Curtin University
Malaysia

Motion Estimation Using Smart Watch Data

Matthew Law Ik Soon

23 OCTOBER 2022

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Motion Estimation Using Smart Watch Data

Matthew Law Ik Soon

A thesis submitted for the degree of Bachelor of Engineering
(Electronic and Electronic Engineering) (Honours)

23 October 2022

Synopsis

A wearable sensor-based technique for estimating lower limb motion based on machine learning and smartwatch data is discussed in this thesis. Long-term data collection is not feasible with the majority of traditional wearable motion analysis sensors, such as inertial measurement unit (IMU) sensors and kinetics foot insoles. In order to get an accurate measurement of the motion of the lower limb, multiple sensors must be mounted to the different joints and segment locations. In addition, the attachment of several sensors results in discomfort for the wearer. IMU sensors, such as smartwatches, have seen increased interest in motion analysis applications in recent years due to their convenience. Smartwatches comprise built-in IMU sensors, multi-functional and easy on the wearer. As a result, smartwatches are suitable for everyday motion analysis applications. This thesis discussed the various types of sensors and their characteristics. Furthermore, the research paper analyses the differences between several machine learning approaches with regard to the accuracy of network estimation.

Matthew Law Ik Soon,
Lot 4188, Jalan Spring,
Lutong Baru,
98000 Miri,
Sarawak, Malaysia.

November 8, 2022

A/Prof. Dr. Ling Huo Chong,
Head of Department,
Department of Electrical & Computer Engineering,
Faculty of Engineering and Science,
Curtin University, Malaysia Campus
CDT 250, 98009 Miri
Sarawak, Malaysia.

Dear A/Prof. Dr. Ling Huo Chong,

I, Matthew Law Ik Soon, hereby submit my thesis entitled "Motion Estimation Using Smart Watch Data" as part of my requirements for completion of the Bachelor of Engineering in Electronic and Electronic Engineering.

I declare that this thesis is entirely my own work with the exception of the acknowledgements and references mentioned.

Yours sincerely,

. . . .

Matthew Law Ik Soon
700028118

TITLE
Motion Estimation Using Smart Watch Data

AUTHOR
Family Name: Law Ik Soon
Given Name: Matthew

DATE 23 October, 2022	**SUPERVISOR** Dr. Saaveethya Sivakumar
DEGREE Bachelor of Engineering	**OPTION** Electronic and Electronic Engineering

ABSTRACT

In recent years, estimation-based gait analysis has acquired increasing popularity. Current trends in long-term gait monitoring rely on machine learning-based estimations. The potential among different machine learning is examined in this study through a comprehensive analysis of current gait estimates based on those methods. As a result, this research assessed the feasibility of FFNN, LSTM, and CNNLSTM for estimating lower limb motion using wrist acceleration acquired from a smartwatch sensor. The wrist acceleration is obtained from a publicly accessible database, which was selected for network training. The gait event is walking, and 32 participants are involved in the gait event. The best result of the lower limb motion acceleration are estimated with $p > 0.929$ and RMSE:0.36°±0.11% in left ankle acceleration x-axis, 0.22°±0.08% in y-axis, 0.20°±0.06% in z-axis, 0.29°±0.11% in right ankle acceleration x-axis, 0.2°±0.09% in y-axis, and 0.23°±0.07% in z-axis. Besides, the smartwatch sensor method may minimise the requirement for several wearable sensors and provide lower limb estimation on a long-term basis using a simplified sensor design. Consequently, using a smartwatch sensor, the proposed study analyses the viability of employing three different machine-learning models for lower limb estimates based on wrist acceleration.

INDEXING TERMS

Smartwatch; Wearable Sensors; IMU sensor; Acceleration; Machine Learning; Gait; Lower Limb Motion; Feed Forward Neural Network(FFNN); Long Short-Term Memory(LSTM); Convolutional Long Short-Term Memory(CNNLSTM).

	GOOD	AVERAGE	POOR
Technical Work			
Report Presentation			

Examiner	Co-Examiner

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for award of any other degree or diploma in any university. This document also complies with the rules of Curtin University on copyright and plagiarism.

....

Matthew Law Ik Soon

23/10/2022

....

Date

Acknowledgement

I would like to take this opportunity to thank every Department of Electrical and Computer Engineering staff in Curtin University, and especially my supervisor Dr Saaveethya Sivakumar, for leading me through my Engineering Year 4 Degree. I would also like to express my gratitude to Curtin University for providing me with the opportunity to acquire engineering expertise. Lastly, I want to express my appreciation to Curtin University for the chance it has provided me to compose a thesis based on the engineering skills I have acquired there.

Contents

Abstract **i**

Declaration **iv**

Acknowledgement **v**

List of Figures **x**

List of Tables **xi**

Abbreviations **xii**

1 INTRODUCTION **1**

1.1 Project Overview 1

1.2 Motivation 3

1.3 Problem Statement 4

1.4 Objectives 5

1.5 Thesis Structure 5

2 LITERATURE REVIEW **6**

2.1 Gait Analysis Sensor 6

2.1.1 Wearable Sensor 7

2.1.2 Fabric-Based Strain Sensor 7

2.1.3 Surgace Electromyogram Sensor 8
2.1.4 Delsys Trigno Sensor 8
2.1.5 Smartwatch Sensor 8

2.2 Smartwatch Based on Gait Analysis 8

2.3 Comparison Between Different Machine Learning Available 9

3 METHODOLOGY 15

3.1 Chapter Overview 15

3.2 Smartwatch Dataset 16

3.3 Pre-processing the Data 17

3.4 Machine Learning Model 19

3.5 Estimation Lower Limb Motion 21

3.6 Chapter Summary 21

4 ANKLE ACCELERATION ESTIMATION USING FFNN 22

4.1 FFNN Implementation Ideas 22

4.2 FFNN Model with 2 Hidden Layers 24

4.3 FFNN Model with 3 Hidden Layers 25

4.4 FFNN model with 4 Hidden Layers 27

4.5 Chapter Summary 29

5 ANKLE ACCELERATION ESTIMATION USING LSTM 30

5.1 LSTM Implementation Ideas 30

5.2 LSTM Model with 1 Hidden Layer 32

5.3 LSTM Model with 2 Hidden Layers 34

5.4 LSTM Model with 3 Hidden Layers 35

5.5 Chapter Summary 37

6 ANKLE ACCELERATION ESTIMATION USING CNN-LSTM 38

6.1 CNNLSTM Implementation Ideas 38

6.2 CNNLSTM Result and Discussion 40

6.3 Chapter Summary 43

7 CONCLUSION AND FUTURE WORK 44

7.1 Conclusion 44

7.2 Future Work 45

List of Figures

Figure 2.1 Sensor available for gait estimation 7

Figure 3.1 Project Workflow 15
Figure 3.2 Input and Output for the project model 16
Figure 3.3 32 participants information 17
Figure 3.4 Average graph of 32 participant 18
Figure 3.5 Graph of relationship between age, weight and height . . . 18

Figure 4.1 Layer Structure of FFNN Model with 2 Hidden Layers . . 24
Figure 4.2 Loss Graph for Training FFNN Model with 2 Hidden Layers 24
Figure 4.3 Bar Chart of RMSE with The Error Bar for 2 Hidden Layers FFNN Model 25
Figure 4.4 Layer Structure of FFNN Model with 3 Hidden Layers . . 25
Figure 4.5 Loss Graph for Training FFNN Model with 3 Hidden Layers 26
Figure 4.6 Bar Chart of RMSE with The Error Bar for 3 Hidden Layers FFNN Model 26
Figure 4.7 Layer Structure of FFNN Model with 4 Hidden Layers . . 27
Figure 4.8 Loss Graph for Training FFNN Model with 4 Hidden Layers 27
Figure 4.9 Bar Chart of RMSE with The Error Bar for 4 Hidden Layers FFNN Model 28

Figure 5.1 LSTM Structure 30
Figure 5.2 Layer Structure of LSTM Model with 1 Hidden Layer . . . 32
Figure 5.3 Loss Graph for Training LSTM Model with 1 Hidden Layer 32

Figure 5.4 Bar Chart of RMSE with The Error Bar for 1 Hidden Layer LSTM Model 33
Figure 5.5 Layer Structure of LSTM Model with 2 Hidden Layers . . 34
Figure 5.6 Loss Graph for Training LSTM Model with 2 Hidden Layers 34
Figure 5.7 Bar Chart of RMSE with The Error Bar for 2 Hidden Layers LSTM Model 35
Figure 5.8 Layer Structure of LSTM Model with 3 Hidden Layers . . 35
Figure 5.9 Loss Graph for Training LSTM Model with 3 Hidden Layers 36
Figure 5.10 Bar Chart of RMSE with The Error Bar for 3 Hidden Layers LSTM Model 37

Figure 6.1 Structure of CNNLSTM Model 38
Figure 6.2 Layer Structure of CNNLSTM Model 39
Figure 6.3 Loss Graph for Training CNNLSTM Model 40
Figure 6.4 Accuracy Graph for Training CNNLSTM Model 40
Figure 6.5 Bar Chart of RMSE with The Error Bar for CNNLSTM Model 41
Figure 6.6 CNNLSTM Estimation Result for Model Outputs 42

List of Tables

Table 2.1 Types of sensor 6
Table 2.2 Summary of Gait Estimation Method 10

Table 4.1 Mean RMSE with Standard Deviation and Correlation Coefficient Results for FFNN Model with 2 Hidden Layers 24
Table 4.2 Mean RMSE with Standard Deviation and Correlation Coefficient Results for FFNN Model with 3 Hidden Layers 26
Table 4.3 Mean RMSE with Standard Deviation and Correlation Coefficient Results for FFNN Model with 4 Hidden Layers 28

Table 5.1 Mean RMSE with Standard Deviation and Correlation Coefficient Results for LSTM Model with 1 Hidden Layers 33
Table 5.2 Mean RMSE with Standard Deviation and Correlation Coefficient Results for LSTM Model with 2 Hidden Layers 34
Table 5.3 Mean RMSE with Standard Deviation and Correlation Coefficient Results for LSTM Model with 3 Hidden Layers 36

Table 6.1 CNNLSTM Training and Validation Result 40
Table 6.2 Mean RMSE with Standard Deviation and Correlation Coefficient Results for CNNLSTM Model 41

Abbreviations

ACL	Anterior Cruciate Ligament
ANNs	Artificial Neural Networks
CNNLSTM	Convolutional Neural Network Long Short-Term Memory
CNN	Convolutional Neural Network
Conv1D	1D Convolutional Neural Networks
DeepConvLSTM	Deep Convolutional Recurrent Neural Network With Long-short Term Memory
EMG	Electromyography
GRU	Gated Recurrent Units
IMU	Inertial Measurement Unit
KFJ	Knee Joint Force
LSTM	Long-short Term Memory
MLP	Multilayer Perceptron
NRMSE	Normalized Root Mean Square Error
OA	Osteoarthritis
PCA-RELM	Principal Component Analysis-Regularized Extreme Learning Machine

PCA	Principal Component Analysis
RF	Random Forest
RMSE	Root Mean Square Error
sEMG	Surface Electromyogram
Tanh	Hyperbolic Tangent

Chapter 1

INTRODUCTION

1.1 Project Overview

Walking is fundamental to human locomotion, requiring the coordination of the muscles, brain, and nerves [1]. Gait is the locomotion achieved by the movement of the limbs, whereas gait analysis evaluates the patterns, whether normal or abnormal, based on the gait cycle. A gait cycle refers to the sequence of limb motion throughout an individual's walking pattern to attain movement by propelling the body's centre of gravity forward [2]. According to Gurchiek et al.[3], there are approximately 250,000 people under the age of 30 have a fracture to their Anterior Cruciate Ligament (ACL) on an annual basis. Nearly half of the people who choose surgical reconstruction may develop post-traumatic osteoarthritis (OA), which may result from changing movement patterns and cause abnormal knee joint loading [3]. In our daily life, gait motion is difficult to capture and analyze.

Gait analysis is the systematic study of human movement. It can be implemented by assessing changes in joint kinematics relative to dynamics acting on the foot externally [4]. In other words, kinematics and kinetics play an important role in gait analysis. Human gait kinematics indicates the motions of the major joints and lower extremity components in the human gait, such as segment angles, joint angles, and angular motion. Gait kinetics studies forces and moments generated by human segment movement, in which the orientation of all leg segments is determined by gait kinematics. [5]. In order to achieve gold standard motion analysis, kinematics data would require six tracking cameras to capture the movement of reflection attached to segments and joints of the lower limb. In short, a 3-dimensional motion capture system is the gold standard for kinematics measurements [6]. Nevertheless, multiple marker attachments restrain human movement and impact natural human locomotion [4]. Therefore, the experiment is not feasible for long-term gait monitoring.

In recent years, the popularity of smart wearable sensors such as smartwatches has gradually extended and inspired different research directions, which include monitoring human health conditions. For instance, smart wearable sensors can track patient health, aid in illness diagnosis, and anticipate patient outcomes [7]. Smartwatch sensors can also be used for long-term motion capture in an environment outside the laboratory. The smartwatch's built-in Inertial Measurement Unit (IMU) sensor is involved in the gait analysis for motion data collection. Furthermore, wearable sensors such as smartwatches require just one sensor attached to the wrist, reducing human movement constraints.

IMU sensor is an electronic device that measure and compute an object's specific force and angular rate. Typical configurations contain one accelerometer, gyroscope, and magnetometer per axis for each of the three principal axes: pitch, roll, and yaw. When IMU sensors are attached to the human body, the sensors track and measure the signal from human motion. Some IMUs-based motion analysis systems are available such as Xsens, Invemsense, Microstrian, PX4, VectorNav, Sparkfun,and ADIS [8]. Nevertheless, IMU sensor are susceptible to error accumulation over time, known as "drift". Since the device continuously measures the change in relation to itself, the IMU sensor keeps rounding up small fractions in its calculations over time. Therefore, these minor inaccuracies might lead to a significant error. Research tends to overcome the IMU sensor drift error using machine learning methods, such as Artificial Neural Networks.

