STEERING CONTROL ASSISTANCE FOR A POWERED WHEELCHAIR USING DEEP NEURAL FRAMEWORKS

A PROJECT REPORT

submitted by

BIJEESH B
(Reg. No. TKM20EEII09)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology
in
Electrical and Electronics Engineering

with specialisation in

Industrial Instrumentation and Control

Department of Electrical and Electronics Engineering
TKM College of Engineering
Kollam - 691005
KERALA

JULY 2022

DECLARATION

I undersigned hereby declare that the project report entitled "**Steering Control Assistance for a Powered Wheelchair Using Deep Neural Frameworks**", submitted for partial fulfillment of the requirements for the award of degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of *Prof. Sumayya Jaleel*, Assistant Professor, Department of Electrical and Electronics Engineering. This submission represents my ideas in my own words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam
July 01, 2022

BIJEESH B

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM - 691005

CERTIFICATE

This is to certify that the report entitled " **Steering Control Assistance for a Powered Wheelchair Using Deep Neural Frameworks**" submitted by **BIJEESH B** , **(Reg.No. TKM20EEII09)** of fourth semester to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, is a bonafide record of the project work done by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Sumayya Jaleel
Project Supervisor
Assistant Professor
Department of EEE
TKM College of Engineering

Prof. Sumayya Jaleel
Project Coordinator
Assistant Professor
Department of EEE
TKM College of Engineering

Prof. Shanavas T N
PG Coordinator
Associate Professor
Department of EEE
TKM College of Engineering

Dr. Sabeena Beevi K
Head of the Department
Associate Professor
Department of EEE
TKM College of Engineering

Acknowledgement

A lot of effort and hard work has been put into this project in course of its presentation. However, it would not have been possible without the kind support and help of many individuals and other sources. I would like to extend my sincere thanks to all of them. I take this opportunity to express my deep sense of gratitude and sincere thanks to all who helped me to complete this project report successfully.

I express my sincere thanks to *Dr. T A Shahul Hameed*, Principal, TKM College of Engineering for his encouragement in the completion of my project.

I thank *Dr. Sabeena Beevi K*, Head of the Department, Department of Electrical and Electronics Engineering, *Dr. Imthias Ahamed T P*, Professor, Department of Electrical and Electronics Engineering and *Prof. Shanavas T N*, PG Coordinator, Department of Electrical and Electronics Engineering for their support and cooperation.

I am greatly thankful to my Project Supervisor and Coordinator *Prof. Sumayya Jaleel*, Assistant Professor, Department of Electrical and Electronics Engineering for her supervision, assistance and helpful suggestions.

I am deeply indebted *Prof. Amal A*, Assistant Professor, Department of Electrical and Electronics Engineering, for his excellent guidance, positive criticism and valuable comments.

Finally I thank my parents, friends, near and dear ones who directly and indirectly contributed to the successful completion of my project project.

BIJEESH B

Abstract

People with intellectual, motor, sensory damage always faces difficulty in their safe mobility. Thus, for their safe and secure navigation caregivers are needed always. To help these disabled persons for an independent mobility, a new deep learning technique for steering control assist for a powered wheelchair is proposed. In this work, the system learns to direct a powered wheelchair with user satisfaction in new environments. Long-Short Term Memory (LSTM), Bidirectional LSTM (BLSTM) and Gated Recurrent Unit (GRU) neural networks are used for assisting the guidance of a powered wheelchair. The inputs to these neural networks is dataset of ranges produced by ultrasonic sensors considered to be devoted on the wheelchair and output is the six direction classes. The model predicts the wheelchair direction on the basis of the input data. Later the direction predicted by neural network techniques is been combined with user defining direction which produces the resultant direction so as to avoid objects in the path. The differently-abled person or patients, uses an input method to provide desired speed along with direction and the neural network provides a safe direction for the wheelchair avoiding objects in vicinity. A powered wheelchair model simulation is also done and results are verified with the expected outcomes. The comparison study on numerous parameters concludes that GRU model showed the best performance with overall accuracy of 97.38%. The accuracies obtained from LSTM and BLSTM models are 95.17% and 95.89% respectively.

Contents

Abstract

List of Tables **i**

List of Figures **ii**

Abbreviations **iv**

Notations **v**

1 INTRODUCTION **1**

1.1 General Background 1

1.2 Objectives 3

1.3 Scheme of Project Work 3

2 LITERATURE REVIEW **4**

2.1 Overview 4

2.2 Literature Survey 4

2.3 Concluding Remarks 7

3 NEURAL NETWORKS **8**

3.1 Overview 8

3.2 Artificial Neural Networks (ANN) 8

3.3 Concluding Remarks 13

4 DEEP NEURAL NETWORKS **14**

4.1 Overview 14

4.2 Recurrent Neural Networks (RNN) 14

4.3 Long-Short Term Memory (LSTM) 15
4.4 Bidirectional Long-Short Term Memory (BLSTM) 17
4.5 Gated Recurrent Unit (GRU) 18
4.6 Concluding Remarks 19

5 METHODOLOGY **20**
5.1 Overview 20
5.2 Processes Followed 20
5.3 Network Model Implementation 22
5.3.1 Model Parameter and Hyperparameter 22
5.3.2 Hyperparameter tuning 22
5.4 Design of PW Simulation Model 24
5.4.1 Wheelchair Kinematics 24
5.5 Concluding Remarks 26

6 EXPERIMENTS **27**
6.1 Overview 27
6.2 Powered Wheelchair Simulation 27
6.3 Expected Control Assistance 28
6.4 Concluding Remarks 30

7 RESULTS AND DISCUSSION **31**
7.1 Overview 31
7.2 Results from LSTM Neural Network 31
7.3 Results from BLSTM Neural Network 33
7.4 Results from GRU Neural Network 34
7.5 Confusion Matrix Evaluation 35
7.6 Prediction of Direction with GRU Network 37
7.6.1 Simulated Model 39
7.7 Comparison Study 40
7.8 Concluding Remarks 41

8 CONCLUSION AND FUTURE SCOPE **42**

References **43**

List of Publications **46**

List of Tables

7.1 Comparison of various deep learning techniques 41

List of Figures

1.1 Powered wheelchair and smart wheelchair 1
1.2 Relationship between AI, ML and DL 2

3.1 Single neuron of BNN 9
3.2 Architecture of ANN 9

4.1 Architecture of RNN) 14
4.2 Architecture of LSTM 16
4.3 Architecture of BLSTM 17
4.4 Architecture of GRU 18

5.1 Four element matrix layout 20
5.2 Flowchart of the proposed system 21
5.3 Wheelchair kinematics diagram 24

6.1 Resultant direction for each scenarios 29
6.2 Overall expected wheelchair model navigation 30

7.1 Accuracy graph of LSTM model 32
7.2 Loss graph of LSTM model 32
7.3 Accuracy graph of BLSTM model 33
7.4 Loss graph of BLSTM model 33
7.5 Accuracy graph of GRU model 34
7.6 Loss graph of GRU model 34
7.7 Confusion matrix depicting classes 35
7.8 Confusion matrix of LSTM network 36
7.9 Confusion matrix of BLSTM network 36

7.10 Confusion matrix of GRU network 37
7.11 Model output as forward 38
7.12 Model output as right turn 38
7.13 Model output as left turn 38
7.14 Model output as right spin 38
7.15 Model output as left spin 38
7.16 Model output as stop 39
7.17 Resultant PW direction as forward 39
7.18 Resultant PW direction as left turn 40
7.19 Resultant PW direction as right turn 40

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BCI	Brain Computer Interface
BLSTM	Bidirectional Long-Short Term Memory
CNN	Convolutional Neural Network
DBN	Dynamic Bayesian Network
DL	Deep Learning
GRU	Gated Recurrent Unit
HIS	Hybrid Intelligent System
HRBN	Heuristic-Rule Based Network
LSTM	Long-Short Term Memory
ML	Machine Learning
PNM	Peter Net Model
PW	Powered Wheelchair
ReLU	Rectified Linear Unit
SCAD	Scanning Collison Avoidance Device
SDA	Steering Direction Assist
SW	Smart Wheelchair

Notations

C_t	Cell state
d	Distance between wheels, m
D_c	Distance from center, cm
D_r	Distance from right side, cm
D_l	Distance from left side, cm
f_t	Forget gate
h_t	Hidden state
l_c	Distance center moved, cm
l_l	Distance left wheel moved, cm
l_r	Distance right wheel moved, cm
o_t	Output gate
r_t	Reset gate
θ	Wheel heading
V_l	Velocity of left wheel, m/s
V_r	Velocity of right wheel, m/s
x_t	Input state
z_t	Update gate

Chapter 1

INTRODUCTION

1.1 General Background

In hospitals and clinics, wheelchairs are the most often utilised patient transportation method. Wheelchairs are necessary for those who have motor impairments. Disabilities may result from certain diseases, inherent health conditions, or accidents. A third or less of wheelchair users require assistance with daily duties or mobility. Power wheelchairs (PW) are a choice for these people (Figure 1.1). The majority of PW navigate via joystick. Alternative control methods, such as brain control, chin joysticks, sip-and-puff, and head joysticks, are available for some users who are unable to operate a standard joystick. Some users require assistance with daily activities like eating, drinking and communicating, etc., with the family members in additional to navigation.

Figure 1.1: Powered wheelchair and smart wheelchair

Many advancements on adapting traditional wheelchairs to PW users were successful in advancing technology for PW users to carry out daily duties. The intention was to create a PW by

combining smart and intelligent technologies, but instead a smart wheelchair (SW) was created (Figure 1.1). Researchers combined a variety of technologies used for mobile robot movement in order to produce a smart wheelchair for this user population that finds it challenging to maneuver a PW. An SW is essentially a powered wheelchair base that has a computer and a number of sensors included in, or it might just be a mobile robot with a seat on it. A technology shift in PW could have a major negative impact on users if it doesn't collaborate with the user as a manual control in PW.

