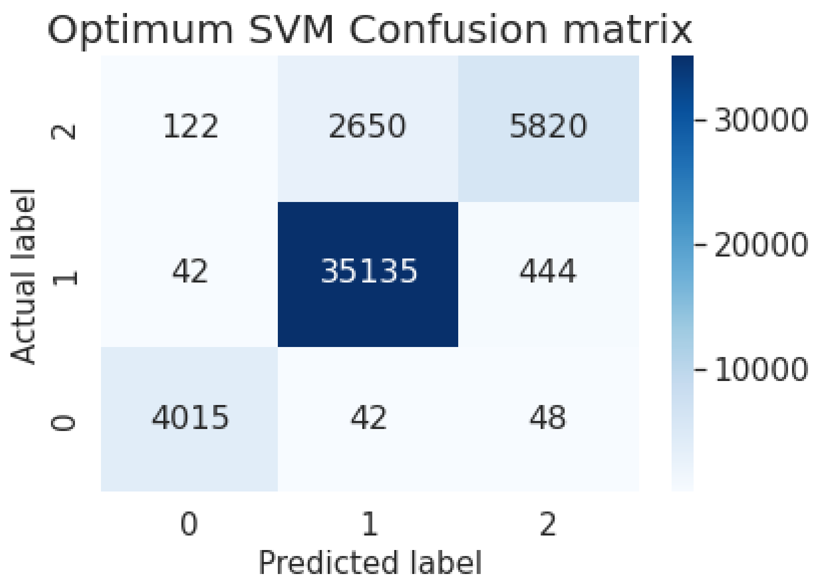


Figure 4.8: Confusion matrix of Support Vector Machine optimum configuration. The labels 0, 1, and 2 referred to old healthy, young healthy, and Parkinson's disease.

4.3 Artificial Neural Network classification model

In this section, the performance results of SVM model with different configurations and using three different datasets were illustrated. The manipulating hyperparameter is the number of neurons in the layer. The constant hyperparameters include ReLU activation function, cross-entropy loss function, ADAM optimiser, 200 training epochs and batch size of 16 as mentioned in Section 3.3.2 in Methodology.

4.3.1 Pretesting to decide the constant hyperparameters

4.3.1.1 Deciding loss function

Figure 4.9 compares the accuracy score of loss functions using different number of neurons in ANN classifier. The loss functions compared are the cross-entropy loss and KL-divergence loss. The other hyperparameters are kept constant as mentioned before. From Figure 4.9, it is observed that KL-divergence loss produces higher accuracy than cross-entropy loss at a smaller number of neurons; while cross-entropy loss performs better at a greater number of neurons. Although the accuracy difference between these two loss functions is relatively small (the difference is less than 2%). But since higher performance is desired, thus cross-entropy loss is used.

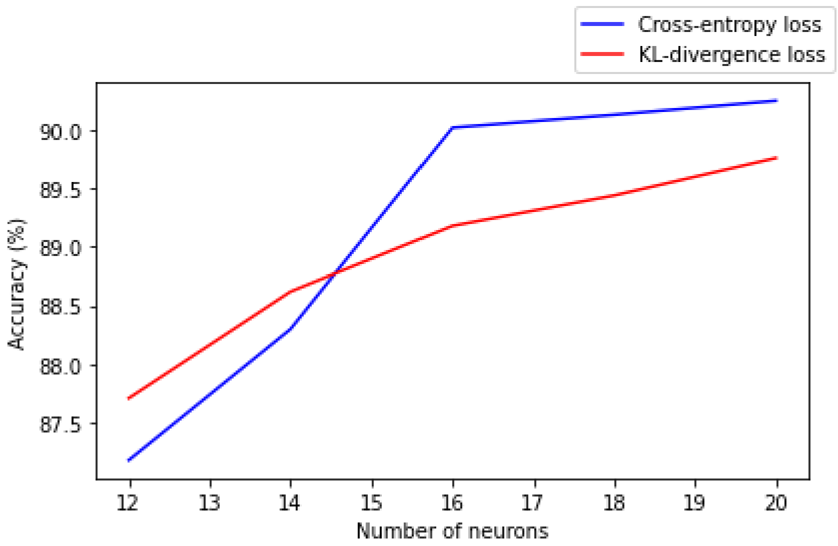


Figure 4.9: Comparison of accuracy score of different loss functions.

4.3.1.2 Deciding training epochs

Table 4.2 and Figure 4.10 show the results of applying different epochs in the ANN classifier for one iteration in 10-folds cross-validation. ANN classifier is applying gradient descent; thus to achieve better prediction results, we want to minimize the losses.

From Figure 4.10, the learning rate (the slope of the curve) decreases as the number of epochs increases. This shows that the training of the classifier is getting less effective with the increase in number of epochs. From Table 4.2, the training loss and validation loss of 300 epochs decreased by 8.85% compared to 200 epochs, but the computational time increased by 100%. Therefore, the number of epochs used for the project is fixed to 200. The computational time will be too long when performing K-fold cross-validation and the increase in performance is not significant.

Table 4.2: Losses using a various number of epochs in Artificial Neural Network classifier.

Epochs	Training loss	Validation loss	Computational time (minutes)
100	0.3561	0.3496	6
200	0.2881	0.2881	10
300	0.2626	0.2657	20

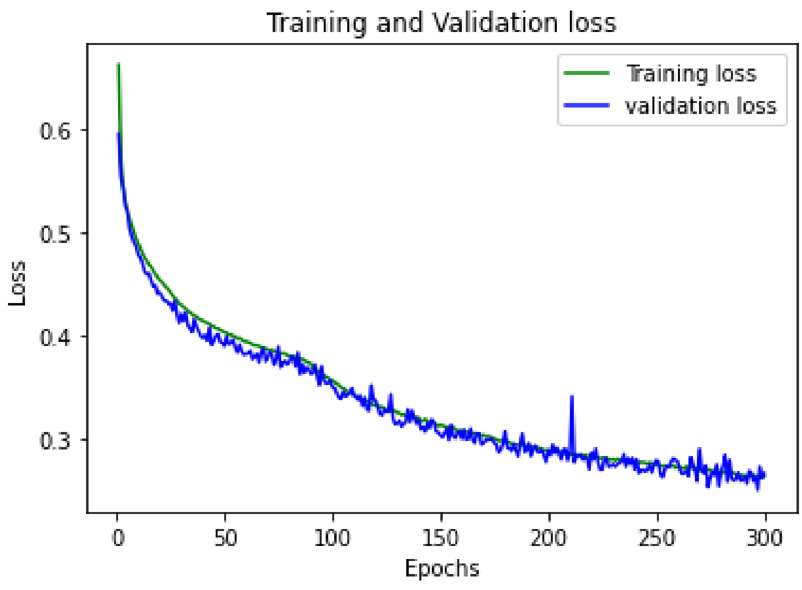


Figure 4.10: Plot of loss against epochs.

