DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DESIGN OF WEARABLE SYSTEMS FOR ACTIVITY MONITORING AND THEIR APPLICATIONS USING NEURAL NETWORKS AND DATA FUSION TECHNIQUES

by
Gökmen AŞCIOĞLU

October, 2021
İZMİR

DESIGN OF WEARABLE SYSTEMS FOR ACTIVITY MONITORING AND THEIR APPLICATIONS USING NEURAL NETWORKS AND DATA FUSION TECHNIQUES

A Thesis Submitted to the
Graduate School of Natural And Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Electrical and Electronics Engineering

by
Gökmen AŞCIOĞLU

October, 2021
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**DESIGN OF WEARABLE SYSTEMS FOR ACTIVITY MONITORING AND THEIR APPLICATIONS USING NEURAL NETWORKS AND DATA FUSION TECHNIQUES**" completed by **GÖKMEN AŞCIOĞLU** under supervision of **ASSOC. PROF. DR. YAVUZ ŞENOL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

....

Assoc. Prof. Dr. Yavuz ŞENOL

Supervisor

....

Assoc. Prof. Dr. Ahmet ÖZKURT

Thesis Committee Member

....

Prof. Dr. Zeki KIRAL

Thesis Committee Member

....

Prof. Dr. Mustafa GÜNDÜZALP

Examining Committee Member

....

Prof. Dr. Mehmet ENGİN

Examining Committee Member

Prof. Dr. Özgür ÖZÇELİK
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Dr. Yavuz Şenol for his continuous support, guidance, encouragement, help, advice, and endless patience during my study with him. My sincere thanks also go to my thesis committee member Dr. Ahmet Özkurt and Dr. Zeki Kıral. I would also like to express my deepest gratitude to all people who have helped in Dr. Duygu Erdem, Ceylan Kılıç and Onur Çelik.

I would also like to thank for its generous financial support to Dokuz Eylul University under Grant 2020.KB.FEN.011.

Finally, I would like to thank my family and my darling for supporting me spiritually throughout writing this thesis and my life in general.

Gökmen AŞCIOĞLU

DESIGN OF WEARABLE SYSTEMS FOR ACTIVITY MONITORING AND THEIR APPLICATIONS USING NEURAL NETWORKS AND DATA FUSION TECHNIQUES

ABSTRACT

Motion analysis assists experts in the evaluation of health problems including depression and immobility. Continuously monitoring is important for early detection and to prevent irreversible deformities. Video-based systems are popular due to their high accuracy. However; they are expensive and convenient for indoor use. To overcome these challenges, we presented various wearable devices; namely a smart garment, wireless smart insoles, and wireless inertial measurement units (IMUs).

Designed wearable systems provide valuable information for gait analysis and activity recognition. Firstly, lower limb joint angles were predicted using smart garment and fusion techniques during activities. Video-based system was used as a reference device to evaluate the performance of applied techniques. Secondly, it was planned to determine the ideal sensor locations for activity recognition. IMU sensors were placed on various points of the human body and collected data was fed into artificial neural networks for evaluation. Thirdly, activity recognition were aimed using various deep learning neural networks, namely convolutional neural networks, long-short term neural networks, and convolutional long-short term neural networks. Finally, activity recognition performance was evaluated for different sensor modalities including smart insoles and IMU sensors using neural networks.

The results of four studies show that; i) our smart garment presents similar performance with a video based system, ii) chest and leg are the most ideal positions for activity recognition, iii) convolutional long short term neural networks outperform the other tested networks for activity recognition, and iv) smart insoles provide the same activity recognition performance with IMU sensors attached on the leg.

Keywords: Wearable devices, motion monitoring, gait analysis, activity recognition, sensor technologies, neural networks, fusion techniques

AKTİVİTE İZLEME İÇİN GİYİLEBİLİR SİSTEMLERİN TASARIMI VE SİNİR AĞLARI VE VERİ FÜZYON TEKNİKLERİ KULLANILARAK UYGULAMALARI

ÖZ

Hareket analizi, depresyon ve hareketsizlik gibi sağlık sorunlarının değerlendirilmesinde uzmanlara yardımcı olur. Erken teşhis ve geri dönüşü olmayan deformiteleri önlemek için sürekli izleme önemlidir. Video tabanlı sistemler, yüksek doğrulukları nedeniyle popülerdir. Ancak, maliyetleri yüksek ve iç mekan kullanımı için uygundurlar. Bu zorlukların üstesinden gelmek için; akıllı giysi, kablosuz akıllı tabanlık ve kablosuz atalet ölçüm birimleri gibi çeşitli giyilebilir cihazlar tasarlandı.

Tasarlanan giyilebilir sistemler, yürüyüş analizi ve aktivite tanıma için değerli bilgiler sağlamıştır. İlk olarak; akıllı giysi ve füzyon teknikleri kullanılarak alt ekstremite eklem açıları tahmin edildi. Uygulanan tekniklerin performansını değerlendirmek için referans cihaz olarak video tabanlı bir sistem kullanılmıştır. İkinci olarak; aktivite tanıma için ideal sensör konumlarının tespiti planlanmıştır. İnsan vücudunun çeşitli noktalarına atalet ölçüm birimi yerleştirildi. Toplanan veriler değerlendirme için yapay sinir ağlarına beslendi. Üçüncü olarak; aktivite tanıma evrişimli sinir ağları, uzun kısa vadeli sinir ağları ve evrişimli uzun-kısa vadeli sinir ağları dahil olmak üzere çeşitli derin öğrenme sinir ağları kullanılarak amaçlandı. Son olarak; sinir ağları ile akıllı tabanlık ve kablosuz atalet ölçüm birimleri kullanılarak farklı sensör modaliteleri için aktivite tanıma performansı değerlendirildi.

Dört çalışmanın sonuçları gösteriyor ki; i) akıllı giysimiz video tabanlı bir sistemle benzer performans göstermektedir, ii) göğüs ve bacak, aktivite tanıma için en ideal konumlar, iii) konvolüsyonel uzun kısa vadeli sinir ağları, aktivite tanıma için test edilen diğer ağlardan daha iyi performans sunmaktadır ve iv) akıllı tabanlık, bacağa takılan IMU sensörleri ile benzer aktivite tanıma performansını sağlamaktadır.

Anahtar kelimeler: Giyilebilir cihazlar, hareket izleme, yürüme analizi, aktivite tanıma, sensör teknolojileri, sinir ağları, füzyon teknikleri

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Motion Analysis

There is a relationship between human motion and disease symptoms as a result of nervous and musculoskeletal systems. Motion analysis provides characteristic and valuable information to experts for assessment of health-related applications such as fall detection in elderly people, evaluation of rehabilitation process, obesity, cardiac diseases, and so on (Wang & Zhou, 2015; Jarchi et al., 2018; Qi et al., 2018; Lee & Chung, 2009). As such, reliable and accurate motion monitoring can assist an early and precise diagnosis of the mentioned disorders.

Gait analysis and activity recognition are unique topics to diagnose some diseases and evaluate the treatment process. Gait analysis parameters such as stride length, duration of gait phases, and joint angles can be indicators in order to evaluate problems including Parkinson disease and walking abnormalities (Hausdorff, 2009; Chambers & Sutherland, 2002). Activity recognition is potentially useful for the detection of mobility-based problems such as falling and obesity (Sansrimahachai & Toahchoodee, 2016; Artikis et al., 2010).

Motion monitoring is performed using camera-based and sensor-based systems. Camera-based systems provide the highest accuracy (Cuadrado et al., 2021). However, they have some disadvantages such as cost, need a special environment, and useful for only indoor applications (Tao et al., 2012; Bhosale et al., 2015). Moreover, these systems suffer from issues such as privacy and ethics (Lago et al., 2017). Sensor-based systems allow monitoring daily movements with superior advantages including low-cost, lightweight, and small-size (Mukhopadhyay, 2014; Zhang et al., 2012). Moreover, they enable continuous monitoring with low-power consumption even if the user is in the outdoor environment.

1.2 Gait Analysis

Human gait analysis is a mathematical analysis and modeling of human walking. It characterizes human walking in terms of measurement and quantities such as step length, walking speed and so on (Ghoussayni et al., 2004). Gait analysis has focused on three domains including kinetic, kinematic, and spatio-temporal parameters (Tao et al., 2012). The kinetic analysis examines the required forces for walking movement. This type of analysis is usually utilized from ground-reaction forces. The kinematic analysis evaluates body motions through space. The body position and movement of each segment are converted into angular displacements of joints over time. Spatio-temporal analysis is rhythm-based parameters including cadence and stride length.

Clinical motion analysis laboratories equipped with many cameras and force plates are considered a gold standard of human gait analysis because of high accuracy; however, they are expensive, have limited moving area, and need lengthy setup duration (Cuesta-Vargas et al., 2010; Zhang et al., 2017). Moreover, kinetic analysis is limited to only those steps that are applied on a force plate. To overcome these challenges, many researchers has presented wearable gait monitoring systems with the recent developments in sensor technology (Bilro et al., 2011; Low et al., 2020; Gonçalves et al., 2021). They particularly have been used inertial measurement units and force sensors owing to easy placement on the human body, low-cost, and lightweight.

IMUs are particularly used for joint angle estimation (Majumder & Deen, 2020; Jeon & Lee, 2018). Two IMUs are placed on the upper and lower leg to measure the Euler angles of each segment consist of joint. Euler angles can be calculated using only accelerometer and only gyroscope. To improve the accuracy, fusion algorithms are popularly used. Moreover, machine learning methods are popular for joint angle prediction (Coker et al., 2021; Lim et al., 2020).

Force sensors are used to design smart insoles (Tan et al., 2015; Cho, 2017; Manupibul et al., 2014). These insoles have the ability to gather information about ground reaction forces. Particularly, they can be beneficial to detect gait abnormalities. Many researchers presented smart insole designs using a variety of sensors including force-sensitive resistors (FSRs) and also IMUs (Mustufa et al., 2015; Lin et al., 2016).

1.3 Activity Recognition

Human activity recognition (HAR) purposes detection of activities performed in daily life. However, It is challenging since daily activities are complex and highly diverse. For this purpose; a variety of sensors and camera can be popularly used.

In the literature, human activities are categorized as basic and complex. There are different comments about these concepts. A researcher explains that basic activities are repetitive motions such as walking, writing, and sitting. Complex activities rely on a variety of non-repetitive hand movements such as smoking (Shoaib et al., 2016). Another researcher categorizes activities as stationary (e.g. sitting and standing), simple (e.g. ascending and descending stairs), and complex (e.g. cooking and cleaning) (Bharti et al., 2018). In this thesis, we have combined given these definitions and commented that basic activities are repetitive activities and not include any hand gestures, whereas complex activities are repetitive or non-repetitive motions including hand gestures. According to our definition, walking, running, and jumping for basic activities, whereas smoking, writing, and drinking coffee for complex activities can be given as examples.

Activity monitoring is popularly performed using various wearable devices such as smart clothes, smartphones, and so on. They can be placed, mounted, or attached on various parts of the human body. Ideal sensor position for activity recognition is still a debated topic. Researchers have been used various body points for this purpose. An accelerometer mounted on the waist provides applicable information about various activities such as sitting, walking, scrubbing, and vacuuming (Karantonis et al., 2006). Accelerometers placed on various body points including chest, thigh, and trunk is useful for recognition of many activities such as walking, running, and working on a personal computer (Olguın & Pentland, 2006). Moreover, an accelerometer placed on necklace has been used to identify ascending and descending stairs (Pirttikangas et al., 2006).

The studies on activity recognition are popularly performed using various traditional and modern machine learning methods. Among the most popular traditional machine learning techniques are support vector machine (SVM), k-nearest neighbor (kNN), and artificial neural network (ANN) (Jobanputra et al., 2019; Kaghyan & Sarukhanyan, 2012; Mehr et al., 2016). The common aspects of these machine learning methods are the lack of feature extraction capability. These methods require explicit human intervention in the form of feature extraction for training. However, it is known that manual feature extraction technique limits generalization capability since it is based on human experience and domain of knowledge. Furthermore, this process causes time-consuming. Recently, deep learning (DL) branch of modern machine learning has become a popular choice as a powerful tool for activity recognition (Wang et al., 2019; Ronao & Cho, 2016). DL extracts features automatically through hierarchical architectures and offers superior performance. It does not need specialized knowledge and expertise.

There are numerous studies on human activity recognition using machine learning. In particular, many researchers have used public benchmark datasets including OPPORTUNITY, UCI-HAR, and WISDM for their studies (Chavarriaga et al., 2013; Anguita et al., 2013; Weiss, 2019). OPPORTUNITY dataset was built from collected data from a special environment equipped with sensors. It has 17 different activities including opening a dishwasher, toggle switch, etc. Total four subjects were attended data collection processes. For this purpose, a variety of sensors were placed on the objects, environment, and the human body. UCI-HAR dataset was recorded from 30 subjects aged 19-48 years. A smartphone with a built-in inertial sensor was mounted around their waist and subjects performed gait-based six activities including walking, stair descent, and stair ascend. WISDM dataset was collected with a smartphone, which is placed on front leg pockets. 36 subjects were attended data collection processes. The datasets were collected and recorded for standing, sitting, walking, upstairs, downstairs, and jogging. Researchers have been done performance comparisons for activity recognition in terms of different metrics such as recall, precision, and accuracy between traditional and modern machine learning methods. The results show that deep learning neural network outperforms to traditional machine learning methods (Wan et al., 2020; Zebin et al., 2016; Chikhaoui & Gouineau, 2017).

1.4 Wearable Devices

Advance development in sensor and connectivity technologies enables the design of various wearable devices including smartwatches, smartphones, smart insoles, and smart clothes for continuous motion monitoring. Popular sensors used in these devices are accelerometer, gyroscope, force-sensitive resistor, global position system, and electromyography. Using outputs of these sensors and data processing methods, motion-based information such as walking speed, joint angles, stride length, and ground reaction forces can be derived.

Smartphones are equipped with a variety sensors including IMU and global position system. These devices also include internal memory and battery required for the design of HAR system. Sensor data acquired from these devices enables to identify many daily activities (Guo et al., 2016; Hassan et al., 2018). Smartphones suffer from integration on the human body in particular. Moreover, the arbitrary orientation of smartphones needs coordinate transformation to achieve HAR (Sun et al., 2010).

Smartwatches include integrated sensors that provide acceleration and gyroscope information transfer via a connectivity type to a personal computer or a smartphone. Their disadvantages can be placed on limited locations of the human body including the wrist or ankle. However, they are worn even when at home and during nighttime in comparison to smartphones. There are various studies using smartwatches to classify daily activities (Chernbumroong et al., 2013; Mortazavi et al., 2015).

Smart insoles are equipped with sensors inside insoles or base of shoes (Xu et al., 2012; Wang et al., 2015). They can measure force information applied by foot during movement. These devices get easier data collection processes for outdoor applications. Various sensors such as a force-sensitive resistor, pressure sensor, gyroscope, and accelerometer are used for insoles. These systems can be particularly useful for the recognition of gait-based activities.

Smart clothes are convenient to acquire information by placing many sensors. They allow long-term monitoring due to easy wearing and comfort (Adaskevicius, 2014). There are various types of smart clothes for different applications. Smart t-shirts can monitor sleeping and cardiac status with heart rate and electrocardiogram sensors. Sensing gloves and smart trousers equipped with inertial sensor and goniometers can evaluate rehabilitation processes of stroke patients. There are also a variety of smart clothes inform about body position and breathing. Smart clothes bring sensors into contact with various parts of the human body. However, this interaction may affect adversely comfort.

1.5 Thesis Motivation

Motion analysis helps professionals assessment of health problems including falling, depression, and immobility. Continuously monitoring provides valuable information for early detection and prevention of progressive irreversible deformities. Video-based systems are gold standard because of high accuracy. However; they have expensive equipment, a limited moving area, and are convenient for indoor use. Moreover, they affect privacy life adversely and suffer from ethical implications. To overcome these challenges, we have focused on the design of various wearable devices, which are low-cost, lightweight, easy attachable/detachable, and convenient for outdoor applications.

To evaluate the performance of designed wearable systems, we have presented various applications of gait analysis and activity recognition. These are i) prediction of knee and ankle joint angles using smart garment with fusion techniques and comparing the obtained results with a video-based reference system, ii) to evaluate ideal sensor positions on the human body for activity recognition, iii) performance comparison of various deep learning neural networks for activity recognition, and iv) to compare neural network performance using different sensor modalities including smart insole and IMU sensor outputs for activity recognition.

1.6 Thesis Outline

Chapter 2 offers theoretical background about methods used in experiments.

Chapter 3 introduces the equipment including smart garment and designed wireless sensors used in experiments.

Chapter 4 presents some mathematical methods for joint angle estimation and obtained results were compared with respect to reference device.

Chapter 5 determines weights of sensor positions for activity recognition.

Chapter 6 presents a study about basic and complex activity recognition using deep learning neural networks and inertial measurement unit data.

Chapter 7 presents a study about recognition of gait-based activities using deep learning neural network and sensor data including foot contact states and inertial measurement unit data.

Chapter 8 presents outcomes about all study.

CHAPTER TWO
THEORETICAL BACKGROUND

In this chapter, theoretical information shall be given about the techniques used in experiments. For this purpose; information about orientation estimation techniques using inertial measurement sensor, sensor fusion techniques, and mainly applied neural network types for activity recognition were presented. Moreover, a piece of short information has been given for data processing methods.

2.1 Inertial Measurement Unit

Inertial measurement units are an electronic device capable of measuring a specific force and angular rate with respect to a reference frame (Ribeiro & Santos, 2017). They mainly encapsulate accelerometer, gyroscope, and sometimes magnetometer in order to generate information about entire movement.

Nowadays, many of IMUs are based on microelectromechanical system (MEMS) technology. MEMS components as shown in Figure 2.1 are small, lightweight, low-cost, and consume low power. Moreover, they present high accuracy (Renkoski, 2008). These sensors are used in many fields such as aerospace, structural monitoring, or consumer electronics (Janson, 2005; Sekiya et al., 2016).

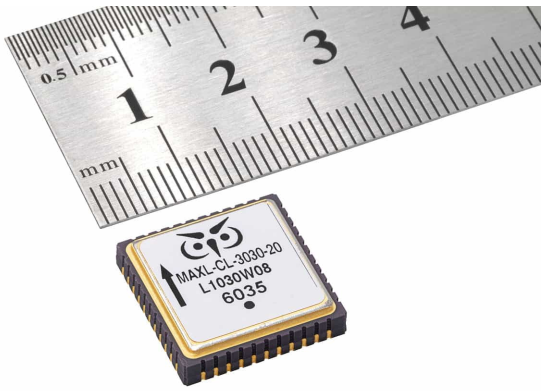

Figure 2.1 MEMS IMU

Accelerometer measures linear acceleration along x, y, and z-axis in terms of g or m/s^2 corresponding to applied force. Gyroscope is a sensor type that measures angular velocity in x, y, and z-axis in terms of rad/s or $degree/s$. Using outputs of these sensors, to make angle prediction is also possible.

2.2 Orientation Estimation using Acceleration and Gyroscope Outputs

Orientation is described as rotations in x, y, and z-axis. Rotation around x-axis, y-axis, and z-axis can be defined as roll (ϕ), pitch (θ), and yaw (ψ). Both linear accelerations (a_x, a_y, and a_z) and angular rates (ω_x, ω_y, and ω_z) can be used for calculation of rotation angles.

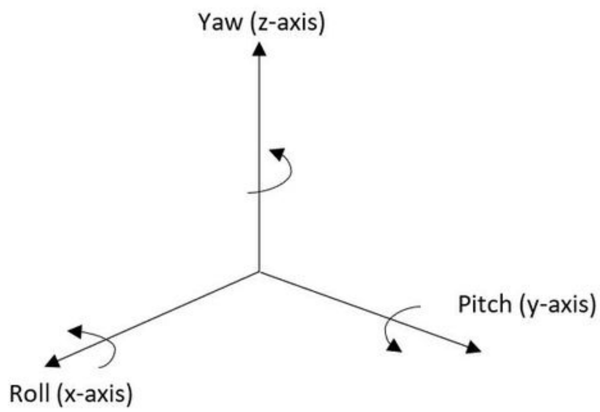

Figure 2.2 Orientation angles

Accelerometers are used for estimation of pitch and roll angles (Luczak et al., 2006). Yaw angle can not be calculated since the linear acceleration around z-axis is fix during rotation (Promrit et al., 2018). Equation 2.1 and Equation 2.2 present the formula for calculation of roll and pitch angles, respectively.

$$\phi_{Acc} = \arctan(-\frac{a_x}{a_z}) \tag{2.1}$$

$$\theta_{Acc} = \arctan(\frac{-a_y}{\sqrt{a_x^2 + a_z^2}}) \tag{2.2}$$

Gyroscope can be used to estimate the pitch, roll, and yaw angles (Wang et al., 2010). These estimations can be performed with a simple integration in respect to time a shown in Equation 2.3, Equation 2.4, and Equation 2.5.

$$\phi_{Gyro} = \phi_{Gyro_0} + \omega_{\phi}\Delta t \tag{2.3}$$

$$\theta_{Gyro} = \theta_{Gyro_0} + \omega_{\theta}\Delta t \tag{2.4}$$

$$\psi_{Gyro} = \psi_{Gyro_0} + \omega_{\psi}\Delta t \tag{2.5}$$

2.3 Sensor Fusion

Accelerometers offer good performance for orientation calculation during long-term, but they can affect adversely from noise in short-term. However, gyroscopes present good performance for short-term, but they have drift problem for long-term (Kok et al., 2017). To overcome these challenges, popular fusion techniques including complimentary filter and extended Kalman filter (EKF) are used.