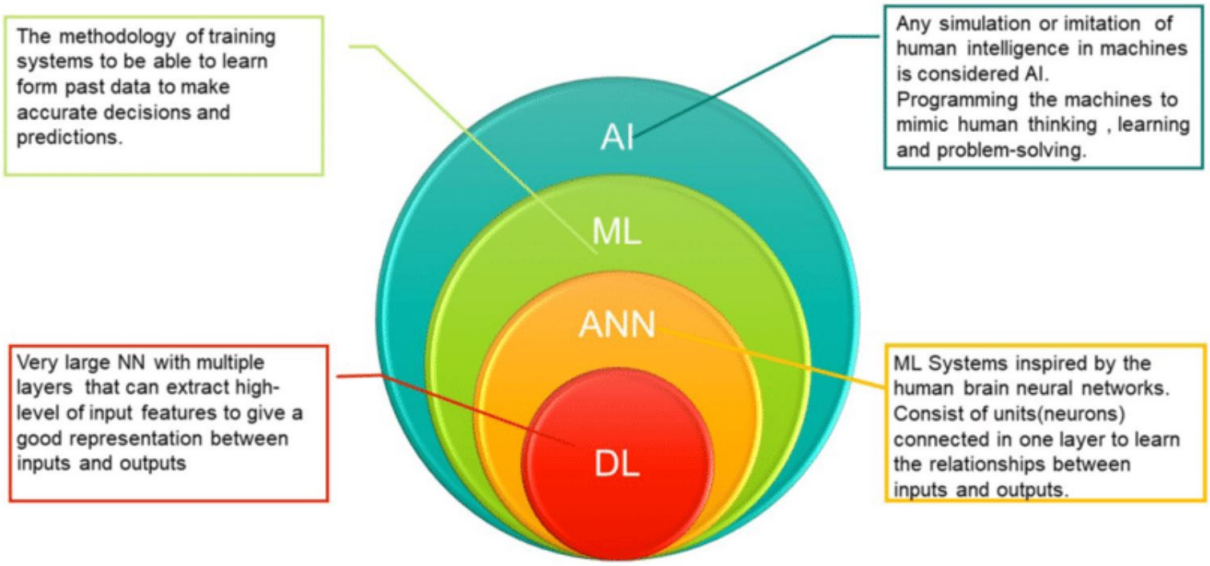

Figure 1.2: Relationship between AI, ML and DL

Artificial Intelligence, known as AI, is the attempt to make computers intelligent enough to mirror human cognitive abilities. It is the emulation of human thinking processes by machines, particularly computer systems. For the creation and training of machine learning algorithms, AI requires a foundation of specialised hardware and software. Expert systems, summary language processing, speech recognition, creativity, and computer vision are just a few of the fields in which AI has broad applications.

The area of artificial intelligence known as machine learning(ML) deals with the statistical component. This is the science of operating a computer without writing any code. The computer is taught to solve issues by examining at hundreds of thousands of examples, learning from them, and then applying what it has learned to similar problems in different contexts.ML includes classification, clustering, etc. Three different categories of machine learning algorithms exist:

- **Supervised Learning**: Patterns can be found and applied to label new data sets, data sets are labelled.
- **Unsupervised Learning**: The order of the data sets is based on similarities or differences rather than on labels.
- **Reinforcement Learning**: There are no labels on the data sets, yet the AI system receives feedback after executing one or more actions.

Deep Learning (DL) is a very specialised area of Machine Learning (ML), where computers can genuinely continue to learn and develop their capacity for reasoned decision-making. Compared to most machine learning algorithms, DL involves a higher level of automation. Various DL methods are explained in the study. The Relationship between Artificial Intelligence(AI), Machine Learning(ML),and Deep Learning(DL) are shown in Figure.1.2.

1.2 Objectives

The objective is to develop a deep neural network that is more accurate and efficient for steering control assistance and powered wheelchair direction prediction. In order to do this, deep neural network modelling and modelling of the powered wheelchair simulation model is to be done followed by powered wheelchair model simulation describing real world situations. Finally comparison of various deep learning approaches and concluding the better framework among them.

1.3 Scheme of Project Work

The project work is organised in 8 chapters. General overview, project objectives, and scheme of project work are included in chapter 1 titled as introduction. The literature review conducted for this work is covered in Chapter 2. The fundamentals of neural networks and ANN are covered in Chapter 3. Deep neural networks, which were taken into consideration for the work, are described in depth in chapter 4. The proposed system's methodology is presented in chapter 5. Chapter 6 of the report describes the experiments in parts and also the parameters taken into account. The model results, simulation results, and comparison-based analysis are all covered up in chapter 7 and finally Chapter 8 deals with the conclusion of the report.

Chapter 2

LITERATURE REVIEW

2.1 Overview

This chapter mainly describes a literature review on the Powered wheelchair and Smart wheelchair technologies. The traditional methods and advanced methods that include probabilistic methods, heuristic methods, rule-based method, Fuzzy based development etc. are discussed.

2.2 Literature Survey

There has generally been a significant investment in research and development (RD) of PWs, SWs, and assistive technology in general. In[1], a system described that makes use of neural networks. By using deep learning technique, the neural network classifies the input data and produce steering instructions. Another study [1], proposed a deep learning approach using LSTM. A self-reliance factors that share control between the sensor and user systems in order to assist users who are mentally or physically differently abled [3]. In [2] decision making using Preference Ranking Organization METHod for Enrichment of Evaluations II (PROMETHEE II) was explained. This method was the first kind of it. PROMETHEE I partial outranking and PROMETHEE II total ranking of alternatives are types of PROMETHEE methods. Preference functions and simplified function equations are utilised, mainly this method conduct comparisons between different possibility regarding each condition and then widely compare with respect to all conditions. The human driver calculates and uses a self-reliance factor, and the controller establishes the control gains automatically to make up for the user's limitations when using the wheelchair. The self-reliance factor, which has three primary components, and

the driver's confidence in operating a powered wheelchair were both estimated to determine the importance of the user's directives. The factor was set to specific values to represent the capabilities of the user in various conditions. The various circumstances that were taken into account to determine the self-reliance factor for prevention, protection, and support, and finally the overall self-reliance factor. Infrared light and regulated veer are explained in [6].

The technology offers increased energy saving with less effort, more precise control, and less swerve on slopes. The system used caster-steering position as well as caster angle measurement, which offers feedback to limit drift away from a set route. A localization strategy utilizing IR-reflecting man-made landmarks was put out in [4]. The ceiling was covered with IR light-reflecting landmarks, which enabled localization. The photographs were taken with a CMOS camera together with USB line. Kinematic analysis was carried out to eliminate issues with application brought on by image distortion. Drive Safe Device (DSS), a motorised wheelchair collision prevention system, was suggested in [10]. DSS uses a distributed method to offer a navigation help solution. A point-to-point navigation method was proposed in [11], the system used a magnetic marker track or barcode and a coloured lane. Track-following technology was truly needed to generate significant changes to the user's environment and when the wheelchair was being used. The wheelchair track system's course direction was extended so that a junction occurred in the study [7], a novel Scanning Collision Avoidance Device (SCAD) system was present at Chailey Heritage. In [23] Scanning Collision Avoidance Device (SCAD) which was attached to power wheelchair prevent driving into things or obstacles. New variable switches were created in order to provide variable speed control than the standard digital switches. The system considered the drawback of mechanical bumper and provided a solution. In [23], it was suggested that mobile robots may be teleoperated by humans to do jobs. By placing an artefact into the floor, such as a vehicle track, a guide wire, or an optically visible line that is painted, formed with tape, or the floor, humans can provide guidance.

In [8] proposed a technique that plan path based on data from sensors for avoiding obstacles and at range goal idea on basis of human perception- heuristic information. Heuristic- Rule Based Network (HBRN) consists of simple algorithm that produce predefined estimation function. HRBN-optimum trajectory is provided. In manifold robot setting a Peter Net Model (PNM). A non-contact multi-layered impedance model for controlling an electric wheelchair with a bio

signal-based obstacle avoidance strategy was proposed in [19]. The suggested technology calculates a virtual repulsive force prior to the wheelchair colliding with the multi-layered impedance fields surrounding it. The wheelchair, which is based, experienced a virtual external force act as a result of this non-contact impedance control technology. The control of wheelchair navigation via a Brain Computer Interface (BCI) based on Steady State Visual Evoked Potential signal (SSVEP) was proposed in [20]. With the reactive navigation paradigm, the system was dependable and safer. Users with spinal cord injuries can get navigational help from the SSVEP evoked BCI system (SCIs). In order to fulfil the needs of the user and the user's capacity to prevent dissatisfaction, [16] suggested a probabilistic framework to recognise local navigation plans in a user specific manner. Dynamic Bayesian Network (DBN) was used to supply the user model with the problem translation of the plan and learn the plan recognition models while driving. Another probabilistic approach for user-centered plan recognition was put forth in [17]. It was a form of mixed control model. This motion planner takes into account the kinetic and dynamic restrictions. A medium fidelity driving simulator was the technology under discussion when [22] proposed a haptic combination control. The longitudinal distance between the host vehicle and the pedestrian, the horizontal position of the pedestrian, and the moving speed are all measured by sensors built into the Steering Direction Assist (SDA) system. The Time to Collision (TTC) longitudinal distance between the host vehicle and pedestrian divided by vehicle speed determines the assistive functionality.

Incorporating more sophisticated AI algorithms or techniques to assist differently abled users drive their wheelchair will improve their quality of life by boosting their self-confidence and independence thus preventing them from the need of caregiver assistance, as opposed to the earlier methods or approaches, which had many limitations. Using a hybrid intelligent system (HIS) based on fuzzy logic (FL) and ANN, [21] offered a system that generated a virtual E-Puck mobile robot with an obstacle avoidance controller. A comparison research was conducted using two controllers, namely the Multiple Layer Perceptron (MLP) and HIS controllers.

In this paper, a system that combines a new rule-based methodology with deep neural networks is described. To provide steering instructions, the Long Short-Term Memory(LSTM), Bidirectional LSTM (BLSTM) and Gated Recurrent Unit (GRU) Neural Networks are used to process input data from ultrasonic sensors using a deep learning algorithms based on a rule based ap-

proach. The network's different layers process the sensor output, classifying the information into various steering directions based on rules. This direction is combined with user input to get the desired direction. Model Simulations and outcomes are compared between LSTM, BLSTM and GRU for identifying the more efficient and accurate deep learning framework among them.