Neural Networks, also known as Artificial Neural Networks (ANNs), is a subset of machine learning where the structure of the human brain inspires the algorithms [9]. ANNs are known for predicting the desired output through the learning process [6]. ANNs are proven to be powerful tools, especially for IMU sensor data-based applications. Therefore, in this project, ANNs will be proposed as a tool to estimate lower limb joint angles from the measure of smartwatch IMU sensor data. As some predictive modelling of many variables, it increases model training time and requires a vast number of system memory. From that, the user will encounter a high-dimensional data analysis in machine learning [10]. The feature selection method is an alternative way to overcome the high dimensional data analysis [11].

Feature selection is an excellent technique to handle the problem by removing unnecessary and redundant data. Feature selection analysis reduces the number of input variables during data training [10], [11]. As a result, it may enhance learning accuracy, reduce computation time, and promote a better understanding of learning data or models [9]. Hence, the feature selection method will be proposed as a tool in this project to reduce redundant input data and simplify learning results.

1.2 Motivation

The typical gait is evaluated using different types of wearable sensors, and in most cases, these sensors are directly connected to various segments and joints to evaluate motion. In order to get an accurate reading of the motion being performed by the lower limb, it is necessary to connect multiple sensors to various joints and segments. The wearer experiences discomfort due to the attachment of several sensors. Smartwatches offer a solution that can be utilised daily and estimate motion by just attaching to the user's wrist. Hence, it eliminates the need for additional sensors.

As human dependence on technology grows yearly, smartwatches present an opportunity for new fields in estimating gait motion. Moreover, a smartwatch can be utilised for the purpose of long-term motion capture in an environment other than a laboratory environment. In addition, the rising popularity of smart sensors has gradually expanded into other areas of study, such as sporting activity or clinical applications. However, recent research has shown that fewer people implement smartwatch sensors for gait analysis. It is likely since smartwatch sensors are a newer technology than traditional wearable sensors. This research proposed a preliminary study using data from smartwatches to investigate motion analysis by contrasting several machine learning models.

1.3 Problem Statement

Research on human gait involves performing qualitative and quantitative analyses of the various factors that contribute to its overall characteristics. The variables of interest might differ from one area of research to another. For example, for security purposes, the focus may be on differentiating and recognising individuals based on a broad description of their silhouette and the motions between the subject's various body parts when walking. Gait disorders affect a significant portion of the world's population and are major contributors to the progression of neurodegenerative diseases, including Parkinson's disease, cerebellar ataxia, myelopathies, neuromuscular diseases (myopathies), and spinal amyotrophy. From a clinical perspective, the significance of human gait analysis lies in the fact that gait disorders affect a substantial portion of the world's population.

In addition, each year, there are many people suffer a fracture to their anterior cruciate ligament. Nearly half of the people who choose surgery to fix their knees may get post-traumatic osteoarthritis, which can happen when people modify movement patterns and cause abnormal knee joint loading. The investigation of human gait characteristics has been the focus of much research because of the potential utility of this knowledge in clinical applications. Acceleration motion analysis is utilised in human gait to explore and analyse the systematic examination of human gait motion. Acceleration is the rate at which the velocity of an object changes over time, and it is dependent on position, time, speed, and direction.

1.4 Objectives

This research project has objectives to:

1. Develop machine learning models to train the data for estimating gait motion based on wrist acceleration
2. Identify salient inputs that best correlate with the smartwatch data and the lower limb motion
3. Compare different machine learning models that best suites for estimating gait motion from smartwatch data

1.5 Thesis Structure

The rest of the paper is structured as follows: Chapter 2 indicates the literature review of the gait analysis sensor, a comparison of the various machine learning available, a problem statement and the project's objective. The methodology taken to complete the project is detailed in Chapter 3. Chapters 4, 5 and 6 discuss various machine learning models for lower limb acceleration estimation and their respective model results. Chapter 7 indicates the conclusion and future work of the project, whereas Chapter 8 includes the appendix of the project.

Chapter 2

LITERATURE REVIEW

2.1 Gait Analysis Sensor

Table 2.1 summarize different types of sensors for the gait analysis, the sensors attached to the body parts and the data collected by the sensors. More and more sensors have been designed and invented to estimate human locomotion in recent years. Figure 2.1 shows five types of gait analysis sensors: wearable sensor, fabric-based strain sensor, Surface Electromyogram (sEMG) sensor, Delsys Trigo sensor, and smartwatch sensor [12]–[16].

Table 2.1: Types of sensor

Sensor	**Attached body part**	**Acquired data**
Wearable Sensor	Lower extremity or arm or both	Acceleration and gyroscope
Fabric-Based Strain Sensor	Lower extremity	Acceleration and gyroscope
Surface Electromyogram Sensor	Lower extremity	Surface electromyogram
Delsys Trigno Sensor	Lower extremity and wrist	Electromyography
Smartwatch Sensor	Wrist	Acceleration and angular velocity

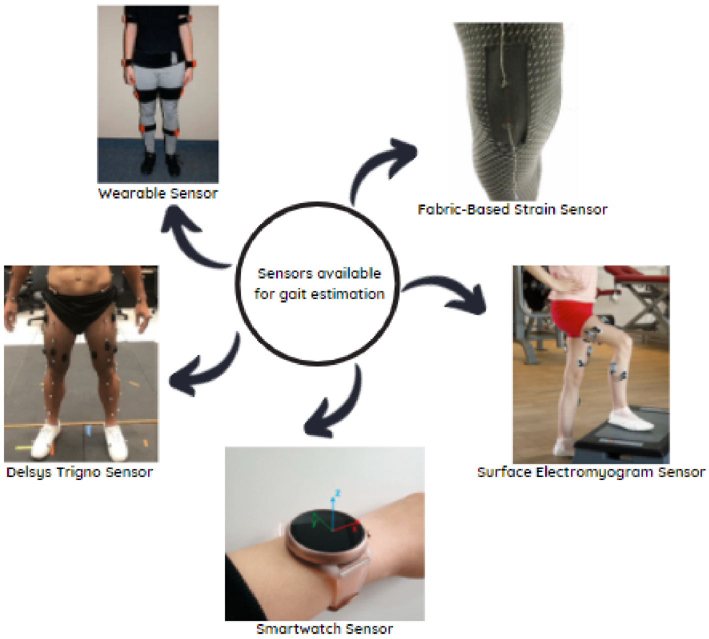

Wearable Sensor
Fabric-Based Strain Sensor
Sensors available for gait estimation
Delsys Trigno Sensor
Surface Electromyogram Sensor
Smartwatch Sensor

Figure 2.1: Sensor available for gait estimation

2.1.1 Wearable Sensor

Wearable sensors are attached to the whole human body but mainly on the lower limb body and arm. Wearable sensors are generally used to collect the acceleration and gyroscope data for estimating gait motion. Multiple sensors are attached to different human body parts to collect more data and information. Therefore, a better and more accurate result is achieved when numerous wearable sensors are connected.

2.1.2 Fabric-Based Strain Sensor

The fabric-based strain sensor is a sensor consisting of thermoplastic and nano-conductive particles in a filament [13]. Gholami et al. [13] sewed the sensor onto the cloth and subsequently put it onto a commercially stretchable pant. The conductive thread was attached to the sensor at each end using conductive paint and super glue. Conductive thread is used to make electrical connections [13]. The limitation of the sensor is minor adjustment is required to precisely position the sensor on the knee for different people.

2.1.3 Surgace Electromyogram Sensor

A surface electromyogram sensor is used to collect the surface electromyogram signal for gait estimation. The surface electromyogram signal is a physiological signal produced by muscle contraction or relaxation on the skin [14]. Deng et al. [14] employed a sEMG sensor to estimate the knee joint angle from surface electromyography signals.

2.1.4 Delsys Trigno Sensor

Besides, the Delsys Trigno sensor is a sensor to collect Electromyography (EMG) signal. EMG is a signal that is often used to determine the electrical activity of the skeletal muscles [15]. EMG signal may be exploited to estimate human intent for the purpose of controlling.

2.1.5 Smartwatch Sensor

The smartwatch sensor is one type of IMU sensor, built into the smartwatch. Smartwatch users can collect the data such as acceleration and angular velocity data when immobile. Joo et al. [16] used a smartwatch to predict the foot striking pattern from the smartwatch data. A smartwatch sensor is convenient for people to make gait estimations as the estimation can be done both inside and outside the laboratory. Compared to other sensors, some of the sensors cannot be used outside the laboratory. As the dependence of technology increases over the year, smartwatch is being highly used as a sensor in the future for the gait analysis.

2.2 Smartwatch Based on Gait Analysis

The smartwatch is one of the latest devices for gait analyses and an auspicious mobile tool with a wide range of onboard sensing capabilities in motion estimation. Most current smartwatches have both an accelerometer and gyroscope sensor to support the experiment that relies on sensor-based activity identification. However, the smartwatch is a wrist-mounted device without any other sensors. It is a challenge for the smartwatch to detect standard metrics for gait analysis [17]. In a further study, the capability of the smartwatch is investigated to determine the standard gait analysis.

Smartwatch-based biometrics has gained popularity due to the low cost of smartwatches and the fact that smartwatches are convenient for people to wear (i.e., smartwatches are mobile). Besides, the smartwatch can transmit data to other devices such as smartphones via Bluetooth or the Internet [18]. The smartwatch provides numerous significant benefits over other mobile devices like smartphones for gait-based biometrics. For instance, the smartwatch is attached to the most common location on the body: the wrist. When the user is walking, there is substantially more movement at the wrist than smartphone position, located at the upper thigh in a trouser pocket [18]. In short, the smartwatch shows that it is preferable for biometric identification.

2.3 Comparison Between Different Machine Learning Available

Table 2.2 summarize the literature review on human gait estimations using the wearable sensors presented from 2015 to 2022. From Table 2, we can see that different machine learning types are performed to train the data for estimating gait motion.

Table 2.2: Summary of Gait Estimation Method

Author	**Type**	**Feature Selection**	**Input**	**Output**	**Results**
Poitras et al. [12]	IMUs	-	Acceleration and gyroscopes	3D joint angular kinematics	WRIST -RMSE: 3 to 30 -p: 0.62 to 0.99
Gholami et al. [13]	Scikit-Learn module of python	Random forest algorithm	Acceleration and gyroscope	Knee joint angle	Intra-subject evaluation: -Walking & flexion exercises:MAE=1.94&3.02 -Coefficient of determination, R(square) = 0.97 Inter-subject evaluation: -Walking & flexion exercises:MAE=4.14&6.97 -Coefficient of determination, R(square) = 0.9
Deng et al. [14]	RELM	PCA	Surface electromyogram	Knee joint angle	RMSE = 0.8478 & p = 0.954
Coker et al. [15]	ANN	-	Electromyography (EMG)	Knee joint angle	Knee flexion angle prediciton: p <0.001 & RMSE = 4.62
Joo et al. [16]	LSTM, GRU and Conv1D	Random forest algorithm	Acceleration and angular velocities	Foot striking pattern	Fm (weighted average of F1-scores): -Walking pattern data: Fm = 96.379 -Running pattern data: Fm = 98.056
Stetter et al. [19]	ANN	-	Acceleration and angular velocity	Knee joint forces (KJF)	Vertical KJF: 0.6<r<0.94 ∼Highest correlation: (0.94± 0.33) ∼rRMSE for Fv* and Fv: 14.2% and 25.9% Anterior-posterior KFJ: 0.64<r<0.90 ∼rRMSE for Fap* and Fap: 17.4% and 27.1% Medial-lateral KFJ: 0.25<r<0.60 ∼rRMSE for Fap* and Fap: 27.7% and 45.9% where *=predicted; no include*=calculated
Hernandez et al. [20]	DeepConvLSTM	-	Acceleration and gyroscope	Lower body joint angles	All speed conditions: 0.67<r<0.99 Hip right rotation&lumbar extension:0.4<r<0.7 Hip left rotation&ankle inversion:0.7<r<0.9 Knee and hip joints: RMSE = 2.4 Walking and running: RMSE = 1.9 and 1.6
Sung et al. [21]	RNN LSTM	Filter feature selection method	Acceleration and angular velocity	Ankle, knee, and hip joint angle	R(square) >0.74, RMSE <7 & NRMSE <9.87
Rapp et al. [22]	LSTM and Conv1D	-	Accelerometer and gyroscope	Joint kinematics	Lower extremity kinematics: RMSE (<1.27 ± 0.38) for flexion/extension RMSE (<2.52 ± 0.98) for abduction RMSE (<3.34 ± 1.02) for int/ext rotation
Mundt et al. [23]	ANN, LSTM, MLP and CNN	LOSO validation correlation coefficients	Acceleration and angular rate	Joint angle	Flexion-extension angles: r >0.911 Knee angles: 0.389 <r <0.625
Ardestani et al. [24]	3-layer FFNN	PCA	Acceleration and displacement	DSP: GRF & M	Training data: p = 0.98 Validation data: p = 0.97
Argent et al. [25]	Machine learning (ML) algorithm	Random forest algorithm	Kinematics	Joint angle	RMSE = 4.81 (SD = 1.89) RMSE = 4.99 (SD = 1.83) (specific additional variable included)
Wouda et al. [26]	MLP	-	-Orientations of lower body -Acceleration (vertical)	-Knee joint angle -Vertical ground reaction forces	Knee joint angles: p >0.9 & RMSE <5 Ground reaction forces: p >0.9 & RMSE <0.27BW
Majumder et al. [27]	IMUs	Filter Robustness	Accelerometer and gyroscope	Joint angle	Average RMSE = 2.5 & r = 0.951
Alcaraz et al. [28]	CNN, FNNs, GRNN, RNN, NARX, and LSTM	Hibert-Huang transformation (HHT)	3D linear acceleration and 3D angular velocity	Joint angle	Hip joint angle: average RMSE = 1.91 Knee joint angle: average RMSE = 2.12 Ankle joint angle: average RMSE = 2.57

Stetter et al. [19] employed an Artificial Neural Network (ANNs) to estimate knee joint force (KFJ) during various sports movements based on acceleration and angular velocity data obtained from the wearable sensor. Based on the results, ANN predicted a strong KFJ correlation for vertical KFJ and anterior-posterior KFJ. The research summarised the link between wearable sensors and ANNs for evaluating joint reaction estimation during sports movement. In contrast to Stetter et al.[19], Coker et al. [15] applied ANNs to train data to predict knee flexion angles during gait at different speeds (in ms). As the number of training trials increased, the accuracy of angle predictions in ANNs increased. The proposed ANN-based technique demonstrated that ANNs is able to map non-linearity between input and output. However, the amount of training data influences the relationship's robustness between the input and output variables. As a result, ANN is not recommended to use in a large sample size to improve its robustness of input and output variables. Therefore, the drawback of ANNs is the limited sample size and narrow demographics of the subjects [15], [19]. The authors have suggested additional research into the combining the proposed method for centroids estimation.