2.3.1 Complementary Filter

The complementary filter is a simple and popular technique to fuse gyroscope and acceleration data (Higgins, 1975). It is useful to obtain orientation angles such as pitch and roll combining the strength point of one sensor and the weak point of another sensor. Accelerometer presents better performance for slow moving signals. On the other hand, gyroscope offers better performance for fast moving signals. With this purpose, complementary filter combines them using a low-pass filter for the accelerometer and a high-pass filter for the gyroscope. Orientation angles can be fused using Equation 2.6.

$$\theta_{complimentary} = (1 - \alpha) * \theta_{Acc} + \alpha * \theta_{Gyro} \tag{2.6}$$

where θ_{Acc} is pitch or roll estimated from accelerometer, θ_{Gyro} is pitch or roll estimated from gyroscope, α is coefficient between 0 and 1.

2.3.2 Extended Kalman Filter

EKF is a type of Kalman Filter used to model a non-linear dynamic system that suffers from noise or environment (Keatmanee et al., 2013). Kalman Filters predict the state of the system in the past, present, and future. Their operation principle is to find the optimal probability of the hypothesis of the predicted state due to hypothesis of previous state. Then, hypothesis of current state is corrected to have the best estimation using the sensor data for each time step.

EKF linearizes a nonlinear function using Jacobian matrix about a current estimate. The rest of the process is to apply standard Kalman Filter. The most important part of the EKF application is model building using mathematical knowledge to construct a nonlinear transition function for the unknown parameter in each prediction state. There are two models including state and measurement for EKF, respectively given in Equation 2.7 and Equation 2.8.

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \tag{2.7}$$

$$z_k = h(x_k) + v_k \tag{2.8}$$

where f is the function of the previous state, x_{k-1} is previous state, u_{k-1} is control input, x_k is current state, h is the measurement function of current state, x_k is current state, and z_k is measurement. w_{k-1} and v_k are Gaussian noises for process model with covariance Q and measurement model with covariance R, respectively.

EKF equations are based on two parts consist of prediction and correction state, which derived from state and measurement model. At the correction state, Kalman gain (K_k) is calculated with $\hat{x}_k^-$ and P_k^- which are obtained from state transition model and defined by user, respectively. When covariance R approaches to 0, measurement variable outcomes better results than state model. When P_k^- approaches to 0, state model presents more trustable result. Estimated data is updated using K_k value. Moreover, error covariance and all parameter is updated for each iteration and estimated values become more accurate.

Extended Kalman Filter linearizes nonlinear model using Jacobian matrix, which is first-order partial derivative of a vector function with respect to a vector. These are given in Equation 2.9 and Equation 2.10.

$$F_{k-1} = \frac{\partial f}{\partial x}|_{\hat{x}_{k-1}^+, u_{k-1}} \tag{2.9}$$

$$H_k = \frac{\partial h}{\partial x}|_{\hat{x}_k^-} \tag{2.10}$$

Filter algorithm is given in Equation 2.11, Equation 2.12, Equation 2.13, Equation 2.14, Equation 2.15, and Equation 2.16 (Valade et al., 2017).

Predicted state estimate:

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}) \tag{2.11}$$

Predicted error covariance:

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q \tag{2.12}$$

Measurement residual:

$$\tilde{y}_k = z_k - h(\hat{x}_k^-) \tag{2.13}$$

Kalman gain:

$$K_k = P_k^- H_k^T (R + H_k P_k^- H_k^T)^{-1} \tag{2.14}$$

Updated state estimate:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{y} \tag{2.15}$$

Updated error covariance:

$$P_k^+ = (I - K_k H_k) P_k^- \tag{2.16}$$

The hat operator, $\hat{}$ is used to define an estimate of a variable. In other words, $\hat{x}$ points out an estimate of x. x. The superscripts $-$ and $+$ expresses for predicted and updated estimates, respectively.

2.4 Neural Networks

Neural networks are mathematical model represents the central nervous system (Kuo, 2016). They play a key role in the design of intelligent systems with the specifications such as classification, prediction, and optimization. Many disciplines apply them to solve various engineering problems in the field of health, sport science, automotive, etc.

2.4.1 Biological Neuron

The human brain contains billions of neurons. Each neuron makes a connection to a large number of neighbor neurons for information exchange (Suzuki, 2013). A biological neuron comprises many dendrites, a soma (cell body), an axon, and many synapses as shown in Figure 2.3. Each dendrite is connected to a large number of neighbor neurons. They apply a multiplication operation by its own weight value to incoming information. Soma processes input signals and determines whether the neuron produces an electrical stimulation. Axon takes the processed information and transmits it to relevant cells. Synapses provide a connection between axons and other neuron dendrites.

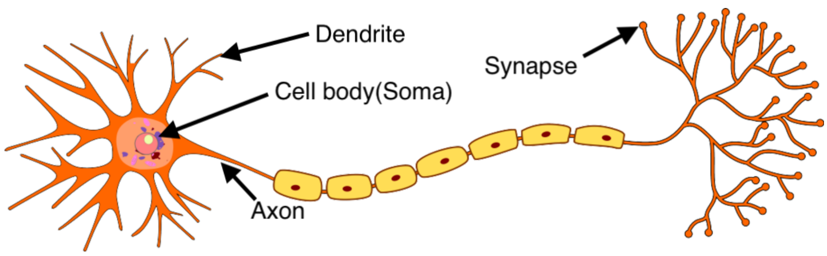

Figure 2.3 Biological neuron and components (Wikimedia, n.d.)

2.4.2 Artificial Neuron

An artificial neuron is a mathematical model that represents the capabilities of a biological neuron. It takes signals and multiplies them separately by weight values (dendrites). Then, multiplied outputs are summed and passed through a nonlinear function to produce output (soma). Finally, the output layer (axon) gives an output for neighbor neurons or is taken as the final output value if there is no connection. Artificial neurons have four basic components, namely weights, bias, summation

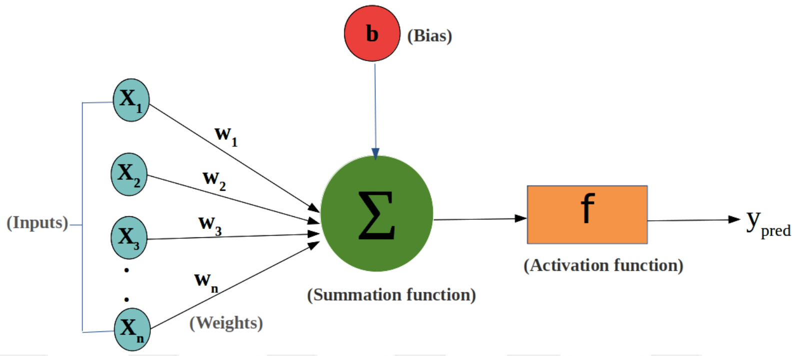

Figure 2.4 Artificial neuron and components (Satya Ganesh, 2020)

function, and activation function as shown in Figure 2.4. The values of input vector can be considered as $X = [X_1, X_2, X_3, ..., X_n]$. Each input is multiplied with relevant weight factor $W = [W_1, W_2, W_3, ..., W_n]$. Bias value (b) is used to adjust shift activation function to either left or right. It can be a positive or negative value. Activation function determines the output of network mapping a value range between 0 to 1 or -1 to 1 etc. The mathematical representation of a neuron is given in Equation 2.17.

$$y = f(A^T W + \theta) \tag{2.17}$$

2.4.3 Activation Functions

Activation functions are used to limit neuron outputs. The most popular types used in neural networks are nonlinear functions including log-sigmoid, hyperbolic tangent sigmoid, rectified linear unit (RELU), and softmax activation functions.

Log-Sigmoid activation function converts input values between $-\infty$ and $+\infty$ into range of 0 and 1 (Hagan et al., 1997). The mathematical formula for this function is shown in Equation 2.18. It is particularly used in multilayer perceptron neural networks (MLP NNs) since this transfer function is differentiable.

$$y = \frac{1}{1 + e^{-x}} \tag{2.18}$$

which x is equal to $A^T W + \theta$.

Hyperbolic tangent sigmoid function maps output range from -1 to 1 (Hagan et al., 1997). It has greater range than log-sigmoid activation. Its advantage is that negative inputs can be mapped negative values and zero inputs will be mapped near zero. The mathematical formula is given in Equation 2.19.

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.19}$$

Both log-sigmoid and hyperbolic tangent sigmoid function are used in feed-forward neural networks.

RELU is popular activation function used in deep learning neural networks. It is linear for all positive values, and zero for all negative values (Lin & Shen, 2018). Mathematically, it is defined as Equation 2.20.

$$y = max(0, x) \tag{2.20}$$

Softmax activation function returns a probability distribution using cross-entropy method for target classes in classification problems (Nwankpa et al., 2018).

2.4.4 Artificial Neural Networks

2.4.4.1 Multilayer Perceptron Neural Networks

MLP NNs are a type of feed forward neural networks (Nielsen, 2015). They consist of three stages; namely input layer, one or more hidden layers, and output layer as shown in Figure 2.5. Input layer is an entry point and receives raw or processed features. Hidden layers transfer the incoming information by processing from input layer to output layer. Output layer performs prediction or classification task. The information in MLP NNs only moves into forward direction. A neuron can not make a cycle itself. Each layer is connected to the previous layer.

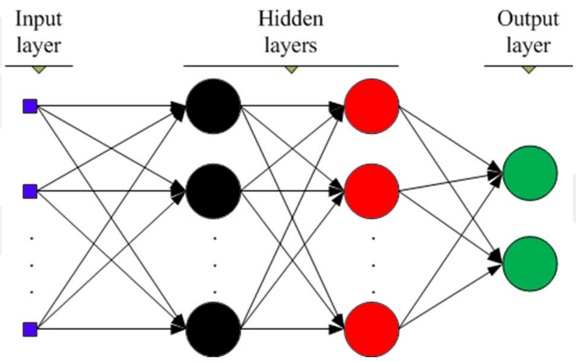

Figure 2.5 MLP NN architecture

MLP NNs are used to solve supervised learning problems. They train neural network model depending on a known set of input-output relations and learn dependencies between those inputs and outputs. Training process aims to minimize error by adjusting the hyperparameters. Backpropagation algorithm is always used to make weight and bias adjustments relative to the error (Leung & Haykin, 1991). Network performance can be evaluated in a variety of terms such as mean squared error (MSE) or accuracy.

2.4.5 Deep Learning Neural Networks

2.4.5.1 Convolutional Neural Networks

Convolutional neural network is a type of deep learning neural network that can automatically extract the features from hierarchical architectures (Albawi et al., 2017; Jolly et al., 2018). It is particularly used for image classification. The overall structure of this network as shown in Figure 2.6 comprises of six layers as convolutional, pooling, flattened, fully-connected, classification layers, and activation functions (RELU, softmax, etc.).

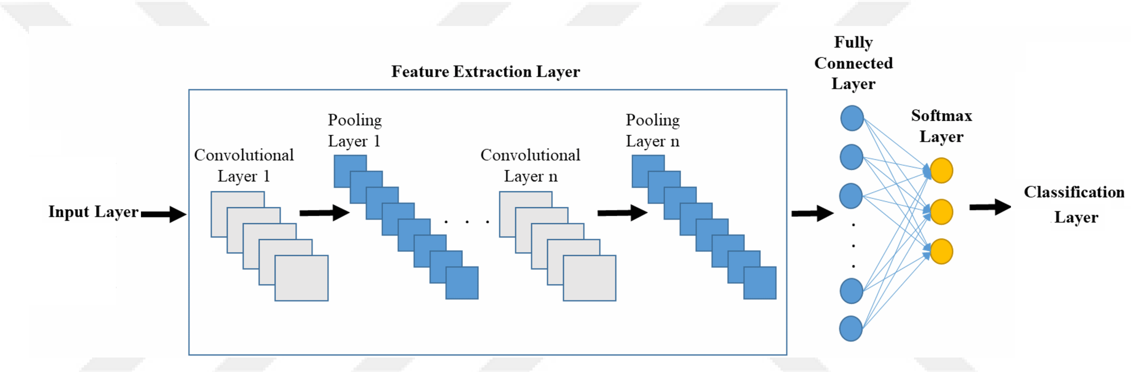

Figure 2.6 CNN architecture

Convolutional layers are used to extract feature maps from the input dataset using one or more filters. Each filter with n-dimension makes convolution operation by sliding with determined strides on the input dataset and a feature map is created corresponding to each filter.

Pooling layers are used to reduce the dimensions of feature maps generated by convolutional layers. It allows to reveal dominant characteristics using n-dimension filters, which slides on each channel of feature maps with determined strides. This process also significantly decreases computational power of neural network. The most common approach for pooling is maximum method.

Flatten layer converts the two or more pooled feature maps into a single size vector. Fully connected layer is a type of feed-forward layer. Each neuron in the layer has connection to each neuron in previous layer. Classification layer produces final class type using incoming class score probabilities from softmax activation function.

RELU is especially used in deep learning. It accepts negative values as 0 and keep positive values without changing.

2.4.5.2 Long-Short Term Neural Networks

Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber in 1997 (Hochreiter & Schmidhuber, 1997). They are a type of recurrent neural network (RNN) capable of learning long-term dependencies.

RNNs have a context layer different from feed-forward neural networks. This context layer keeps unit outputs for an amount of time. They have a basic structure to process cycling inputs and this operation principle can cause exploding and vanishing problems when the influence of a given cycle input in context layer decays or rises up exponentially. Unlike RNN, LSTM units have a special structure to regulate past information flow to overcome exploding and vanishing problems (Pascanu et al., 2013; Bengio et al., 1994). Each LSTM unit has a memory and three gates as input, forget, and output. The input gate decides whether information needs to be passed. The forget gate chooses the information that should be forgotten from the previous states. The output gate determines information to be passed.

LSTM NNs have five layers consist of input layer, one or more LSTM layers, fully-connected layers, one softmax activation function, and one classification layer (Sherstinsky, 2020). LSTM layers have many particular memory cells, which include gates such as input, forget, and output. Fully-connected layer is a feed-forward network that uses the extracted features by LSTM layers. The softmax activation function makes probability distribution for class labels and classification layer decides the final class. The architecture of LSTM NN is given in Figure 2.7.

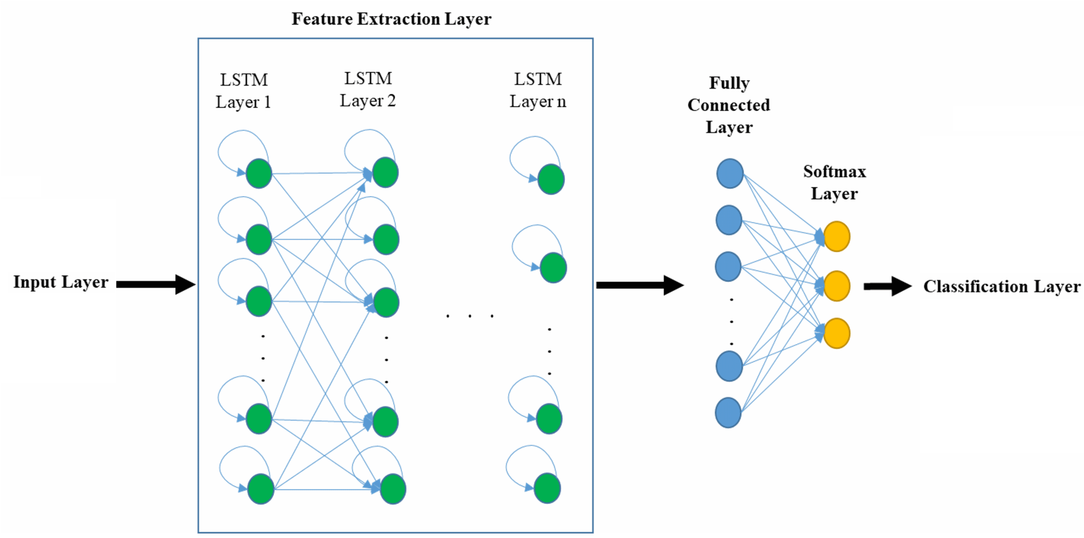

Figure 2.7 LSTM NN architecture

2.4.5.3 Convolutional Long-Short Term Neural Networks

Convolutional Long-Short Term Neural Networks (ConvLSTM NN) is a hybrid model, which combines the feature extraction part of CNN and LSTM NN, as seen in Figure 2.8 (Tsironi et al., 2016). In CNN part, feature maps are extracted using convolutional and pooling layers. These maps are given as input for LSTM NN that models the temporal dynamics of the extracted features. The outputs of LSTM NN passing through one or more fully-connected layers, softmax activation function, and classification layer produce class label.

2.4.6 Hyperparameters

Hyperparameters are the variables which define the network structure and controls training algorithm. Choosing appropriate values of hyperparameters play a key role in the success of neural network performance (Smith, 2018; Claesen & De Moor, 2015). Otherwise, it may cause underfitting or overfitting that can affect adversely network performance. Some critical hyperparameters are learning rate, number of hidden layers, neuron number in hidden layers, momentum coefficient, optimization algorithm, number of epochs, and batch size.

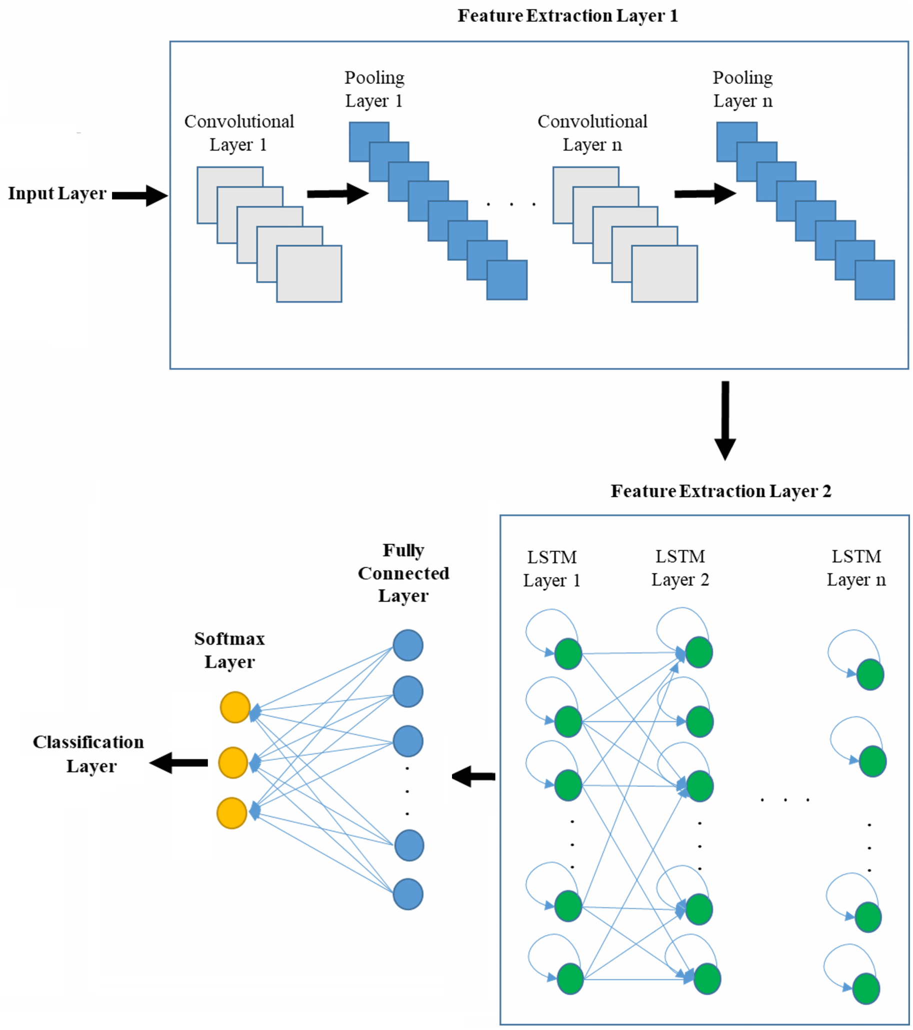

Figure 2.8 Convolutional LSTM NN architecture

The learning rate determines the rate at which a network updates its parameters. Low learning rate decelerates the training process but converges smoothly. High learning rate accelerates learning process, however it may not converge. The batch size is the number of subsamples dedicated to the network where the parameter update takes place. The epoch number refers to the number of times that all training data is presented to the network during the learning process. As long as training performance improves, the number of epochs may increase unless validation accuracy decreases. Momentum coefficient determines the direction of next step to prevent oscillations using information belong to the previous steps.