2.3 Concluding Remarks

This chapter examines a literature review on both conventional and some cutting-edge powered wheelchair technologies that have been created to date. It has been noted that the majority of works use a combination of self-reliance, probabilistic, heuristic, and rule-based methodologies as well as fuzzy theories and intelligent system approaches. The following chapter addresses the methods and techniques used in the work.

Chapter 3

NEURAL NETWORKS

3.1 Overview

This chapter deals with the foundations of Artificial Neural Networks (ANN), as well as their theory and structure which is the main part, also discusses a few terms that are related to ANN, advantages and disadvantages of ANNs.

3.2 Artificial Neural Networks (ANN)

In the modern meaning, a neural network is an artificial neural network made up of synthetic neurons or nodes. A neural network is a network of biological neurons, shown in Figure 3.1. As a result, a neural network can be a Biological Neural Network (BNN) consisting of biological neurons or an artificial neural network that is employed to solve artificial intelligence (AI) challenges.

In ANN, the connections between biological neurons are represented as weights between nodes. Inhibitory connections are shown by negative values, and excitatory connections are indicated by positive weights. The weighted sum of all the inputs, which is weighted by the weights of the connections from the inputs to the neuron, is used to determine the neuron's output. A bias term is then added to this sum to determine the neuron's output. Artificial neural networks are renowned for their ability to adapt, which means that they change as they gain knowledge from initial training and additional data from later runs. The most fundamental learning model is based on the concept of input stream weighting, where each node assigns a value to the signifi-

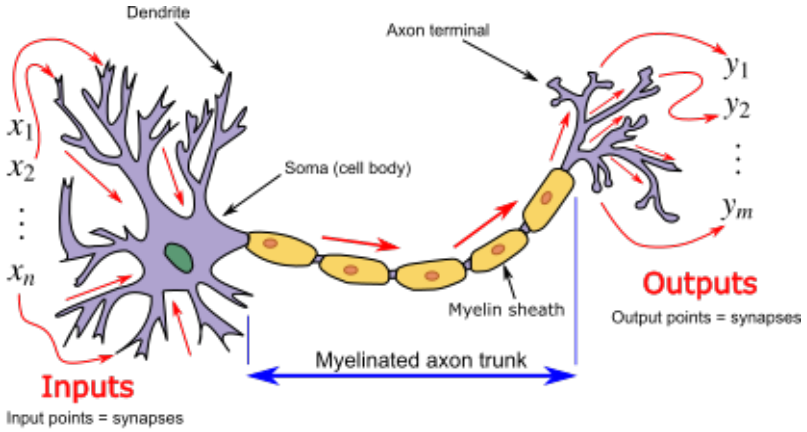

Figure 3.1: Single neuron of BNN

cance of the data input from each of its forebears. The weight of inputs that are higher produce accurate answers.

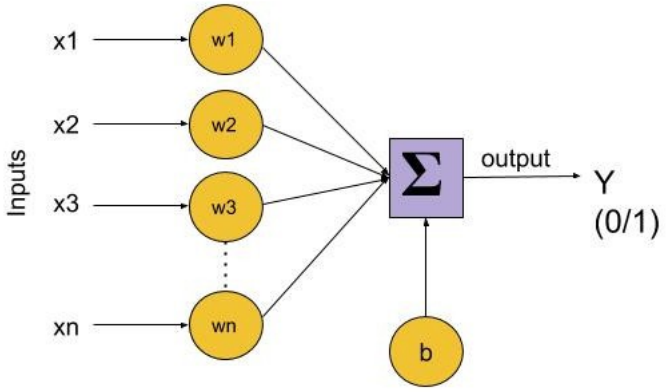

Figure 3.2: Architecture of ANN

The Neural Network which is depicted in Figure 3.2, is described in Equation 3.1. where x_1,x_2,..x_n are the inputs to the input nodes and w_1,w_2..w_n are the weights of input.

$$\sum_{i=1}^{n} w_i x_i + b = w_1 x_1 + w_2 x_2 + w_3 x_3 + ... + w_n x_n + b \tag{3.1}$$

Output is produced after performing the activations to the input.

$$\text{output } = f(x) = \begin{cases} 1 \text{ if } \sum w_i x_i + b \geq 0 \\ 0 \text{ if } \sum w_i x_i + b < 0 \end{cases} \tag{3.2}$$

Diverse types of neural networks with different applications can be categorised. The first neural network was the perceptron, developed in 1958 by Frank Rosenblatt. The simplest type of neural network has a single neuron. An input layer, one or more hidden layers, and an output

layer are the three main components of feedforward neural networks, also known as Multi-Layer Perceptrons (MLPs). Because most actual issues are nonlinear, sigmoid neurons are used in these neural networks instead of perceptrons, which are also frequently referred to as MLPs. These models serve as the basis for neural networks used in computer vision, language processing, as well as other fields. Data is typically input into them to train them. Similar to feedforward networks, Convolutional Neural Networks (CNNs) are being used for image recognition, pattern identification, and/or computer vision. To find patterns in a picture, these networks use ideas from linear algebra, mainly matrix multiplication.
Advantages of ANNs are:

- It is possible to simulate the real-world linkages between input and output by learning and modelling complicated, nonlinear relationships.
- ANNs can forecast the results of unseen data due to their capacity to generalise and infer unknown associations on unknown data.
- The input variables are not constrained in any way, including how they should be distributed.

Disadvantages of ANNs are:

- The right artificial neural network structure can only be established by trial and error plus experience because there are no guidelines for selecting the right network layout.
- Lack of understanding of the why as well as how the answer results in a lack of confidence in the network.
- Unable to handle sequential data and don't share parameter across time.

Common terms in ANN

(i) **Input layer**

Input nodes are another name for input layer. The inputs or information are given to the model at these nodes, where it can learn and draw conclusions. Input nodes transmit data to hidden layers, the following layer.

(ii) **Hidden layer**

The group of neurons known as the "hidden layers" is where all calculations on the input

data are made. A neural network may contain any number of hidden layers. One hidden layer makes up the simplest network.

(iii) **Output layer**

The model's findings or output, as obtained from all the calculations, are found in the output layer. In the output layer, there may be one or more nodes.

(iv) **Weights**

This demonstrates the significance of that specific neuron. A higher value indicates more relationship importance.

(v) **Bias**

Comparable to a constant in a linear equation, bias plays a similar role in neural networks.

(vi) **Activation Function**

The neural networks now have non-linearity thanks to this function. It also determines whether a neuron may add to the layer below. Some of the activation mechanisms that are frequently used are:

- Step function
- Linear function
- Sigmoid function
- Tanh function
- Rectified Linear Unit (ReLU) function
- Leaky ReLU function

(vii) **Loss Function**

The difference between predicted and expected output is quantified by the loss function. Starting with random weights and biases, neural networks learn by making predictions, comparing those predictions to the expected results, and making adjustments to the weights and biases as necessary. Depending on the situation, many loss function types are utilised, including:

- Regression: Squared error loss
- Binary classification: Binary cross-entropy loss

- Multi class classification: Multi-class cross-entropy loss

(viii) **Learning rate**

The model's ability to adapt to errors in each observation depends on how quickly it learns. A low learning rate takes more time but may result in greater accuracy, while a high learning rate shortens training period but reduces overall accuracy.

(ix) **Optimizers**

The optimizer shape and mould model into more accurate model by adjusting the weights and biases, the loss function, which is the mathematical ways of measuring how wrong predictions made by neural network, is its guide. It tells the optimizer whether it's moving in the right or wrong direction. It tie together the loss function and model parameters by updating the network based on the top of the loss function. RMS prop and Adam are the most commonly used Optimizers.

(x) **Forward propagation**

The database's information is sent to several neurons, which are located in the input layer and number from x_1,x_2,...x_n. In this layer, no calculations are made. These are linked to neurons in the following layer through channels. Each neuron in the hidden layer is connected with a numerical value termed the bias, which is subsequently added to the input total, after the input is multiplied by weights. This weighted sum is then run through a nonlinear function, which effectively determines whether or not that specific neuron can contribute to the subsequent layer. It basically functions as a form of probability in the output layer. The ultimate output is determined by the neuron with the highest value.

(xi) **Back propagation**

Back propagation function in reverse direction. Information is sent from the output layer to the hidden layers instead of the input layers. New neural networks are extremely effective because of back propagation. In the final phase before propagation, a new network spits out a prediction, demonstrating how new neural networks may learn on their own. In back propagation, a new system assesses its own performance and determines if it is correct or incorrect. If it is incorrect, the network utilises something called a loss function to quantify the departure from the expected output, and this information is passed back to the hidden layers for the weights and biases to be modified so that network accuracy level rises.

3.3 Concluding Remarks

The fundamentals of Artificial Neural Networks (ANNs), their theory, and structure are covered in this chapter. For ease of understanding, some terms related to are briefly defined. The benefits and drawbacks of ANN that promoted development to the next level are also covered in this chapter.

Chapter 4

DEEP NEURAL NETWORKS

4.1 Overview

The basics of Recurrent Neural Network (RNN) and details of Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (BLSTM), and Gated Recurrent Unit (GRU) are covered in this chapter.

4.2 Recurrent Neural Networks (RNN)

A neural network is a collection of interconnected layers that mimics the organisation and operation of the human brain. It trains a neural network using sophisticated algorithms and learns from enormous amounts of data. A neural network's primary function is to process input and required output data, then change its internal state using several network nodes to generate output that is more closely aligned with the desired output.