Hernandez et al. [20] estimated lower body joint angles from lower limb kinematic data (acceleration and angular velocity data) using a deep convolutional recurrent neural network with long-short term memory (DeepConvLSTM). According to the research results, DeepConvLSTM predicted a high correlation for hip left rotation, ankle right, and left inversion while a moderate correlation for hip right rotation and lumbar extension. Without any specific knowledge of the wearer's body features or the orientation of the IMU relative to the attached segment, the proposed model can consistently predict joint kinematics for walking, running, and gait changes. In contrast, Sung et al. [21] developed a one-layer long-short term memory (LSTM) model with extracted feature input to estimate multi-joint angles. In the LSTM model, hyperbolic tangent (Tanh) was used as an activation function. The results depicted a strong coefficient of determination for angle estimation with a low normalized root mean square error (NRMSE). Nevertheless, LSTM models are intensive to train and require large datasets. Therefore, it requires much more time to train the input variables to identify the patterns in the data and forecast the output.

Joo et al. [16] employed 1D convolutional neural networks (Conv1D) models to estimate the foot striking pattern from the smartwatch data. In this study, the authors applied the Conv1D model as machine learning to learn complicated and hierarchical features and train the obtained data from raw sensor signals in an end-to-end learning method. The results showed a robust classification performance with a high weighted-average F1 score through Conv1D and gated recurrent units (GRU) models. In contrast, Rapp et al. [22] proposed a kinematics estimation from IMUs data using combined deep learning, including Conv1D, and LSTM. The authors employed one-dimensional Conv1D to predict a time window that glides through the whole time series, making offline processing more efficient. To summarise the paper, Conv1D models with 2 to 3 hidden layers, a window size with more than 55ms, more than 50 nodes per layer, 5000 iterations, and a learning rate greater than 0.001 produced the best results. The result indicated the lower extremity kinematics were predicted with a minimal root mean square error (RMSE).

Besides, Mundt et al. [23] compared between three types of machine learning for estimating joint angles and moments from IMUs data: multilayer perceptron (MLP), LSTM, and convolutional neural network (CNN). In this investigation, three neural networks were given the identical dataset. Overall, the result shown all ANN had strong correlations between IMUs data and predicted kinematic and kinetic data. First and foremost, MLP networks are used to flatten the image of time series data in order to handle time-sequence input data. MLPs are simple to train, but they are computationally costly and are not ideal for time-normalized data. Furthermore, CNNs were created with the goal of mapping image data to a single output variable. CNNs learn from raw image data by leveraging correlations between local pixels. CNNs are suited to data with spatial relationships. CNNs are suitable for predicting time series human movements due to the spatial relationship seen in time series data. LSTM networks were designed to solve sequence prediction problems. The success of LSTM in natural language processing can be attributed to its ability to exploit temporal dependencies in the data. Therefore, they are suitable for tasks involving motion sequence prediction. The

CNN network is the best machine learning for training data since it predicted the most accurate predictions among the three-machine learning. However, the CNN network requires a huge dataset and elaborate preprocessing operation to convert IMUs data to images. In contrast to CNN, MLP had only marginally lower prediction capabilities with significantly fewer pre-processing stages. In comparison to CNN and LSTM, the performance of LSTM was poor. Nevertheless, it requires less pre-processing and facilitates the real-time joint angle and joint moment prediction [23].

Deng et al. [14] proposed an angle estimation algorithm of the continuous motion for the feature selection based on Principal Component Analysis (PCA) feature selection. This paper collected surface electromyogram (sEMG) signals during lower limb movements [14]. The signals were then pre-processed, and feature samples were retrieved from the obtained sEMG signals. The authors used the PCA method to reduce the feature sample dimensions. PCA is used to extract and analyse the complex interactions between highly coupled variables because of its inherent capability to differentiate and identify the underlying basic patterns of a data space. Using Principal Component Analysis-Regularized Extreme Learning Machine (PCA-RELM) algorithms, the result depicted a strong correlation coefficient for the knee joint angle estimation from the EMG signal. In contrast to Deng et al. [14], Ardestani et al. [24] proposed that PCA identifies the contributions of joint kinematics to joint moments. In most cases, PCA is used to reduce the dataset such that it may be represented in terms of a lower-dimensional dataset while retaining as much information as possible from the original dataset. However, Ardestani et al. [24] employed PCA to generate a large dataset of probabilistic kinematics from a small set of experimental databases.

In another study, Gholami et al. [13] proposed a random forest algorithm as a machine-learning algorithm to estimate the knee joint angle from IMUs data.

Random Forest (RF) algorithms and neural networks were developed by using the scikit-Learn module of python in this study. The authors raised the number of trees in the RF algorithm until no more improvements in outcomes were detected. To compute the optimal parameter, the RF algorithm's parameters such as maximum features and minimum sample leaf were altered. However, the calculated parameters were deemed slightly insensitive, hence the default parameters were used. Overall, the results depicted an excellent coefficient of determination for walking and flexion exercise. Compared to the traditional linear technique, the knee joint angle estimation using the RF algorithm was more accurate. In contrast, Argent et al. [25] proposed a few forms of feature selection to estimate the joint angle, one of which was RF regression. RF models were employed to optimise the accuracy of a machine learning model. Some of the parameters, known as hyperparameters, cannot be learned using machine learning, and they were chosen before the initial training takes place. The value of hyperparameters were adjusted over several trains and test iterations to identify the ideal model configuration, which was then utilized for analysis by machine learning algorithms. When compared to the gold standard, the results of this paper indicated a high degree of accuracy.

Most of the previous research were mainly focused on joint angle estimation using wearable sensor data [12], [26]–[28]. Because the smartwatch is a newer technology than the wearable sensor, fewer reviews of smartwatch data on joint angle estimation have been published. Besides, because the smartwatch is a wrist-mounted device, performing standard gold analysis for gait motion is a challenging. However, the measurements in lower limb motion are regarded time demanding and complex in nature. Smartwatch is more convenient than a wearable sensor for deploying mobile motion based behavioural biometrics. The advancement of the smartwatch offers long-term motion capture in an environment outside the laboratory.

Chapter 3

METHODOLOGY

3.1 Chapter Overview

The project overview method is depicted in Figure 3.1. The project is classified into four categories: smartwatch dataset, data pre-processing, machine learning model and estimation lower limb motion. At the start of the project, the acceleration dataset acquired from the smartwatch sensor is collected. Next, the dataset is pre-processed by being normalised before being fed into the model. Different machine learning models are developed for data training purposes. The data are trained to estimate the lower limb motion with the test result.

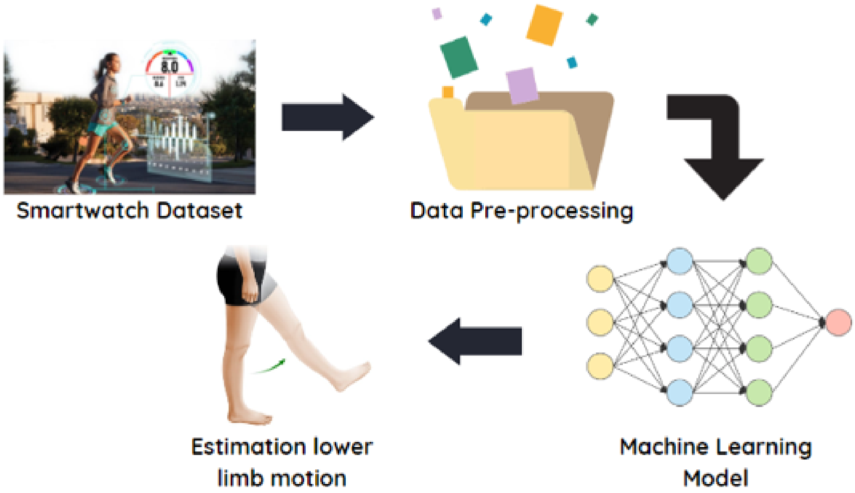


Figure 3.1: Project Workflow

3.2 Smartwatch Dataset

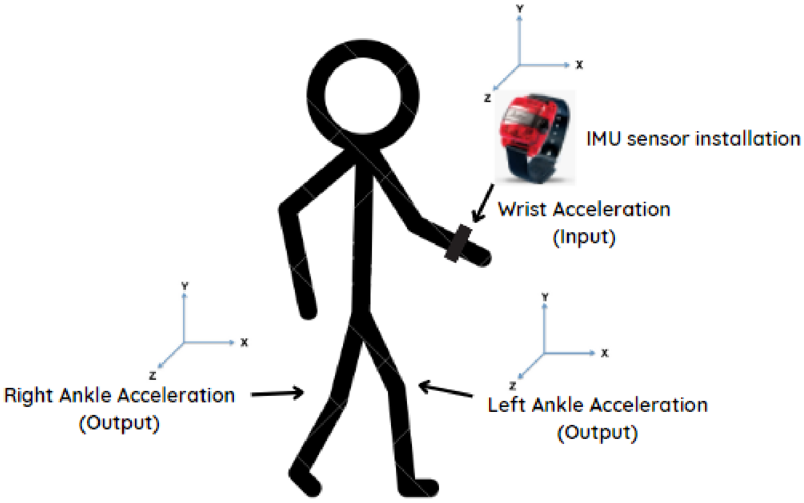
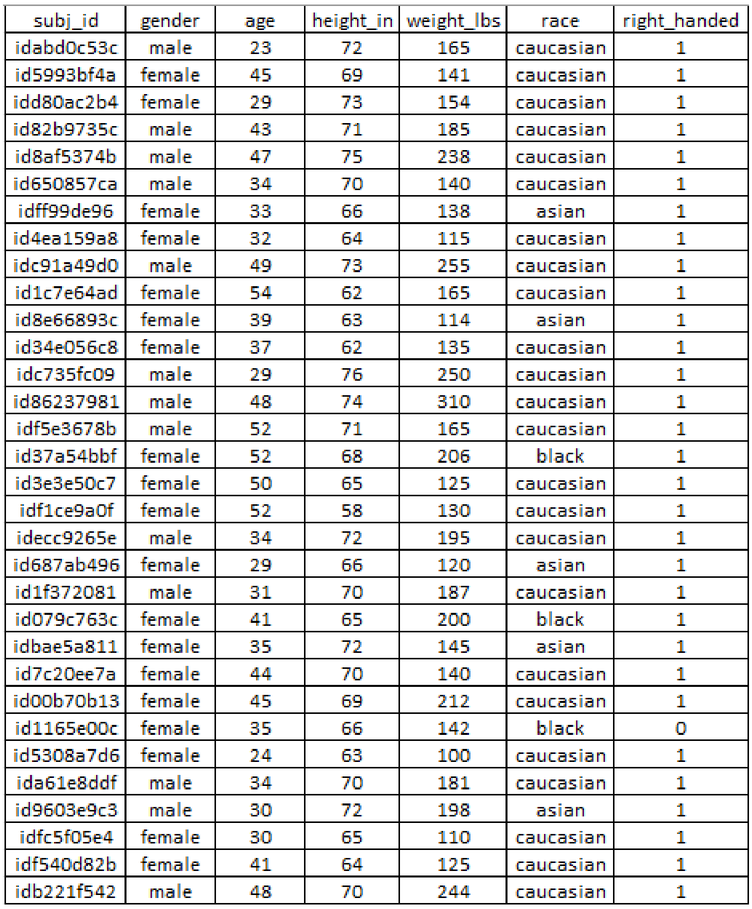


Figure 3.2: Input and Output for the project model

The acceleration data set is acquired from the publicly available database, named 'Raw Accelerometry Data' published by Marta Karas [29]. The public database consists of acceleration data acquired during the gait motion such as walking [29]. The acceleration data comprise of left wrist, left ankle and right ankle at 100Hz sampling frequency. The data set was collected from 32 healthy adults with 3-axial ActiGraph GT3+ devices and wearable accelerometers devices. These devices consist of IMU sensor and the devices were attached to their bodies. The IMU device is a wearable accelerometer device that can estimate and compute an object's specific force and angular rate of an object. This project utilized wrist acceleration as input while left and right ankle acceleration as output for the data set. Figure 3.2 shows the visualization input and output of the project, whereas Figure 3.3 indicated the information of 32 participants.



subj_id	gender	age	height_in	weight_lbs	race	right_handed
idabd0c53c	male	23	72	165	caucasian	1
id5993bf4a	female	45	69	141	caucasian	1
idd80ac2b4	female	29	73	154	caucasian	1
id82b9735c	male	43	71	185	caucasian	1
id8af5374b	male	47	75	238	caucasian	1
id650857ca	male	34	70	140	caucasian	1
idff99de96	female	33	66	138	asian	1
id4ea159a8	female	32	64	115	caucasian	1
idc91a49d0	male	49	73	255	caucasian	1
id1c7e64ad	female	54	62	165	caucasian	1
id8e66893c	female	39	63	114	asian	1
id34e056c8	female	37	62	135	caucasian	1
idc735fc09	male	29	76	250	caucasian	1
id86237981	male	48	74	310	caucasian	1
idf5e3678b	male	52	71	165	caucasian	1
id37a54bbf	female	52	68	206	black	1
id3e3e50c7	female	50	65	125	caucasian	1
idf1ce9a0f	female	52	58	130	caucasian	1
idecc9265e	male	34	72	195	caucasian	1
id687ab496	female	29	66	120	asian	1
id1f372081	male	31	70	187	caucasian	1
id079c763c	female	41	65	200	black	1
idbae5a811	female	35	72	145	asian	1
id7c20ee7a	female	44	70	140	caucasian	1
id00b70b13	female	45	69	212	caucasian	1
id1165e00c	female	35	66	142	black	0
id5308a7d6	female	24	63	100	caucasian	1
ida61e8ddf	male	34	70	181	caucasian	1
id9603e9c3	male	30	72	198	asian	1
idfc5f05e4	female	30	65	110	caucasian	1
idf540d82b	female	41	64	125	caucasian	1
idb221f542	male	48	70	244	caucasian	1

Figure 3.3: 32 participants information

3.3 Pre-processing the Data

Before the data being in the model, the data were pre-processed. Pre-processing was done in Google Colaboratory and Python. Firstly, the data was visualized to study the acceleration graph pattern. The upper part of Figure 3.4 illustrated the average graph of left wrist x-axis, y-axis and z-axis acceleration acquired from 32 participants. Meanwhile, the bottom part of Figure 3.4 illustrated the average graph of left and right ankle x-axis, y-axis and z-axis acceleration acquired from 32 participants.