2.5 Data Segmentation

Data segmentation works mainly in partitioning data and dividing them into different classes based on cluster form. (Nitzberg et al., 1993). Figure 2.9 describes how sliding window are applied to segment the collected data. Window size is set to θ seconds and size of overlapping is set to δ ($0 \leq \delta \leq \theta$). In case δ is equal to zero, segmentation is performed for non-overlapping. Otherwise, it is overlapping.

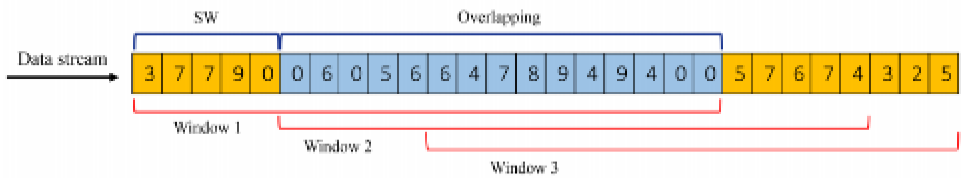

Figure 2.9 Data segmentation (Bersch et al., 2014)

2.6 Data Augmentation

Data augmentation is a method used for increasing the amount of data by adding slightly modified copies of existing data or created newly synthetic data from available data (Pawara et al., 2017). It helps to augment the diversity of dataset without need an effort to collect more data in order to help improve accuracy and prevent the overfitting of machine learning model. There are various methods such as data flip, data rotation, adding gaussian noise to data, and data crop to increase the size of the data.

2.7 Evaluation Metrics

There are various metrics to evaluate the algorithm performance. In this thesis; accuracy, recall (sensivity), precision, and F_{score} were used for evaluation. The mathematical definitions of them is given in Equation 2.21, Equation 2.22, Equation 2.23, and Equation 2.24.

$$Recall(Sensivity) = \frac{TP}{TP + FN} \tag{2.21}$$

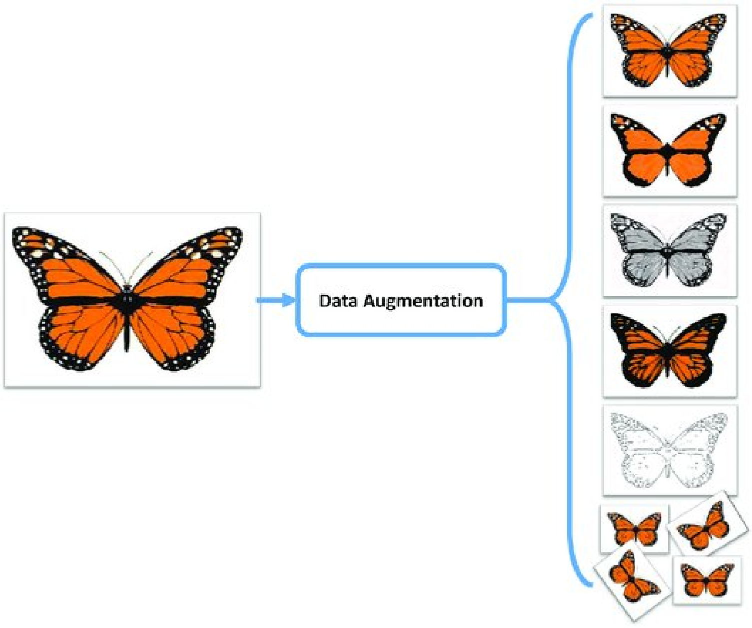

Figure 2.10 Data augmentation (Ahmad et al., 2017)

$$Precision = \frac{TP}{TP + FP} \tag{2.22}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.23}$$

$$F_{score} = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{2.24}$$

which TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

CHAPTER THREE
EXPERIMENTAL SETUP

In this chapter; a smart garment, wearable wireless IMU sensors, and wireless smart insoles designed for our experiments are introduced.

3.1 Smart Garment

A smart garment was designed to predict and record knee and ankle joint angles in daily activities. The design process consists of two main steps as textile and electronic parts. Textile part of the design has two components including a tight and a

Figure 3.1 Placements of IMU sensors sewn on velcro

belt. They were equipped with IMU sensors, wire connections, an electronic board and some accessories. IMU sensors were positioned on lower and upper parts of relevant joints along the same line of the tight. They were sewn on Velcro instead of sewing directly on tight to adjust the location for different users as shown in Figure 3.1 and connected to electronic board via wires.

Electronic board was designed to collect and record incoming IMU sensor data. Main components used in the design are a microcontroller (Mbed NXP LPC1768), six IMUs (MPU6050), a memory module and a battery unit. Microcontroller collects and records raw accelerometer and gyroscope data with 50Hz sampling rate in memory card. It communicates with sensor through I2C bus and memory module through SPI bus.

Circuit design of the electronic board was performed in Autodesk EAGLE PCB. Its schematic part is given in Figure 3.2. Software implementation was done on MBED online compiler with C/C++ programming language.

To cover electronic components on tight and provide aesthetic appearance, a cloth line was placed throughout the middle of the front part of legs. One side of cloth line was sewn on tight fabric for durability and Velcro was sewn inside of the other side of cloth line to mount IMU sensors at any desired points on tight. It provides easy attachment/detachment feature for electronic components. This feature also enables the garment washable as a conventional textile product. In addition to tight, a belt with a kangaroo pocket was designed to place the electronic board. The pocket provides aesthetic appearance like cloth line. Moreover, it is quite useful to improve comfort and mobility of user. Final design of smart garment can be seen in Figure 3.3.

3.2 Wireless Sensor Network

A wireless sensor network (WSN) was purposed to collect and record data from wearable IMU sensors and smart insoles equipped with force-sensitive resistors during

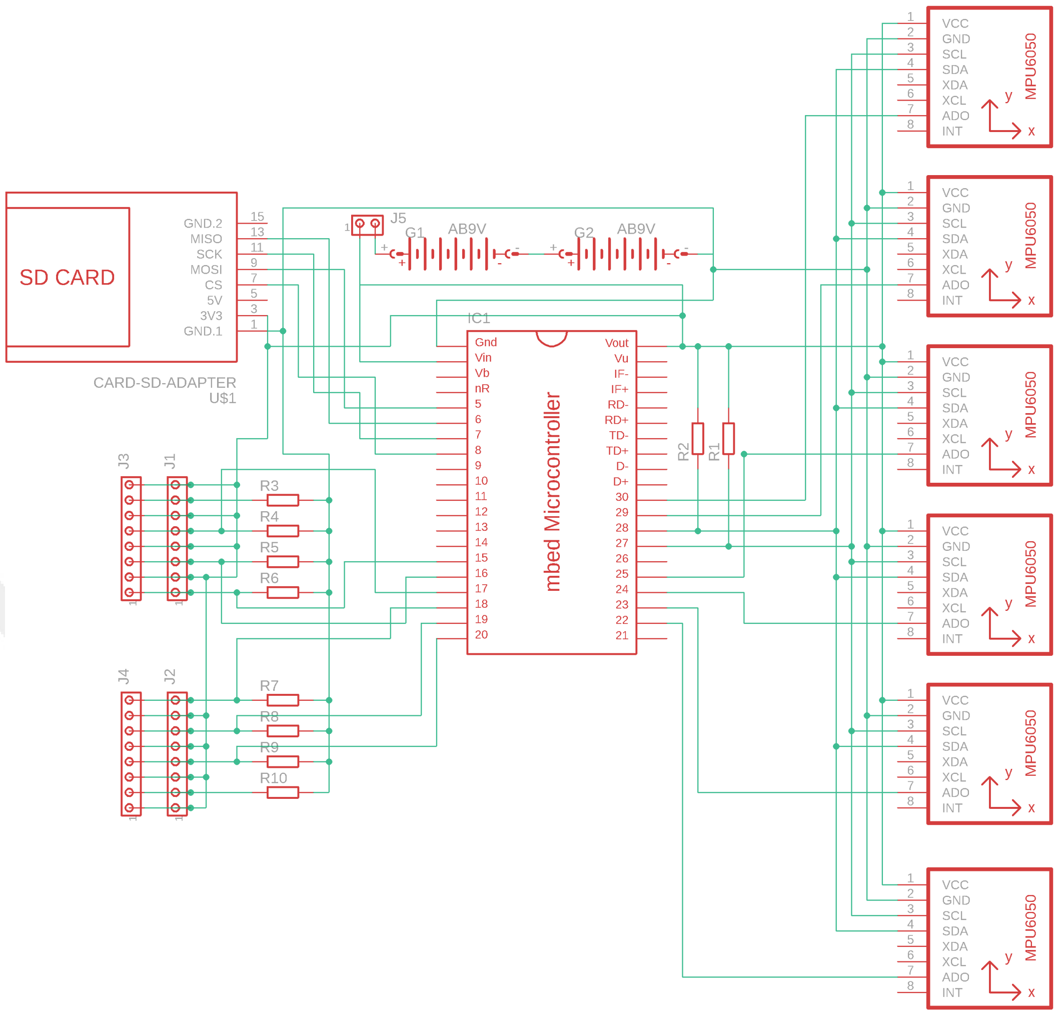

Figure 3.2 Schematic circuit of electronic board

various daily activities. The network consists of a master and many slave devices. Figure 3.4 shows block diagram of the designed WSN system.

The master device establishes and manages a wireless network, as well as records incoming signals from the slave devices. Raspberry Pi Zero Wireless was the chosen master device due to its small size, low-cost, and internal Wi-Fi module. It supports Linux Operating System with readily available programmable tools required to operate the master device.

The master device generates a network to which slave nodes whose MAC addresses are registered can connect and are assigned static IP addresses as shown in Figure 3.5. It manages all slave devices by broadcasting various commands.

Figure 3.3 Smart garment (Personal archive, 2018)

Master-slave communication employs User Datagram Protocol (UDP). Software implementation of master device was performed using Phyton script.

The software on the master device (Python script) also accepts interactively various management commands which can be submitted via a web interface in Figure 3.6 that runs on the same wireless network. Management commands range from broadcasting option to viewing connected devices on the network.

Slave devices connect to the network created by master device via Wi-Fi communication and send sensor data to the master. For the slave devices, NodeMCU Wi-Fi development modules were selected. It has a sufficiently small size and low cost. Furthermore, NodeMCU allows the utilization of built-in microcontroller of the

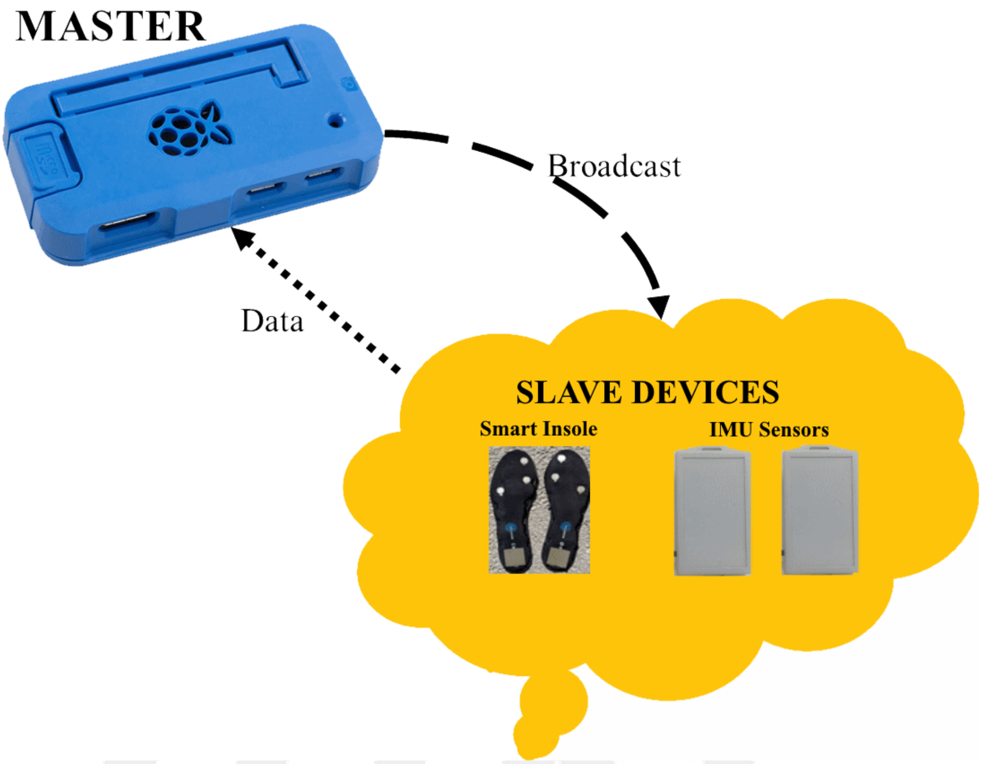

Figure 3.4 Block diagram of wireless sensor networks

Wi-Fi module, dispensing with the need for external microcontroller. NodeMCU development module has analogue input, SPI and I2C bus interface and it makes communication easily with the sensors.

The slave level acquisition frequency was configured as 40Hz. Slave devices run for 6 hours on a full charge with lithium polymer batteries. When the switch is in OFF mode, external power can be connected to charge the battery via the micro USB on the integrated circuit. Charging does not require batteries to be removed from its slot.

When the battery switch set to ON mode, the slave devices connect to the sensor network and await commands from the master device; this command is broadcast to all slaves. The master device periodically broadcasts counter to the slave devices, which prompt the slaves to collect and return sensor data. These broadcasts also serve to synchronize slave devices. Slave devices add data counter and checksum to each sensor data. Master device collects and after verifying the data records it in a file. Each sensor

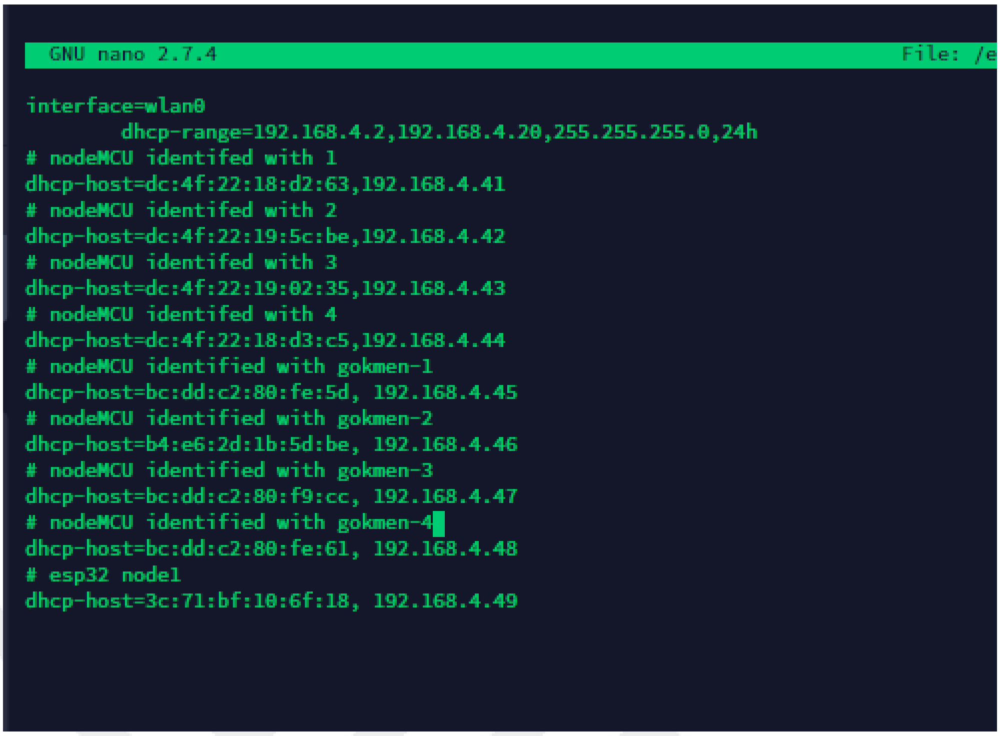

Figure 3.5 Python script assigning static IP address to slave devices

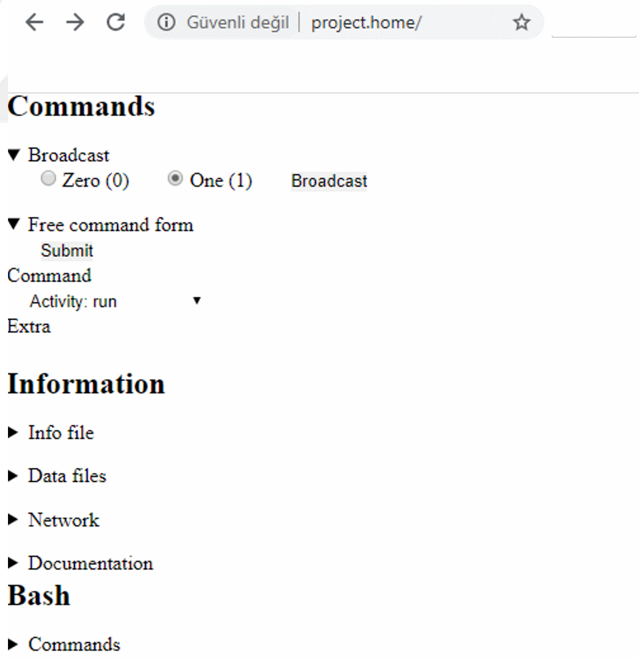

Figure 3.6 Software management tool of master device

data is saved in a separate file defined by the global name specified by the user and the IP assigned to the slave devices.

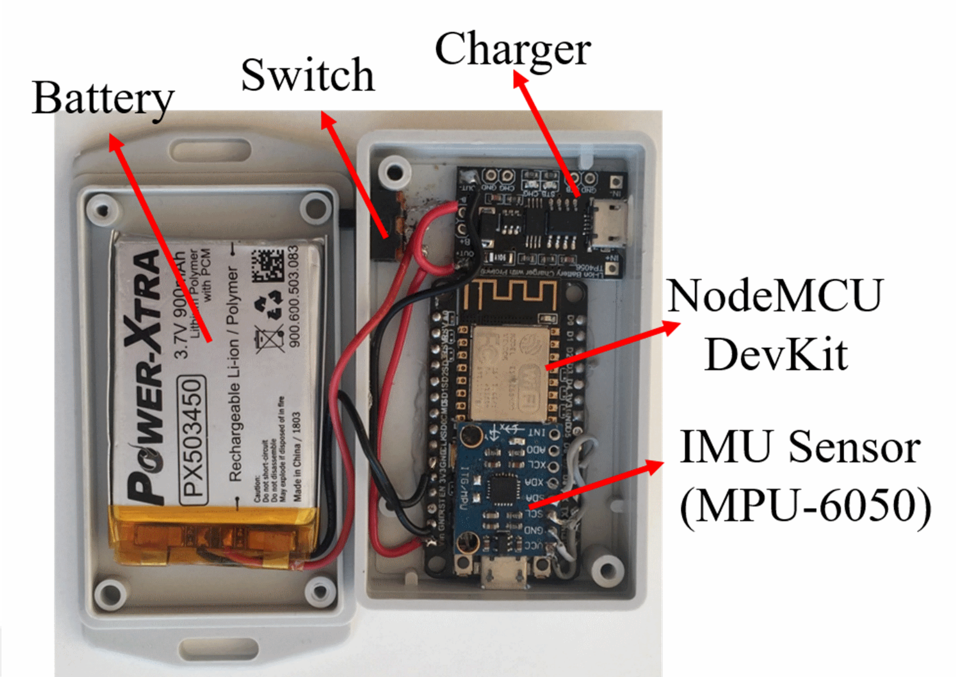

Figure 3.7 The design of wireless IMU slave device

Communication protocol between master and slave devices was chosen as IEEE 802.11n. The protocol has advantages such as high data rate and longer distance at indoor and outdoor when compared with 802.11 b/g.