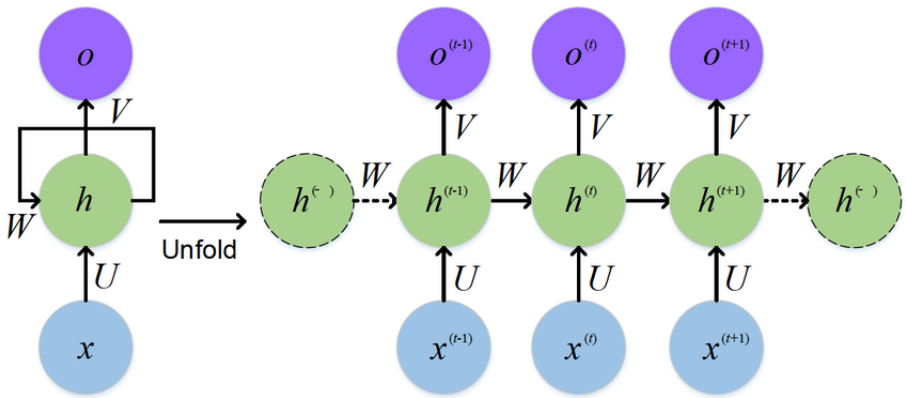

Figure 4.1: Architecture of RNN)

Neural network that forms connections between nodes into a directed or undirected structure along a temporal sequence is called a recurrent neural network (RNN). RNN uses a feedback loop in hidden layer. It can display temporal dynamic behaviour in a way of this. RNNs, which are derived from feedforward neural networks, may process input sequences of different lengths by using their internal state (memory). They can therefore be used for tasks like linked, unsegmented handwriting recognition or speech recognition. RNN are theoretically capable of running any program to handle any input sequence because they are turning complete.

At $t-1$ time step the hidden node h_{t-1} uses the input x_{t-1} to produce output o_{t-1} as seen in feed forward neural networks. In the next time step, hidden node at h_t uses both new input x_t as well as the state from previous time step h_{t-1} as input to make new prediction. This means that current neural network usse knowledge of its previous states as input for its current prediction and we can repeat this process for an arbitrary number of steps allowing for the network to propagate information by its hidden states throughout time. The following equations descibes RNN shown in Figure 4.1

$$\begin{aligned} h_t &= \sigma_h \left(x_t W_x + h_{t-1} W_h + b_i \right) \\ y_t &= \sigma_y \left(h_t W_h + b_y \right) \end{aligned} \tag{4.1}$$

h_t is the hidden state
W is the weight matrix
b is the bias

RNNs operate on remembering their past and their decisions are influenced by what it learned from past basic feed forward, output of one layer is saved and fed back into the input in order to anticipate the output of that layer. A Feed-Forward Neural Network can be changed into a Recurrent Neural Network as shown fig.1. With backpropagation there are certain issues: vanishing and exploding gradients.

4.3 Long-Short Term Memory (LSTM)

A deep learning model called the LSTM Neural Network gathers input features from many layers of data and uses these characteristics as input for classification. The LSTM model was created to process the network utilising the past and future. It is a sort of multilayer neural

network and is a subset of recurrent neural networks (RNN). Exploding and vanishing gradient issues are displayed by RNN. The long-term dependency issue is solved by the LSTM model by using a memory cell to save crucial information about earlier states. Hochreiter and Schmidhuber presented this method in 1997. LSTMs are structured like a chain as shown in Figure4.2.

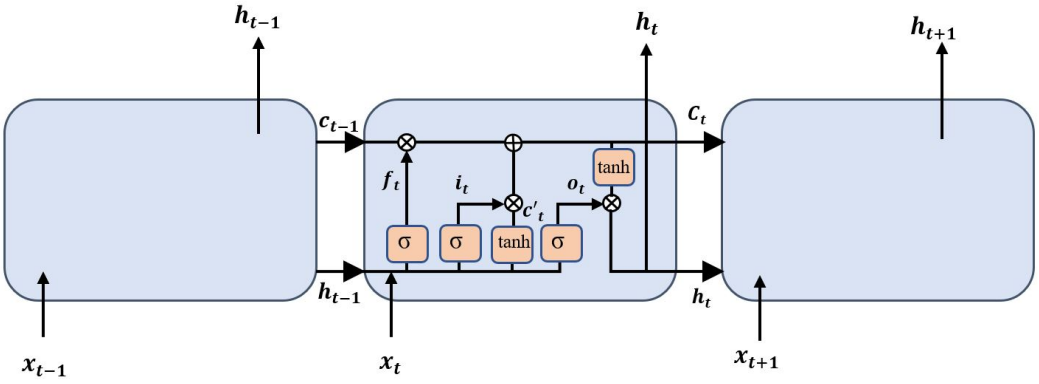

Figure 4.2: Architecture of LSTM

There are four interacting networks engaging in a unique way rather than just one neural network. The cell state, which may be compared to a conveyor belt running directly down the chain with only a few modest linear interactions, is a key component of LSTM.

The first stage in LSTM is to determine which information is unnecessary and will be removed from the cell state. A sigmoid layer termed the "forget gate layer" makes this choice.

$$f_t = \sigma_g \left(x_t W_f + h_{t-1} U_f + b_f \right) \tag{4.2}$$

Here, f_t is the forget gate, σ is sigmoid function, x_t is the new input, h_{t-1} is the output from previous time stamp W, U are the weight matrix and b_f is the bias.

The next step is to choose which new data will be stored in the cell state. The steps that make up this entire process are as follows: The choice of which values will be updated is made by an input gate layer, a sigmoid layer. A tanh layer provides a vector of new potential candidate values to be added to the state next.

$$\begin{aligned} i_t &= \sigma_g \left(x_t W_i + h_{t-1} U_i + b_i \right) \\ \tilde{C}_t &= \tanh \left(x_t W_c + h_{t-1} U_c + b_c \right) \end{aligned} \tag{4.3}$$

Here, i_t is the input gate, σ is sigmoid function, x_t is the new input, $\tanh$ is squashing function.

Then, in order to update the old cell state, C_{t-1} into the new cell state C_t, first multiply the old state C_{t-1}, by the earlier decided-to-forget values, add $i_t * \tilde{C}_t$ after that. This is a scaled-down version of the new candidate values based on how much each state value was updated. Put the new subject information here and remove the old subject.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4.4}$$

Run a sigmoid layer later to determine what portions of the cell state will be output. Then, to ensure that only the predetermined portions are output, add the cell state using tanh and multiply it by the sigmoid gate's output. Although the output will be filtered, it will depend on the cell status.

$$\begin{aligned} o_t &= \sigma_g \left(x_t W_o + h_{t-1} U_o + b_o \right) \\ h_t &= \tanh \left(C_t \right) * o_t \end{aligned} \tag{4.5}$$

Here, o_t is the output gate, C_t is the new cell state, and h_t is the new output gate.

4.4 Bidirectional Long-Short Term Memory (BLSTM)

Another variant of the RNN is the BLSTM Neural Network, shown in Figure 4.3.

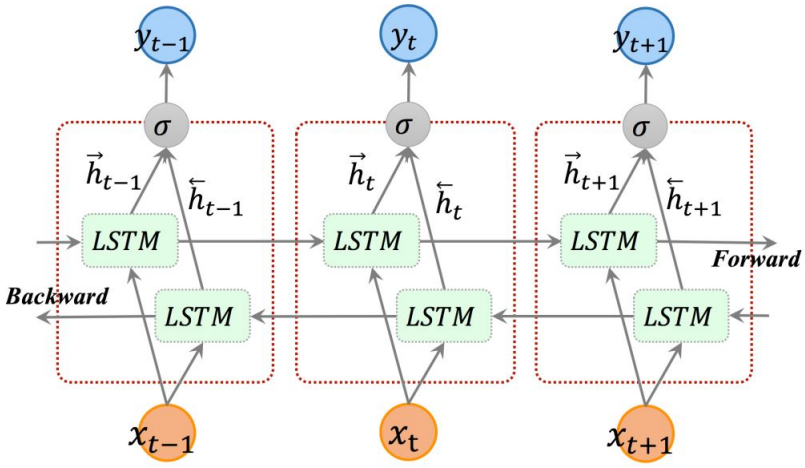

Figure 4.3: Architecture of BLSTM

Traditional unidirectional LSTM has been improved by BLSTM. Only the past and present are used in the unidirectional LSTM model. However, BLSTM processes data in both directions, from future to past and from past to present. Both forward and backward data are used to train the BLSTM model, which is then used for prediction and classification.

4.5 Gated Recurrent Unit (GRU)

GRU Neural Network is as a gating technique which is a variation of RNN, shown in Figure 4.4. Compared to LSTM and BLSTM, this version is newer. GRU was released in 2014 and is becoming more well-known. RNN's short-term memory issue is resolved by GRU. GRU only has two gates, which makes it different from LSTM or BLSTM. Because it is simple to change and does not require memory units, GRU is faster and more accurate than LSTM. It also responds quickly. It was discovered that the performance of GRU and LSTM were comparable for a few tasks involving polyphonic music modelling, speech signal modelling, and natural language processing. It has been demonstrated that GRUs perform better on some smaller and less frequent datasets. The term "minimal gated unit" refers to a brief overview of the full gated unit that uses gating performed using the prior hidden state and the bias in various combinations.The Hadamard product is indicated in the following equation by the dot operator by GRU.