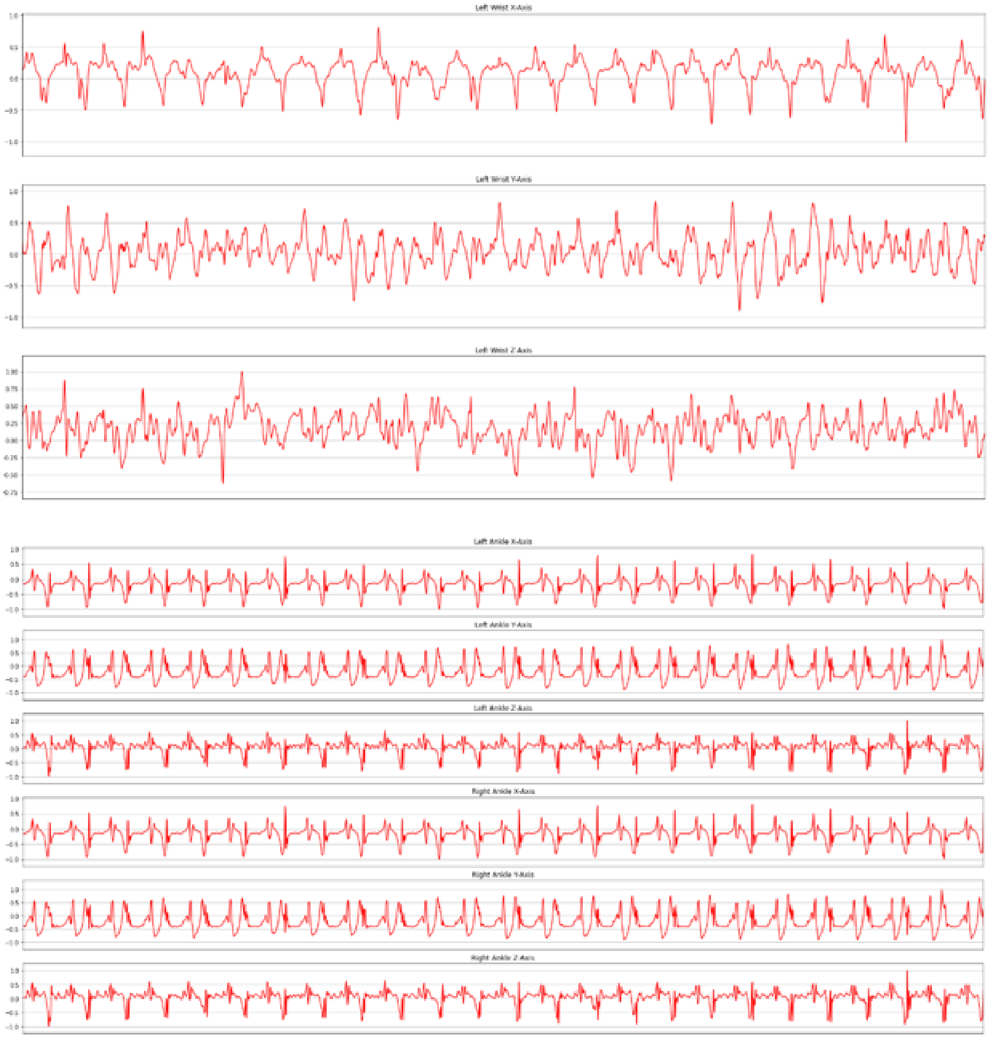


Figure 3.4: Average graph of 32 participant

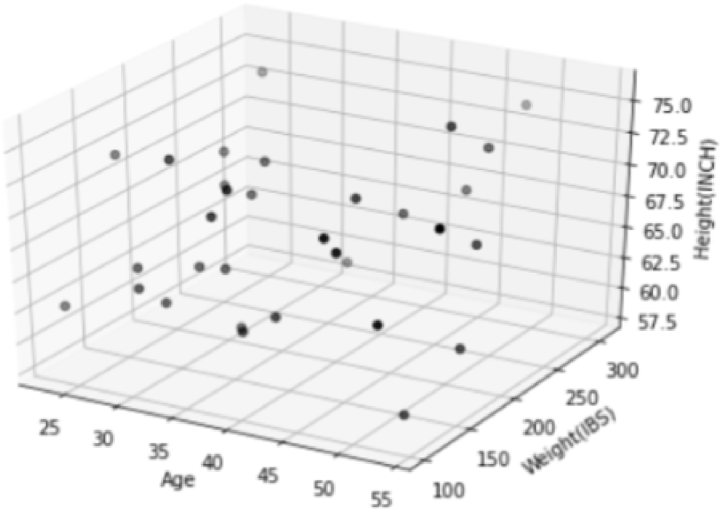


Figure 3.5: Graph of relationship between age, weight and height

Left ankle 3-axis acceleration were set as input, whereas left and right 3-axis acceleration were set as output for the model. Besides, the relationship between age, weight, and height for the 32 participants is shown in Figure 3.5.

$$x_{norm} = \frac{x - min\ x}{max\ x - min\ x} \tag{3.1}$$

not using absolute is to ensure the normalized data can have the negative value which indicating the direction

$$x_{norm} = 2\frac{x - min\ x}{max\ x - min\ x} - 1 \tag{3.2}$$

Next, the data was normalized between -1 and 1 as the dataset contained negative values. The function of data normalization is to avoid data over-fitting. The general equation of the data normalization is shown in Equation (3.1). However, for the dataset of negative values, Equation (3.2) is implemented to normalize the data between -1 and 1. The dataset was then divided into 32 chunks (32 samples). The first 26 chunks (80% dataset) are apportioned to training, and the last 6 chunks (20% dataset) are apportioned to testing.

3.4 Machine Learning Model

The machine learning model was trained using the Sequential method. Wrist acceleration was set as input (X value), while left and right ankle accelerations were set as output (y value). The target of the model outputs is salient data of left and right ankle acceleration. The initial network weights and biases were chosen at random. eLu functions were employed to activate the hidden node and calculate the desired outcome. The model keeps adjusting the weights and biases until it cannot improve model better. The model learned the relationship between the input and output on how they work. Data were taken in the ratio of 80% for training and 20% for testing. The remaining 20% of input data were taken for testing after the training using the first 80% of the data had been thoroughly trained. After testing the 20% data input, the model generated a predicted 20% data output value. The predicted output value was compared to the actual output value.

$$\omega_{t+1} = \omega_t - \eta \frac{V_t}{\sqrt{s_t + \epsilon}} g_t \tag{3.3}$$

where η is initial learning rate. g_t is gradient at time t along ω. v_t is exponential average of gradients along ω. s_t is exponential average of squares of gradient along ω [30]

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2 \tag{3.4}$$

where n is the sample size. $\hat{Y}_i$ is the i^{th} predicted value. Y_i is the i^{th} actual value.

$$Accuracy = \frac{True\ Negative + True\ Positive}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \tag{3.5}$$

This project employed Adam optimizer (Equation (3.3)) as a training algorithm to train the data and used Mean Square Error (MSE) (Equation (3.4)) to calculate the loss in training model. The accuracy algorithm that used to estimate the accuracy of training model is shown in Equation (3.5). The model training using Sequential was done in Google Colaboratory and Python. Further information of the model will be discussed in Chapters 4 to 6.

3.5 Estimation Lower Limb Motion

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y}_i - Yi)^2}{n}} \tag{3.6}$$

where n is the sample size. $\hat{Y}_i$ is the i^{th} predicted value. Y_i is the i^{th} actual value

$$r = \frac{\sum(Y_i - \bar{Y})(\hat{Y}_i - \bar{\hat{Y}})}{\sum(Y_i - \bar{Y})^2 \sum(\hat{Y}_i - \bar{\hat{Y}})^2} \tag{3.7}$$

where Y_i is the i^{th} actual value. $\hat{Y}_i$ is the i^{th} predicted value.

The network was tested with 20 percent of the dataset. RMSE and correlation coefficient were employed in this model to evaluate the differences between the predicted output and the actual output. The formula of RMSE is shown in Equation (3.6), whereas the formula of correlation coefficient is shown in Equation (3.7). Correlation coefficient evaluations are divided into 5 categorises: very weak (0 to 0.19), weak (0.20 to 0.39), moderate (0.40 to 0.59), strong (0.60 to 0.79), and very strong (0.8 to 1.0).

3.6 Chapter Summary

This chapter summarises the general overflow of the proposed machine learning models, including FFNN, LSTM, and CNNLSTM, from the beginning with data collection and ending with the estimation of lower limb acceleration. Each machine learning model will be described in the following few chapters respectively with further specific knowledge implementations.

Chapter 4

ANKLE ACCELERATION ESTIMATION USING FFNN

4.1 FFNN Implementation Ideas

The Feed Forward Neural Network (FFNN) model is a commonly used ANN model architecture. In other words, the FFNN model is a basic ANN model consisting of an input layer, an output layer, and at least one hidden layer. The most recent developments in gait tracking over a long period rely on machine learning-based estimations. Consequently, this study investigates the feasibility of FFNN for lower limb motion estimation using wrist acceleration acquired from the smartwatch sensor. Various hidden layers are implemented in the model to determine the optimal number of hidden layers for FFNN model estimation.

This study introduces the FFNN model for estimating lower limb motion from wrist acceleration. X represents the model's input, whereas y represents the model's output. k, i and j represent the input, hidden, and output node numbers respectively. Moreover, ω denotes the weight that connects between layers, whereas b denotes biases between layers. Before training the network, the network's weights and biases are set to random values. During the data training, the network transmits a combination of the weighted sum of inputs (Equation (4.1)) to all linked neurons.

$$W_i = \sum_{k=1}^{N_{IN}} X_k(n)\omega_{ik} + b_1 \tag{4.1}$$

where n is the sample number (n=1,2,3,...). N_{IN} is the number of input node. b_1 is bias for the hidden layer

$$eLu = \sigma = \alpha(exp^x - 1)\ for\, x < 0,\ eLu = \sigma = x\ \ for\, x \geq 0 \tag{4.2}$$

where eLu is Activation function. α is equal 1.0.

$$y_{ji} = \sigma(\sum_{k=1}^{N_{IN}} X_k(n)\omega_{ik} + b_1) \tag{4.3}$$

eLu function (Equation (4.2)) is employed as the activation function to allow the network to learn non-linear pattern.The network computes the signal with an activation function to activate the neurons can be expressed as Equation (4.3).

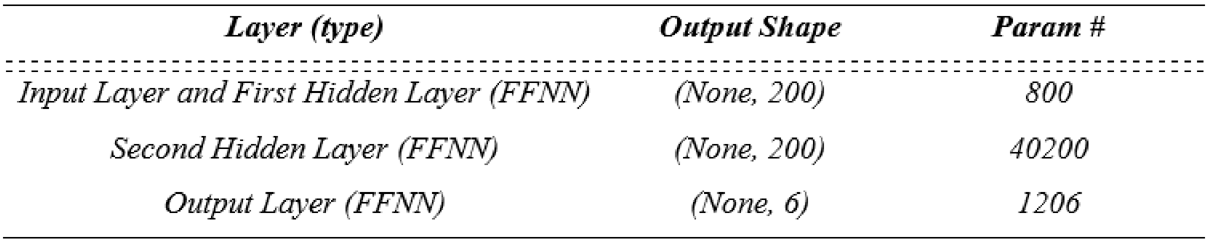
$$y(n) = \sum_{j=1}^{N_{ON}} y_{ji}(n)\omega_{ji} + b_2 \tag{4.4}$$

where N_{ON} is the number of output node. b_2 is bias for the output layer

At the output layer, the weighted sum of all hidden neurons are transferred by (Equation (4.4)). Adam optimization (Equation 3.3)is employed to improve the weights of the network by adjusting network weights and biases. MSE (Equation 3.4) is used as loss function to estimate the performance of the optimizer. Therefore, this metric is minimized during the data training. The model stops learning until obtains the final output with possible minor errors or reaches 20 epochs.

This study uses 32 samples, which are divided as follows: 60% for training, 20% for validation, and 20% for testing.