3.2.1 IMU Sensor Design

IMU Slave device consists of ESP8266 development board, IMU sensor (MPU6050), a battery, and a charge unit. The development board makes communication with IMU sensors via I2C bus and send incoming data to master device. The components are protected by a box to prevent hardware problems and isolate the circuit from users. The enclosure box has attached elastic rope and stopper to allow the user to comfortably mount and secure the sensor on the human body. Figure 3.7 shows the hardware design of the IMU slave devices.

3.2.2 Smart Insole Design

Wireless smart insoles were designed to collect foot contact states during activities. The electronic parts of the insole consist of FSRs, a small-size and low-capacity battery, and an development board supporting Wi-Fi communication. Four piece FSRs were used in each insole design. These sensors were placed on heel, medial forefoot (MF), lateral forefoot (LF), and toe area of the foot by examining many studies in the literature (Tee et al., 2017; Muñoz-Organero et al., 2017; Lin et al., 2016). FSRs transmit their outputs via analogue interface to the development board, which processes the incoming FSR outputs and transfer the master device via Wi-Fi connectivity.

We have used a memory foam sole in the insole design. It has advantages including flexibility, cushioning, and moisture absorption. This type of sole helps to reduce shocks and vibration and prevents perspiration as well as providing optimal walking comfort. The designed smart insole can be seen in Figure 3.8.

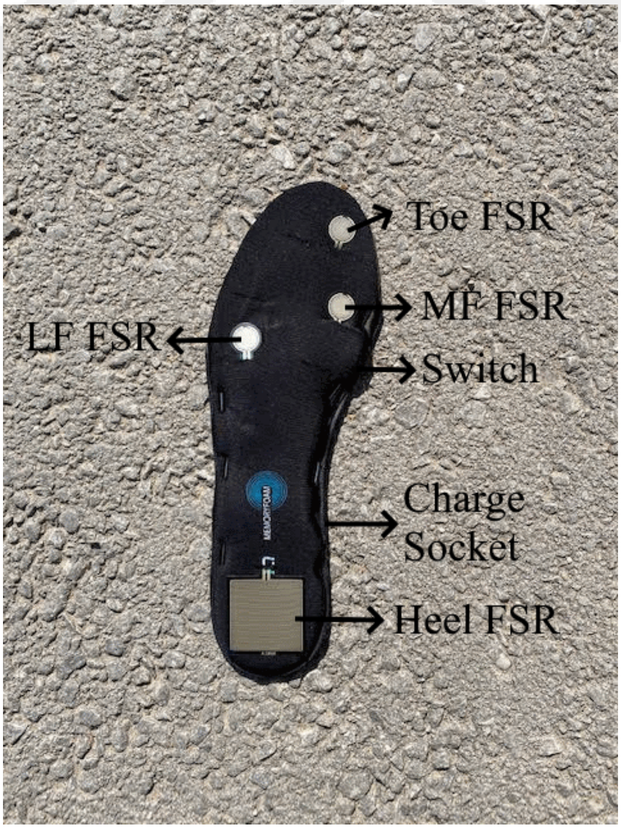

Figure 3.8 The designed smart insole and position of FSRs

CHAPTER FOUR
JOINT ANGLE ESTIMATION USING VARIOUS FUSION TECHNIQUES AND COMPARISON OF OBTAINED RESULTS WITH A REFERENCE SYSTEM

Joint angles reveal useful information for diagnosis and treatment of some diseases. They are always collected with camera-based systems in private rooms. However, these systems have disadvantages including expensive equipment, limited movement area and only indoor use. To overcome these issues, a smart garment is presented to collect knee and ankle joint angles of both legs. With this purpose, IMU sensors were placed into garment and obtained raw data was converted into joint angles using various estimation techniques including complementary and extended Kalman filter. Volunteers participated in experiments performed activities including level walking and sit-down/stand-up activities. A comparison was done between calculated joint angles by designed system and measured joint angles by Kinect Sensor.

4.1 Method

There are camera and sensor-based systems that can estimate joint angles using various data processing techniques. The most reliable systems are based on the camera. However, they have disadvantages such as expensive and convenient for only indoor use. To overcome these challenges, we have focused on the smart garment design introduced in Chapter 3.

The smart garment was designed to estimate knee and ankle joint angles. For this purpose; two IMU sensors justified each other were placed on lower and upper parts of the relevant joint. Joint angles are calculated by subtracting angles from each other obtained by these sensors. For our system; since the knee and ankle joints are sequential, a total of three sensors, one common, were used for our system as seen from Figure 4.1.

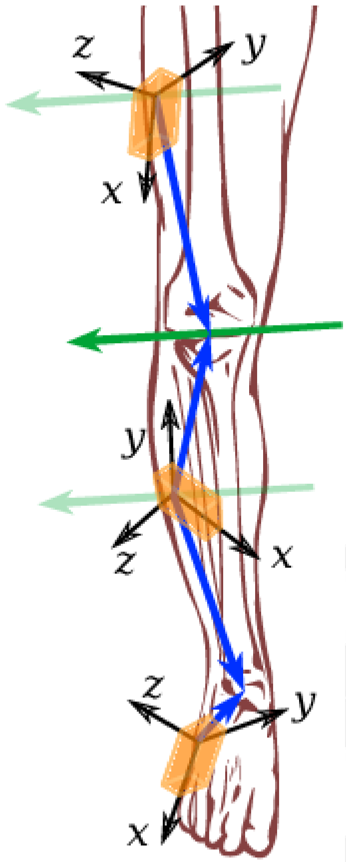

Figure 4.1 IMU sensor positions for knee and ankle joint angle estimation (Seel et al., 2014)

There are various methods to estimate joint angles. Three-axis acceleration and three-axis angular rate values generated by IMU sensors are separately used to convert into angle values in terms of degree utilizing mathematical formulas as explained in Chapter 2. Moreover, sensor fusion techniques including complementary and extended Kalman filter are also used to improve the estimated results. In this study, we have focused on these four techniques to calculate joint angles and evaluate results. To verify obtained results from our smart garment, Kinect v2 sensor given in Figure 4.2 was chosen as a reference device. It is a compact system, consisting of two cameras, namely an RGB camera and an infrared camera, which serve as depth sensors. Features of Kinect v2 sensor are presented in Table 4.1.

Kinect sensor tracks movements of the human extremities by computing the position of various skeleton joints based on the depth information and triangulation. It determines the origin as hip center for tracking algorithm. Using the respective

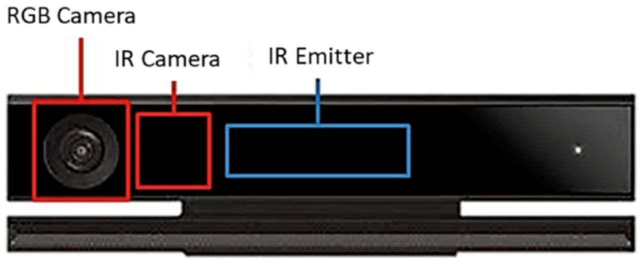

Figure 4.2 Kinect v2 sensor

Table 4.1 The summarized features of kinect v2 sensor

IR camera resolution	512 x 424 pixels
RGB camera resolution	1920 x 1080 pixels
Field of view	70 x 60 degrees
Frame rate	30 frames per second
Operative measuring range	from 0.5 to 4.5m

functions provided by Kinect SDK, position of 20 points as shown in Figure 4.3 can be obtained.

Five subjects (3 males and 2 females) from Dokuz Eylul University with age ranging from 20 to 35 (mean 30±3.6), height 160-192 cm (mean 175±3.4) and body mass 52-85 kg (mean 70±3.2) participated to data collection process. They performed daily activities consist of walking on the ground and sit-down/stand-up. Each one made a practice for approximately 2 minutes to acquire reliable data sets. Subjects have neither injury nor gait abnormality.

Both Kinect sensor and IMU sensors on smart garment recorded the data simultaneously. Sampling rates were adjusted to 30Hz for both devices. Figure 4.4 shows an example from captured images during experiments.

4.2 Results

To examine the performance of our designed system, Microsoft Kinect sensor was taken as a reference platform. Since experiments are done in the limited moving area and indoor, daily activities including walking on the ground and sit-down/stand-up

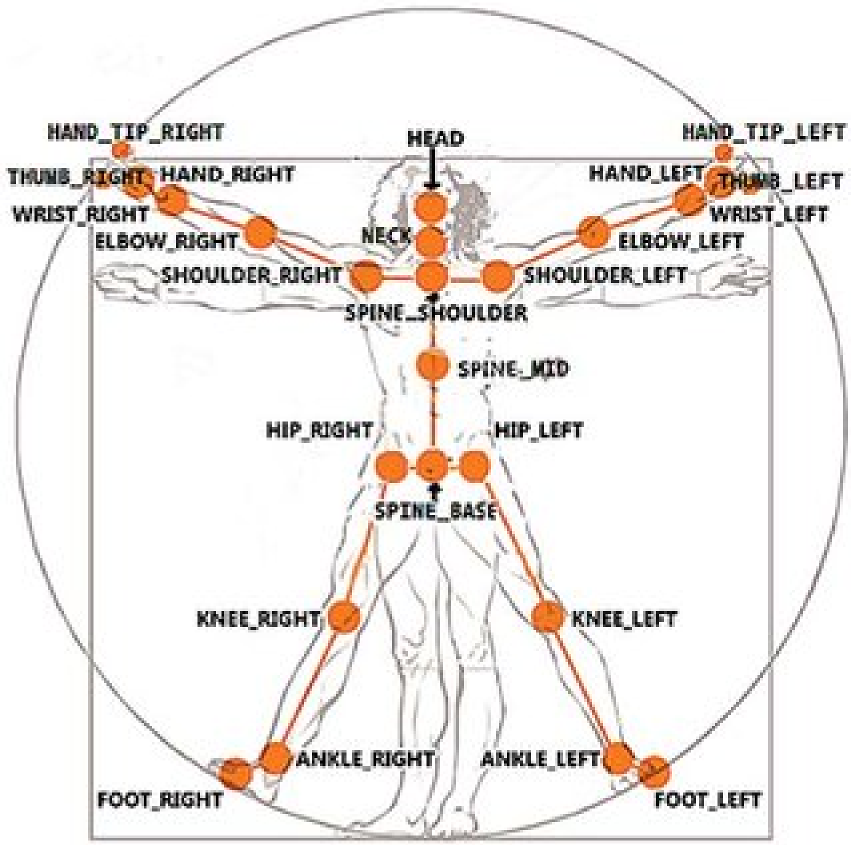

Figure 4.3 Joint angles measured by kinect v2 sensor (Stack overflow, 2018)

were chosen for test processes. Subjects, which wear the smart garment, were performed these activities in view angle of the Kinect sensor. All data collection processes were simultaneously performed by both systems.

Kinect sensor transfer obtained data to computer via USB. Then, functions provided in the Kinect SDK are used to calculate joint angles.

Using IMU sensor data, we have applied four techniques to calculate joint angles. Firstly, joint angles were obtained using only accelerometer outputs. Secondly, only gyroscope outputs, namely angular rates, were used to calculate joint angles. Thirdly, calculated joint angles by accelerometer and gyroscope outputs were combined using complimentary filter. Finally, produced angles by accelerometer and gyroscope outputs were fused using Extended Kalman filter.

Figure 4.5 shows a comparison of estimated joint angles for the right knee joint using

Figure 4.4 An image captured by kinect v2 during experiment (Personal archive, 2018)

Table 4.2 Absolute error rates of smart garment when only accelerometer outputs were used for joint angle prediction and kinect sensor taken as reference

	Knee Angle	Ankle Angle
Activity Types	**Right Leg**	
Sit-down/Stand-up	2.08%	0.5%
Walking on the ground	4.19%	2.17%
	Left Leg	
Sit-down/Stand-up	2.21%	0.34%
Walking on the ground	4.08%	2.19%

only acceleration and only angular rates with provided knee joint angle by the reference device during sit-down/stand-up activity. The results show that accelerometer affected adversely from noise for short-term and gyroscope has drift problem for long-term.

To overcome these challenges, fusing algorithms were used. Figure 4.6 presents a comparison during sit-down/stand-up activity between joint angles provided by the complementary filter, extended Kalman filter, and reference device. The results show that fusing techniques remove noise and drift problems and increase accuracy.

To make a more clear comparison, accuracy performance of smart garment for each method with respect to reference system was presented in Table 4.2, Table 4.3, Table 4.4, and Table 4.5. Performance analyses were separately offered for each leg, each activity type, and each technique.

The results show that the best results were obtained using Kalman Filter for each

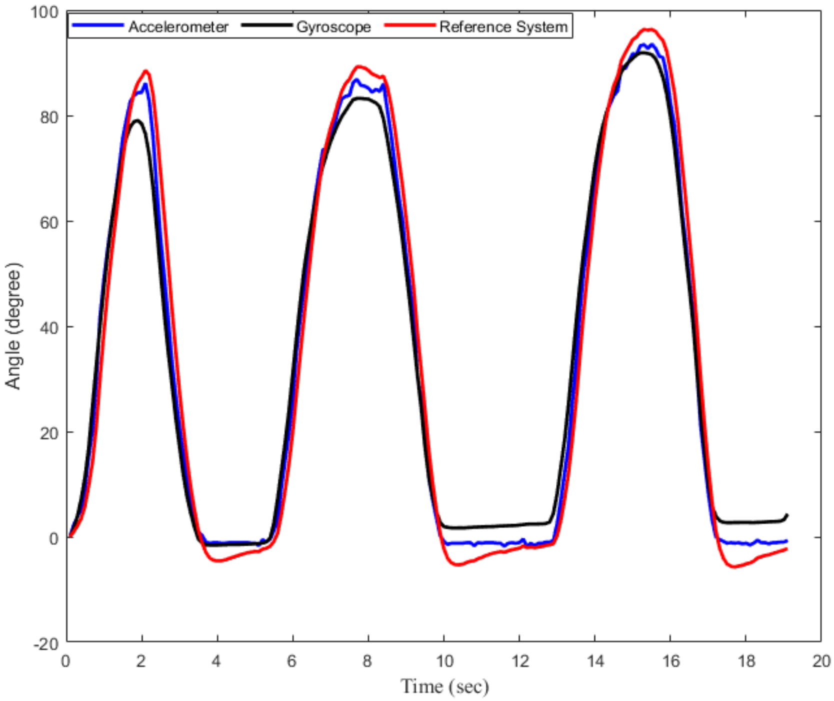

Figure 4.5 Knee joint angles provided by only acceleration signals, only angular rates, and reference device

Table 4.3 Absolute error rates of smart garment when only gyroscope outputs were used for joint angle prediction and kinect sensor taken as reference

	Knee Angle	Ankle Angle
Activity Types	**Right Leg**	
Sit-down/Stand-up	2.91%	0.55%
Walking on the ground	4.25%	2.25%
	Left Leg	
Sit-down/Stand-up	2.35%	0.49%
Walking on the ground	4.28%	2.42%

Table 4.4 Absolute error rates of smart garment when complementary filter was used for joint angle prediction and kinect sensor taken as reference

	Knee Angle	Ankle Angle
Activity Types	**Right Leg**	
Sit-down/Stand-up	1.3%	0.33%
Walking on the ground	2.14%	1.37%
	Left Leg	
Sit-down/Stand-up	1.12%	0.27%
Walking on the ground	2.24%	1.54%

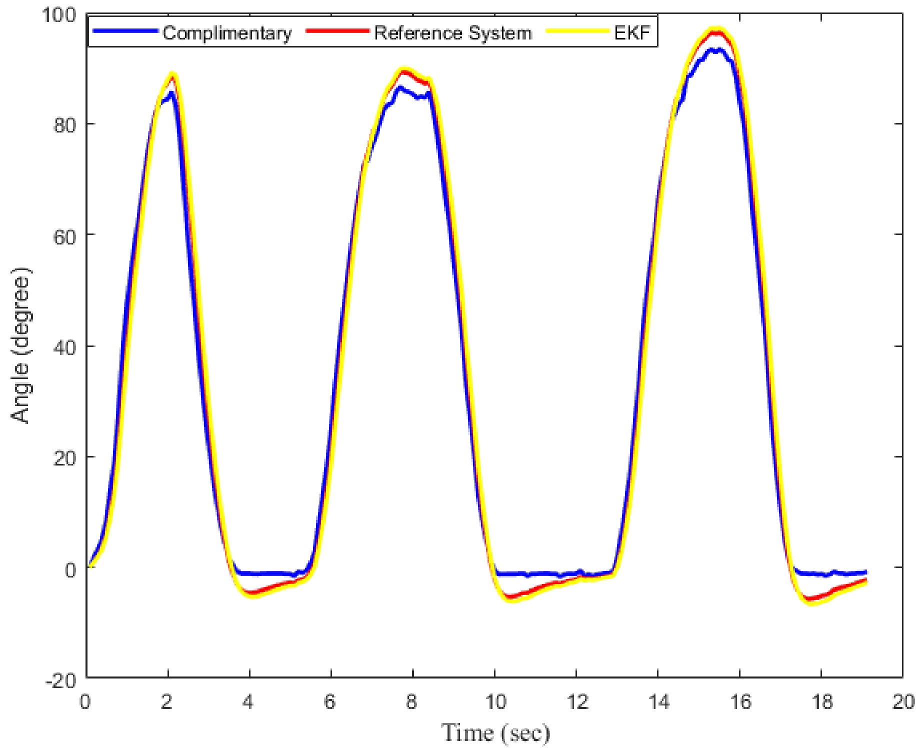

Figure 4.6 Knee joint angles provided by complementary filter, extended kalman filter, and reference device

Table 4.5 Absolute error rates of smart garment when EKF was used for joint angle prediction and kinect sensor taken as reference

	Knee Angle	Ankle Angle
Activity Types	**Right Leg**	
Sit-down/Stand-up	1.02%	0.25%
Walking on the ground	1.08%	0.89%
	Left Leg	
Sit-down/Stand-up	0.97%	0.22%
Walking on the ground	1.41%	1.03%

activity type. The accuracy for this method changes from 95.9% to 99.75%. To fuse accelerometer and gyroscope outputs improves the performance. To use only accelerometer or only gyroscope output has poor performance when compared with fusion algorithms. Moreover results present that predicted joint angles during Sit-down/Stand-up activity is higher accuracy when compared with walking on ground.

CHAPTER FIVE
EVALUATION OF SENSOR POSITIONS ON HUMAN BODY FOR ACTIVITY RECOGNITION USING ARTIFICIAL NEURAL NETWORKS AND ACCELEROMETER DATA

Sensor position is an important challenge that affects the performance of activity recognition. For this purpose; a total of four wireless IMU sensors were placed on the various parts of the human body including the chest, arm, shoulder, and leg. Accelerometer data were collected and recorded for various activities including walking, running, jumping, and sit down/stand up. Then, the obtained dataset from each sensor was fed into MLP NNs separately or fusing each other to evaluate sensor positions.

5.1 Method

Total four wearable wireless IMU sensors introduced in Chapter 3 were used to collect and record 3-axis acceleration information. These sensors were placed on the chest, arm, leg, and shoulder area of the human body as shown in Figure 5.1.

A total of twenty healthy people (12 male and 8 female) with ages from 24 to 30, height from 158cm to 192cm, and weight from 48kg to 98kg attended data collection processes. Each user was informed about the purpose of the study and asked to perform four activity types. These activities and its abbreviations are given in Table 5.1.

Table 5.1 Activities and abbreviations

	Type	Abbreviation
Activities	Walking on flat ground	WG
	Running	RN
	Jumping	JP
	Sit down/Stand up	SS

The sampling frequency of IMU sensors was configured as 40Hz. Each individual performed activities for a total of four minutes as one minute for each activity. A total

Figure 5.1 IMU sensor positions on the human body (Personal archive, 2020)

of 192000 acceleration data for each axis was obtained.

Recorded accelerometer outputs were noisy. To remove it, a low-pass filter was applied to the obtained dataset. Then, the magnitude of acceleration was calculated using the given formula in Equation 5.1 and added as a new feature to the current feature set including x, y, and z-axis values of the accelerometer.

$$| a |= \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{5.1}$$

The obtained dataset was normalized to eliminate the dominant characteristics of large values over small values before neural network training and test. For this purpose, min-max normalization was applied to features and the data range was scaled between 0 and 1.