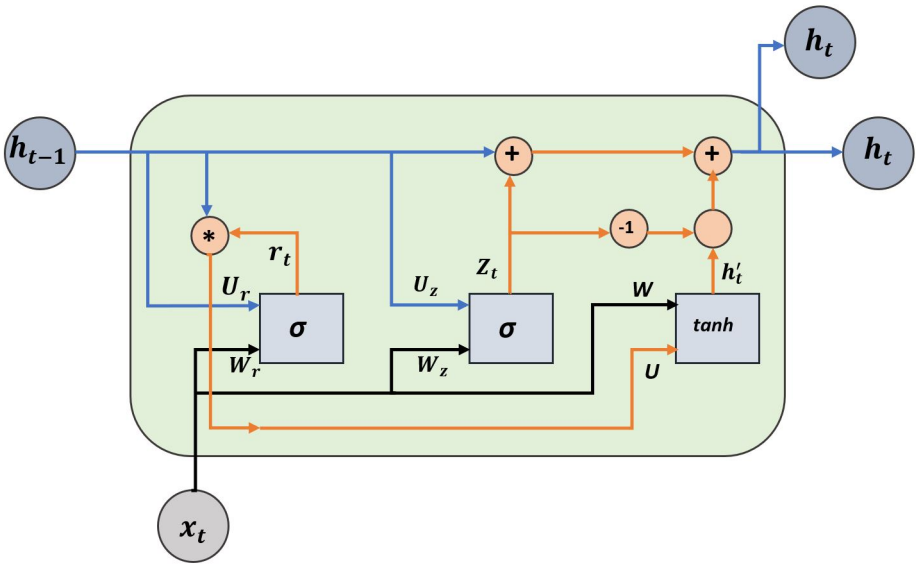

Figure 4.4: Architecture of GRU

A reset gate will perform a mathematical operation like weighted sum applied sigmoid activation on top of hidden state h_{t-1} and current input x_t. The reset gate value is this r_t value that is received described in equations below. Update gates are used to store memory and apply it. Accordingly, the update gate will function identically like the reset gate while using various weights. For example, in the equation, W_z are different but the same weighted sum of $W_z x_t$ and $U_z h_{t-1}$ is applied, and the result is z_t get here, which is the value of the update gate. z_t, r_t,

h_{t-1} and x_t are used to produce new hidden state and output. Here the multiplication operation is hadamard product.

$$\begin{aligned} r_t &= \sigma_g \left(W_r x_t + U_r h_{t-1} + b_r \right) \\ z_t &= \sigma_g \left(W_z x_t + U_z h_{t-1} + b_z \right) \\ \hat{h}_t &= \phi_h \left(W_h x_t + U_h \left(r_t \odot h_{t-1} \right) + b_h \right) \\ h_t &= z_t \odot \hat{h}_t + (1 - z_t) \odot h_{t-1} \end{aligned} \tag{4.6}$$

Here, variables x_t, h_t and $\hat{h}_t$t are used to represent input, output vectors, and candidate activation vectors. z_t and r_t are update gate vectors and reset gate vectors, parameter matrices, and vector activation functions, respectively. W, U, b are used to represent parameter matrices and vector. σ_g is a sigmoid function and hyperbolic tangent ϕ_h. The update and reset gate vectors are combined into a forget gate in the minimal gated unit, which is similar to the fully gated unit. This indicates that the output vector's equation must be modified as well. These Deep learning models are considered and comparison and evaluation are done.

4.6 Concluding Remarks

This chapter provided explanations for Recurrent Neural Networks (RNN), Long short Term Memory (LSTM), Bidirectional Long Short Term Memory (BLSTM), and Gated Recurrent Rnits (GRU) neural networks. It is also discussed how each method's cell structure and associated equations connect to its operation.

Chapter 5

METHODOLOGY

5.1 Overview

In-depth discussion of the steps in the system's data collection, processing, parameters, deep learning model building, and integrated control is covered in this chapter.

5.2 Processes Followed

Sensors are essential for helping wheelchairs navigate safely. Various studies have looked into a variety of sensor types, including infrared, ultrasonic, structured light, and various local architectures distinct from global systems, such as tilt sensors, gyroscopes, etc [16]-[22]. Due to the physical characteristics of ultrasonic sensors, which were tested and shown to be free of cross talk and side lobe interference, these sensors were taken into consideration for this investigation.

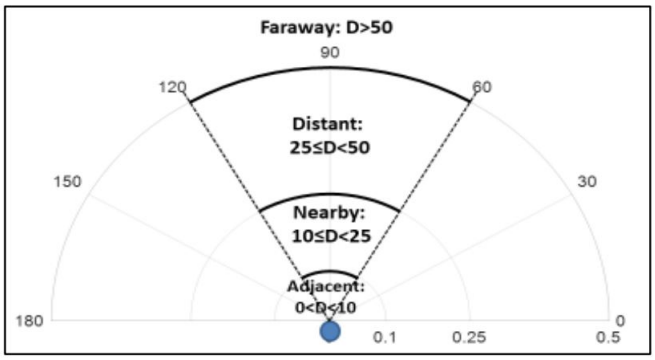

Figure 5.1: Four element matrix layout

A powered wheelchair's general direction is determined using a rule-based approach based on configuration and sensor readings [1]. On the front of the wheelchair, three ultrasonic sensors are thought to be placed. The first sensor determines the distance to the wheelchair's right-side nearest obstacles (D_r). The second sensor calculates the distance to nearest obstacle to the wheelchair's left (D_l), while the third sensor measures the distance to nearest obstacle to the wheelchair's front (D_c). When an obstacle or object is present, an ultrasonic sensor will measure the time that takes for a pulse to travel there and return. For the creation of the study matrix, that has four components—adjacent, nearby, distant, and far away, same as in[1], shown in Figure 5.1.

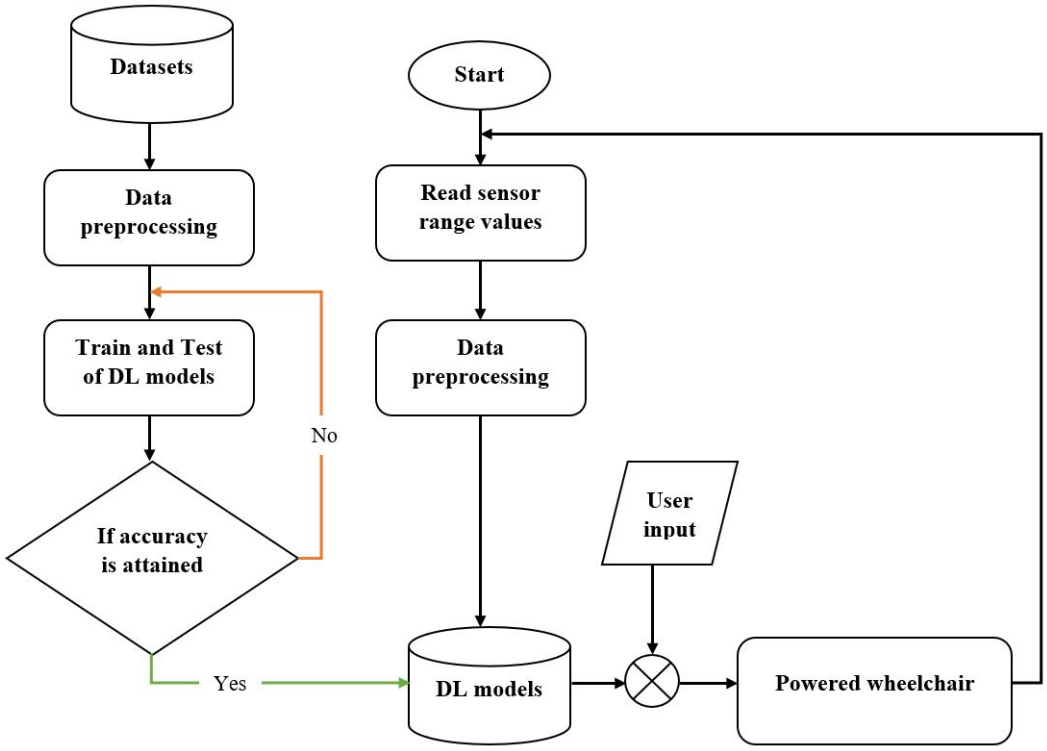

Figure 5.2: Flowchart of the proposed system

Then an dataset containing these sensor readings based on a set of rules[1], as D_r, D_c, and D_l values, is processed. This process makes use of deep learning neural networks. The sensor readings are transmitted through various layers of the deep learning neural networks to identify their features, processing the data as a result. The missing values in the dataset are either ignored after importing it or filled in using an iterative imputer function, followed by binary encoding and reshaping. The data set is divided into training and test sets. The system then

determines the relevant identifiers for classifying the processed sensor readings into six directions. Thus deep learning model is utilised for prediction after these steps. Different powered wheelchair steering directions are represented by the six classes: forward, right spin, left spin, right turn, left turn, and stop.The network and user input combination approach is used to employ the deep learning model's processed direction output for combined or shared control of a powered wheelchair, and the resulting direction is provided for the powered wheelchair, providing a closed loop control. Figure 5.1 explains the proposed methodology of the study in brief.

5.3 Network Model Implementation

Some concepts are required prior to the model's execution because they serve as the process's crucial foundation which are discussed below. These are called the model parameters and hyperparameters, and they vary depending on the goal.

5.3.1 Model Parameter and Hyperparameter

A model parameter is an internal neural network variable whose value can be inferred directly from the data. The values of model parameters, which the model needs in order to generate predictions, determine the model's skill. Instead than being manually set, these parameters are saved as part of the learnt model. Weights and prejudices are two examples.

Since the neural network's model hyperparameter setup is external, values cannot be directly inferred from the data. Finding the best bargain is not an easy task. Model hyperparameters are frequently referred to as parameters since they are the machine learning components that require manual setup and tuning. When the Deep Learning algorithm tuned, it actually means the hyperparameters are tuned. As a result, a parameter needs to be manually specified for instance, learning rate, etc are hyperparameters.

5.3.2 Hyperparameter tuning

Trial and error approach is used to tune the hyperparameters.

1. **Epochs:** One epoch occurs when the complete dataset is processed by the neural network both forward and backward just once. The use of many epochs improves the generalize ability of our model. As the number of epochs rises, more adjustments are made to the parameters, improving performance. Varying models use different numbers of epochs.

2. **Batch and Batch sizes:** In order to update the internal model parameters, big datasets are broken up into smaller batches, and the size of these samples is determined as the batch size. Batch size is the total number of trainees in a training session.

3. **Iterations:** One epoch must be completed in the specified number of batches, or the number of batches must match the specified number of epoch iterations.

4. **Train-Test ratio:** Ratio used to divide the total dataset into training and testing datasets.

The dataset as explained earlier is imported as a csv file and utilized as training and testing datasets, with a split of the training and testing data of 8:2 respectively. The network is trained over 100 epochs with 64 batches. When the final deep neural network model is completed, overall model accuracy is examined. Accuracy and loss graphs, and confusion matrices are obtained as outputs. Based on sensor readings, the trained and tested neural network model produces an overall result. There are six possibilities for the general course of a powered wheelchair: forward, right turn, left turn, right spin, left spin, and stop, for easiness represented as f, rt, ls, rs, ls, s respectively. The powered wheelchair's direction is predicted using this finished efficient neural network model.

User input is combined with the neural network model's output to achieve the desired direction. Because the user must be able to override the system, this is done to provide a shared control. This also allows for the prediction of navigation using human skill and sensor information. For the purpose of controlling speed and direction, human input serves as an interface between the powered wheelchair user and the wheelchairs. As a result, people with disabilities can use driving skills and sensor control when they lack awareness or competence when moving in a wheelchair from one location to another. Additionally, offer guidance in challenging and new circumstances.