4.3.2 Performance results

4.3.2.1 Tabulation of performance results

Table 4.3 shows the performance results of ANN classifier on D1, D2, and D3 datasets.

Table 4.3: Performance results of Artificial Neural Network classification model.

Dataset	Number of neurons	Computational time (minutes)	Performance metrics	
			Accuracy (%)	F1 score (%)
D1	6	20	94.7	94.53
	8	17	97.17	97.15
	10	20	98.26	98.24
	12	22	98.47	98.45
	14	20	98.79	98.79
	16	20	98.79	98.78
	18	22	98.87	98.86
	20	23	98.8	98.8
D2	6	90	86.97	85.43

	8	84	87.64	86.08
	10	100	88.20	86.64
	12	100	88.24	86.69
	14	103	89.4	88.12
	16	83	89.94	88.83
	18	100	89.91	88.77
	20	84	90.47	89.5
D3	6	100	82.55	79.98
	8	108	85.12	83.28
	10	120	87.06	85.51
	12	100	88.45	87.15
	14	120	89.12	87.95
	16	100	89.58	88.5
	18	110	90.03	89.13
	20	112	90.56	89.69

4.3.2.2 Graphs of performance results

Figure 4.11 to Figure 4.13 show the graphs of ANN performance results of D1, D2 and D3 training datasets, respectively.

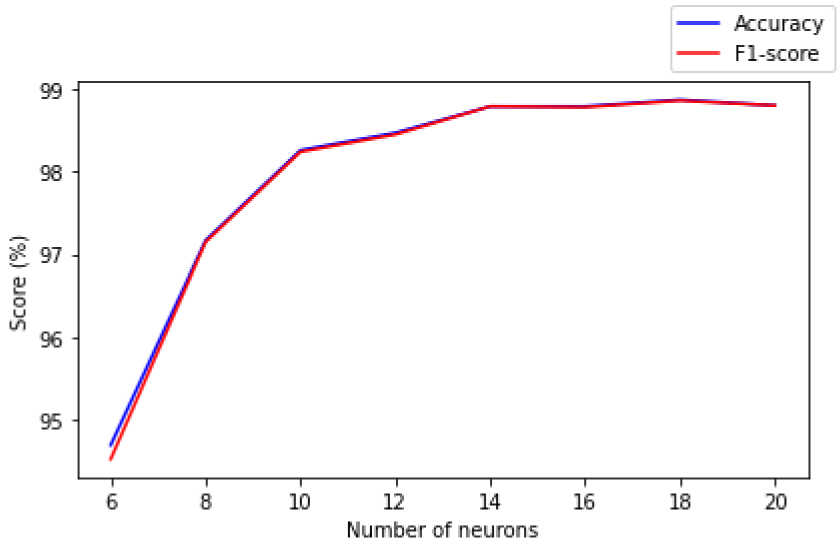


Figure 4.11: Artificial Neural Network performance metrics of D1 dataset.

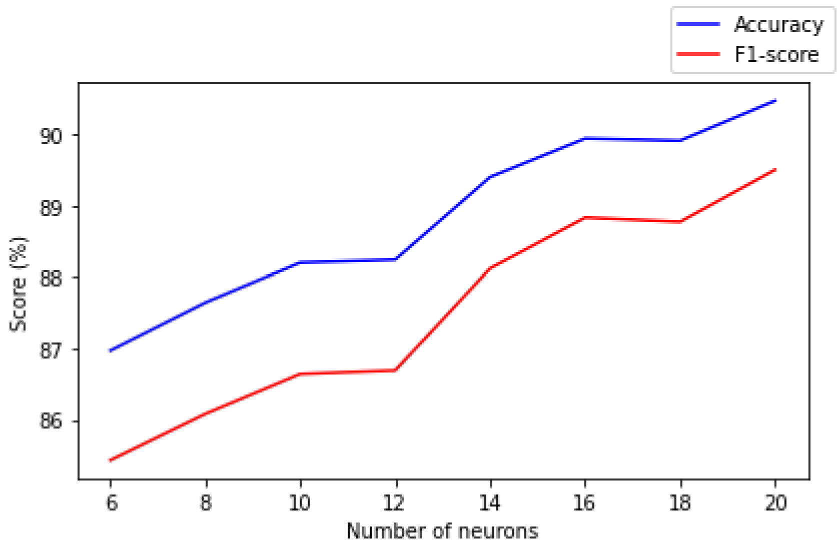


Figure 4.12: Artificial Neural Network performance metrics of D2 dataset.

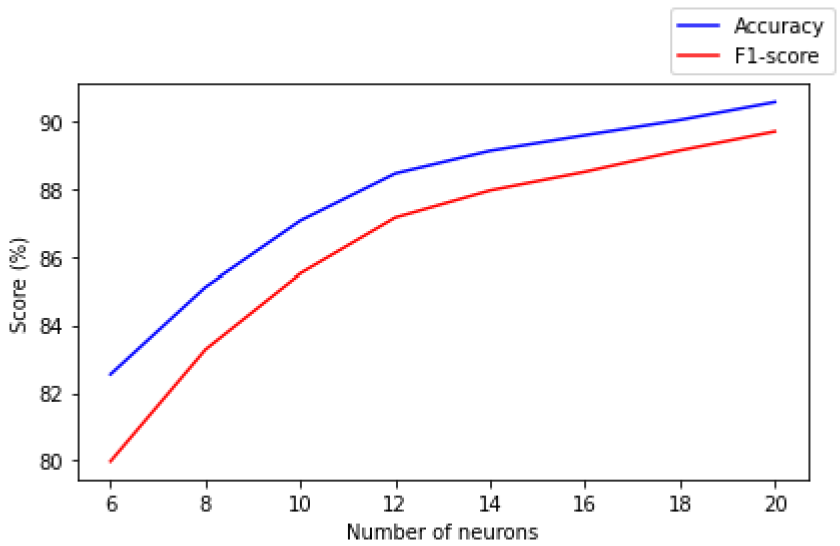


Figure 4.13: Artificial Neural Network performance metrics of D3 dataset.