All recorded data from all volunteers were divided into frames using 1.5-sec window sizes with 50% overlapping. The window-size was chosen due to the average cycle duration of activities. The frames were divided into two parts as 70% training and 30%

Table 5.2 Network names and input combinations

	Network Name	Input Set
Single sensor	N1	Chest
	N2	Shoulder
	N3	Leg
	N4	Arm
Two-sensors	N5	Chest-Leg
	N6	Chest-Arm
	N7	Chest-Shoulder
	N8	Leg-Shoulder
	N9	Leg-Arm
	N10	Shoulder-Arm
Three-sensors	N11	Chest-Shoulder-Leg
	N12	Leg-Shoulder-Arm
	N13	Leg-Arm-Chest
	N14	Shoulder-Arm-Chest
All sensors	N15	Shoulder-Arm-Chest-Leg

test.

To evaluate sensor positions for human activity recognition, a total of 15 different input sets were obtained using single or multiple sensor combinations. These combinations and neural network names corresponding to the input dataset are given in Table 5.2.

All input datasets were separately fed into MLP NNs for recognition of activities including walking on flat ground, running, jumping, and sit down-stand up. All hyperparameters including the number of the hidden layers, the neuron number in hidden layers, learning rate, optimization algorithm were determined by fine-tuning to have the best results.

Neural network applications and data processes were conducted on MATLAB 2018b. Network performances were evaluated in terms of precision, recall, and accuracy.

5.2 Results

To evaluate sensor positions for activity recognition, fifteen eural network models were proposed. Each input set was fed into MLP NNs to recognize the activity types including walking on flat ground, running, jumping, and sit down/stand up. A total of 6396 frames as 4477 frames for training and 1919 frames for the test was used. Each proposed neural network were trained and tested to have the best results by adjusting hyperparameters via fine-tuning.

Firstly, each sensor output of placed on various areas on the human body separately was fed into the proposed four MLP NNs. Table 5.3, Table 5.4, Table 5.5, and Table 5.6 presents the obtained results in terms of recall and precision. Each network has the best recognition results for sit down/stand up activities with respect to both recall and precision. In terms of accuracy; N1, N2, and N3 presents same results around %94. However, N4 consists of sensor outputs on shoulder has 89.8% success rate and offer the poorest results than other three networks.

Secondly, six different input sets were created with the dual combinations of sensors placed on the human body. Each input set was separately fed into MLP NNs. Table 5.7, Table 5.8, Table 5.9, and Table 5.10 present test results of the proposed models. Each network has the best recognition performance for sit-down/stand-up like MLP NNs consist of single-sensor inputs. N5, N6, N7, N8, N9, and N10 have accuracy rates at 95.67%, 94.9%, 95.45%, 95.65%, 94.0%, and 94.0%, respectively. The network results show that dual combinations of sensor outputs offer better results than single sensor outputs.

Table 5.3 MLP neural network performance for inputs which provided by IMU sensor on chest

		Actual Outputs				
	N1	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	94.8%	2.7%	2.2%	1.2%	93.9%
	RN	2.1%	95.0%	0.7%	0.6%	96.5%
	JP	2.8%	2.3%	94.1%	3.1%	91.9%
	SS	0.3%	0.0%	3.0%	95.1%	96.6%
	Recall	94.8%	95.0%	94.1%	95.1%	

Table 5.4 MLP neural network performance for inputs which provided by IMU sensor on shoulder

		Actual Outputs				
	N2	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	93.9%	4.2%	1.8%	2.1%	92.0%
	RN	3.5%	92.7%	2.5%	1.9%	92.1%
	JP	1.7%	2.3%	94.5%	1.6%	93.4%
	SS	0.9%	0.8%	1.3%	94.4%	96.9%
	Recall	93.9%	92.7%	94.5%	94.4%	

Table 5.5 MLP neural network performance for inputs which provided by IMU sensor on leg

		Actual Outputs				
	N3	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	95.5%	1.4%	2.7%	1.2%	94.7%
	RN	3.1%	93.6%	2.9%	1.4%	92.6%
	JP	1.3%	4.4%	92.8%	0.5%	93.7%
	SS	0.1%	0.6%	1.6%	96.9%	97.6%
	Recall	95.5%	93.6%	92.8%	96.9%	

Table 5.6 MLP neural network performance for inputs which provided by IMU sensor on arm

		Actual Outputs				
	N4	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	89.4%	5.7%	4.9%	4.0%	85.9%
	RN	4.2%	88.1%	3.7%	2.1%	89.8%
	JP	3.0%	4.6%	88.3%	0.5%	91.5%
	SS	3.4%	1.6%	3.1%	93.4%	92.0%
	Recall	89.4%	88.1%	88.3%	93.4%	

Table 5.7 MLP neural network performance for inputs which provided by IMU sensors on chest and leg

		Actual Outputs				
	N5	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	96.4%	2.3%	1.4%	1.4%	94.9%
	RN	2.0%	96.1%	3.2%	2.2%	92.8%
	JP	1.4%	1.5%	94.9%	1.1%	95.9%
	SS	0.2%	0.1%	0.5%	95.3%	99.1%
	Recall	96.4%	96.1%	94.9%	95.3%	

Thirdly, four different input sets were created with triple combinations of sensors placed on the human body. Each input set was fed into MLP NNs. Obtained results are presented in Table 5.13, Table 5.14, Table 5.15, and Table 5.16. Sit-down/Stand-up activity is the most recognizable activity in terms of precision and recall like neural networks networks consist of single sensor outputs and dual combinations of sensor

Table 5.8 MLP neural network performance for inputs which provided by IMU sensors on chest and arm

		Actual Outputs				
	N6	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	94.8%	3.1%	3.9%	0.8%	92.3%
	RN	1.6%	95.2%	1.9%	1.5%	95.0%
	JP	2.4%	1.2%	93.1%	1.2%	95.0%
	SS	1.2%	0.5%	1.1%	96.5%	97.1%
	Recall	94.8%	95.2%	93.1%	96.5%	

Table 5.9 MLP neural network performance for inputs which provided by IMU sensors on chest and shoulder

		Actual Outputs				
	N7	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	95.1%	1.9%	0.9%	1.6%	95.5%
	RN	3.2%	95.3%	2.6%	1.4%	92.9%
	JP	1.4%	1.8%	96.0%	1.6%	95.2%
	SS	0.3%	1.0%	0.5%	95.4%	98.1%
	Recall	95.1%	95.3%	96.0%	95.4%	

Table 5.10 MLP neural network performance for inputs which provided by IMU sensors on leg and shoulder

		Actual Outputs				
	N8	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	94.9%	3.2%	2.9%	0.5%	93.4%
	RN	1.7%	94.1%	1.4%	0.6%	96.2%
	JP	3.1%	1.5%	95.5%	0.8%	94.6%
	SS	0.3%	1.2%	0.2%	98.1%	98.2%
	Recall	94.9%	94.1%	95.5%	98.1%	

Table 5.11 MLP neural network performance for inputs which provided by IMU sensors on leg and arm

		Actual Outputs				
	N9	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	93.8%	1.1%	2.4%	1.3%	95.1%
	RN	3.7%	92.9%	2.8%	2.1%	90.6%
	JP	1.9%	4.7%	94.1%	1.4%	92.1%
	SS	0.6%	1.3%	0.7%	95.2%	97.3%
	Recall	93.8%	92.9%	94.1%	95.2%	

outputs. We see that the network N11 that use shoulder, chest, and leg provided the best accuracy with 97.27%. The accuracy is at least 95.35% for triple combinations of sensor outputs.

Table 5.12 MLP neural network performance for inputs which provided by IMU sensors on shoulder and arm

		Actual Outputs				
	N10	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	93.7%	2.1%	1.9%	2.5%	93.5%
	RN	3.3%	94.0%	3.2%	1.8%	91.8%
	JP	2.1%	3.7%	93.8%	1.2%	93.0%
	SS	0.9%	0.2%	1.1%	94.5%	97.7%
	Recall	93.7%	94.0%	93.8%	94.5%	

Table 5.13 MLP neural network performance for inputs which provided by IMU sensors on chest, shoulder, and leg

		Actual Outputs				
	N11	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	97.1%	2.4%	1.6%	0.8%	94.3%
	RN	1.0%	96.9%	0.8%	0.3%	97.8%
	JP	1.2%	0.4%	96.4%	0.2%	98.1%
	SS	0.7%	0.3%	1.2%	98.7%	97.8%
	Recall	97.1%	96.9%	96.4%	98.7%	

Table 5.14 MLP neural network performance for inputs which provided by IMU sensors on leg, shoulder, and arm

		Actual Outputs				
	N12	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	95.4%	2.4%	1.2%	1.7%	94.7%
	RN	1.9%	95.7%	2.9%	1.1%	94.1%
	JP	2.4%	1.2%	94.8%	1.3%	95.0%
	SS	0.3%	0.7%	1.1%	95.9%	97.8%
	Recall	95.4%	95.7%	94.8%	95.9%	

Table 5.15 MLP neural network performance for inputs which provided by IMU sensors on leg, arm, and chest

		Actual Outputs				
	N13	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	95.0%	0.9%	1.8%	0.8%	96.4%
	RN	2.3%	95.4%	1.9%	1.2%	94.6%
	JP	2.6%	2.3%	96.1%	3.1%	92.3%
	SS	0.1%	1.4%	0.2%	94.9%	98.2%
	Recall	95.0%	95.4%	96.1%	94.9%	

Finally, all sensor outputs placed on the human body was combined and feature number was maximized with totally 12 different inputs. These dataset was fed into

Table 5.16 MLP neural network performance for inputs which provided by IMU sensors on shoulder, arm, and chest

		Actual Outputs				
	N14	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	95.1%	1.8%	3.4%	0.4%	94.4%
	RN	1.9%	96.4%	2.0%	2.1%	94.1%
	JP	2.4%	1.2%	93.7%	1.3%	95.0%
	SS	0.6%	0.6%	0.9%	96.2%	97.8%
	Recall	95.1%	96.4%	93.7%	96.2%	

MLP NNs. Table 5.17 presents the obtained results. It show that increment in the feature numbers was affected adversely the accuracy. It has reduced around 2% and has obtained as 95.2%.

Table 5.17 MLP NN performance when all sensor outputs are combined

		Actual Outputs				
	N15	**WG**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	95.0%	2.3%	0.8%	1.4%	95.4%
	RN	4.3%	94.7%	2.2%	2.1%	91.6%
	JP	0.6%	1.8%	95.2%	0.6%	96.9%
	SS	0.1%	1.2%	1.8%	95.9%	96.8%
	Recall	95.0%	94.7%	95.2%	95.9%	

Table 5.18 shows the average accuracy and F_{score} values of all networks. N11 consisting of the chest, leg and shoulder sensor data presents the best results 97.27% accuracy rate and 0.97 F_{score} value.

Table 5.18 Evaluation of activity recognition performance for each neural network in terms of accuracy and F_{score}

Network Name	Accuracy	F_{score}
N1	94.75%	0.947
N2	93.87%	0.937
N3	94.70%	0.946
N4	89.80%	0.898
N5	95.67%	0.956
N6	94.90%	0.948
N7	95.45%	0.954
N8	95.65%	0.956
N9	94.00%	0.938
N10	94.00%	0.940
N11	97.27%	0.970
N12	95.45%	0.954
N13	95.35%	0.953
N14	95.35%	0.953
N15	95.20%	0.951

CHAPTER SIX
ACTIVITY RECOGNITION USING IMU SENSOR DATA AND DEEP LEARNING NEURAL NETWORKS

In this chapter, basic and complex activity recognition was performed using IMU sensor outputs and deep learning neural networks. For this purpose; wireless IMU sensors were placed on arm, chest, shank, and thigh parts of human body. Accelerometer and gyroscope sensor outputs were collected and recorded during various types of activity including writing on paper and hand-washing. Determined thirteen activities were divided into three categories as basic, complex, and all. The data belong to each activity category were fed into three neural network types of deep learning, namely CNN, LSTM NN and ConvLSTM NN. Then, the network performances were separately examined for each category.

6.1 Method

To gather acceleration and gyroscope outputs, designed wearable wireless IMU sensors introduced in Chapter 3 was used. IMU sensors were attached on the thigh, shank, arm and chest parts of the human body as seen from Figure 6.1.

IMU sensor data was collected from 60 healthy users (37 males and 23 females) with age ranging from 20 to 40 (mean 25.9±3.8), height 160-195 cm (mean 179±3.0) and weight 50-103 kg (mean 76.8±3.4). Each user was briefed on what the study is about and was asked to perform a total of thirteen different activities, which is combination of eight basic and five complex. Table 6.1 gives performed all activities and their abbreviations.

Figure 6.1 Placements of IMU slave devices on human body (Personal archive, 2020)

Table 6.1 Categories and activities

	All Activities	Abbreviation
Basic Activities	Walking on flat ground	WG
	Walking uphill	WU
	Walking downhill	WD
	Ascending stairs	ASC
	Descending stairs	DES
	Running	RN
	Jumping	JP
	Sit down - Stand up	SS
Complex Activities	Cleaning table	CT
	Drink from cup	DC
	Open-close refrigerator	OCR
	Writing on paper	WP
	Hand washing	HW

Figure 6.2 shows several pictures of the data collection moment for four different activity types, which is descending stairs, writing on paper, walking downhill, and running.

For each subject, 65-minutes of data were collected for thirteen different activities by recording 5 minutes of data per activity. Before the actual activity, each sensor was calibrated by gathering data while the subject is in a standing position. A capture of

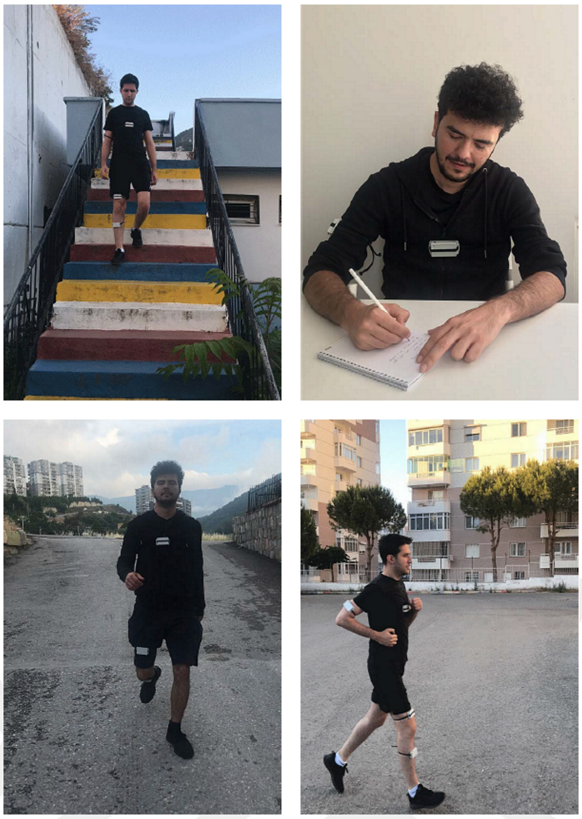

Figure 6.2 Data collection moment during some activities (Personal archive, 2020)

activity signals taken from thigh part of leg and arm during walking uphill and cleaning table is given in Figure 6.3 and Figure 6.4, respectively.

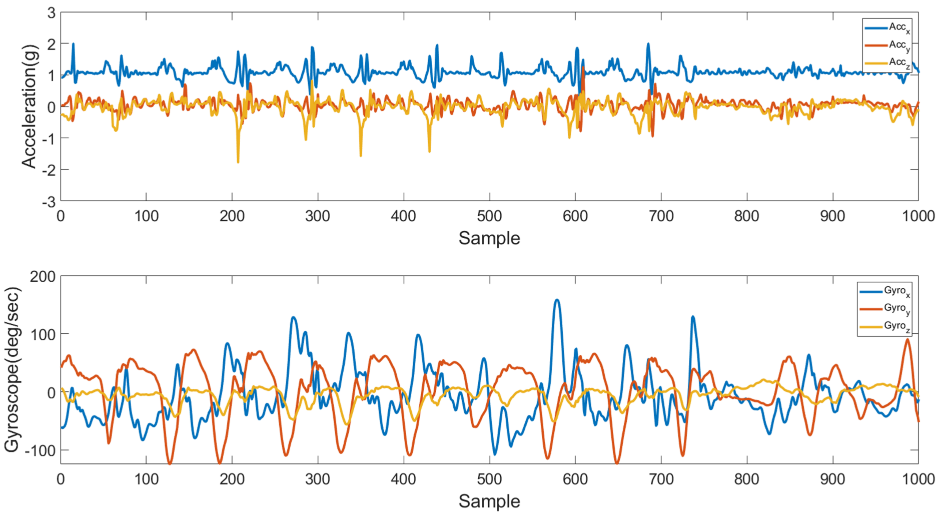

Figure 6.3 A capture of raw acceleration and gyroscope data for walking uphill from thigh part of leg

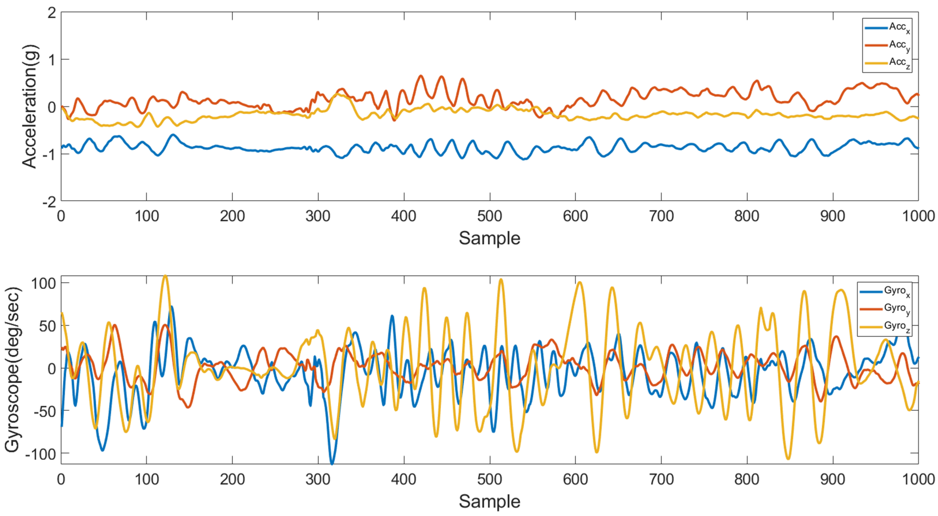

Figure 6.4 A capture of raw acceleration and gyroscope data for cleaning table from arm

During activities, incoming accelerometer signals from sensors were noisy. To remove noise, moving average filter was used. The data range of accelerometer and gyroscope signals was not at the same scale. To uniform data range, min-max normalization was separately applied on each feature of the dataset. This is important to prevent dominant characteristics of large values according to fewer values during training. Then, the signals were divided into frames with non-overlapping sliding windows. The main challenge was to find the correct window size. In the literature, researchers have used different window sizes to segment data such as 1-second window size without overlapping, 2-seconds window size with 1-second overlapping and so on (Karantonis et al., 2006; Preece et al., 2008). The small window size affects adversely computational cost, and wide window size may cause overfitting. In this study, we determined window size as 1.5-seconds by observing the average period duration of activities. Due to this selected window-size, we obtained total of 12000 frames for each activity and totally 156000 frames from all activities. Then, data augmentation was performed by adding noise into frames and number of frames was increased five times.

The produced frames were divided into two parts as 70% training and 30% testing. Then, the frames belong to each category fed into three different deep learning neural networks; CNN, LSTM NN, and ConvLSTM NN. Hyperparameters were calibrated to obtain the best network performances by fine-tuning.

Training and test processes of deep learning applications were performed on graphical processor unit (GPU) to accelerate working time (Cheng et al., 2012; Kouris & Koutsouris, 2011). The specifications of the chosen GPU are given in Table 6.2. Processes were conducted in deep learning toolbox and Experiment Manager of MATLAB 2020a.

Table 6.2 Specification of GPU in experiments

Parameters	Value
Graphic Card Chip	Geforce RTX2070 Super
Graphic Memory Capacity	8GB DDR6
Clock Frequency	1770MHz
Cuda Kernels	2560
Memory Bandwidth	448GB/sec
Memory Speed	14Gbps

6.2 Results

The recognition performances of the proposed models for the test frames are evaluated in this section. We divided activities into three categories as basic, complex, and all. The datasets belong to each category were separately applied to CNN, LSTM NN, and ConvLSTM NN. To have the best performances from networks, different architectures were constructed and the values of hyperparameters were determined by fine-tuning. For each activity, 42000 and 18000 frames were taken as training and test sets, respectively. Total frame numbers used in networks changed depending on the number of activities in each category. The network performances for the test frames will be discussed for each category.