5.4 Design of PW Simulation Model

The output of the deep neural network model is coupled with user input to provide a result that satisfies the user. In order to provide shared control as opposed to autonomous control, when the user has no part in direction instruction, the user will be able to override the system. Additionally, it enables the prediction of navigation for users with various abilities or slow reaction times utilising human skill and sensor data. So, utilising a visual simulation platform, the study's shared control of the user and deep learning model is carried out. For the development of the powered wheelchair simulation model, the following kinematics are considered:

5.4.1 Wheelchair Kinematics

Modeling is required for powered wheelchair control. Therefore, a powered wheelchair is compared to a differential drive wheeled mobile robot, which is a typical type. Wheel speeds can be adjusted so that the wheelchair moves straight ahead when the right and left wheels are rotating at the same speed. Turning in the direction of the slower-turning wheel occurs if one wheel is rotating more slowly than the other. This is how user can actually move the powered wheelchair around.

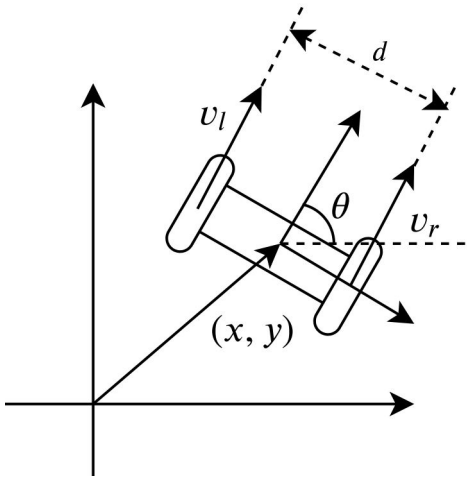

Figure 5.3: Wheelchair kinematics diagram

Two parameters must be known when thinking about wheelchair model dimensions: the first is wheel base, which refers to the distance between the wheels; in this case, it is defined as d. The second parameter is the radius of the wheel, which indicates how large the wheel is; it is

considered here as r. Although simple to calculate, these variables don't signify much when designing controllers. The available control signals for controlling motion are V_r and V_l. Where V_r is the rate at which the right wheel is turning and V_l is the rate at which left wheel is turning. So these are the two inputs to the system.

The x, y positions of the system as well as the system's orientation or heading θ are all factors in its state. So powered wheelchair model needs to connect the inputs which is V_r and V_l to the states which is x, y and θ. Considering some kind of derivations the relationship is described as follows [13].

$$\begin{aligned} \dot{x} &= \frac{V_r + V_l}{2} \cos\theta \\ \dot{y} &= \frac{V_r + V_l}{2} \sin\theta \\ \dot{\theta} &= \frac{V_r - V_l}{d} \end{aligned} \tag{5.1}$$

Here, $\dot{x}$ is how the x position of mobile robots changes, $\dot{y}$ is how the y position of mobile robots changes and $\dot{\theta}$ measures how the robot turns.

The powered wheelchair's state is x, y and θ, which means that its location is x, y, and its heading or orientation is θ. Thus a mechanism to know where it is needed, so that the state information can be collected. It is easy to determine the distance each wheel has travelled. Thus, it is possible to determine how far the centre has moved. Let l_r be the distance right wheel has moved and l_l be the distance left wheel has moved and l_c which is distance the center of the system moved can be calculated as,

$$l_c = \frac{l_r + l_l}{2} \tag{5.2}$$

Start at x and after a time interval, a calculated l_c can compute the new x' and y' positions of the wheelchair as well as new orientation or heading θ' if measuring how far the wheels have travelled during a time interval and can be defined as,

$$\begin{aligned} x' &= x + l_c \cos\theta \\ y' &= y + l_c \sin\theta \\ \theta' &= \theta + \frac{l_r - l_l}{d} \end{aligned} \tag{5.3}$$

5.5 Concluding Remarks

This chapter focused on familiarising with the proposed methodology, which covered the steps taken, the network model's implementation and key parameters, the tuned parameters, and the mechanics applied to simulate a powered wheelchair model. The experimental results and their analysis are covered in the following chapter.

Chapter 6

EXPERIMENTS

6.1 Overview

The work is separated into two experimental sections, the first of which involves the development of a deep neural network models using LSTM, BLSTM, and GRU to which the dataset has been loaded and processed, each model's direction prediction is then determined. The combined or shared control approach created with deep learning and user-specific instructions is included in the second section. This chapter includes an insight of how the visual environment and powered wheelchair simulation model were designed to produce the desired resultant control output.

6.2 Powered Wheelchair Simulation

In the virtual studio platform, the powered wheelchair was simulated using python programming, python modules and other software tools. This chapter's prior sections outline the kinematics under consideration. Using MS Paint, the simulation environment was built with the appropriate size and attributes. Obstacles are made both indoors and outside. Another design is a motorised wheelchair with a body and four wheels, two of which are caster wheels and the other two are control wheels. For replicating the powered wheelchair, three classes were developed, including:

First-class which characterises the simulation's visuals, which include the environment's dimensions, a powered wheelchair model, and map images. The group of colours is declared and

the environment's requirements are loaded using python modules. The loaded environment is named and created using python tools as part of the visual window configuration.

The powered wheelchair employs the python module rotozoom function, which rotates the wheelchair model proportionally to the heading, in the second class, which is the ultrasonic sensor class. Here, the class is initialised using the map and sensor range to detect nearby objects. For scanning the objects, there are start and finish angles that are just the heading plus or minus the sensor range, and the sensor provides a point cloud that contains the measured objects.

The wheelchair is initialised using a start location and wheel velocities with maximum and minimum values in the first, which is the model third class. The previously described kinematic model is used to determine the kinematics part, and the lowest and maximum V_r and V_l velocities are declared. The closest object is initialised to zero for a forward motion using a technique that defines the point cloud supplied by the sensor. For situations where the point cloud is not a none factor, conditions are developed.

The three main classes for the simulation are declared, and then the main functional part is built by loading the necessary python modules, the classes described above, as well as the deep learning model for combining user input and deep learning neural network model output and building a shared control approach. The virtual environment must then be initialised using the graphics class before the start position is declared. Loops are used to provide pause or direction-changing alternatives for the user's delight. Kinematics class is called for motorised wheelchair navigation. The point cloud created by the sensing obstacle approach is then employed by the deep learning method to predict the direction of travel. This point cloud is provided by the ultrasonic class. lastly, screen update commands are provided.

6.3 Expected Control Assistance

A graphical representation of required and expected powered wheelchair navigation is represented in Figure 6.2. The black arrow represents the resultant direction, the red arrow represents user defined direction and green is the GRU predicted direction.

For combining user input with a deep neural model to simulate a powered wheelchair, an environment containing objects is taken into account. So In the study, three scenarios were consideration, and in each of them, user input points in the direction of the destination. When the powered wheelchair starts moving in the first scenario with nothing in its path. In the second scenario, the powered wheelchair detects an object to its right as it moves to the front. Finally, in the third situation, items are detected in front and to the left of the powered wheelchair.

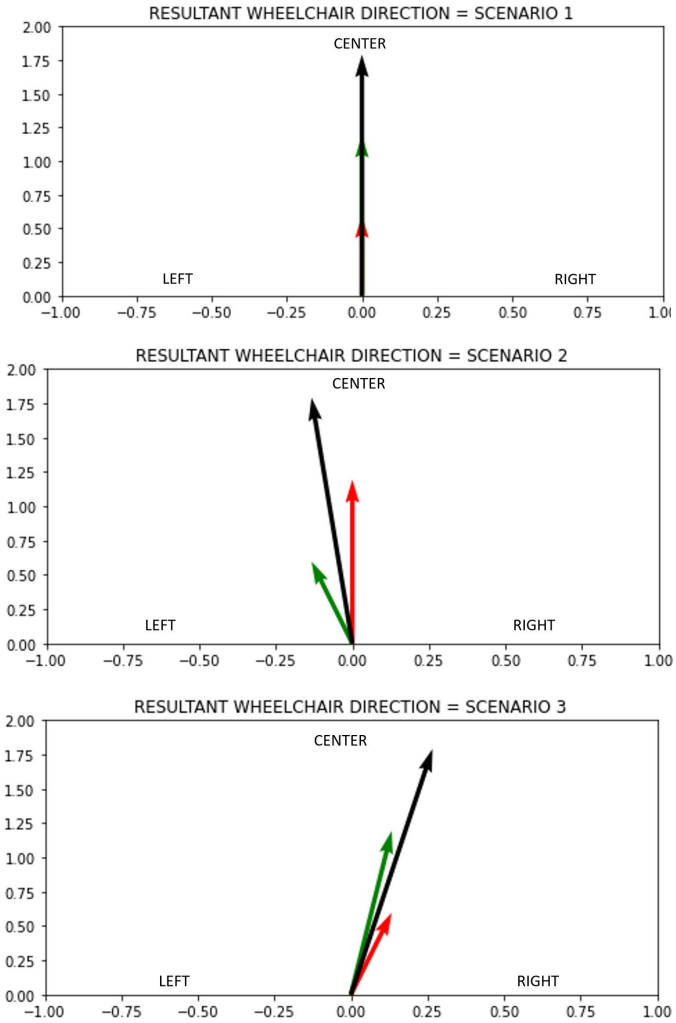

Figure 6.1: Resultant direction for each scenarios

The above mentioned scenarios is shown in Fig.6.1. as vector diagrams. The green arrow indicates the anticipated output of the deep learning model, the red arrow indicates the user input, and the black arrow indicates the projected resultant direction. Figure 6.2 illustrates the overall direction guidance as anticipated by using the shortest route,taking objects in surrounding into consideration.