4.3.2.3 Graphs of computational time

Figure 4.14 to Figure 4.16 show the graphs of ANN computational time of D1, D2, and D3 training datasets, respectively.

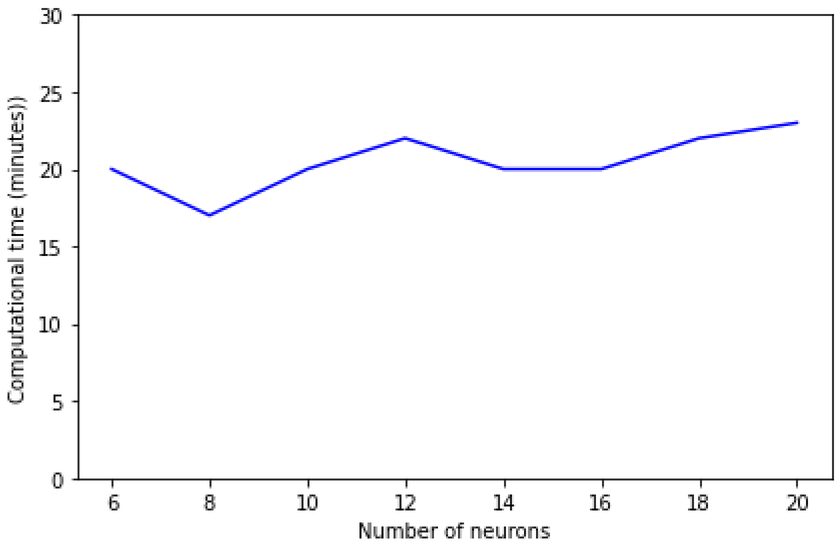


Figure 4.14: Artificial Neural Network computational time of D1 dataset.

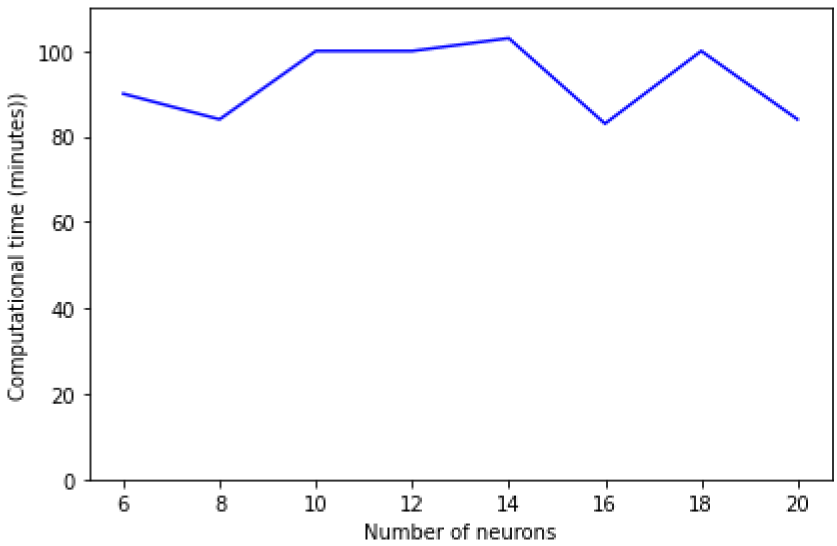


Figure 4.15: Artificial Neural Network computational time of D2 dataset.

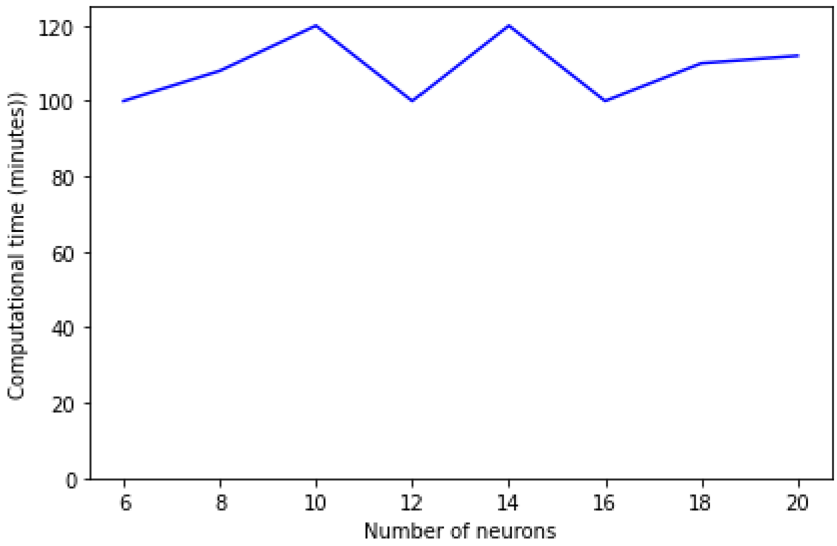


Figure 4.16: Artificial Neural Network computational time of D3 dataset.

4.3.3 Results discussion

4.3.3.1 Performance metrics results

The presentation of performance results was illustrated in section 4.3.2. The section demonstrated the tabulation of detailed values of performance metrics and the graphs to illustrate the performance metrics and computational time of ANN classifier. The graphs can give an overview of performance metrics results when manipulating the configuration of the ANN classifier.

In this study, the number of neurons is the manipulating variable for configuring the ANN model. The constant hyperparameters are ReLU activation function, cross-entropy loss function, ADAM optimiser, 200 training epochs and batch size of 16.

From the graphs in section 4.3.2.2, it is observed that increasing the number of neurons of hidden layers in ANN classifier improves the score of performance metrics. However, there are cases where the performance metrics score did not grow. For example, in D2 dataset, increasing the number of neurons to 12 did not improve the score compared to 10 neurons. However, the metrics curves show a growing trend for all datasets D1, D2 and D3. Besides, the accuracy scores are greater than F1 score. This may be because accuracy simply measures all the correct predicted cases over all the cases; whereas F1 score takes into account precision and recall while measuring the classifier's performance.