6.2.1 Classification of Basic Activity Category

Basic activities consist of walking on flat ground, walking uphill, walking downhill, ascending stairs, descending stairs, running, jumping, and sit down - stand up. These motions are gait-based and approximately periodic signals. The motion speed causes minor changes in the duration of a period.

Firstly, CNN was applied for the classification of basic activities. The best network performance was obtained from architecture; consists of three convolutional layers, three RELU activation functions, three pooling layers, three fully-connected layers, one softmax activation function, and one classification layer. In the first convolutional layer, 96 kernels with 3×3 size were applied to the input frameset. The output of the convolutional layer went through the RELU activation function for non-linearity. In the pooling layer, 2x2 maximum pool filters reduced the dimension of extracted frames from the convolutional layer. 128 kernels with 3x3 size and 2x2 maximum filters for second convolutional and pooling layers and, 256 kernels with 3x3 size and 2x2 maximum filters for third convolutional and pooling layers were used for feature extraction. Sliding steps of filters were adjusted as one and two in convolutional layers and pooling layers, respectively. The extracted information was taken as input for neural network with three fully-connected layers. The number of neurons in layers were determined as 128, 192, and 8, respectively. The output of the last fully-connected layer by passing through softmax activation function fed into classification layer for decision. The other important hyperparameters as well as size and numbers of filters are learning rate, weight optimization algorithm, and mini-batch size for the network performance. These parameters were determined as 0.008, adaptive moment optimization, and 64, respectively.

Table 6.3 Confusion matrix obtained from CNN for classification of basic activities

		Target Class								
		WG	**WU**	**WD**	**ASC**	**DES**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	91.0%	1.5%	2.4%	1.2%	3.0%	1.5%	0.8%	0.2%	89.5%
	WU	2.7%	90.2%	3.6%	2.5%	2.0%	1.2%	0.7%	0.3%	87.4%
	WD	2.0%	2.2%	90.4%	2.0%	1.2%	1.3%	0.3%	1.4%	89.6%
	ASC	2.2%	2.4%	1.2%	91.0%	4.7%	2.0%	1.0%	0.3%	86.8%
	DES	1.9%	3.0%	1.9%	2.8%	88.0%	1.9%	0.2%	1.7%	86.7%
	RN	0.2%	0.3%	0.2%	0.1%	0.7%	90.1%	4.8%	0.7%	92.7%
	JP	0.0%	0.4%	0.2%	0.0%	0.0%	1.9%	92.0%	0.9%	96.4%
	SS	0.0%	0.0%	0.1%	0.4%	0.4%	0.1%	0.2%	94.5%	98.7%
	Recall	91.0%	90.2%	90.4%	91.0%	88.0%	90.1%	92.0%	94.5%	

Secondly, LSTM NN was applied for classification. The neurons in layer have particular memory units to remember information for a longer time. The most successful network performance was obtained from architecture with three LSTM layers, three fully-connected layer, one softmax activation function, and one classification layer. The neuron numbers in LSTM layers were taken as 128, 192 and 256, respectively. The output of the last LSTM layer fed into the fully-connected layer with 64, 96, and 8 neurons, respectively. Softmax activation function determined the probabilistic distribution of potential outcomes. Classification layer decided on an activity label using income information from activation function. The other hyperparameters; weight optimization algorithm, learning rate, and mini-batch size were set to adaptive moment estimation, 0.007, and 128, respectively.

Table 6.4 Confusion matrix obtained from LSTM NN for classification of basic activities

		Target Class								
		WG	**WU**	**WD**	**ASC**	**DES**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	90.5%	2.2%	1.4%	2.6%	0.8%	1.2%	1.4%	0.5%	89.9%
	WU	2.8%	90.1%	2.9%	1.9%	2.2%	1.0%	1.7%	0.1%	87.7%
	WD	2.4%	1.4%	89.0%	2.1%	1.8%	1.6%	1.2%	1.4%	88.2%
	ASC	1.4%	2.7%	2.7%	90.9%	3.3%	1.3%	2.8%	2.3%	84.6%
	DES	2.0%	1.9%	2.5%	1.4%	90.7%	1.4%	0.2%	2.7%	88.2%
	RN	0.3%	0.8%	0.8%	0.5%	0.8%	91.5%	1.7%	1.7%	93.2%
	JP	0.2%	0.4%	0.3%	0.6%	0.3%	1.1%	90.3%	1.8%	95.0%
	SS	0.4%	0.5%	0.4%	0.0%	0.1%	0.9%	0.7%	94.5%	96.7%
	Recall	90.5%	90.1%	89.0%	90.9%	90.7%	91.5%	90.3%	94.5%	

Thirdly, ConvLSTM NN was applied for classification. In this neural network, convolutional and pooling layers of CNN was added to LSTM layers for feature extraction. Extracted features from CNN layers were used as input for LSTM layers. The architecture of feature extraction layers, filter sizes, and numbers used in CNN was not changed. The number of neurons was determined as 64 and 96 in LSTM layers. The outputs of last LSTM layer fed into the fully-connected layer with 8 neurons. The classification results were obtained with passing through softmax activation function of the fully-connected layer. Hyperparameters were tuned to stochastic gradient descent algorithm, 0.03, 64, and 0.8 for weight optimizer, learning rate, mini-batch size, and momentum coefficient, respectively.

The classification performance of each neural network based on eight defined activity types is given in Table 6.3, Table 6.4, and Table 6.5. ConvLSTM improves

Table 6.5 Confusion matrix obtained from ConvLSTM NN for classification of basic activities

		Target Class								
		WG	WU	WD	ASC	DES	RN	JP	SS	Precision
Predicted Class	WG	96.0%	1.2%	1.8%	1.5%	1.4%	0.7%	0.5%	0.8%	92.3%
	WU	0.8%	94.0%	0.8%	2.2%	1.9%	2.6%	1.8%	0.3%	90.0%
	WD	0.9%	0.8%	93.7%	1.7%	1.4%	1.2%	1.1%	0.2%	92.7%
	ASC	0.8%	1.9%	2.2%	92.5%	2.0%	1.8%	2.1%	0.5%	89.1%
	DES	1.2%	1.5%	0.4%	1.2%	91.9%	1.0%	1.5%	0.4%	92.7%
	RN	0.2%	0.4%	0.6%	0.3%	0.4%	92.3%	0.3%	0.3%	97.3%
	JP	0.1%	0.0%	0.5%	0.4%	0.6%	0.2%	92.5%	0.6%	97.4%
	SS	0.0%	0.2%	0.0%	0.2%	0.4%	0.2%	0.2%	96.9%	98.7%
	Recall	96.0%	94.0%	93.7%	92.5%	91.9%	92.3%	92.5%	96.9%	

the recognition of each activity between 0.5% and 5.5% with respect to the other two networks. In Figure 6.5, the comparison of classification accuracy is given for all three networks. This shows that LSTM NN and CNN have almost the same average performances for basic activity recognition. However, ConvLSTM NN outperforms other two networks and gives 93.7% average classification accuracy.

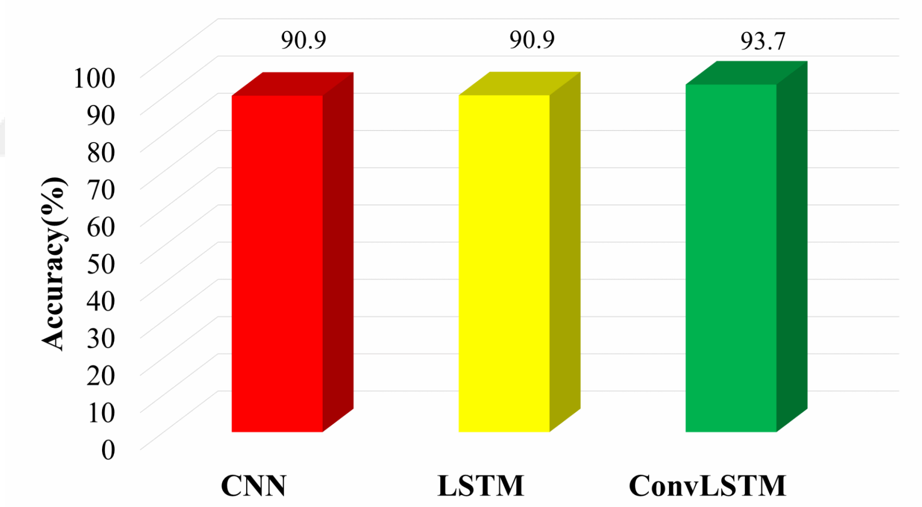

Figure 6.5 Performance comparison of neural networks in term of accuracy for basic activities

6.2.2 Classification of Complex Activity Category

Complex activities consist of cleaning table, drink from cup, open-close refrigerator, hand washing, and writing on paper. These activities are based on hand gesture.

Firstly, CNN was used for classification. The network has presented the best performance with the same CNN architecture that was used for the classification of basic activities. The only difference in this CNN comes from hyperparameter settings. The number of kernels was decreased to 64, 96, and 192 for convolutional layers and, kernel sizes were fixed as 3x3. In pooling layers, 2x2 maximum pooling was used for sub-sampling. The sliding steps were adjusted as one stride for kernels and two strides for maximum pooling. The extracted features fed into the neural network with three fully-connected layers. The neuron numbers of fully-connected layers were adjusted as 64, 96, and 5, respectively. Optimization algorithm, learning rate, and mini-batch size were chosen as adaptive moment estimation, 0.01, and 64, respectively.

Table 6.6 Confusion matrix obtained from CNN for classification of complex activities

		Target Class					
		CT	**DC**	**OCR**	**WP**	**HW**	**Precision**
Predicted Class	**CT**	90.1%	2.5%	3.1%	1.4%	2.1%	90.8%
	DC	2.1%	91.5%	2.5%	2.3%	3.0%	90.2%
	OCR	2.3%	2.2%	90.9%	2.1%	2.7%	90.7%
	WP	1.8%	1.8%	1.0%	92.0%	2.3%	93.0%
	HW	3.7%	2.0%	2.5%	2.2%	89.9%	89.6%
	Recall	90.1%	91.5%	90.9%	92.0%	89.9%	

Secondly, LSTM NN was used for classification. The most successful results were obtained from architecture consists of three LSTM layers, two fully-connected layers, softmax activation function, and classification layer. The neuron numbers in LSTM layers were determined for 128, 256, and 96. The output in LSTM layers fed into fully-connected layer with 32 and 5 neurons, respectively. The other parameters were determined as adaptive moment estimation, 0.001, and 64 for optimizer, learning rate, and mini-batch size, respectively.

Table 6.7 Confusion matrix obtained from LSTM NN for classification of complex activities

		Target Class					
		CT	**DC**	**OCR**	**WP**	**HW**	**Precision**
Predicted Class	**CT**	94.7%	1.6%	2.0%	2.1%	2.0%	92.4%
	DC	0.8%	95.0%	2.3%	2.4%	0.9%	93.6%
	OCR	2.5%	1.2%	91.5%	1.2%	0.9%	94.0%
	WP	0.7%	1.2%	1.2%	93.7%	1.1%	95.7%
	HW	1.3%	1.0%	3.0%	0.6%	95.1%	94.1%
	Recall	94.7%	95.0%	91.5%	93.7%	95.1%	

Thirdly, ConvLSTM NN was used for classification. The extracted features in CNN fed into LSTM. The kernel sizes and number of kernels has the same hyperparameter settings used in feature extraction layer of CNN architecture. LSTM were adjusted as two layers, and the neuron numbers in LSTM layers were determined as 128 and 128. The last output of LSTM fed into fully connected layer with 5 neurons. The best results were obtained with adaptive moment estimation, 0.009 learning rate, and 128 mini-batch size.

Table 6.8 Confusion matrix obtained from ConvLSTM NN for classification of complex activities

		Target Class					
		CT	**DC**	**OCR**	**WP**	**HW**	**Precision**
Predicted Class	**CT**	90.3%	3.5%	2.2%	2.3%	1.0%	90.9%
	DC	1.7%	91.0%	3.4%	4.1%	1.3%	89.6%
	OCR	3.9%	1.7%	89.7%	1.9%	2.0%	90.4%
	WP	1.8%	1.6%	2.3%	89.9%	3.7%	90.5%
	HW	2.3%	2.2%	2.4%	1.8%	92.0%	91.3%
	Recall	90.3%	91.0%	89.7%	89.9%	92.0%	

The classification performance of neural networks was separately given for each activity type in Table 6.6, Table 6.7, and Table 6.8. ConvLSTM NN gives better performance between 0.6% and 6.2% than the other two networks for each activity. Figure 6.6 shows average classification performance for all networks. ConvLSTM gives the best performance with an average success rate of 94.0%. CNN and LSTM NN have approximately same average classification performance as 90.8% and 90.5%, respectively.

6.2.3 Classification of All Activity Category

All activity category consists of both basic and complex activities.

Firstly, CNN was applied for classification. The best results were obtained from the architecture, which made up of three convolutional layers, three RELU activation functions, three pooling layers, four fully-connected layers, one softmax activation, and one classification layer. Kernel sizes were set to 3x3, 4x4, and 3x3, and number

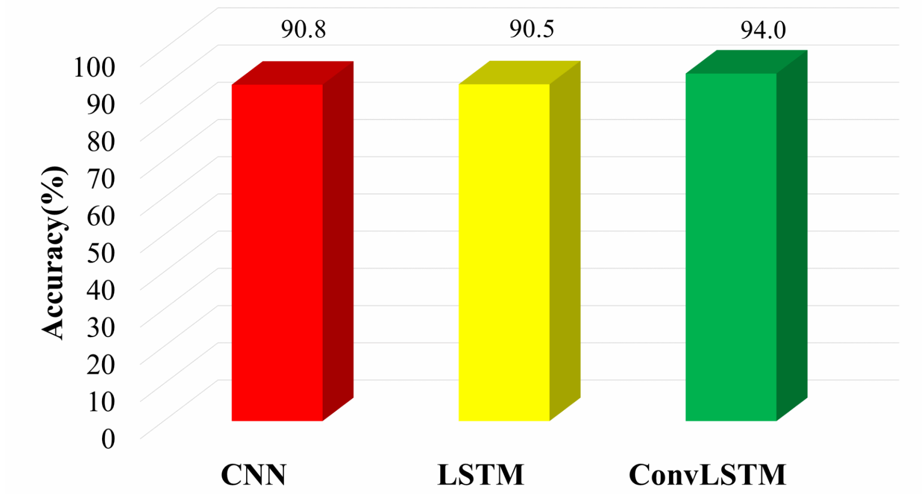

Figure 6.6 Performance comparison of neural networks in term of accuracy for complex activities

of kernels was adjusted as 128, 192, and 256 for convolutional layers. Activation functions were applied to the outputs of all convolutional layers for non-linearity. Three maximum pooling filters with 3x3, 2x2, and 2x2 size were used to reduce the dimension of features and choose dominant characteristic of extracted features from convolutional layers. Sliding strides were tuned to one and two for each convolutional layer and each pooling layer, respectively. The extracted frames fed into neural network include four fully-connected layers. From the beginning to the end of the network, there are 192, 256, 128, and 13 neurons in these layers. The other hyperparameters such as learning rate, weight optimization algorithm, momentum coefficient, and mini-batch size were taken as 0.005, stochastic gradient descent momentum, 0.6, and 128, respectively.

Table 6.9 Confusion matrix obtained from CNN for classification of all activities

		Target Class													
		WG	WU	WD	ASC	DES	RN	JP	SS	CT	HW	DC	WP	OCR	Precision
Predicted Class	WG	86.1%	4.0%	4.9%	2.1%	2.2%	1.9%	1.1%	1.0%	0.0%	0.0%	0.0%	0.0%	0.2%	83.2%
	WU	3.2%	80.3%	5.2%	3.9%	2.3%	1.0%	0.9%	0.6%	0.2%	0.1%	0.2%	0.0%	0.2%	81.9%
	WD	2.4%	3.6%	84.1%	4.1%	1.2%	3.1%	2.0%	1.0%	0.0%	0.0%	0.2%	0.2%	0.1%	82.4%
	ASC	3.9%	3.5%	1.4%	86.1%	6.9%	2.1%	1.7%	0.9%	0.4%	0.1%	0.0%	0.1%	0.1%	80.3%
	DES	2.8%	2.7%	2.8%	1.8%	84.7%	8.4%	0.8%	0.8%	0.0%	0.0%	0.3%	0.2%	0.3%	80.2%
	RN	1.5%	3.7%	0.5%	0.2%	1.5%	82.8%	0.4%	2.7%	0.1%	0.1%	0.3%	0.4%	0.2%	87.7%
	JP	0.0%	1.1%	0.2%	1.0%	0.8%	0.4%	92.8%	3.3%	0.5%	0.0%	0.7%	0.3%	0.5%	91.3%
	SS	0.1%	0.7%	0.4%	0.4%	0.1%	0.0%	0.1%	89.1%	0.0%	0.0%	0.2%	0.0%	0.1%	97.7%
	CT	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.1%	0.2%	80.2%	4.9%	5.1%	2.3%	2.5%	84.0%
	HW	0.0%	0.1%	0.1%	0.0%	0.1%	0.1%	0.0%	0.0%	3.7%	87.0%	6.9%	2.4%	4.9%	82.6%
	DC	0.0%	0.0%	0.1%	0.0%	0.2%	0.2%	0.0%	0.1%	4.1%	3.1%	80.1%	5.9%	4.2%	81.7%
	WP	0.0%	0.2%	0.2%	0.1%	0.0%	0.0%	0.1%	0.0%	2.7%	1.8%	2.6%	83.9%	6.6%	85.4%
	OCR	0.0%	0.1%	0.1%	0.1%	0.0%	0.0%	0.0%	0.3%	8.1%	2.9%	3.4%	4.2%	80.1%	80.7%
	Recall	86.1%	80.3%	84.1%	86.1%	84.7%	82.8%	92.8%	89.1%	80.2%	87.0%	80.1%	83.9%	80.1%	

Table 6.10 Confusion matrix obtained from LSTM NN for classification of all activities

Predicted Class	WG	WU	WD	ASC	DES	RN	JP	SS	CT	HW	DC	WP	OCR	Precision
WG	88.2%	2.1%	4.2%	3.7%	3.2%	1.9%	1.3%	2.6%	0.3%	0.2%	0.4%	0.4%	0.3%	81.1%
WU	1.8%	87.3%	2.2%	3.1%	3.3%	2.5%	2.1%	2.0%	0.4%	0.1%	0.2%	0.3%	0.2%	82.7%
WD	2.4%	1.6%	86.9%	2.3%	2.1%	2.0%	2.4%	2.4%	0.1%	0.0%	0.2%	0.5%	0.4%	84.1%
ASC	2.3%	3.1%	1.6%	84.0%	4.6%	2.9%	1.5%	2.1%	0.2%	0.1%	0.0%	0.1%	0.1%	81.9%
DES	3.0%	2.2%	1.9%	3.2%	83.9%	1.2%	2.3%	2.7%	0.2%	0.0%	0.3%	0.7%	0.3%	82.3%
RN	2.1%	1.3%	1.2%	2.7%	1.7%	85.3%	2.3%	3.4%	0.4%	0.2%	0.3%	0.6%	0.7%	83.5%
JP	0.0%	1.2%	0.9%	0.3%	0.3%	1.3%	86.0%	3.3%	0.3%	0.1%	0.3%	0.3%	0.5%	91.6%
SS	0.1%	0.7%	1.4%	0.2%	0.4%	2.1%	1.3%	80.9%	0.3%	0.1%	0.2%	0.7%	0.7%	90.8%
CT	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%	0.2%	0.4%	80.4%	2.2%	4.3%	1.3%	3.2%	87.1%
HW	0.0%	0.1%	0.3%	0.2%	0.0%	0.2%	0.3%	0.1%	4.1%	91.1%	5.9%	3.4%	2.9%	83.9%
DC	0.0%	0.1%	0.2%	0.1%	0.4%	0.2%	0.1%	0.0%	3.7%	3.2%	82.1%	3.6%	5.4%	82.8%
WP	0.1%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	3.5%	0.9%	2.8%	84.0%	4.3%	87.6%
OCR	0.0%	0.1%	0.1%	0.1%	0.0%	0.3%	0.2%	0.0%	6.1%	1.8%	3.0%	4.1%	81.0%	83.7%
Recall	88.2%	87.3%	86.9%	84.0%	83.9%	85.3%	86.0%	80.9%	80.4%	91.1%	82.1%	84.0%	81.0%	