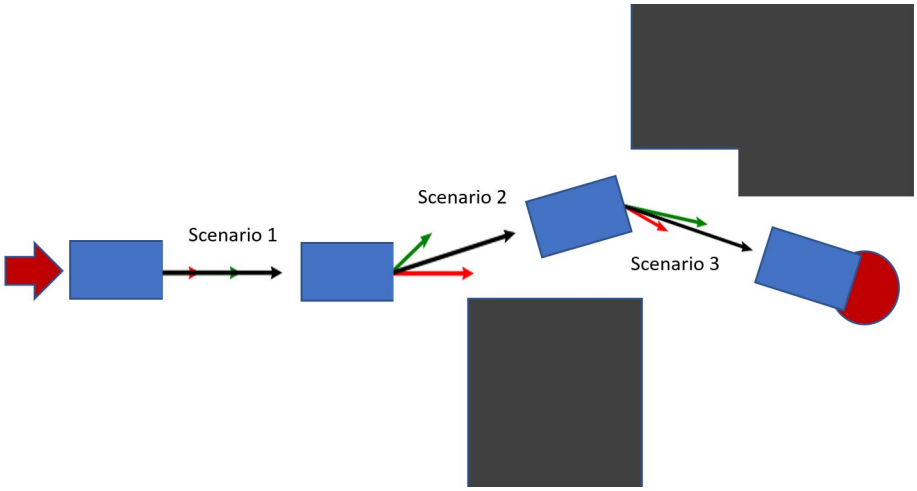

Figure 6.2: Overall expected wheelchair model navigation

6.4 Concluding Remarks

The experimental parts of the study and the powered wheelchair simulation are studied also expected control assistance is also mentioned in this chapter. The next chapter depicts results and discussion of this study.

Chapter 7

RESULTS AND DISCUSSION

7.1 Overview

The performance of each strategy and the accuracy of several deep learning models are examined in this chapter. The output was obtained by performing the steps discussed in earlier chapters. As previously stated, the experiment is divided into two parts: the output of deep learning neural networks and the simulation of a powered wheelchair model to provide combined control.

7.2 Results from LSTM Neural Network

Models are built using python programming, and then they are assessed. The inputs to the neural networks are the sensor readings that indicate how far the wheelchair is from the nearest object to its right, infront, and left.

It is evident from the accuracy and loss graphs that the LSTM network is not a good method. The graphs are distorted right from the beginning of the network training progress. LSTM method produces only accuracy of 95.17%. When using prediction trials, the approach does not produce an accurate direction. The output graphs obtained is shown in Figure 7.1 and Figure7.2.

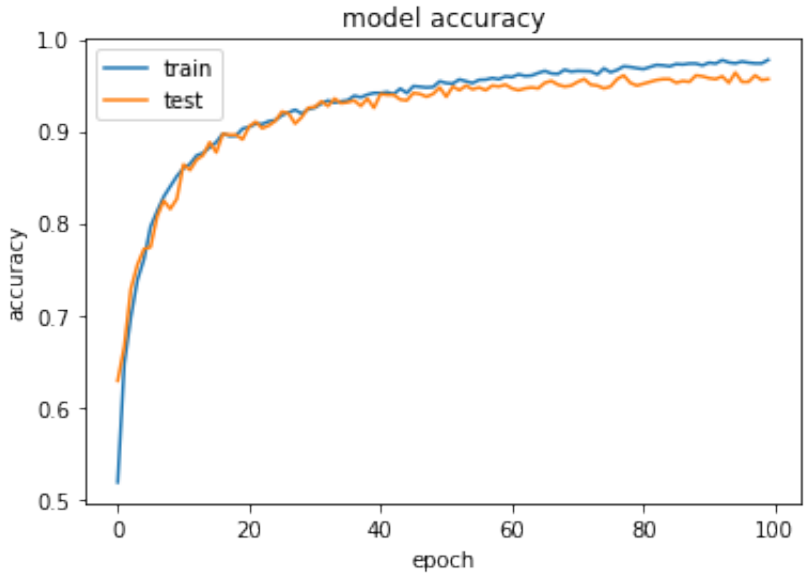

Figure 7.1: Accuracy graph of LSTM model

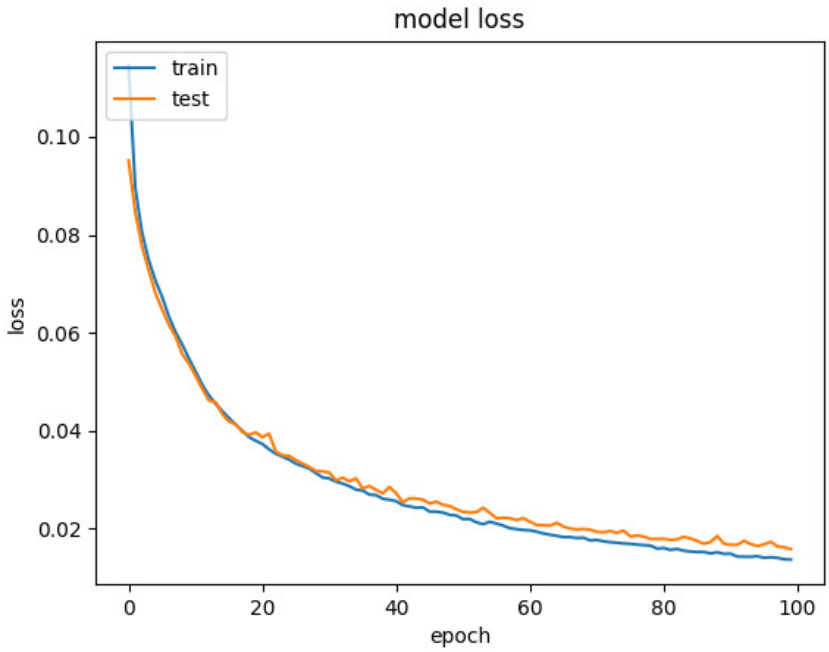

Figure 7.2: Loss graph of LSTM model

7.3 Results from BLSTM Neural Network

In the beginning of training, the BLSTM network almost always responds favourably, but after a certain epoch, the network responds unfavourably, decreasing accuracy, shown in Figure 7.3. This distortion is also present in the mean squared error response, as in Figure 7.4. The accuracy of the BLSTM approach is 95.89%.

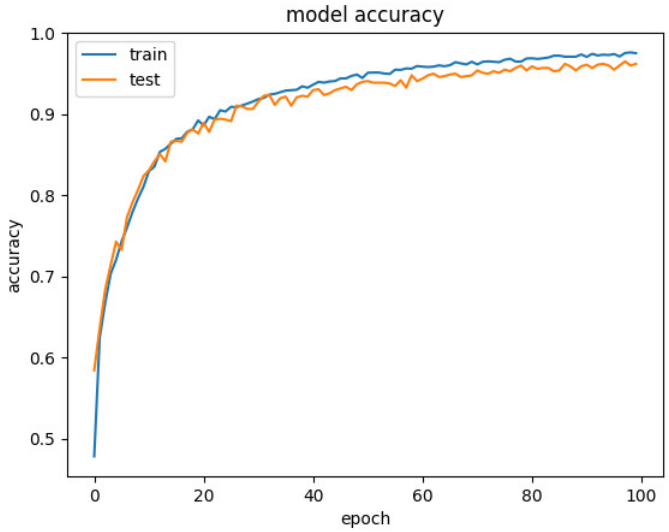

Figure 7.3: Accuracy graph of BLSTM model

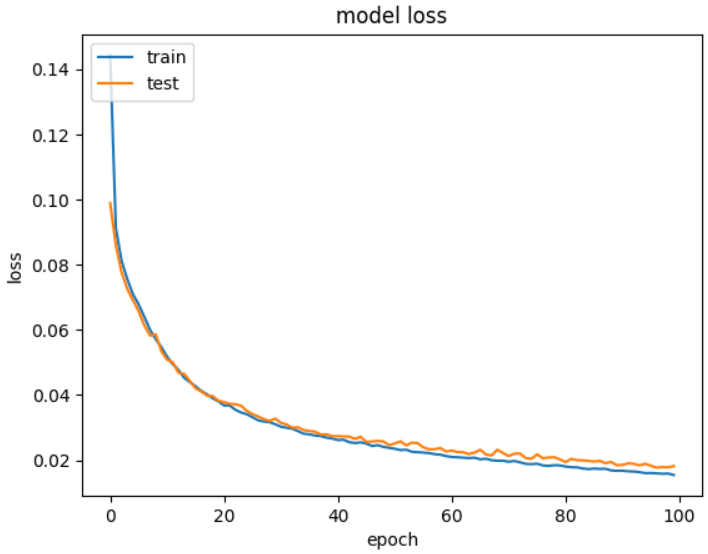

Figure 7.4: Loss graph of BLSTM model

7.4 Results from GRU Neural Network

GRU neural network was the third deep learning technique taken into account. GRU responded favourably in terms of accuracy and loss. The resulting graphs are overlapping one another, as shown in Figure 7.5 and Figure 7.6. This approach produces accuracy of 97.38 percent.