4.3.3.2 Computational time

From section 4.3.2.3, the computational time of different configurations of ANN classifier are shown. The factors that affect the computational time include the size of the dataset and the number of neurons.

From the tabulation of data and graphs, the computational time of D1 dataset that carries the least examples (9129 examples) is the shortest, ranged from 17 to 22 minutes. On the other hand, the D2 dataset (39189 examples) and D3 dataset (48318 examples) require computational time ranging from 83 to 100 minutes and 100 to 120 minutes, respectively. As the number of training examples increases, the classification model needs to take more time to calculate the losses and update the parameters for every iteration. Hence, the larger the dataset size, the longer the computational time.

Next, when the number of neurons increases, the computational time shows a fluctuation trend. Unlike SVM model, where increasing the c and gamma value will increase the computational time, ANN model does not show a patterned trend of computational time when the number of neurons increases. Take D3 dataset as an example, the computational time fluctuates in the range of 100 to 120 minutes. Hence, from the observation of the results, the computational time of ANN model does not positively proportional to number of neurons.

4.3.4 Optimum practical configuration

This section suggested and justified the optimum practical configuration for the ANN classifier.

In previous sections, the performance metrics results and computational time of different configurations of ANN classifier were discussed thoroughly. To summarize, increasing the number of neurons can help improve the classifier's performance, and computational time is directly influenced by the size of the training dataset, but not directly influenced by the number of neurons. Therefore, the resource and performance trade-offs were considered thoroughly when deciding the optimum configuration. The performance refers to the performance metrics scores which we want to achieve as high as possible, and the resource refers to the computational time.

Figure 4.17 below shows the relationship between accuracy score and computational time with the number of neurons for D3 dataset. When applying 20 neurons in hidden layers for ANN classifier produced the highest results among all the configurations. Next, the longest computational time is 120 minutes using ten and 14 neurons. The shortest computational time is 100 using 12 neurons and 16 neurons.

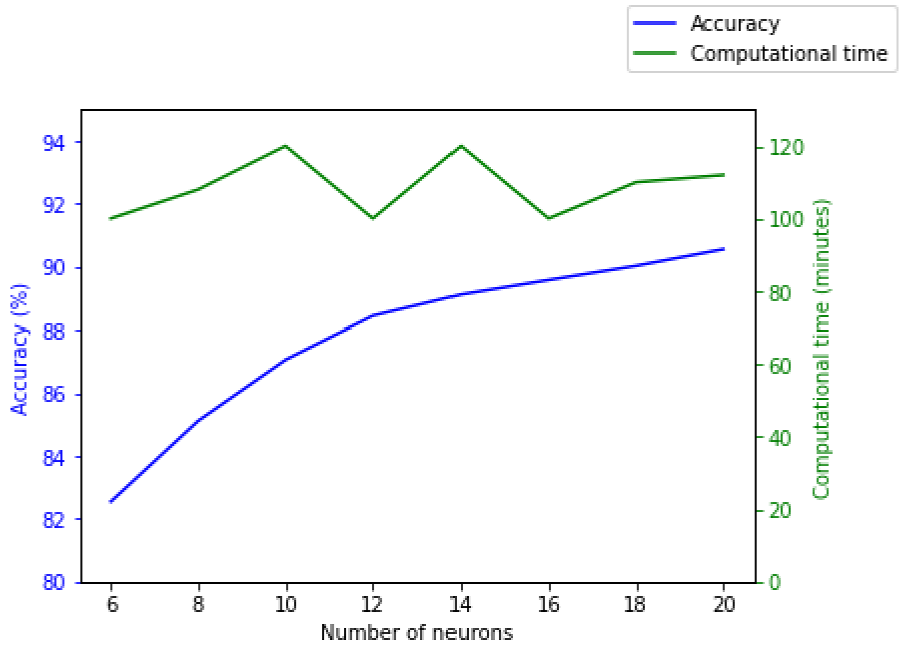


Figure 4.17: Graph of accuracy and computational time against number of neurons of D3 dataset.

In short, the optimum configuration suggested is using 20 neurons for hidden layers in ANN classifier, which can produce 90.56% accuracy and a computational time of 112 minutes. Based on Figure 4.17, ANN classifier with 20 neurons consumes reasonably resources (around the average computational time) and produces the best results. The results of the D3 dataset are used to justify the selection of optimum configuration because the D3 dataset has the largest training examples to provide the most comprehensive reference.

Figure 4.18 below shows the confusion matrix of prediction results from optimum ANN configuration. The precision score for class old healthy, young healthy, and Parkinson's disease are 94.78%, 90.37% and 88.62%, respectively, and the overall precision is 91.25%. Besides, the recall score for class old healthy, young healthy, and Parkinson's disease are 97.66%, 98.16%

and 55.66%, respectively, and the overall recall score is 83.97%. Like the optimum SVM classifier, the recall score for class Parkinson's disease is significantly lower than the other two classes in the ANN classifier. This shows that the trained ANN model faces some problems identifying the input that is class Parkinson's. 55.66% out of all the actual Parkinson's disease examples were predicted correctly using ANN classifier, which is lower than SVM model.

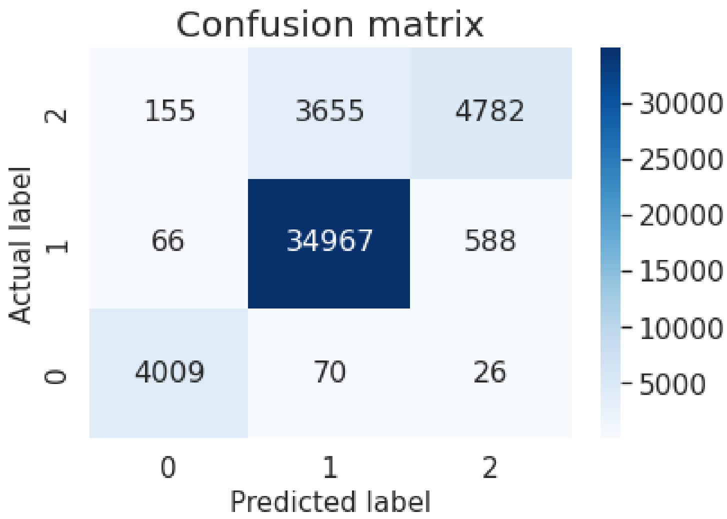


Figure 4.18: Confusion matrix of Artificial Neural Network optimum configuration. The labels 0, 1, and 2 referred to old healthy, young healthy, and Parkinson's disease.