(Column headers: **Target Class**)

Table 6.11 Confusion matrix obtained from ConvLSTM NN for classification of all activities

Predicted Class	WG	WU	WD	ASC	DES	RN	JP	SS	CT	HW	DC	WP	OCR	Precision
WG	90.4%	1.4%	1.6%	2.3%	3.5%	1.0%	0.8%	1.4%	1.0%	0.2%	0.1%	0.3%	0.2%	86.8%
WU	1.4%	90.2%	1.8%	2.8%	1.3%	0.9%	1.4%	2.3%	0.3%	0.1%	0.3%	0.2%	0.3%	87.3%
WD	1.3%	0.4%	91.2%	1.4%	2.2%	1.1%	0.3%	3.5%	0.2%	0.0%	0.4%	0.2%	0.1%	89.1%
ASC	2.4%	1.4%	1.3%	89.5%	2.1%	1.3%	2.0%	2.0%	0.8%	0.1%	0.2%	0.6%	0.2%	86.1%
DES	2.9%	1.9%	1.1%	2.0%	86.1%	1.2%	1.9%	2.9%	0.4%	0.0%	0.3%	0.2%	0.3%	85.1%
RN	0.3%	0.8%	1.4%	1.2%	1.4%	92.4%	1.5%	0.2%	0.4%	0.1%	0.2%	0.1%	0.5%	91.9%
JP	0.4%	1.0%	0.3%	0.2%	2.1%	0.4%	91.0%	0.4%	0.3%	0.1%	0.2%	0.2%	0.1%	94.1%
SS	0.1%	1.0%	0.6%	0.1%	1.0%	1.3%	0.6%	86.7%	0.2%	0.1%	0.7%	0.2%	0.2%	93.4%
CT	0.0%	1.2%	0.2%	0.0%	0.1%	0.1%	0.0%	0.1%	88.6%	0.3%	2.7%	1.8%	3.8%	89.6%
HW	0.2%	0.2%	0.1%	0.0%	0.0%	0.1%	0.1%	0.2%	1.5%	93.6%	3.2%	2.4%	2.7%	89.9%
DC	0.5%	0.1%	0.2%	0.2%	0.1%	0.0%	0.2%	0.1%	2.4%	0.8%	88.1%	2.5%	3.1%	89.6%
WP	0.1%	0.2%	0.1%	0.2%	0.1%	0.2%	0.0%	0.1%	2.0%	2.4%	1.6%	87.8%	2.1%	90.6%
OCR	0.2%	0.2%	0.1%	0.1%	0.0%	0.0%	0.2%	0.1%	1.9%	2.2%	2.0%	3.5%	86.4%	89.2%
Recall	90.4%	90.2%	91.2%	89.5%	86.1%	92.4%	91.0%	86.7%	88.6%	93.6%	88.1%	87.8%	86.4%	

(Column headers: **Target Class**)

Secondly, LSTM NN was applied for classification. The architecture consists of five LSTM layers, three fully connected layers, one softmax activation function, and one classification layer. The neuron numbers used in LSTM layers are 256, 256, 324, 128, and, 128, respectively. Neuron numbers of fully connected layers set to 128, 64, and 13. Learning rate, optimization algorithm, and mini-batch size was taken as 0.006, adaptive moment estimation, and 128, respectively.

Thirdly, ConvLSTM NN was applied for classification. The feature extraction layer of CNN has been fully taken over with its own entire setup. LSTM layers have 96, 192 and, 128 neurons, respectively. The neuron numbers in fully connected layers were set to 32 and 13. Weight optimization algorithm, learning rate, and mini-batch size were set to adaptive moment estimation, 0.008, and 128, respectively.

Confusion matrices in Table 6.9, Table 6.10, and Table 6.11 shows the classification performance of all the networks concerning to the defined architecture and hyperparameters. When the classification accuracies are examined, it can be seen that ConvLSTM NN performs better than the other networks and gives an average of 89.3% success rate considering all activities. Figure 6.7 presents illustrative accuracy performance comparison for all three networks. Here, general performance of the networks for all activities is given.

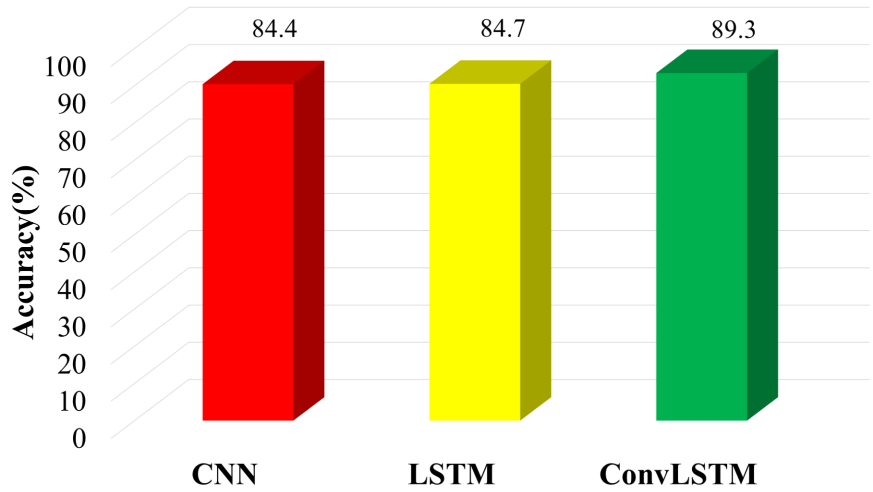

Figure 6.7 Performance comparison of neural networks in term of accuracy for basic activities

6.2.4 Performance Analysis of ConvLSTM NN During A Sequential Daily Activity

To analyze neural network response during sequential daily activities, test data was collected and recorded. The subject has performed in order of sit down - stand up (SS), walking on the ground (WG), ascending stairs (ASC), running (RN), and cleaning table (CT) activities during an average of 20-seconds. Recorded data were fed into ConvLSTM NN. Figure 6.8 presents the predicted classes of neural network and target classes.

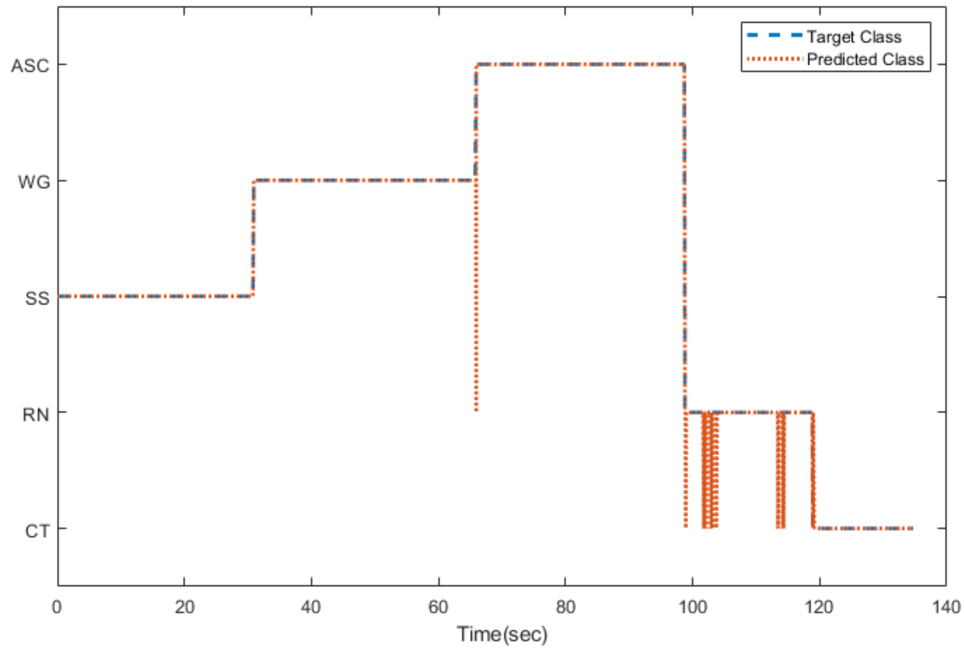

Figure 6.8 Response of ConvLSTM NN during sequential performed daily activities

The results show that neural networks may exhibit momentarily unexpected responses during the transition from one activity to another activity. As can be seen in Figure 6.8, neural networks misclassify at two transition points between classes. The first error occurred at the time of transition from WG to ASC and the second error occurred at the time of transition from ASC to RN. It has been interpreted that the reason for these errors is that the data belonging to both class types form the relevant frame. On the other hand, RN class errors are mistakes that occur in the performance of neural networks.

CHAPTER SEVEN
ACTIVITY RECOGNITION USING DIFFERENT SENSOR MODALITIES AND DEEP LEARNING NEURAL NETWORKS

In Chapter 5, activity recognition performance of deep learning neural networks was examined for different sensor inputs. For this purpose; an IMU sensor for each leg and a smart insole for each foot were used in experiments. With these sensors placed on the human body, subjects performed gait-based activities including ascending stairs and running. Obtained data has been categorized into three parts including only IMU sensor data, only smart insole data, and a combination of these sensor data. Each category was fed into ConvLSTM NN to compare network performances for different sensor modalities.

7.1 Method

In this study, totally four slave devices consist of two smart insoles and two IMU sensors introduced in Chapter 3 were used. As seen from Figure 7.1, smart insoles were placed on the shoe sole of each foot and IMU sensors were attached to the thigh part of each leg. The sampling frequency of all devices was configured as 40Hz.

Force-sensitive resistors used in smart insole design are a type of sensor that measures applied force based on physical pressure. These sensors have significant advantages including low cost, simple use, and thin surface. They can measure weight between 0.01kg to 10kg. However, applied force during performed activities may cause saturation. To overcome this challenge, FSR outputs were converted into foot contact states as “0” or “1” by using a threshold value. In other words, weight measurements were accepted as “No contact” when is less than a threshold value, otherwise accepted as “Contacted”. There are many similar types of FSRs that have different-size surface areas. We have preferred using circular and small-size FSRs for toe, LF, and MF areas while square and wider-size FSR is used for heel area in our insole design. These sensors have the same specifications except for size.

Figure 7.1 Placement of IMU sensors and smart insoles on the human body (Personal archive, 2021)

Total 40 healthy volunteers (30 males and 10 females) with age ranging from 20 to 32 (mean 26.4), height 163-195 cm (mean 178) and weight 50-103 kg (mean 78). Each user was informed about aim of the study and was requested to realize gait-based eight different activities given in Table 7.1.

Table 7.1 Performed gait-based activities

	Type	**Abbreviation**
Activities	Walking on flat ground	WG
	Walking uphill	WU
	Walking downhill	WD
	Ascending stairs	ASC
	Descending stairs	DES
	Running	RN
	Jumping	JP
	Sit down / Stand up	SS

For each subject, total 40-minutes of data were recorded for eight different activities by collecting 5 minutes of data per activity. Accelerometer outputs are noisy. A moving average filter was used to cancel out it. The data values of 3-axis accelerometer and 3-axis gyroscope outputs, and also foot contact states was not at the same range. To uniform data range, min-max normalization was separately applied for each feature of accelerometer and gyroscope. This is important during neural network training and testing to eliminate the dominant characteristics of large values over small values.

All recorded data from all volunteers was divided into frames using 1.5sec non-overlapping window sizes due to average cycle duration of activities. 8000 frames for each activity and total 64000 frames were obtained. Then, frames are divided into two sections as 70% training and 30% test. In other words, the data of 28 volunteers were used for training and the data of 12 volunteers were used for test processes.

Processed data was separated into three categories as only foot contact state, accelerometer and gyroscope outputs, and the combination of all data due to feature type. Each category fed into ConvLSTM NN, respectively. A comparison of network performances was done in terms of precision, recall, f-score, and accuracy for different input set.

The most popular deep learning neural network types are convolutional neural networks and long-short term memory neural networks. However, ConvLSTM NN has presented superior performance compared to these two neural networks in both the study performed in Chapter 4 and many studies in the literature (Ordóñez & Roggen, 2016; Wan et al., 2020). Therefore, we have used the hybrid model ConvLSTM NN. All hyperparameters were adjusted to have the best performance for each input category by fine-tuning.

All data processes and deep learning applications were conducted in MATLAB 2020a. To speed up the process duration, we have used GPU. Its specifications are NVIDIA RTX 2070 model, 8GB memory capacity and 14Gbps memory speed.

7.2 Results

In this study, a wireless sensor network was established with smart insoles and wearable IMU sensors. From 40-healthy people, datasets were collected and recorded from sensors placed sole of each foot and thigh part of each leg during eight basic activities including walking on flat ground, walking uphill, walking downhill, ascending stairs, descending stairs, running, jumping, and sit down/stand up. The obtained data sets were converted into foot contact state, acceleration, and angular rate. Figure 7.2, Figure 7.3, and Figure 7.4 illustrate a part of activity signals which are obtained from IMU sensor of left leg and FSRs of left foot during walking on flat ground.

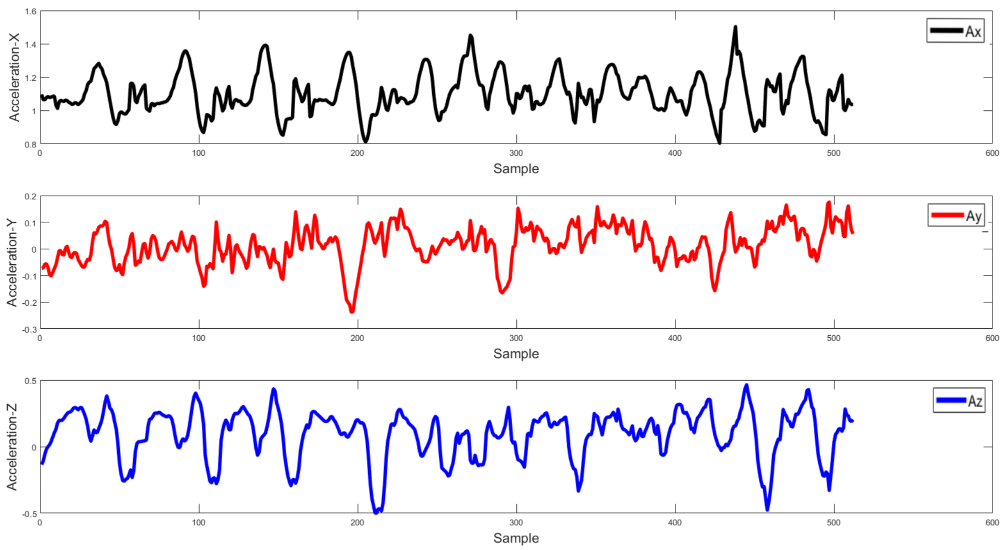

Figure 7.2 A part of acceleration data collected from thigh part of left leg during walking

Datasets were divided into three categories. The first category is based on only smart insole outputs, the second category is based on only IMU sensor outputs, and the third category is based on a combination of all sensor outputs. Each category was divided into 70% training and 30% test. The dataset that belongs to each category was fed into ConvLSTM NN to examine and compare the activity recognition performance. To have the best results, hyperparameters were determined by fine-tuning. The values of some hyperparameters that belong to architecture are given in Table 7.2.

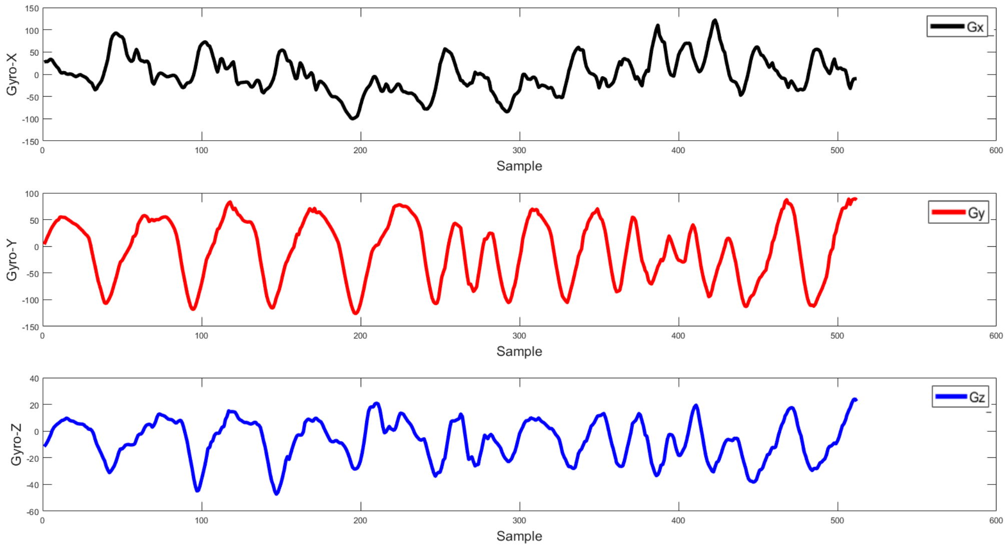

Figure 7.3 A part of gyroscope data collected from thigh part of left leg during walking

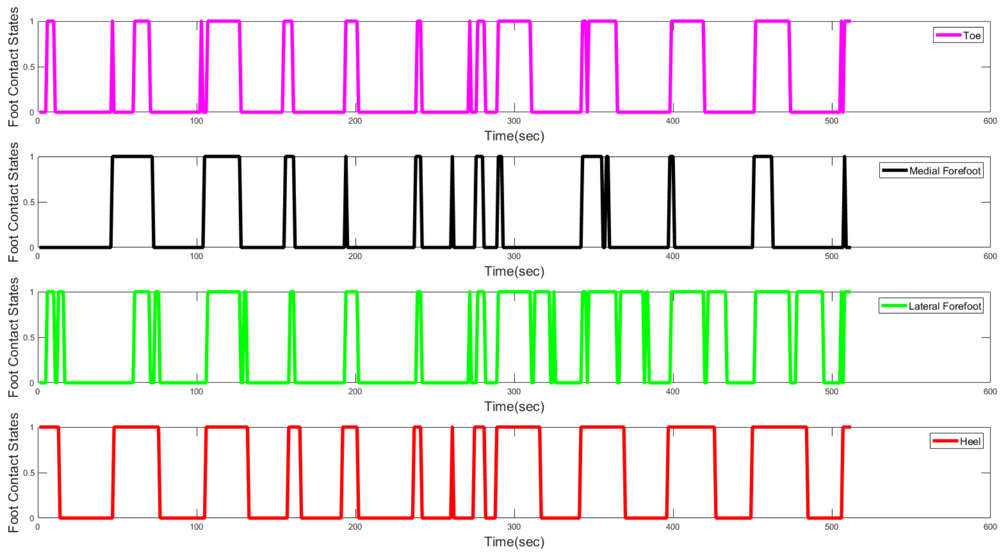

Figure 7.4 A part of foot contact states collected from sole of left foot during walking

The recognition performance of the neural networks for each category were presented in Table 7.3, Table 7.4, and Table 7.5. The results show that the neural network has the best performance for each activity when the input set is third category, which consists of combination of both IMU sensor and smart insole outputs. The sit-down/stand-up activity has the highest rate for each neural network in terms of precision and recall. The average accuracy of neural networks is 90.1%, 88.6%, and 93.4% for categories, respectively. Moreover, ConvLSTM NN with foot contact state input set offers better performance around 1.5% than ConvLSTM NN with IMU

Table 7.2 The values of some hyperparameters used in neural networks

Hyperparameter	FCS	AG	All data
Number of Convolutional layers	2	96-64	3x3
Number of Kernel filters in CNN	2	128-64	3x3
Number of Kernel filters in CNN	3	128-96-64	2x2
Kernel sizes in CNN	2	96-64	3x3
Filter sizes in pooling layers	2	128-64	3x3
Number of Pooling Layers	3	128-96-64	2x2
Number of LSTM Layers	2	128-64	3x3
Number of FC Layers	3	128-96-64	2x2

sensor dataset.