4.2 FFNN Model with 2 Hidden Layers



Layer (type)	*Output Shape*	*Param #*
Input Layer and First Hidden Layer (FFNN)	*(None, 200)*	*800*
Second Hidden Layer (FFNN)	*(None, 200)*	*40200*
Output Layer (FFNN)	*(None, 6)*	*1206*

Figure 4.1: Layer Structure of FFNN Model with 2 Hidden Layers

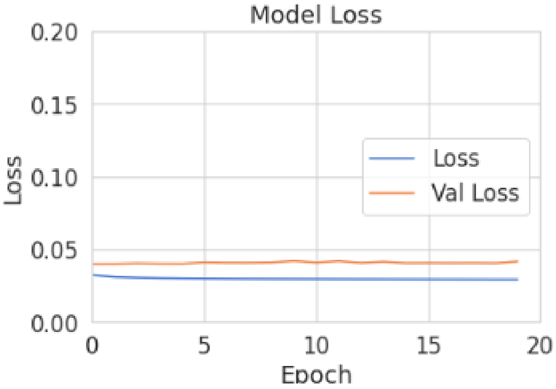


Figure 4.2: Loss Graph for Training FFNN Model with 2 Hidden Layers

The layer structure of the 2 hidden layers is shown in Figure 4.1. This model consists of 4 layers, which are divided as follows: input layer with 3 nodes, first hidden layers with 200 nodes, second hidden layers with 200 nodes and output layer with 6 nodes. Figure 4.2 depicts the training and validation loss curve during the data training. The training loss is 0.0293, whereas the validation loss is 0.0417.

Table 4.1: Mean RMSE with Standard Deviation and Correlation Coefficient Results for FFNN Model with 2 Hidden Layers

Model Outputs	Left Ankle Acceleration			Right Ankle Acceleration		
	x	y	z	x	y	z
Mean RMSE with standard deviation	1.916 (±0.58)	2.461 (±0.82)	1.123 (±0.32)	1.739 (±0.50)	2.369 (±0.55)	1.803 (±0.28)
Correlation Coefficient	0.423	0.797	0.457	0.496	0.421	0.410

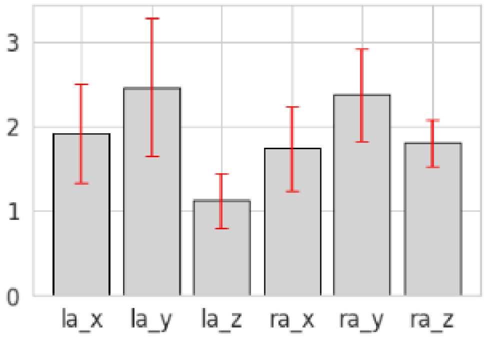
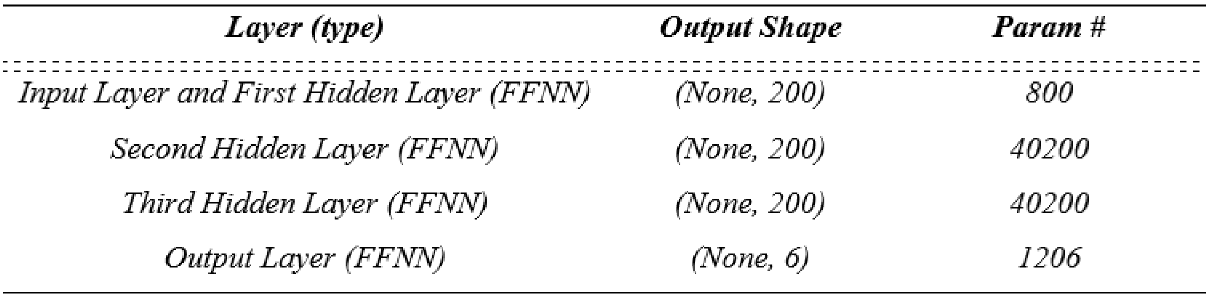


Figure 4.3: Bar Chart of RMSE with The Error Bar for 2 Hidden Layers FFNN Model

Table 5.2 indicated the RMSE, standard deviation, and correlation coefficient results for the six outputs. Besides, Figure 4.3 illustrates the RMSE result with error bar in bar chart. In this model, lower limb motion can be predicted with a mean (±STD) RMSE of less than 1.916° (±0.58°) in left ankle acceleration x-axis, 2.461° (±0.82°) in y-axis, 1.123° (±0.32°) in z-axis, 1.739° (±0.5°) in right ankle acceleration x-axis, 2.369° (±0.55°) in y-axis, and 1.803° (±0.28°) in z-axis across the walking trials. Besides, the model predicted moderate correlation coefficient for 5 outputs (left ankle x-axis, z-axis and right ankle x-axis, y-axis, z-axis) and strong correlation coefficient for 1 output (left ankle y-axis).

4.3 FFNN Model with 3 Hidden Layers



Layer (type)	***Output Shape***	***Param #***
Input Layer and First Hidden Layer (FFNN)	*(None, 200)*	*800*
Second Hidden Layer (FFNN)	*(None, 200)*	*40200*
Third Hidden Layer (FFNN)	*(None, 200)*	*40200*
Output Layer (FFNN)	*(None, 6)*	*1206*

Figure 4.4: Layer Structure of FFNN Model with 3 Hidden Layers

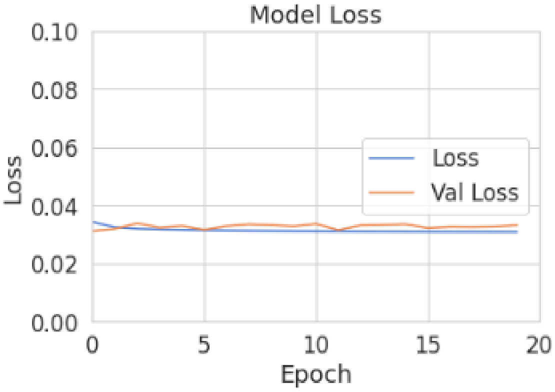


Figure 4.5: Loss Graph for Training FFNN Model with 3 Hidden Layers

Figure 4.4 illustrated the three hidden layers structure. This model comprises of 5 layers, including input layer with 3 nodes, first hidden layers with 200 nodes, second hidden layers with 200 nodes, third hidden layer with 200 nodes and output layer with 6 nodes. In addition, the training and validation loss curves for the data training are shown in Figure 4.5. The training loss is 0.0308, while the validation loss is 0.0331.

Table 4.2: Mean RMSE with Standard Deviation and Correlation Coefficient Results for FFNN Model with 3 Hidden Layers

Model Outputs	Left Ankle Acceleration			Right Ankle Acceleration		
	x	y	z	x	y	z
Mean RMSE with standard deviation	2.086 (±0.58)	2.114 (±0.82)	1.860 (±0.30)	1.910 (±0.89)	3.296 (±0.99)	2.215 (±0.28)
Correlation Coefficient	0.437	0.537	0.436	0.468	0.468	0.402

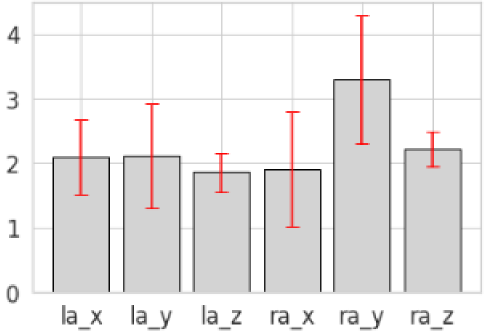
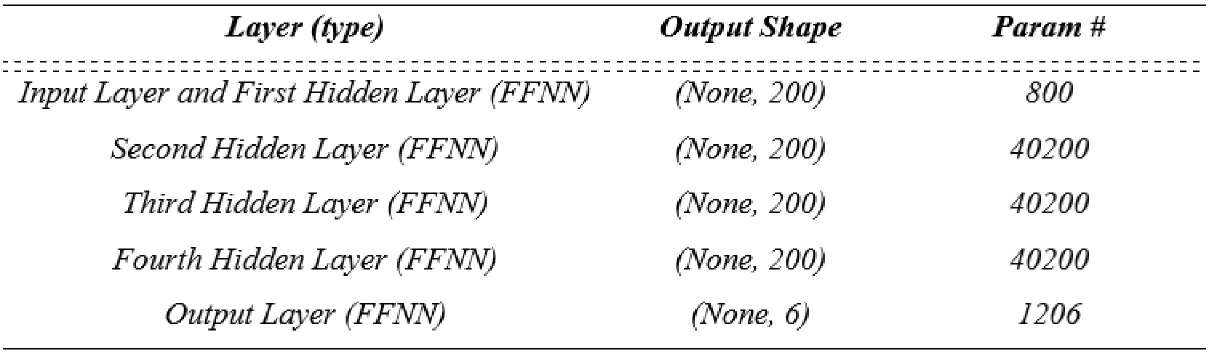


Figure 4.6: Bar Chart of RMSE with The Error Bar for 3 Hidden Layers FFNN Model

Table 4.2 depicted the RMSE, standard deviation, and correlation coefficient results for the six outputs. Furthermore, Figure 4.6 illustrates the RMSE result with error bar in bar chart. In this model, lower limb motion can be predicted with a mean (±STD) RMSE of less than 2.086° (±0.58°) in left ankle acceleration x-axis, 2.114° (±0.82°) in y-axis, 1.86° (±0.3°) in z-axis, 1.910° (±0.89°) in right ankle acceleration x-axis, 3.296° (±0.99°) in y-axis, and 2.215° (±0.28°) in z-axis across the walking trials. On the other hand, the model estimated moderate correlation coefficient for 6 outputs (left ankle x-axis, y-axis, z-axis and right ankle x-axis, y-axis, z-axis).

4.4 FFNN model with 4 Hidden Layers



Layer (type)	*Output Shape*	*Param #*
Input Layer and First Hidden Layer (FFNN)	*(None, 200)*	*800*
Second Hidden Layer (FFNN)	*(None, 200)*	*40200*
Third Hidden Layer (FFNN)	*(None, 200)*	*40200*
Fourth Hidden Layer (FFNN)	*(None, 200)*	*40200*
Output Layer (FFNN)	*(None, 6)*	*1206*

Figure 4.7: Layer Structure of FFNN Model with 4 Hidden Layers

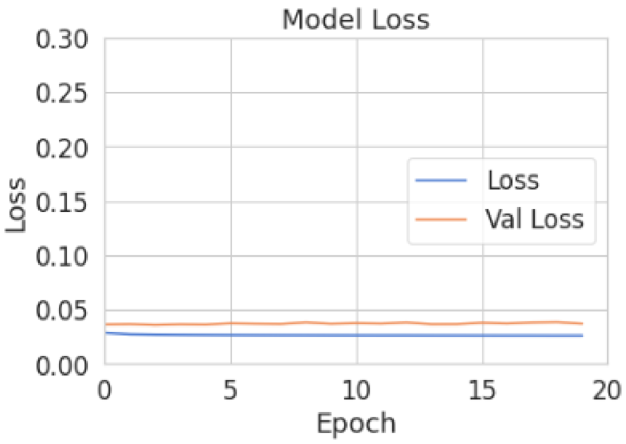


Figure 4.8: Loss Graph for Training FFNN Model with 4 Hidden Layers

The layer structure of the 4 hidden layers is shown in Figure 4.7. This model consists of 6 layers, which are divided as follows: input layer with 3 nodes, first hidden layers with 200 nodes, second hidden layers with 200 nodes, third hidden layers with 200 nodes, fourth hidden layers with 200 nodes and output layer with 6 nodes. Figure 4.8 depicts the training and validation loss curve during the data training. The training loss is 0.0293, whereas the validation loss is 0.0417.

Table 4.3: Mean RMSE with Standard Deviation and Correlation Coefficient Results for FFNN Model with 4 Hidden Layers

Model Outputs	Left Ankle Acceleration			Right Ankle Acceleration		
	x	y	z	x	y	z
Mean RMSE with standard deviation	2.54 (±0.76)	2.73 (±0.90)	2.22 (±0.37)	2.31 (±0.55)	3.05 (±0.86)	2.38 (±0.43)
Correlation Coefficient	0.461	0.517	0.471	0.455	0.556	0.478

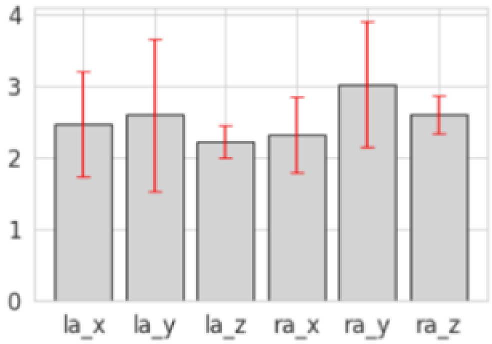


Figure 4.9: Bar Chart of RMSE with The Error Bar for 4 Hidden Layers FFNN Model

Table 4.3 indicated the RMSE, standard deviation, and correlation coefficient results for the six outputs. Besides, Figure 4.9 depicted the RMSE result with error bar in bar chart. In this model, lower limb motion can be predicted with a mean (±STD) RMSE of less than 2.54° (±0.76°) in left ankle acceleration x-axis, 2.73° (±0.90°) in y-axis, 2.22° (±0.37°) in z-axis, 2.31° (±0.55°) in right ankle acceleration x-axis, 3.05° (±0.86°) in y-axis, and 2.38° (±0.43°) in z-axis across the walking trials. Besides, the model predicted moderate correlation coefficient for 6 outputs (left ankle x-axis, y-axis, z-axis and right ankle x-axis, y-axis, z-axis).

4.5 Chapter Summary

For the FFNN models, the average for two hidden layers of FFNN is 1.96°, and the correlation coefficient is greater than 0.41. Three hidden layers FFNN has an average RMSE of 2.24°, and the correlation coefficient is greater than 0.402. FFNN with four hidden layers has an average RMSE of 2.54°, and correlation coefficient is greater than 0.455. Two hidden layers of FFNN are the best model compared to the other two FFNN models. The models may have too many neurons in the hidden layers, so FFNN with more layers gives worse results. Too many neurons in the hidden layers lead to an issue of over-fitting. Consequently, two hidden layers of FFNN are adequate for the project model predictions.