4.4 Comparison with state-of-the-art of gait classification algorithms

In conclusion, the performance results for both developed classification algorithms were discussed thoroughly. Comparing the optimum configuration of SVM and ANN classifiers, SVM classifier is more suitable and effective for the dataset used in this study. This inference can be justified in terms of performance metrics results and computational time. In terms of performance metrics results, the optimum SVM model can generate an accuracy of 93.01% and F1 score of 92.58%; whereas ANN classifier generates an accuracy of 90.56% and F1 score of 89.69%. On the contrary, the computational time for optimum SVM configuration and ANN configuration is 43 minutes and 112 minutes, respectively. Therefore, the SVM classifier is more effectively than ANN classifier as overall for the gait dataset used in this study.

Next, the results from the SVM classifier developed in this study were compared with the state-of-the-art of gait classification in similar applications. Table 4.4 shows the comparison of state-of-art of gait classification. The accuracy of our study's proposed classifier is not outstanding compared to other classification algorithms developed by other researchers in the table. From the table, the algorithm with the highest accuracy is developed by Samà et al. (2013) using SVM with Gaussian kernel with image processing system dataset. However, our gait classification model can achieve comparable results with other algorithms from other research on similar gait applications. In addition, the dataset used in this study contains lesser information as the number of features and size of the dataset are not large compared to other datasets, such as a image processing dataset and wearable sensors dataset, which have larger data size and more training features.

Table 4.4: Comparison of state-of-art of gait classification.

Studies	Type of gait dataset	Classification algorithms	Accuracy (%)
Koh Chee Hong	Gait in Aging and Disease Database	SVM with RBF kernel	93.01
(Derawi and Bours, 2013)	Wearable system dataset	SVM using Dynamic Time Warping (DTW) distance metric	92.00
(Tien, Glaser and Aminoff, 2010)		SVM	84.60
(Samà et al., 2013)	Image processing system dataset	SVM with Gaussian kernel	96.40
(Abdulhay et al., 2018)		SVM with Gaussian RBF kernel	92.70
(LeMoyne et al.,	Floor-based sensors	SVM	80.00

2015)	dataset		
(Tahir and Manap, 2012)		SVM	95.80

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In conclusion, this report presented the performance of SVM and ANN classifiers on gait classification problems in differentiating three classes of subjects: young healthy, old healthy and Parkinson’s disease. In this project, two classification algorithms were developed. By manipulating and testing different combinations of hyperparameters, the optimum configuration for both classifiers was determined and justified. Then, the performance of the two classifiers was compared, and the SVM model was chosen as the proposed solution for the gait classification problem of this study. The proposed SVM classifier produces accuracy and f1 score of 93.01% and 90.56%, respectively, with a computational time of 43 minutes. After that, the proposed optimum SVM model was compared to other state-of-arts of gait classification algorithms in similar gait applications. Although, the performance results of the proposed SVM classifier were not outstanding compared to other gait classification algorithms developed by other researchers. However, our proposed classification algorithm produced comparable results with other state-of-arts using a smaller dataset with fewer training features. This suggests that the proposed SVM classifier can help in the approach of performing effective and accurate objective gait analysis using simpler gait data as input. Therefore, the problem statement was addressed, and the aim and objectives of this study were achieved.

5.2 Recommendations for future work

Future research of this problem should focus on validating the performance of the proposed classifier in classifying Parkinson’s disease using gait data collected physically in a real-life practical environment. Furthermore, improvements in the classifier's robustness are required by applying more training data. In addition, the training dataset used for training should have more information, such as a wider range in age, race and evenly distributed gender.

REFERENCES

Abdulhay, E., Arunkumar, N., Narasimhan, K., Vellaiappan, E. and Venkatraman, V., 2018. Gait and Tremor investigation using machine learning techniques for the diagnosis of Parkinson disease. *Future Generation Computer Systems*, 83, pp.366–373.

Arnold, Z., LaRose, D. and Agu, E., 2015. A factorial experiment to investigate naturalistic factors affecting smartphone gait analysis. In: *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*. pp.451–454.

Auvinet, B., Berrut, G., Touzard, C., Collet, N., Chaleil, D. and Barrey, E., 1997. Gait abnormalities in elderly fallers. *Journal of Aging and Physical Acliviry*, 11(1), pp.40–52.

Baker, R., 2006. Gait analysis methods in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 3.

Benabdelkader, C., Cutler, R. and Davis, L., 2004. Gait recognition using image self-similarity. *EURASIP Journal on Advances in Signal Processing*, 2004, pp.572–585.

di Biase, L., Di Santo, A., Caminiti, M.L., De Liso, A., Shah, S.A., Ricci, L. and Di Lazzaro, V., 2020. Gait Analysis in Parkinson's Disease: An Overview of the Most Accurate Markers for Diagnosis and Symptoms Monitoring. *Sensors*, 20(12), p.3529.

Brand, R.A., 1989. Can biomechanics contribute to clinical orthopaedic assessments? *The Iowa Orthopaedic Journal*, 9, pp.61–64.

Brownlee, J., 2016. *Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFLow Using Keras*. 7th ed. Melbourne, Australia: Machine Learning Mastery.

Brownlee, J., 2019. *How to Choose Loss Functions When Training Deep Learning Neural Networks*. [online] Deep Learning Performance. Available at: <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/> [Accessed 28 Feb. 2022].

Chen, Y.-C., Lou, S.-Z., Huang, C.-Y. and Su, F.-C., 2010. Effects of foot orthoses on gait patterns of flat feet patients. *Clinical Biomechanics (Bristol, Avon)*, 25(3), pp.265–270.

Cutter, G.R., Baier, M.L., Rudick, R.A., Cookfair, D.L., Fischer, J.S., Petkau, J., Syndulko, K., Weinshenker, B.G., Antel, J.P., Confavreux, C., Ellison, G.W., Lublin, F., Miller, A.E., Rao, S.M., Reingold, S., Thompson, A. and Willoughby, E., 1999. Development of a multiple sclerosis functional composite as a clinical trial outcome measure. *Brain : A Journal of Neurology*, 122(5), pp.871–882.