Table 7.3 The classification performance of ConvLSTM NN when input set is only foot contact states

		Target Class								
		WG	**WU**	**WD**	**ASC**	**DES**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	86.4%	4.5%	2.9%	2.8%	3.3%	1.3%	0.9%	0.6%	84.1%
	WU	4.6%	85.4%	4.6%	2.9%	2.8%	1.8%	0.6%	0.4%	82.8%
	WD	3.7%	2.9%	87.4%	4.1%	3.2%	1.7%	1.2%	1.3%	82.8%
	ASC	2.9%	3.6%	2.1%	85.0%	4.2%	2.2%	1.0%	1.3%	83.1%
	DES	2.2%	3.2%	2.5%	4.7%	85.8%	2.1%	0.2%	1.2%	84.2%
	RN	0.2%	0.3%	0.2%	0.1%	0.5%	90.5%	1.9%	0.5%	96.1%
	JP	0.0%	0.1%	0.2%	0.0%	0.0%	0.3%	93.0%	1.2%	98.1%
	SS	0.0%	0.0%	0.1%	0.4%	0.2%	0.1%	1.2%	93.5%	97.9%
	Recall	86.4%	85.4%	87.4%	85.0%	85.8%	90.5%	93.0%	93.5%	

Table 7.4 The classification performance of ConvLSTM NN when input set is accelerometer and gyroscope outputs

		Target Class								
		WG	**WU**	**WD**	**ASC**	**DES**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	88.5%	1.9%	1.8%	2.6%	2.9%	1.1%	1.0%	0.4%	88.3%
	WU	2.3%	89.2%	2.5%	2.9%	2.2%	0.9%	1.3%	0.2%	88.3%
	WD	2.5%	2.2%	88.3%	1.5%	2.7%	0.9%	1.4%	1.5%	87.4%
	ASC	2.2%	2.8%	3.1%	85.6%	3.3%	0.7%	1.4%	0.8%	85.7%
	DES	2.8%	1.5%	2.1%	6.7%	87.9%	0.9%	0.2%	0.2%	85.9%
	RN	1.3%	1.3%	0.9%	0.3%	0.4%	93.4%	1.9%	1.7%	92.3%
	JP	0.4%	0.5%	0.7%	0.4%	0.3%	0.8%	92.3%	0.5%	96.2%
	SS	0.0%	0.6%	0.6%	0.0%	0.3%	1.3%	0.5%	94.7%	96.6%
	Recall	88.5%	89.2%	88.3%	85.6%	87.9%	93.4%	92.3%	94.7%	

Table 7.6 shows F_{score} values of neural networks for each category. Combination of all dataset has better performance in terms of F_{score} with 0.958 than other two categories.

Table 7.5 The classification performance of ConvLSTM NN when input set is combination of all data

		Target Class								
		WG	**WU**	**WD**	**ASC**	**DES**	**RN**	**JP**	**SS**	**Precision**
Predicted Class	**WG**	93.0%	0.9%	2.3%	1.0%	2.3%	0.2%	0.9%	0.1%	92.4%
	WU	1.3%	94.2%	1.9%	3.1%	1.7%	0.9%	1.2%	0.4%	90.0%
	WD	1.4%	0.7%	92.3%	2.7%	2.4%	1.0%	1.1%	1.1%	92.7%
	ASC	1.7%	2.8%	2.7%	91.4%	2.6%	1.4%	1.3%	0.2%	87.8%
	DES	1.8%	0.7%	0.3%	0.8%	89.1%	1.1%	0.4%	0.5%	94.1%
	RN	0.5%	0.3%	0.1%	0.2%	0.5%	95.0%	0.5%	0.5%	97.3%
	JP	0.3%	0.1%	0.3%	0.6%	0.7%	0.2%	94.0%	0.2%	97.5%
	SS	0.0%	0.3%	0.1%	0.2%	0.7%	0.2%	0.6%	97.0%	97.9%
	Recall	93.0%	94.2%	92.3%	91.4%	89.1%	95.0%	94.0%	97.0%	

Table 7.6 F_{score} comparison of neural network performances for each input category

	FCS	**AG**	**All data**
F-score	0.947	0.939	0.958

CHAPTER EIGHT
CONCLUSION

8.1 Conclusion and Future Works

In this study, we have focused on the design of various wearable systems for motion analysis and their applications. The wearable systems consist of the smart garment, wireless IMU sensors and smart insoles. Applications are joint angle prediction for gait analysis and activity recognition.

Firstly, a smart garment for gait analysis has been presented. This garment was designed as similar as possible to conventional textile clothing to provide washability and maximum comfort to user. These features make the garment feasible for collecting relevant joint angles in daily activities and outdoor conditions. Wearing the garment is quite easy and there is no need any preparation time. Furthermore, presented garment is low-cost and suitable for mass production. The results demonstrated that the present system can be used to collect knee and ankle joint angles during walking on the ground and sit-down/stand-up activities.

Secondly, wearable wireless IMU sensors were designed to increase comfort and portability, and also reduce the duration of the data collection process. The designed monitoring system has superior advantages such as being low-cost, lightweight, and easy placement on various points of the human body. The sensors were placed on different body parts of the human body including the shoulder, chest, leg, and arm to find optimal sensor location for activity recognition. The results show that obtained sensor outputs from the immobile position, namely chest, presented the best performance for the neural network. Moreover, sensor outputs obtained arm part has affected network performance adversely for gait-based activities even if all outputs were combined.

Thirdly, activity recognition performance of various deep learning neural networks including CNN, LSTM, and ConvLSTM NN was examined for thirteen activities including writing on a paper, hand-washing, and walking. Activities were categorized as basic, complex, and all. These categories were fed into three neural network types. The results show that ConvLSTM NN has presented the best results for each category. We can make a comment that a hybrid algorithm that combines CNN and LSTM NN outperforms with respect to the activity recognition results than others. Moreover, the results presented that basic activities consist of gait-based motions are more recognizable with available sensor positions for neural networks.

Finally, we have designed a wireless smart insole. It is a low-cost, lightweight, and compact solution. Our design has no visible wire. Moreover, it provides walking comfort with memory foam soles for users. A smart insole was placed on the sole of each foot and an IMU sensor was attached to each leg. We have collected the data for basic activities. These inputs were separately fed into ConvLSTM NN. The results have shown that provided data by smart insole and provided data by IMU sensors offer similar results. Considering the easier use and performance of the designed smart insole, it is seen that it may be sufficient for gait-based activities. Moreover, increasing the data feature by combining these sensor outputs has improved the results.

As a future work, we are planning to make a study about sport activity recognition such as playing football, basketball, tennis, etc. using different sensor modalities and deep learning. Moreover, a public benchmark dataset by expanding available dataset in respect to activity types and sample size can be presented.

REFERENCES

Adaskevicius, R. (2014). Method for recognition of the physical activity of human being using a wearable accelerometer. *Elektronika ir Elektrotechnika*, *20*(5), 127–131.

Ahmad, J., Muhammad, K., & Baik, S. W. (2017). Data augmentation-assisted deep learning of hand-drawn partially colored sketches for visual search. *PloS one*, *12*(8), 0183838.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1-6.

Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., et al. (2013). A public domain dataset for human activity recognition using smartphones. *European Symposium on Artificial Neural Networks*, *3*, 3.

Artikis, A., Sergot, M., & Paliouras, G. (2010). A logic programming approach to activity recognition. *Proceedings of the 2nd ACM International Workshop on Events in Multimedia*, 3–8.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Bersch, S. D., Azzi, D., Khusainov, R., Achumba, I. E., & Ries, J. (2014). Sensor data acquisition and processing parameters for human activity classification. *Sensors*, *14*(3), 4239–4270.

Bharti, P., De, D., Chellappan, S., & Das, S. K. (2018). Human: Complex activity recognition with multi-modal multi-positional body sensing. *IEEE Transactions on Mobile Computing*, *18*(4), 857–870.

Bhosale, T., Kudale, H., Kumthekar, V., Garude, S., & Dhumal, P. (2015). Gait analysis using wearable sensors. *2015 International Conference on Energy Systems and Applications*, 267–269.

Bilro, L., Oliveira, J., Pinto, J., & Nogueira, R. (2011). A reliable low-cost wireless and wearable gait monitoring system based on a plastic optical fibre sensor. *Measurement Science and Technology*, *22*(4), 045801.

Chambers, H. G. & Sutherland, D. H. (2002). A practical guide to gait analysis. *JAAOS-Journal of the American Academy of Orthopaedic Surgeons*, *10*(3), 222–231.

Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., Millán, J. d. R., & Roggen, D. (2013). The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, *34*(15), 2033–2042.

Cheng, J., Chen, X., & Shen, M. (2012). A framework for daily activity monitoring and fall detection based on surface electromyography and accelerometer signals. *IEEE Journal of Biomedical and Health Informatics*, *17*(1), 38–45.

Chernbumroong, S., Cang, S., Atkins, A., & Yu, H. (2013). Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, *40*(5), 1662–1674.

Chikhaoui, B. & Gouineau, F. (2017). Towards automatic feature extraction for activity recognition from wearable sensors: a deep learning approach. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 693–702.

Cho, H. (2017). Design and implementation of a lightweight smart insole for gait analysis. *2017 IEEE Trustcom/BigDataSE/ICESS*, 792–797.

Claesen, M. & De Moor, B. (2015). Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*.

Coker, J., Chen, H., Schall, M. C., Gallagher, S., & Zabala, M. (2021). Emg and joint angle-based machine learning to predict future joint angles at the knee. *Sensors*, *21*(11), 3622.

Cuadrado, J., Michaud, F., Lugrís, U., & Pérez Soto, M. (2021). Using accelerometer data to tune the parameters of an extended kalman filter for optical motion capture: Preliminary application to gait analysis. *Sensors*, *21*(2), 427.

Cuesta-Vargas, A. I., Galán-Mercant, A., & Williams, J. M. (2010). The use of inertial sensors system for human motion analysis. *Physical Therapy Reviews*, *15*(6), 462–473.

Ghoussayni, S., Stevens, C., Durham, S., & Ewins, D. (2004). Assessment and validation of a simple automated method for the detection of gait events and intervals. *Gait & Posture*, *20*(3), 266–272.

Gonçalves, H. R., Rodrigues, A., & Santos, C. P. (2021). Gait monitoring system for patients with parkinson's disease. *Expert Systems with Applications*, *185*, 115653.

Guo, J., Zhou, X., Sun, Y., Ping, G., Zhao, G., & Li, Z. (2016). Smartphone-based patients' activity recognition by using a self-learning scheme for medical monitoring. *Journal of Medical Systems*, *40*(6), 1–14.

Hagan, M. T., Demuth, H. B., & Beale, M. (1997). *Neural network design*. USA: PWS Publishing Co.

Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, *81*, 307–313.

Hausdorff, J. M. (2009). Gait dynamics in parkinson's disease: common and distinct behavior among stride length, gait variability, and fractal-like scaling. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *19*(2), 026113.

Higgins, W. T. (1975). A comparison of complementary and kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, *3*, 321–325.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Janson, S. W. (2005). Aerospace applications of mems. *MEMS/MOEMS Components and Their Applications II*, *5717*, 1–13.

Jarchi, D., Pope, J., Lee, T. K., Tamjidi, L., Mirzaei, A., & Sanei, S. (2018). A review on accelerometry-based gait analysis and emerging clinical applications. *IEEE Reviews in Biomedical Engineering*, *11*, 177–194.

Jeon, T. H. & Lee, J. K. (2018). Imu-based joint angle estimation under various walking and running conditions. *Journal of the Korean Society for Precision Engineering*, *35*(12), 1199–1204.

Jobanputra, C., Bavishi, J., & Doshi, N. (2019). Human activity recognition: A survey. *Procedia Computer Science*, *155*, 698–703.

Jolly, S., Iwana, B. K., Kuroki, R., & Uchida, S. (2018). How do convolutional neural networks learn design? *2018 24th International Conference on Pattern Recognition (ICPR)*, 1085–1090.

Kaghyan, S. & Sarukhanyan, H. (2012). Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer. *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria*, *1*, 146–156.

Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & 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*, *10*(1), 156–167.

Keatmanee, C., Baber, J., & Bakhtyar, M. (2013). Simple example of applying extended kalman filter. *1st International Electrical Engineering Congress, Chiangmai city, Thailand*.

Kok, M., Hol, J. D., & Schön, T. B. (2017). Using inertial sensors for position and orientation estimation. *arXiv preprint arXiv:1704.06053*.

Kouris, I. & Koutsouris, D. (2011). Activity recognition using smartphones and wearable wireless body sensor networks. *International Conference on Wireless Mobile Communication and Healthcare*, 32–37.

Kuo, C.-C. J. (2016). Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, *41*, 406–413.

Lago, P., Lang, F., Roncancio, C., Jiménez-Guarín, C., Mascu, R., & Bonnefond, N. (2017). The contextact@ a4h real-life dataset of daily-living activities. *International and Interdisciplinary Conference on Modeling and Using Context*, 175–188.

Lee, Y.-D. & Chung, W.-Y. (2009). Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring. *Sensors and Actuators B: Chemical*, *140*(2), 390–395.

Leung, H. & Haykin, S. (1991). The complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, *39*(9), 2101–2104.

Lim, H., Kim, B., & Park, S. (2020). Prediction of lower limb kinetics and kinematics during walking by a single imu on the lower back using machine learning. *Sensors*, *20*(1), 130.

Lin, F., Wang, A., Zhuang, Y., Tomita, M. R., & Xu, W. (2016). Smart insole: A wearable sensor device for unobtrusive gait monitoring in daily life. *IEEE Transactions on Industrial Informatics*, *12*(6), 2281–2291.

Lin, G. & Shen, W. (2018). Research on convolutional neural network based on improved relu piecewise activation function. *Procedia Computer Science*, *131*, 977–984.

Lorussi, F., Carbonaro, N., De Rossi, D., Paradiso, R., Veltink, P., & Tognetti, A. (2016). Wearable textile platform for assessing stroke patient treatment in daily life conditions. *Frontiers in Bioengineering and Biotechnology*, *4*, 28.

Low, J.-H., Chee, P.-S., Lim, E.-H., & Ganesan, V. (2020). Design of a wireless smart insole using stretchable microfluidic sensor for gait monitoring. *Smart Materials and Structures*, *29*(6), 065003.

Luczak, S., Oleksiuk, W., & Bodnicki, M. (2006). Sensing tilt with mems accelerometers. *IEEE Sensors Journal*, *6*(6), 1669–1675.

Majumder, S. & Deen, M. J. (2020). Wearable imu-based system for real-time monitoring of lower-limb joints. *IEEE Sensors Journal*, *21*(6), 8267–8275.

Manupibul, U., Charoensuk, W., & Kaimuk, P. (2014). Design and development of smart insole system for plantar pressure measurement in imbalance human body and heavy activities. *The 7th 2014 Biomedical Engineering International Conference*, 1–5.

Mehr, H. D., Polat, H., & Cetin, A. (2016). Resident activity recognition in smart homes by using artificial neural networks. *2016 4th International Istanbul Smart Grid Congress and Fair (ICSG)*, 1–5.

Mortazavi, B., Nemati, E., VanderWall, K., Flores-Rodriguez, H. G., Cai, J. Y. J., Lucier, J., Naeim, A., & Sarrafzadeh, M. (2015). Can smartwatches replace smartphones for posture tracking? *Sensors*, *15*(10), 26783–26800.

Mukhopadhyay, S. C. (2014). Wearable sensors for human activity monitoring: A review. *IEEE Sensors Journal*, *15*(3), 1321–1330.

Muñoz-Organero, M., Parker, J., Powell, L., Davies, R., & Mawson, S. (2017). Sensor optimization in smart insoles for post-stroke gait asymmetries using total variation and l 1 distances. *IEEE Sensors Journal*, *17*(10), 3142–3151.

Mustufa, Y. A., Barton, J., O'Flynn, B., Davies, R., McCullagh, P., & Zheng, H. (2015). Design of a smart insole for ambulatory assessment of gait. *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 1–5.

Nielsen, M. A. (2015). *Neural networks and deep learning*, *25*. CA: Determination press.

Nitzberg, M., Mumford, D., & Shiota, T. (1993). *Filtering, segmentation and depth*, Berlin: Springer.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Olguın, D. O. & Pentland, A. S. (2006). Human activity recognition: Accuracy across common locations for wearable sensors. *Proceedings of 2006 10th IEEE International Symposium on Wearable Computers, Montreux, Switzerland*, 11–14.

Ordóñez, F. J. & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *International Conference on Machine Learning*, 1310–1318.

Pawara, P., Okafor, E., Schomaker, L., & Wiering, M. (2017). Data augmentation for plant classification. *International Conference on Advanced Concepts for Intelligent Vision Systems*, 615–626.

Pirttikangas, S., Fujinami, K., & Nakajima, T. (2006). Feature selection and activity recognition from wearable sensors. *International Symposium on Ubiquitious Computing Systems*, 516–527.

Preece, S. J., Goulermas, J. Y., Kenney, L. P., & Howard, D. (2008). A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, *56*(3), 871–879.

Promrit, P., Chokchaitam, S., & Ikura, M. (2018). In-vehicle mems imu calibration using accelerometer. *2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, 1–3.

Qi, J., Yang, P., Waraich, A., Deng, Z., Zhao, Y., & Yang, Y. (2018). Examining sensor-based physical activity recognition and monitoring for healthcare using internet of things: A systematic review. *Journal of Biomedical Informatics*, *87*, 138–153.

Renkoski, B. M. (2008). *The effect of carouseling on MEMS IMU performance for gyrocompassing applications*. PhD thesis, Massachusetts Institute of Technology, Cambridge.

Ribeiro, N. F. & Santos, C. P. (2017). Inertial measurement units: A brief state of the art on gait analysis. *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*, 1–4.

Ronao, C. A. & Cho, S.-B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, *59*, 235–244.

Sansrimahachai, W. & Toahchoodee, M. (2016). Mobile-phone based immobility tracking system for elderly care. *2016 IEEE Region 10 Conference (TENCON)*, 3550–3553.

Seel, T., Raisch, J., & Schauer, T. (2014). Imu-based joint angle measurement for gait analysis. *Sensors*, *14*(4), 6891–6909.

Sekiya, H., Kinomoto, T., & Miki, C. (2016). Determination method of bridge rotation angle response using mems imu. *Sensors*, *16*(11), 1882.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., & Havinga, P. J. (2016). Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, *16*(4), 426.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

Sun, L., Zhang, D., Li, B., Guo, B., & Li, S. (2010). Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. *International Conference on Ubiquitous Intelligence and Computing*, 548–562.

Suzuki, K. (2013). *Artificial neural networks: Architectures and applications*. USA: IntechOpen.

Tan, A. M., Fuss, F. K., Weizman, Y., & Troynikov, O. (2015). Development of a smart insole for medical and sports purposes. *Procedia Engineering*, *112*, 152–156.

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

Tee, K. S., Javahar, Y. S. H., Saim, H., Zakaria, W. N. W., Khialdin, S. B. M., Isa, H., Awad, M., & Soon, C. F. (2017). A portable insole pressure mapping system. *Telkomnika*, *15*(4), 1493–1500.

Tsironi, E., Barros, P. V., & Wermter, S. (2016). Gesture recognition with a convolutional long short-term memory recurrent neural network. *European Symposium on Artificial Neural Networks*.

Valade, A., Acco, P., Grabolosa, P., & Fourniols, J.-Y. (2017). A study about kalman filters applied to embedded sensors. *Sensors*, *17*(12), 2810.

Wan, S., Qi, L., Xu, X., Tong, C., & Gu, Z. (2020). Deep learning models for real-time human activity recognition with smartphones. *Mobile Networks and Applications*, *25*(2), 743–755.

Wang, B., Rajput, K. S., Tam, W.-K., Tung, A. K., & Yang, Z. (2015). Freewalker: a smart insole for longitudinal gait analysis. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3723–3726.

Wang, H.-w., Qi, C.-j., Wu, Q.-x., Wang, X.-n., & Xing, Z.-h. (2010). Algorithm of roll, pitch and yaw angular velocity for three-axis gyroscope. *2010 The 2nd Conference on Environmental Science and Information Application Technology*, 2, 395–398.

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, 3–11.

Wang, S. & Zhou, G. (2015). A review on radio based activity recognition. *Digital Communications and Networks*, *1*(1), 20–29.

Weiss, G. M. (2019). Wisdm smartphone and smartwatch activity and biometrics dataset. *UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set*.

Xu, W., Huang, M.-C., Amini, N., Liu, J. J., He, L., & Sarrafzadeh, M. (2012). Smart insole: A wearable system for gait analysis. *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*, 1–4.

Zebin, T., Scully, P. J., & Ozanyan, K. B. (2016). Human activity recognition with inertial sensors using a deep learning approach. *2016 IEEE Sensors*, 1–3.

Zhang, B., Jiang, S., Wei, D., Marschollek, M., & Zhang, W. (2012). State of the art in gait analysis using wearable sensors for healthcare applications. *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, 213–218.

Zhang, Y., Wang, M., Awrejcewicz, J., Fekete, G., Ren, F., & Gu, Y. (2017). Using gold-standard gait analysis methods to assess experience effects on lower-limb mechanics during moderate high-heeled jogging and running. *Journal of Visualized Experiments: JoVE*, (127).