Chapter 5

ANKLE ACCELERATION ESTIMATION USING LSTM

5.1 LSTM Implementation Ideas

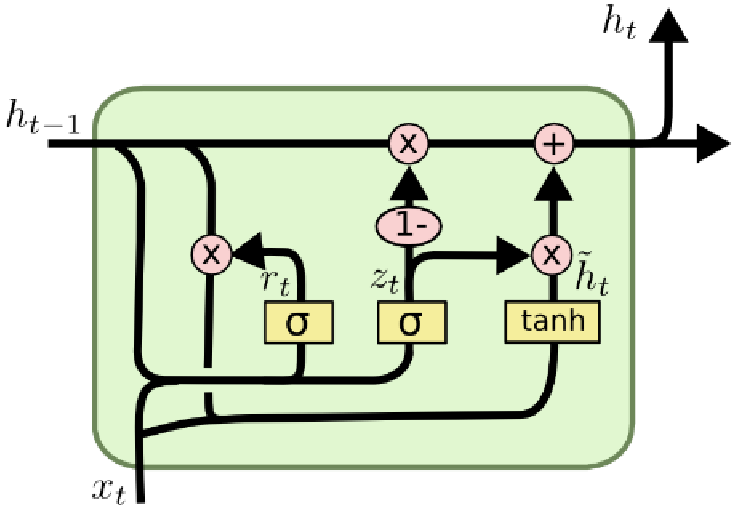


Figure 5.1: LSTM Structure

Long short-term memory (LSTM) networks are the upgrade version of RNN that increase memory capacity. LSTM is employed to construct the layers of an RNN. LSTMs assign ”weights” to data, enabling RNNs to either admit new information, forget information, or give adequate significance to influence the output.

The first stage determines whether the information from the previous timestamp should be remembered or if it is irrelevant and should be discarded. In the second stage, the cell attempts to acquire new information from the input data it has received. In the last stage, the cell transmits the latest information from the present timestamp to the subsequent timestamp. These three LSTM cell stages are known as gates. The first stage is known as the Forget gate, the second as the Input gate, and the third as the Output gate.

The initial step in a cell of an LSTM network is to determine whether information from the initial timestamp should be retained or discarded. The formula of Forget gate is shown in Equation (5.1).

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{5.1}$$

where x_t is input to the current timestamp. b_f is the weight associated with the input. h_{t-1} is the hidden state of the previous timestamp. W_f is the weight matrix associated with hidden state.

Input gate is utilised to measure the significance of new information carried by input. The formula of input gate are shown in Equation (5.2) and Equation (5.3).

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t + b_i]) \tag{5.2}$$

where x_t is the input at the current timestamp, t. b_i is the weight matrix of input. h_{t-1} is the hidden state at the previous timestamp. W_i is the weight matrix of input associated with hidden state.

$$\tilde{C}_t = tanh(W_C \bullet [h_{t-1}, x_t] + b_C) \tag{5.3}$$

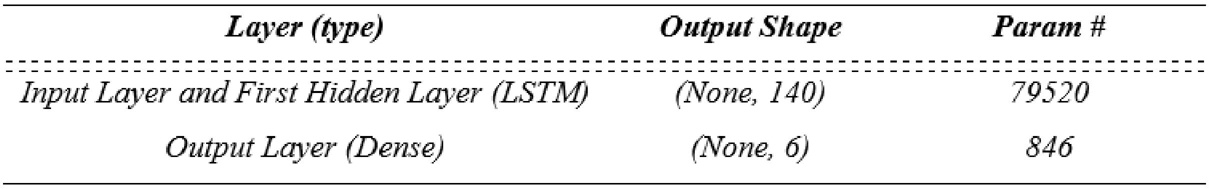
The output gate determines what portion of the current cell is output. The formula of Output gate is shown in Equation (5.4) and Equation (5.5).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5.4}$$

where b_o is the weight associated with the output. W_f is the weight matrix associated with output state.

$$h_t = o_t * tanh(C_t) \tag{5.5}$$

5.2 LSTM Model with 1 Hidden Layer



Layer (type)	***Output Shape***	***Param #***
Input Layer and First Hidden Layer (LSTM)	*(None, 140)*	*79520*
Output Layer (Dense)	*(None, 6)*	*846*

Figure 5.2: Layer Structure of LSTM Model with 1 Hidden Layer

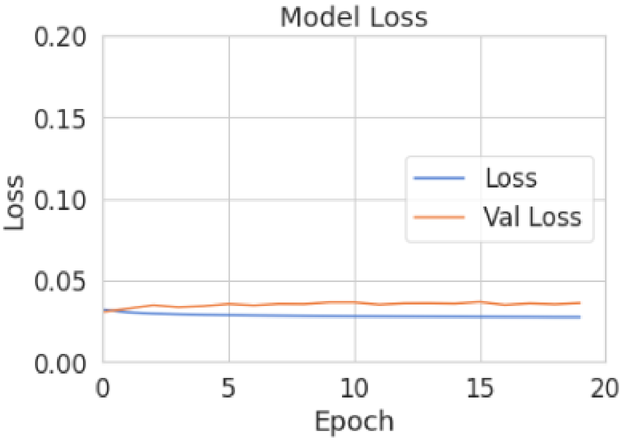


Figure 5.3: Loss Graph for Training LSTM Model with 1 Hidden Layer

The layer structure of the 1 hidden layer is shown in Figure 5.2. This model consists of 3 layers, which are divided as follows: input layer with 3 nodes, first hidden layers with 140 nodes and output layer with 6 nodes. Figure 5.3 depicts the training and validation loss curve during the data training. The training loss is 0.0274, whereas the validation loss is 0.0360.

Table 5.1: Mean RMSE with Standard Deviation and Correlation Coefficient Results for LSTM Model with 1 Hidden Layers

Model Outputs	Left Ankle Acceleration			Right Ankle Acceleration		
	x	y	z	x	y	z
Mean RMSE with standard deviation	1.177 (±0.58)	1.923 (±0.91)	0.357 (±0.18)	1.101 (±0.55)	1.618 (±0.58)	0.429 (±0.22)
Correlation Coefficient	0.423	0.456	0.429	0.409	0.426	0.469

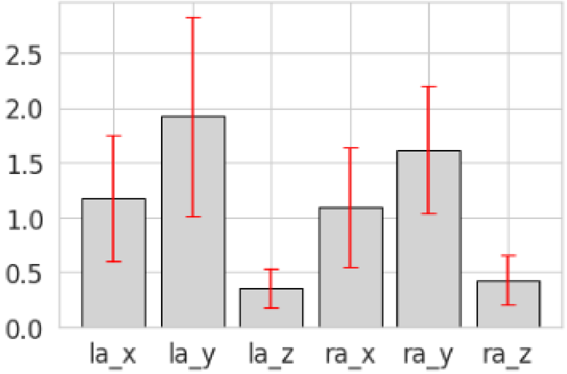
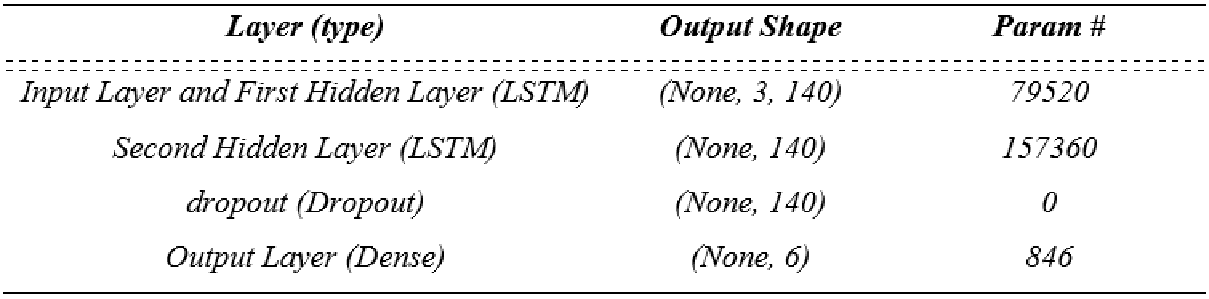


Figure 5.4: Bar Chart of RMSE with The Error Bar for 1 Hidden Layer LSTM Model

Table 5.1 indicated the RMSE, standard deviation, and correlation coefficient results for the six outputs. Besides, Figure 5.4 depicted the RMSE result with error bar in bar chart. In this model, lower limb motion can be predicted with a mean (±STD) RMSE of less than 1.177° (±0.58°) in left ankle acceleration x-axis, 1.923° (±0.91°) in y-axis, 0.357° (±0.18°) in z-axis, 1.101° (±0.55°) in right ankle acceleration x-axis, 1.618° (±0.58°) in y-axis, and 0.429° (±0.22°) in z-axis across the walking trials. In addition, the model predicted moderate correlation coefficient for 6 outputs.

5.3 LSTM Model with 2 Hidden Layers



Layer (type)	*Output Shape*	*Param #*
Input Layer and First Hidden Layer (LSTM)	*(None, 3, 140)*	*79520*
Second Hidden Layer (LSTM)	*(None, 140)*	*157360*
dropout (Dropout)	*(None, 140)*	*0*
Output Layer (Dense)	*(None, 6)*	*846*

Figure 5.5: Layer Structure of LSTM Model with 2 Hidden Layers

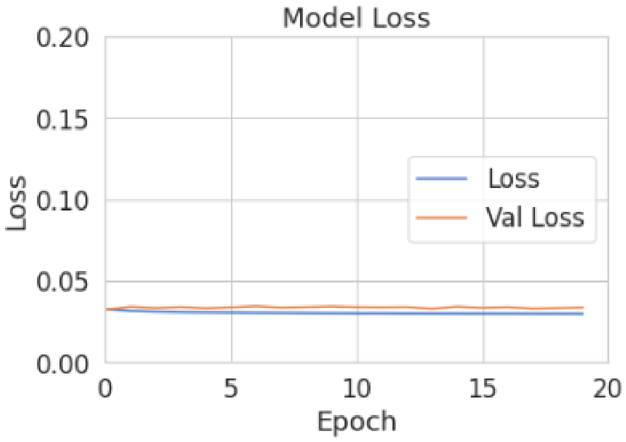


Figure 5.6: Loss Graph for Training LSTM Model with 2 Hidden Layers

Figure 5.5 illustrated the two hidden layers structure. This model comprises of 4 layers, including input layer with 3 nodes, first hidden layers with 140 nodes, second hidden layers with 140 nodes and output layer with 6 nodes. Dropout is used after second layer with 0.5 to avoid over-fitting. In addition, the training and validation loss curves for the data training are shown in Figure 5.6. The training loss is 0.0297, while the validation loss is 0.0335.

Table 5.2: Mean RMSE with Standard Deviation and Correlation Coefficient Results for LSTM Model with 2 Hidden Layers

Model Outputs	Left Ankle Acceleration			Right Ankle Acceleration		
	x	y	z	x	y	z
Mean RMSE with standard deviation	1.14 (±0.59)	1.67 (±0.82)	0.54 (±0.27)	1.10 (±0.54)	2.05 (±0.78)	0.55 (±0.27)
Correlation Coefficient	0.506	0.481	0.438	0.461	0.519	0.405

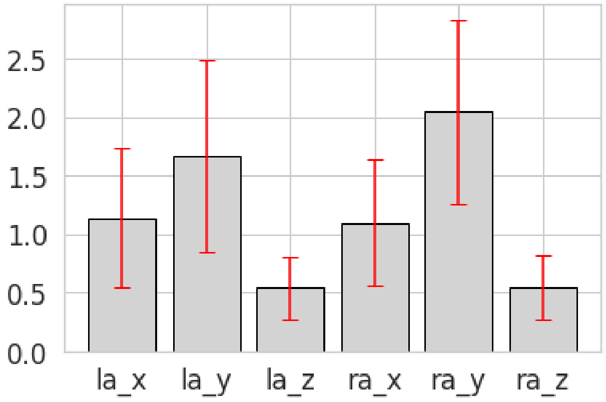
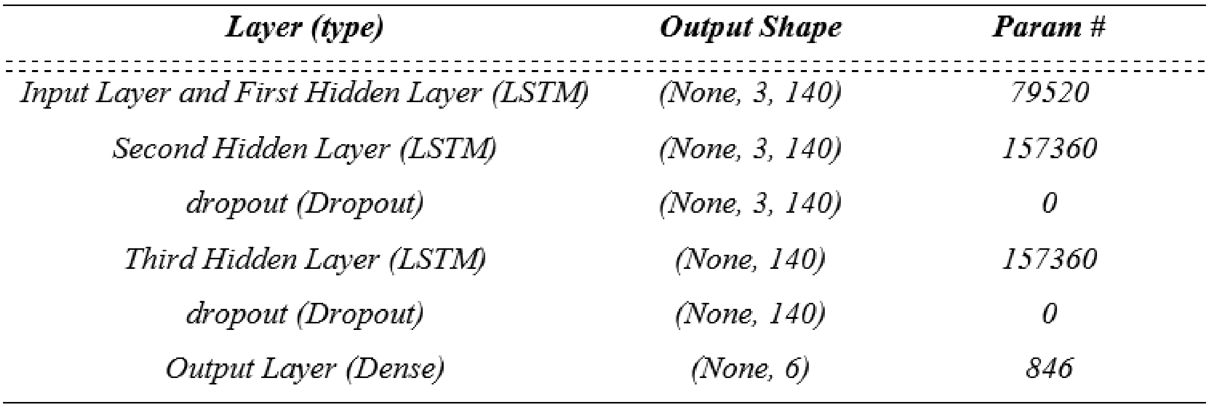


Figure 5.7: Bar Chart of RMSE with The Error Bar for 2 Hidden Layers LSTM Model

Table 5.2 depicted the RMSE, standard deviation, and correlation coefficient results for the six outputs. Moreover, Figure 5.7 illustrates the RMSE result with error bar in bar chart. In this model, lower limb motion can be estimated with a mean (±STD) RMSE of less than 1.14° (±0.59°) in left ankle acceleration x-axis, 1.67° (±0.82°) in y-axis, 0.54° (±0.27°) in z-axis, 1.1° (±0.54°) in right ankle acceleration x-axis, 2.05° (±0.78°) in y-axis, and 0.55° (±0.27°) in z-axis across the walking trials. Furthermore, the model estimated moderate correlation coefficient for 6 outputs.