Demura, T., Demura, S., Uchiyama, M. and Sugiura, H., 2014. Examination of factors affecting gait properties in healthy older adults: focusing on knee extension strength, visual acuity, and knee joint pain. *Journal of Geriatric Physical Therapy (2001)*, 37(2), pp.52–57.

Derawi, M., Ali, H. and Alaya Cheikh, F., 2011. Gait Recognition using Time-of-Flight Sensor. In: *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*. pp.187–194.

Derawi, M. and Bours, P., 2013. Gait and activity recognition using commercial phones. *Computers & Security*, 39, pp.137–144.

Dietz, V., Leenders, K.L. and Colombo, G., 1997. Leg muscle activation during gait in Parkinson’s disease: influence of body unloading. *Electroencephalography and Clinical Neurophysiology*, 105(5), pp.400–405.

Dietz, V., Zijlstra, W., Prokop, T. and Berger, W., 1995. Leg muscle activation during gait in Parkinson’s disease: adaptation and interlimb coordination. *Electroencephalography and Clinical Neurophysiology*, 97(6), pp.408–415.

Djurić-Jovicić, M.D., Jovicić, N.S., Radovanović, S.M., Kresojević, N.D., Kostić, V.S. and Popović, M.B., 2014. Quantitative and qualitative gait assessments in Parkinson’s disease patients. *Vojnosanitetski Pregled*, 71(9), pp.809–816.

Donath, L., Faude, O., Lichtenstein, E., Nuesch, C. and Mündermann, A., 2016. Validity and reliability of a portable gait analysis system for measuring spatiotemporal gait characteristics: Comparison to an instrumented treadmill. *Journal of NeuroEngineering and Rehabilitation*, [online] 13(1), pp.1–9. Available at: <http://dx.doi.org/10.1186/s12984-016-0115-z>.

Duchi, J., Hazan, E. and Singer, Y., 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12, pp.2121–2159.

Elavarasan, N. and Mani, D.K., 2015. A survey on feature extraction techniques. *International Journal of Innovative Research in Computer and Communication Engineering*, 03(01), pp.52–55.

Elomaa, T., 1999. The Biases of Decision Tree Pruning Strategies BT. In: D.J. Hand, J.N. Kok and M.R. Berthold, eds. *Advances in Intelligent Data Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg.pp.63–74.

Engel, D., Hüttenberger, L. and Hamann, B., 2012. A survey of dimension reduction methods for high-dimensional data analysis and visualization. *OpenAccess Series in Informatics*, 27, pp.135–149.

F.Y, O., J.E.T, A., O, A., J. O, H., O, O. and J, A., 2017. Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, 48(3), pp.128–138.

Frigo, C. and Crenna, P., 2009. Multichannel SEMG in clinical gait analysis: a review and state-of-the-art. *Clinical Biomechanics*, 24(3), pp.236–245.

Gabel, M., Gilad-Bachrach, R., Renshaw, E. and Schuster, A., 2012. Full body gait analysis with Kinect. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2012, pp.1964–1967.

Gafurov, D., Bours, P. and Snekkenes, E., 2011. User authentication based on foot motion. *Signal, Image and Video Processing*, 5(4), p.457.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J.M. and Herrera, F., 2016. Big data preprocessing: methods and prospects. *Big Data Analytics*, 1(1), pp.1–22.

Geng, J., 2011. Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics*, [online] 3(2), pp.128–160. Available at: <http://aop.osa.org/abstract.cfm?URI=aop-3-2-128>.

Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K. and Stanley, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, [online] 101(23), pp.e215–e220. Available at: <https://doi.org/10.1161/01.CIR.101.23.e215>.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep learning*. Cambridge, MA: The MIT Press.

Goyal, C., 2021. *Importance of Cross Validation: Are Evaluation Metrics enough?* [online] Available at: <https://www.analyticsvidhya.com/blog/2021/05/importance-of-cross-validation-are-evaluation-metrics-enough/> [Accessed 6 Aug. 2021].

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V., 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, [online] 46(1), pp.389–422. Available at: <https://doi.org/10.1023/A:1012487302797>.

Hall, S.J., 2019. *Basic biomechanics*. 8th ed. New York: McGraw-Hill.

Hastie, T. and Tibshirani, R., 1996. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6), pp.607–616.

Heaton, J., 2008. *Introduction to Neural Networks for Java, 2nd Edition*. 2nd ed. United States of America: Heaton Research, Inc.

Hinton, G.E., Srivastava, N. and Swersky, K., 2012. *Neural Networks for Machine Learning Lecture 6a Overview of mini-batch gradient descent.* [online] COURSERA: Neural Networks for Machine Learning. Available at: <https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf%

0Ahttp://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf>.

Hobart, J.C., Riazi, A., Lamping, D.L., Fitzpatrick, R. and Thompson, A.J., 2003. Measuring the impact of MS on walking ability: the 12-Item MS Walking Scale (MSWS-12). *Neurology*, 60(1), pp.31–36.

Horak, F.B. and Mancini, M., 2013. Objective biomarkers of balance and gait for Parkinson's disease using body-worn sensors. *Movement Disorders : Official Journal of the Movement Disorder Society*, 28(11), pp.1544–1551.

Hunt, M.A., Birmingham, T.B., Giffin, J.R. and Jenkyn, T.R., 2006. Associations among knee adduction moment, frontal plane ground reaction force, and lever arm during walking in patients with knee osteoarthritis. *Journal of Biomechanics*, 39(12), pp.2213–2220.

Karantonis, D.M., Narayanan, M.R., Mathie, M., Lovell, N.H. and Celler, B.G., 2006. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on Information Technology In Biomedicine : A Publication of the IEEE Engineering in Medicine and Biology Society*, 10(1), pp.156–167.

Kesar, T.M., Binder-Macleod, S.A., Hicks, G.E. and Reisman, D.S., 2011. Minimal detectable change for gait variables collected during treadmill walking in individuals post-stroke. *Gait & Posture*, 33(2), pp.314–317.

Ketkar, N., 2017. Introduction to Keras. In: *Deep Learning with Python*. Apress, Berkeley, CA.pp.97–111.

Khalid, S., Khalil, T. and Nasreen, S., 2014. A survey of feature selection and feature extraction techniques in machine learning. *Proceedings of 2014 Science and Information Conference*, pp.372–378.