5.4 LSTM Model with 3 Hidden Layers



Layer (type)	***Output Shape***	***Param #***
Input Layer and First Hidden Layer (LSTM)	*(None, 3, 140)*	*79520*
Second Hidden Layer (LSTM)	*(None, 3, 140)*	*157360*
dropout (Dropout)	*(None, 3, 140)*	*0*
Third Hidden Layer (LSTM)	*(None, 140)*	*157360*
dropout (Dropout)	*(None, 140)*	*0*
Output Layer (Dense)	*(None, 6)*	*846*

Figure 5.8: Layer Structure of LSTM Model with 3 Hidden Layers

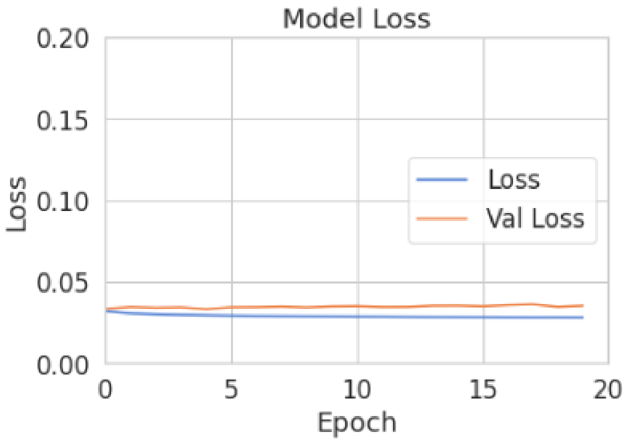


Figure 5.9: Loss Graph for Training LSTM Model with 3 Hidden Layers

The layer structure of the 3 hidden layers is shown in Figure 5.8. This model consists of 5 layers, which are divided as follows: input layer with 3 nodes, first hidden layers with 140 nodes, second hidden layers with 140 nodes, third hidden layers with 140 nodes and output layer with 6 nodes. Dropout were implemented after second hidden layer and third hidden layer with 0.5 and 0.2. Figure 5.9 depicts the training and validation loss curve during the data training. The training loss is 0.0282, whereas the validation loss is 0.0352.

Table 5.3: Mean RMSE with Standard Deviation and Correlation Coefficient Results for LSTM Model with 3 Hidden Layers

Model Outputs	Left Ankle Acceleration			Right Ankle Acceleration		
	x	y	z	x	y	z
Mean RMSE with standard deviation	1.12 (±0.53)	1.74 (±0.84)	0.41 (±0.20)	1.03 (±0.51)	1.35 (±0.61)	0.48 (±0.24)
Correlation Coefficient	0.487	0.469	0.418	0.428	0.568	0.410

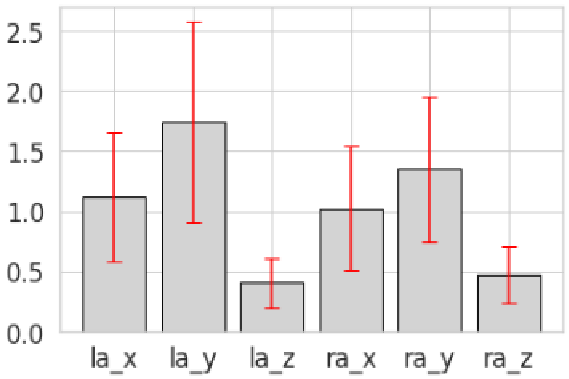


Figure 5.10: Bar Chart of RMSE with The Error Bar for 3 Hidden Layers LSTM Model

Table 5.3 indicated the RMSE, standard deviation, and correlation coefficient results for the six outputs. Besides, Figure 5.10 depicted the RMSE result with error bar in bar chart. In this model, lower limb motion can be estimated with a mean (±STD) RMSE of less than 1.12° (±0.53°) in left ankle acceleration x-axis, 1.74° (±0.84°) in y-axis, 0.41° (±0.20°) in z-axis, 1.03° (±0.51°) in right ankle acceleration x-axis, 1.35° (±0.61°) in y-axis, and 0.48° (±0.24°) in z-axis across the walking trials. Moreover, the model predicted moderate correlation coefficient for 6 outputs.

5.5 Chapter Summary

One hidden layer LSTM has the average RMSE of 1.1° and the correlation coefficient is more than 0.409. Two hidden layers LSTM has the average RMSE of 1.175° and the correlation coefficient is more than 0.405. Three hidden layers LSTM has the average RMSE of 1.02° and the correlation coefficient is more than 0.410. Among the three models, 3 hidden layers LSTM has the best result compared to other two models. This may due to neither one hidden layer LSTM nor two hidden layers LSTM have the sufficient neurons in the layer for training. Therefore, three hidden layers LSTM is the best model among the three LSTM models.

Chapter 6

ANKLE ACCELERATION ESTIMATION USING CNN-LSTM

6.1 CNNLSTM Implementation Ideas

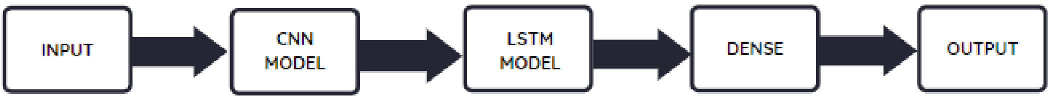
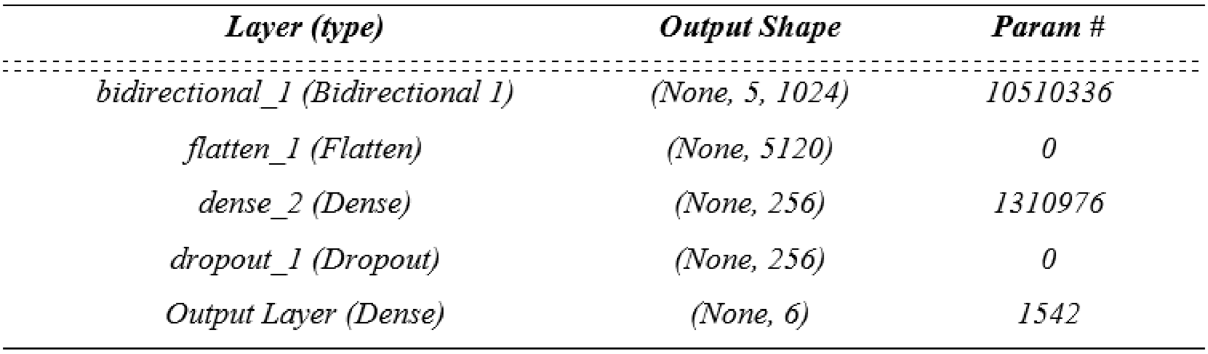


Figure 6.1: Structure of CNNLSTM Model

A Convolutional Neural Network Long Short-Term Memory (CNNLSTM) is constructed by first defining CNN layers, followed by LSTM layers, and a dense layer on the output. CNNLSTM architecture is modifiable based on the type and parameter settings of the network's constituent layers. CNNLSTM comprises a convolutional layer, a pooling layer, an LSTM layer, and a Dense layer. Each layer may change the number of filters, kernel size, and several strides. Depending on the features of the learning data, modifying these parameters might impact learning speed and performance. The sliding window technique is employed to preprocess the input data, a multivariate time series. The size of the kernel is adjusted to keep the loss of temporal information to a minimum. The data then transmits into the LSTM after passing through the convolution and pooling layers. The transmitted data becomes the input data of the LSTM layer. Regarding the LSTM layer, the following implementation is similar to LSTM (Refer to Chapter 5).



Layer (type)	***Output Shape***	***Param #***
bidirectional_1 (Bidirectional 1)	*(None, 5, 1024)*	*10510336*
flatten_1 (Flatten)	*(None, 5120)*	*0*
dense_2 (Dense)	*(None, 256)*	*1310976*
dropout_1 (Dropout)	*(None, 256)*	*0*
Output Layer (Dense)	*(None, 6)*	*1542*

Figure 6.2: Layer Structure of CNNLSTM Model

The designed layer structure of CNNLSTM is illustrated in Figure 6.2. In this model, the convolution layer's filter number was set to 512, the kernel size was set to 5, the number of strides was set to 1, the dropout was set to 0.2, and the eLu function was employed. Flatten layer was used to convert multi-dimensional arrays into a one-dimensional array. The information from a one-dimensional array is then flattened and sent to a classification model. The most important reason that flatten was employed before processing or feeding the input is that multi-dimensional arrays take up more memory than one-dimensional arrays. Afterwards, the subsequent layer was set at 256 nodes, followed by a dropout layer was adjusted to 0.3 to prevent over-fitting. The output layer was configured with 6 nodes as the model had 6 outputs. Adam optimizer was used in this model with 0.001 learning rate. The learning process was terminated when the model loss attained its smallest possible value.

6.2 CNNLSTM Result and Discussion

Table 6.1: CNNLSTM Training and Validation Result

Metrics	Training Result	Validation Result
MSE	0.0025	0.0099
Accuracy	91.10%	82.22%

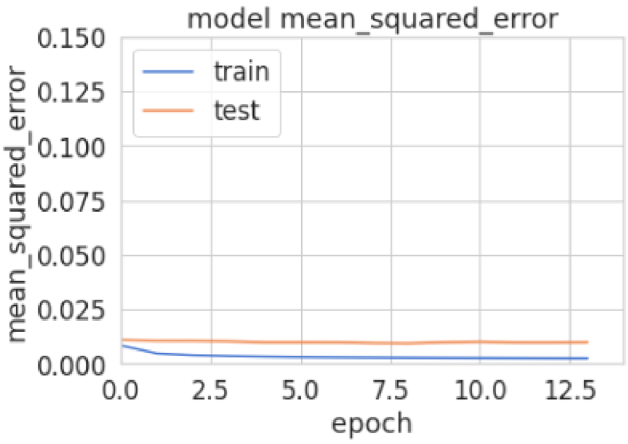


Figure 6.3: Loss Graph for Training CNNLSTM Model

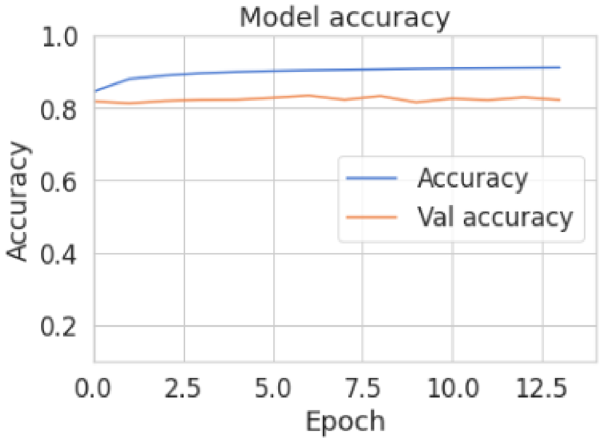


Figure 6.4: Accuracy Graph for Training CNNLSTM Model

Table 6.1 depictde MSE and accuracy values for the data training for CNNLSTM model. Figure 6.3 indicated the model loss against the epochs during data training, whereas Figure 6.4 indicated the model accuracy against epochs during data training. As seen in the Figure 6.3, when the number of epochs increases, the model loss using Equation(3.4) decreases. In contrast to the model loss graph, the accuracy curve rises when the number of epochs increases.

Table 6.2: Mean RMSE with Standard Deviation and Correlation Coefficient Results for CNNLSTM Model

Model Outputs	Left Ankle Acceleration			Right Ankle Acceleration		
	x	y	z	x	y	z
Mean RMSE with standard deviation	0.36 (±0.11)	0.22 (±0.08)	0.20 (±0.06)	0.29 (±0.11)	0.20 (±0.09)	0.23 (±0.07)
Correlation Coefficient	0.949	0.992	0.953	0.968	0.993	0.929

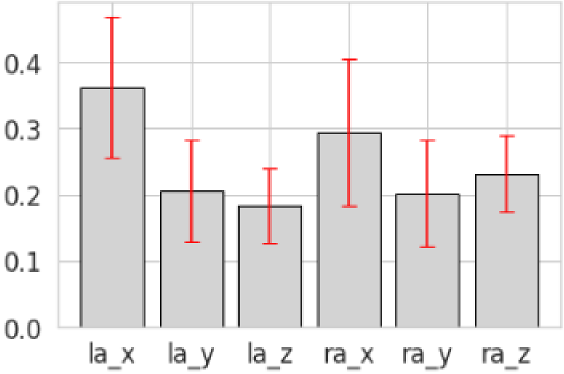


Figure 6.5: Bar Chart of RMSE with The Error Bar for CNNLSTM Model

Table 6.2 depicted the RMSE, standard deviation, and correlation coefficient results for the six outputs. Besides, Figure 6.5 illustrates the RMSE result with error bar in bar chart. In this model, lower limb motion can be estimated with a mean (±STD) RMSE of less than 0.36° (±0.11°) in left ankle acceleration x-axis, less than 0.22° (±0.08°) in y-axis, 0.20° (±0.06°) in z-axis, 0.29° (±0.11°) in right ankle acceleration x-axis, 0.2° (±0.09°) in y-axis, and 0.23° (±0.07°) in z-axis across the walking trials. Furthermore, the model estimated a very strong correlation coefficient for 6 outputs, which means the model can predict the lower limb motion with a correlation coefficient with $p > 0.929$.

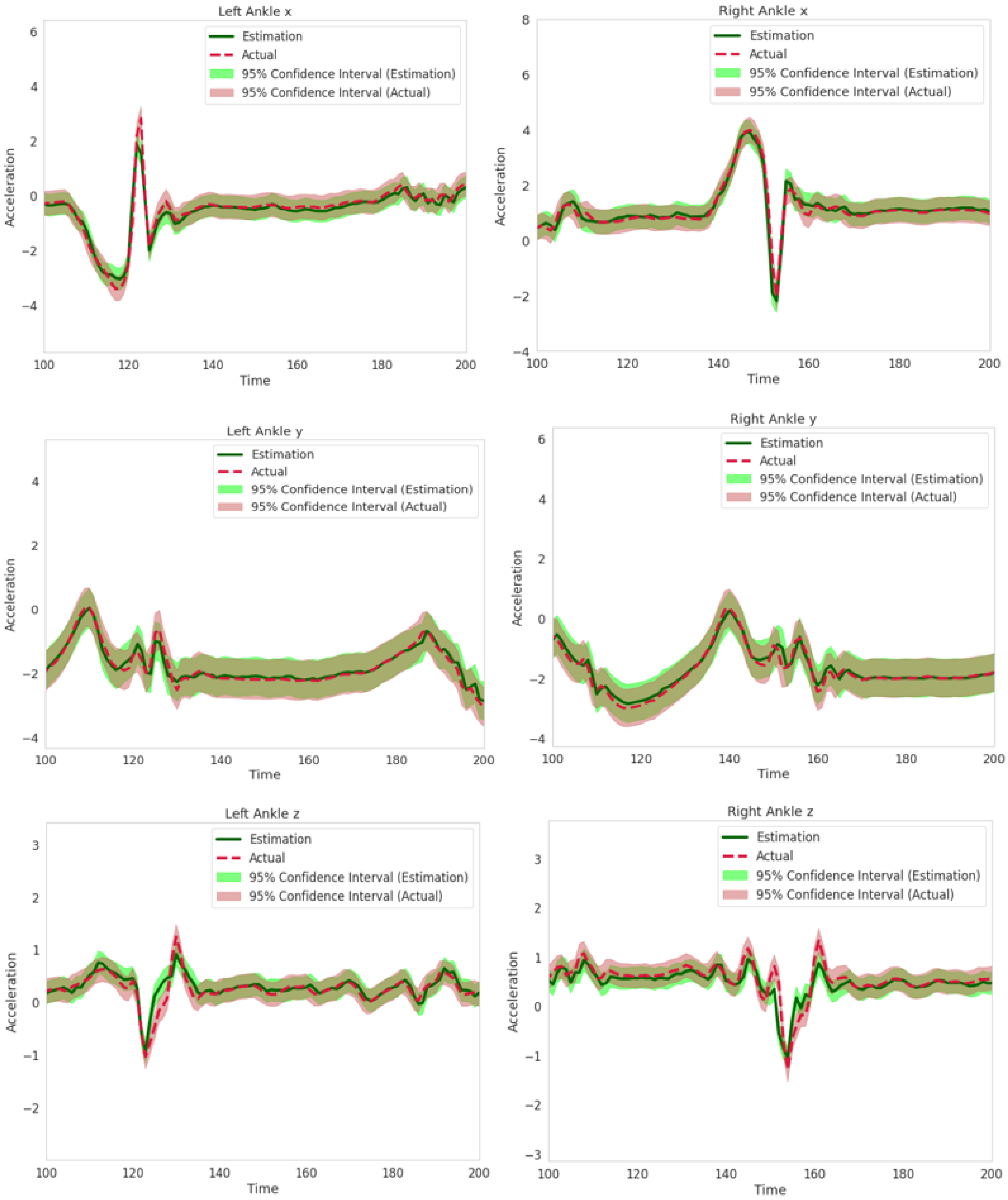


Figure 6.6: CNNLSTM Estimation Result for Model Outputs

95% confidence interval was used to estimate the model output as shown in Figure 6.6. 95% confidence interval means there is 5% chance of being wrong. Dark green line represented the output estimation from the model, whereas the dark red dotted line represented the actual dataset. Light green highlight is the 95% confidence interval for the output estimation, while light red highlight is the 95% confidence interval for the actual dataset. Overall, the machine learning model was able to predict the outputs as the graph of estimation is almost aligned with the actual. However, in the left ankle z-axis graph, the estimation plot rises between 120 and 140 seconds compared to actual plot. In addition, in right ankle z-axis graph, there is a certain difference between the estimation and actual plots between 140 and 160 seconds.

6.3 Chapter Summary

CNNLSTM has the average RMSE of 0.25° and the correlation coefficient is more than 0.929, which mean CNNLSTM has a very strong correlation coefficient. Among the three different machine learning in Chapters 4 to 6, CNNLSTM is the best suites machine learning to estimate the lower limb motion. It has the lowest model loss with 0.0099, highest accuracy with 82.22%, lowest mean RMSE with 0.25°, and correlation coefficient is more than 0.929.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In conclusion, the smartwatch possesses a powerful sensor, which provides an easily accessible platform for implementing and deploying mobile motion-based behavioural biometrics. A smartwatch consists of an IMU sensor, which can assess the lower limb motion of the smartwatch user when they are wearing the smartwatch. In addition, we investigate the feasibility of gait analysis on a wrist-worn device, such as a smartwatch. Furthermore, the comparison of the machine learning method is described in conjunction with an accurate evaluation of the gait estimation. Moreover, this research looked at various sensor types for gait estimation. As the dependence on technology increases over the year, it is potential for future direction on using smartwatch data to estimate gait motion.

Besides, this study implemented the FFNN, LSTM and CNNLSTM models. The network is trained using salient features and targets chosen based on a priori knowledge of the wrist acceleration associated with gait motion. As a result, the best FFNN model estimated ankle acceleration with $p > 0.410$ and

RMSE:1.916°±0.58% in left ankle acceleration x-axis, 2.461°±0.82% in y-axis, 1.123°±0.32% in z-axis, 1.739°±0.50% in right ankle acceleration x-axis, 2.369°± 0.55% in y-axis, 1.803°±0.28% in z-axis. Furthermore, the best LSTM model predicted ankle acceleration with $p > 0.41$ and RMSE:1.12°±0.53% in left ankle acceleration x-axis, 1.74°±0.84% in y-axis, 0.41°±0.20% in z-axis, 1.03°±0.51% in right ankle acceleration x-axis, 1.35°±0.61% in y-axis, 0.48°±0.24% in z-axis. On the other hand, CNNLSTM accurately predicted ankle acceleration with $p > 0.929$ and RMSE:0.36°±0.11% in left ankle acceleration x-axis, 0.22°±0.08% in y-axis, 0.20°±0.06% in z-axis, 0.29°±0.11% in right ankle acceleration x-axis, 0.2°±0.09% in y-axis, 0.23°±0.07% in z-axis. Among the three machine learning, CNNLSTM is the most suitable machine learning for predicting gait motion from smartwatch data.

7.2 Future Work

Future research may implement feature selection (such as PCA and RF algorithms) to eliminate redundant data during data preprocessing. As a result, it will minimise the time required for data training and improve network performance. Additionally, feature selection also reduces the data frame to prevent overfitting. In addition, various machine learning models such as convolution neural network (CNN) and multilayer perceptron (MLP) will be taken into consideration to make the estimation of the lower limb motion.

Bibliography

[1] A. Saboor, T. Kask, A. Kuusik, *et al.*, "Latest research trends in gait analysis using wearable sensors and machine learning: A systematic review," *IEEE Access*, vol. 8, pp. 167 830–167 864, 2020. DOI: `10.1109/ACCESS.2020.3022818`.

[2] K. S. Moon, S. Q. Lee, Y. Ozturk, A. Gaidhani, and J. A. Cox, "Identification of gait motion patterns using wearable inertial sensor network," *Sensors*, vol. 19, no. 22, p. 5024, 2019.

[3] R. D. Gurchiek, R. H. Choquette, B. D. Beynnon, *et al.*, "Remote gait analysis using wearable sensors detects asymmetric gait patterns in patients recovering from acl reconstruction," in *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, IEEE, 2019, pp. 1–4.

[4] S. Sivakumar, A. A. Gopalai, K. H. Lim, D. Gouwanda, and S. Chauhan, "Joint angle estimation with wavelet neural networks," *Scientific reports*, vol. 11, no. 1, pp. 1–15, 2021.

[5] S. McBride, P. Dixon, M. Mokha, and M. S. Cheng, "The relationship between supination resistance and the kinetics and kinematics of the foot and ankle during gait," *Gait & Posture*, vol. 73, pp. 239–245, 2019.

[6] S. Sivakumar, A. A. Gopalai, D. Gouwanda, and L. K. Hann, "Ann for gait estimations: A review on current trends and future applications," in

2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), 2016, pp. 311–316. DOI: `10.1109/IECBES.2016.7843464`.

[7] V. Vijayan, J. P. Connolly, J. Condell, N. McKelvey, and P. Gardiner, "Review of wearable devices and data collection considerations for connected health," *Sensors*, vol. 21, no. 16, p. 5589, 2021.

[8] G. G. Samatas and T. P. Pachidis, "Inertial measurement units (imus) in mobile robots over the last five years: A review," *Designs*, vol. 6, no. 1, p. 17, 2022.

[9] M. Tkáč and R. Verner, "Artificial neural networks in business: Two decades of research," *Applied Soft Computing*, vol. 38, pp. 788–804, 2016.

[10] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[11] S. Salcedo-Sanz, L. Cornejo-Bueno, L. Prieto, D. Paredes, and R. Garcıa-Herrera, "Feature selection in machine learning prediction systems for renewable energy applications," *Renewable and Sustainable Energy Reviews*, vol. 90, pp. 728–741, 2018.

[12] I. Poitras, F. Dupuis, M. Bielmann, *et al.*, "Validity and reliability of wearable sensors for joint angle estimation: A systematic review," *Sensors*, vol. 19, no. 7, p. 1555, 2019.

[13] M. Gholami, A. Ejupi, A. Rezaei, A. Ferrone, and C. Menon, "Estimation of knee joint angle using a fabric-based strain sensor and machine learning: A preliminary investigation," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, IEEE, 2018, pp. 589–594.

[14] Y. Deng, F. Gao, and H. Chen, "Angle estimation for knee joint movement based on pca-relm algorithm," *Symmetry*, vol. 12, no. 1, p. 130, 2020.

[15] J. Coker, H. Chen, M. C. Schall, S. Gallagher, and M. Zabala, “Emg and joint angle-based machine learning to predict future joint angles at the knee,” *Sensors*, vol. 21, no. 11, p. 3622, 2021.

[16] H. Joo, H. Kim, J.-K. Ryu, S. Ryu, K.-M. Lee, and S.-C. Kim, “Estimation of fine-grained foot strike patterns with wearable smartwatch devices,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 3, p. 1279, 2022.

[17] N. S. Erdem, C. Ersoy, and C. Tunca, “Gait analysis using smartwatches,” in *2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, 2019, pp. 1–6. DOI: `10.1109/PIMRCW.2019.8880821`.

[18] A. H. Johnston and G. M. Weiss, “Smartwatch-based biometric gait recognition,” in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2015, pp. 1–6. DOI: `10.1109/BTAS.2015.7358794`.

[19] B. J. Stetter, S. Ringhof, F. C. Krafft, S. Sell, and T. Stein, “Estimation of knee joint forces in sport movements using wearable sensors and machine learning,” *Sensors*, vol. 19, no. 17, p. 3690, 2019.

[20] V. Hernandez, D. Dadkhah, V. Babakeshizadeh, and D. Kulić, “Lower body kinematics estimation from wearable sensors for walking and running: A deep learning approach,” *Gait & Posture*, vol. 83, pp. 185–193, 2021.

[21] J. Sung, S. Han, H. Park, *et al.*, “Prediction of lower extremity multi-joint angles during overground walking by using a single imu with a low frequency based on an lstm recurrent neural network,” *Sensors*, vol. 22, no. 1, p. 53, 2021.

[22] E. Rapp, S. Shin, W. Thomsen, R. Ferber, and E. Halilaj, “Estimation of kinematics from inertial measurement units using a combined deep learning

and optimization framework," *Journal of Biomechanics*, vol. 116, p. 110 229, 2021.

[23] M. Mundt, W. R. Johnson, W. Potthast, B. Markert, A. Mian, and J. Alderson, "A comparison of three neural network approaches for estimating joint angles and moments from inertial measurement units," *Sensors*, vol. 21, no. 13, p. 4535, 2021.

[24] M. M. Ardestani, M. Moazen, and Z. Jin, "Sensitivity analysis of human lower extremity joint moments due to changes in joint kinematics," *Medical engineering & physics*, vol. 37, no. 2, pp. 165–174, 2015.

[25] R. Argent, S. Drummond, A. Remus, M. O'Reilly, and B. Caulfield, "Evaluating the use of machine learning in the assessment of joint angle using a single inertial sensor," *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 6, p. 2 055 668 319 868 544, 2019.

[26] F. J. Wouda, M. Giuberti, G. Bellusci, *et al.*, "Estimation of vertical ground reaction forces and sagittal knee kinematics during running using three inertial sensors," *Frontiers in physiology*, vol. 9, p. 218, 2018.

[27] S. Majumder and M. J. Deen, "Wearable imu-based system for real-time monitoring of lower-limb joints," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8267–8275, 2020.

[28] J. C. Alcaraz, S. Moghaddamnia, and J. Peissig, "Efficiency of deep neural networks for joint angle modeling in digital gait assessment," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, pp. 1–20, 2021.

[29] M. Karas, J. Urbanek, C. Crainiceanu, J. Harezlak, and W. Fadel, *Labeled raw accelerometry data captured during walking, stair climbing and driving*, Jun. 2021. [Online]. Available: `https://www.physionet.org/content/accelerometry-walk-climb-drive/1.0.0/`.

[30] A. Kathuria, *Intro to optimization in deep learning: Momentum, rmsprop and adam*, Dec. 2020. [Online]. Available: `https://blog.paperspace.com/intro-to-optimization-momentum-rmsprop-adam/`.