Kingma, D. and Ba, J., 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Ko, S., Tolea, M.I., Hausdorff, J.M. and Ferrucci, L., 2011. Sex-specific differences in gait patterns of healthy older adults: results from the Baltimore Longitudinal Study of Aging. *Journal of Biomechanics*, 44(10), pp.1974–1979.

Kobayashi, H., Kakihana, W. and Kimura, T., 2014. Combined effects of age and gender on gait symmetry and regularity assessed by autocorrelation of trunk acceleration. *Journal of Neuroengineering and Rehabilitation*, 11, p.109.

Kojima, S., Furuna, T., Ikeda, N., Nakamura, M. and Sawada, Y., 2008. Falls among community-dwelling elderly people of Hokkaido, Japan. *Geriatrics and Gerontology International*, 8(4), pp.272–277.

Kotsiantis, S., Kanellopoulos, D. and Pintelas, P., 2006. Data Preprocessing for Supervised Learning. *International Journal of Computer Science*, 1, pp.111–117.

Kotsiantis, S.B., 2007. Supervised Machine Learning: A Review of Classification Techniques. In: I. Maglogiannis, K. Karpouzis, M. Wallace and J. Soldatos, eds. *Emerging Artificial Intelligence Application in Computer Engineering*. Netherlands: IOS Press.pp.3–24.

Ladha, L., 2011. Feature selection methods and algorithms. *International Journal on Computer Science and Engineering*, 3(5), pp.1787–1797.

Lakshminarayan, K., Harp, S.A. and Samad, T., 1999. Imputation of missing data in industrial databases. *Applied Intelligence*, [online] 11(3), pp.259–275. Available at: <https://doi.org/10.1023/A:1008334909089>.

LeMoyne, R., Mastroianni, T., Hessel, A. and Nishikawa, K., 2015. Implementation of machine learning for classifying prosthesis type through conventional gait analysis. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. pp.202–205.

Leusmann, P., Mollering, C., Klack, L., Kasugai, K., Ziefle, M. and Rumpe, B., 2011. Your floor knows where you are: sensing and acquisition of movement data. In: *2011 IEEE 12th International Conference on Mobile Data Management*. pp.61–66.

Liu, H., Cao, Y. and Wang, Z., 2010. Automatic gait recognition from a distance. *2010 Chinese Control and Decision Conference, CCDC 2010*, pp.2777–2782.

Liu, T., Inoue, Y. and Shibata, K., 2010. A wearable ground reaction force sensor system and its application to the measurement of extrinsic gait variability. *Sensors*, 10(11), pp.10240–10255.

Lopez de Mantaras, R. and Armengol, E., 1998. Machine learning from examples: Inductive and Lazy methods. *Data & Knowledge Engineering*, 25(1), pp.99–123.

Malani, P., 2008. Geriatric physical diagnosis: a guide to observation and assessment. *The Journal of the American Medical Association*, 299, pp.1838–1839.

Massachusetts Institute of Technology, 1999. Gait in Aging and Disease Database. *PhysioNet*. [online] Available at: <https://archive.physionet.org/physiobank/database/gaitdb/>.

Mathie, M.J., Coster, A.C.F., Lovell, N.H. and Celler, B.G., 2004. Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiological Measurement*, 25(2), pp.R1-20.

Middleton, L., Buss, A.A., Bazin, A. and Nixon, M.S., 2005. A floor sensor system for gait recognition. *Proceedings - Fourth IEEE Workshop on*

Automatic Identification Advanced Technologies, AUTO ID 2005, 2005, pp.171–180.

Moissenet, F. and Armand, S., 2015. Qualitative and quantitative methods of assessing gait disorders. In: *Orthopedic Management of Children with Cerebral Palsy: A Comprehensive Approach*. Nova Science Publishers Inc.pp.215–240.

Muñoz Ospina, B., Valderrama Chaparro, J.A., Arango Paredes, J.D., Castaño Pino, Y.J., Navarro, A. and Orozco, J.L., 2019. Age matters: objective gait assessment in early Parkinson's disease using an RGB-D camera. *Parkinson's Disease*, 2019, p.5050182.

Muro-de-la-Herran, A., Garcia-Zapirain, B. and Mendez-Zorrilla, A., 2014. Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications. *Sensors*, 14(2), pp.3362–3394.

Murthy, S.K., 1998. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, [online] 2(4), pp.345–389. Available at: <https://doi.org/10.1023/A:1009744630224>.

Nagano, H., Sarashina, E., Sparrow, W., Mizukami, K. and Begg, R., 2019. General mental health is associated with gait asymmetry. *Sensors*, 19(22).

Ngo, T.T., Makihara, 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.

Nigg, B.M. and Herzog, W., 2007. *Basic biomechanics of the musculoskeletal system*. 3rd ed. Chichester, NJ: John Wiley & Sons.

Norton, R., Campbell, A.J., Lee-Joe, T., Robinson, E. and Butler, M., 1997. Circumstances of falls resulting in hip fractures among older people. *Journal of the American Geriatrics Society*, 45(9), pp.1108–1112.

Pirker, W. and Katzenschlager, R., 2017. Gait disorders in adults and the elderly : a clinical guide. *Wiener Kinische Wochenschrift*, 129(3–4), pp.81–95.

Pratheepan, Y., Condell, J. V and Prasad, G., 2009. The use of dynamic and static characteristics of gait for individual identification. In: *2009 13th International Machine Vision and Image Processing Conference*. pp.111–116.

Precise Biometrics, 2021. *Understanding biometric performance evaluation*. [online] Available at: <https://precisebiometrics.com/wp-content/uploads/2014/11/White-Paper-Understanding-Biometric-Performance-Evaluation.pdf> [Accessed 27 Jun. 2021].

Rabuffetti, M. and Frigo, C., 2001. Ground reaction: intrinsic and extrinsic variability assessment and related method for artefact treatment. *Journal of Biomechanics*, 34(3), pp.363–370.

Reed, L.F., Urry, S.R. and Wearing, S.C., 2013. Reliability of spatiotemporal and kinetic gait parameters determined by a new instrumented treadmill system. *BMC Musculoskeletal Disorders*, 14(1), p.249.

Reynolds, R.A., Woods, R. and Baker, J.D., 2006. *Handbook of Research on Electronic Surveys and Measurements*. *Journal of Computing in Higher Education*, United States of America: Idea Group Reference.

Robertson, D., Caldwell, G., Hamill, J., Kamen, G. and Whittlesey, S., 2013. *Research Methods in Biomechanics*. 2nd ed. United States of America: Edwards Brothers Malloy.

Rubenstein, L.Z., 2006. Falls in older people: epidemiology, risk factors and strategies for prevention. *Age and Ageing*, 35 Suppl 2, pp.ii37–ii41.

Rueterbories, J., Spaich, E.G., Larsen, B. and Andersen, O.K., 2010. Methods for gait event detection and analysis in ambulatory systems. *Medical Engineering & Physics*, 32(6), pp.545–552.

Samà, A., Ruiz, F.J., Agell, N., Pérez-López, C., Català, A. and Cabestany, J., 2013. Gait identification by means of box approximation geometry of reconstructed attractors in latent space. *Neurocomputing*, 121, pp.79–88.

Samson, W., Van Hamme, A., Sanchez, S., Chèze, L., Van Sint Jan, S. and Feipel, V., 2012. Dynamic footprint analysis by time-of-flight camera. *Computer Methods in Biomechanics and Biomedical Engineering*, 15 Suppl 1, pp.180–182.

Scholkopf, B., Burges, C. and Smola, A., 1998. *Advances in Kernel Methods*. *MIT Press*. London, England: The MIT Press.

Scikit-learn, 2021. *Support Vector Machines*. [online] Available at: <https://scikit-learn.org/stable/modules/svm.html> [Accessed 4 Aug. 2021].

Soofi, A. and Awan, A., 2017. Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic & Applied Sciences*, 13, pp.459–465.

Sprager, S. and Zazula, D., 2011. Impact of different walking surfaces on gait identification based on higher-order statistics of accelerometer data. In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. pp.360–365.

Storcheus, Dmitry; Rostamizadeh, Afshin; Kumar, S., 2015. A survey of modern questions and challenges in feature extraction. *The 1st InternationalWorkshop "Feature Extraction: Modern Questions and Challenges"*, 44, pp.1–18.

Sun, B., Wang, Y. and Banda, J., 2014. Gait characteristic analysis and identification based on the iPhone's accelerometer and gyrometer. *Sensors*, 14(9), pp.17037–17054.

Tableau Software, 2021. *Guide to data cleaning: definition, benefits, components, and how to clean your data*. [online] Available at: <https://www.tableau.com/learn/articles/what-is-data-cleaning> [Accessed 12 Jul. 2021].

Tahir, N. and Manap, H., 2012. Parkinson Disease Gait Classification based on Machine Learning Approach. *Journal of Applied Sciences*, 12, pp.180–185.

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

Teknomo, K., 2017. *Strengths and weaknesse of K Nearest Neighbor*. [online] Available at: <https://people.revoledu.com/kardi/tutorial/KNN/Strength and Weakness.htm> [Accessed 20 Jul. 2021].

Tien, I., Glaser, S.D. and Aminoff, M.J., 2010. Characterization of gait abnormalities in Parkinson's disease using a wireless inertial sensor system. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. pp.3353–3356.

Toosizadeh, N., Mohler, J., Lei, H., Parvaneh, S., Sherman, S. and Najafi, B., 2015. Motor performance assessment in Parkinson's Disease: association between abjective in-Clinic, objective in-Home, and subjective/semi-objective measures. *PloS One*, 10(4), p.e0124763.

Uusitalo, L., 2007. Advantages and challenges of Bayesian networks in environmental modelling. *Ecological Modelling*, 203(3), pp.312–318.

Veerabhadrappa and Rangarajan, L., 2010. Bi-level dimensionality reduction methods using feature selection and feature extraction. *International Journal of Computer Applications*, 4, pp.33–38.

Veres, G. V, Gordon, L., Carter, J.N. and Nixon, M.S., 2004. What image information is important in silhouette-based gait recognition? In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* pp.II–II.

Viswanath, P. and Sarma, T.H., 2011. An improvement to k-nearest neighbor classifier. In: *2011 IEEE Recent Advances in Intelligent Computational Systems*. pp.227–231.

Wang, S., Gao, R. and Wang, L., 2016. Bayesian network classifiers based on Gaussian kernel density. *Expert Systems with Applications*, 51, pp.207–217.

Wentink, E.C., Schut, V.G.H., Prinsen, E.C., Rietman, J.S. and Veltink, P.H., 2014. Detection of the onset of gait initiation using kinematic sensors and EMG in transfemoral amputees. *Gait & Posture*, 39(1), pp.391–396.

Whittle, M.W., 1996. Clinical gait analysis: A review. *Human Movement Science*, 15(3), pp.369–387.

Witten, I. and Frank, I.H., 2002. *Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations*. *Morgan Kaufmann*, San Mateo.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J. and Steinberg, D., 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1), pp.1–37.

Xue, Z., Ming, D., Song, W., Wan, B. and Jin, S., 2010. Infrared gait recognition based on wavelet transform and support vector machine. *Pattern Recognition*, 43(8), pp.2904–2910.

Young, W.R., Ellmers, T.J., Kinrade, N.P., Cossar, J. and Cocks, A.J., 2020. Re-evaluating the measurement and influence of conscious movement processing on gait performance in older adults: Development of the Gait-Specific Attentional Profile. *Gait & Posture*, 81, pp.73–77.

Zanardi, A.P.J., da Silva, E.S., Costa, R.R., Passos-Monteiro, E., Dos Santos, I.O., Kruel, L.F.M. and Peyré-Tartaruga, L.A., 2021. Gait parameters of Parkinson’s disease compared with healthy controls: a systematic review and meta-analysis. *Scientific Reports*, 11(1), p.752.

Zeiler, M., 2012. *ADADELTA: An adaptive learning rate method*. arXiv.

Zhong, Y. and Deng, Y., 2014. Sensor orientation invariant mobile gait biometrics. In: *IEEE International Joint Conference on Biometrics*. pp.1–8.

Zhou, S., 2003. Probabilistic analysis of kernel principal components : mixture modeling , and classification. In: *Conference and Workshop on Neural Information Processing Systems*.

Zhu, X. and Wu, X., 2004. Class noise vs. attribute noise: a quantitative study. *Artificial Intelligence Review*, 22(3), pp.177–210.