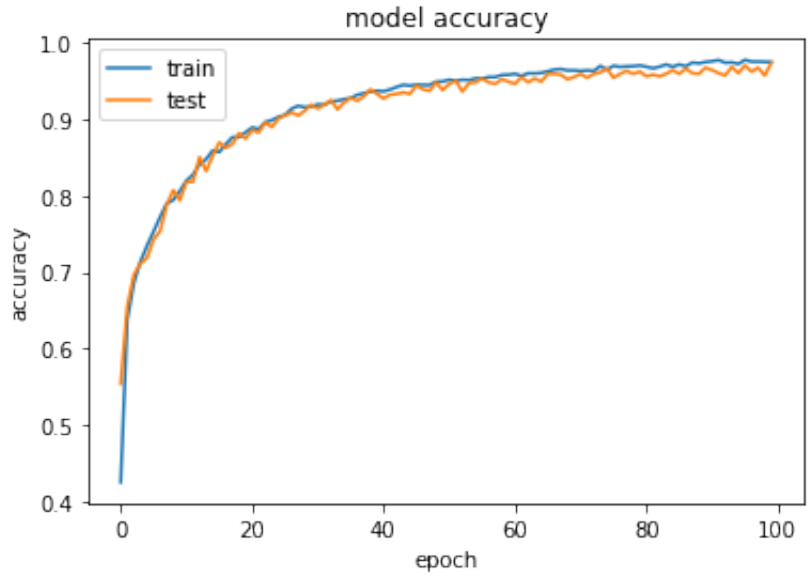

Figure 7.5: Accuracy graph of GRU model

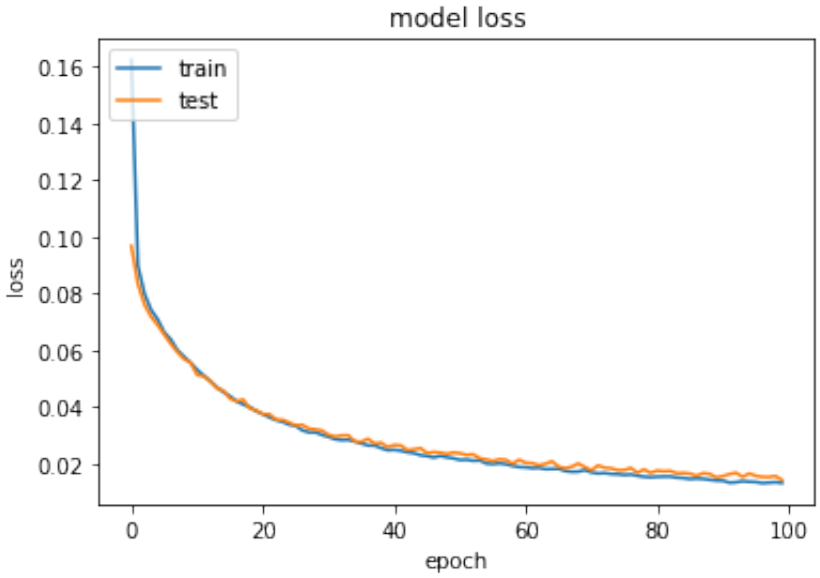

Figure 7.6: Loss graph of GRU model

7.5 Confusion Matrix Evaluation

The efficiency of a prediction model is evaluated using a n x n matrix termed a confusion matrix, where n is the total number of classes. The deep learning model's predicted goal values are compared to the actual goal values in the matrix. This gives us a thorough insight of the errors that our categorization model is committing as well as how effectively it is working.

Precision and recall are both crucial because information retrieval depends on both. When compared to the negative class, it also prioritises the positive classes. In terms of precision and sensitivity, only the true positive (TP), false positive (FP), and false negative (FN) are employed; true negative is not taken into account (TN).

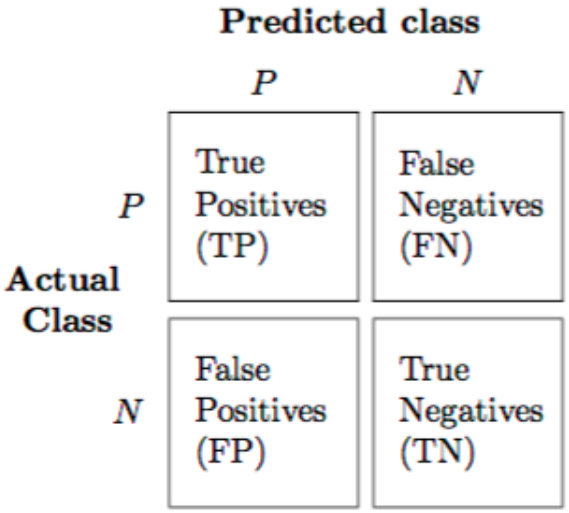

Figure 7.7: Confusion matrix depicting classes

True positive (TP) means that predicted class and actual class both are positive, True negative (TN) define predicted class and actual class both are negative, when predicted class is positive but actual class is negative then it is False positive (FP) and finally False negative (FN) means that predicted class is negative but actual class is positive.The performance of the proposed direction prediction model has been validated using criteria including accuracy, precision, recall, and f1-score. The mathematical representation of these parameters are as,

$$Precision = \frac{TP}{TP + FP} \tag{7.1}$$

Out of all the positive predictions, precision calculates the proportion of actual positive results.

$$Recall = \frac{TP}{TP + FN} \tag{7.2}$$

In contrast, the recall provides the expected positive proportion of all positive results.

$$F1score = 2 * \frac{P * R}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{7.3}$$

The arithmetic mean of sensitivity and precision is known as the F1-score. The figures below displays the confusion matrixes that was obtained for the various deep learning models.

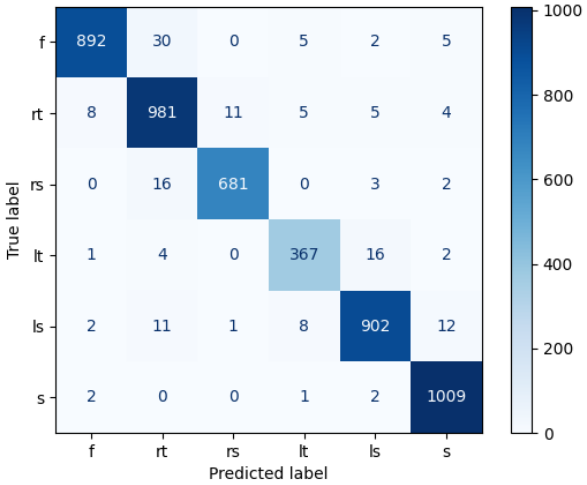

Figure 7.8: Confusion matrix of LSTM network

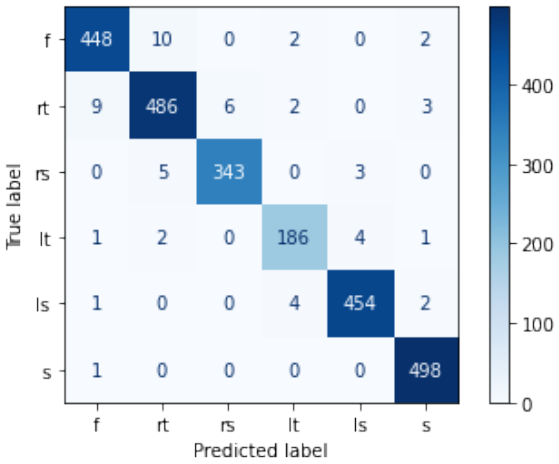

Figure 7.9: Confusion matrix of BLSTM network

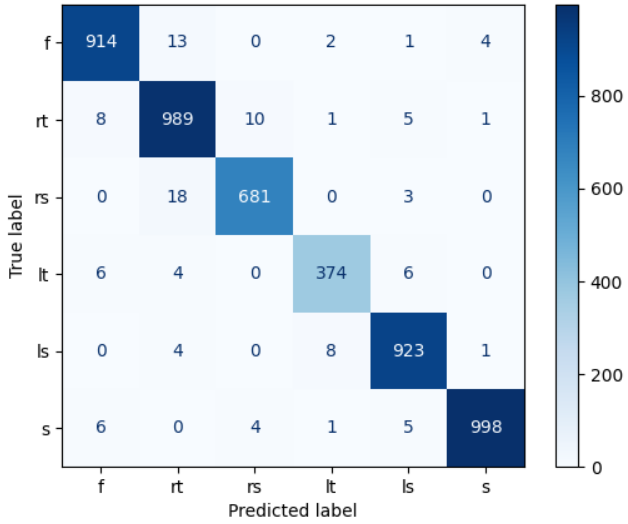

Figure 7.10: Confusion matrix of GRU network

The performance of the models is assessed by determining precision, recall, f1-score, and accuracy of the classes from the derived confusion matrixes illustrated in Figure 7.8–7.10. While recall provides the expected positive percentage out of all the positive, precision determines the proportion of the truly positive out of all the positive predicted. The harmonic mean of recall and precision is known as the f1-score. Recall reveals the probability of false negatives and false positives. The LSTM network has the overall lowest precision score when comparing the precision values of the various classes. Similar results are seen with BLSTM; the parameters differ significantly from LSTM. The results for the parameters selected for the performance analysis are summarised in Table 7.1, with GRU showing a superior response than the other two deep learning models when all deep learning models are taken into account.

7.6 Prediction of Direction with GRU Network

The results of the model implementation and training process clearly demonstrate that the GRU network, with an accuracy of 97.38% and other distinct model parameters, is the most accurate and efficient model. In order to analyse the prediction, the model is therefore validated using random sensor range data. Figure 7.11-7.16 below shows the direction produced by the GRU model in a random fashion.

```
[ 0.936511    0.06007826 -0.02668529 -0.07560565  0.00355033  0.04945274]
0.936511
sensor read= [[[ 80]
  [100]
  [ 60]]]
0
Direction= forward
```

Figure 7.11: Model output as forward

```
[ 0.02116116  1.1479557  -0.08910735 -0.03669929 -0.07382558  0.00429668]
1.1479557
sensor read= [[[80]
  [30]
  [40]]]
1
Direction= right turn
```

Figure 7.12: Model output as right turn

```
[-0.02545607 -0.01810287 -0.02352797  1.0093516  -0.07507244  0.0118413 ]
1.0093516
sensor read= [[[ 5]
  [80]
  [30]]]
3
Direction= left turn
```

Figure 7.13: Model output as left turn

```
[ 0.02240426 -0.02794015  1.0446837   0.01326872 -0.01979962 -0.09058261]
1.0446837
sensor read= [[[40]
  [30]
  [ 5]]]
2
Direction= right spin
```

Figure 7.14: Model output as right spin

```
[ 0.01099796  0.12566508 -0.02535784 -0.1475829   0.94117224  0.0341617 ]
0.94117224
sensor read= [[[15]
  [45]
  [80]]]
4
Direction= left spin
```

Figure 7.15: Model output as left spin

```
[-0.10263832 -0.06782288 -0.03414628 -0.04182535 -0.074838    1.2140853 ]
1.2140853
sensor read= [[[ 5]
  [15]
  [15]]]
5
Direction= stop
```

Figure 7.16: Model output as stop

7.6.1 Simulated Model

As previously mentioned, a virtual environment is created for the simulation of a powered wheelchair model utilising python programming and modules and readily available general equations. All three deep learning network models were loaded, simulated, and confirmed with regard to user input and output control instructions. For direction prediction, it was discovered that GRU network accuracy finally reached 97.38% and the output graphs and confusion matrix produced for the same is shown in Figure 7.5-7.6 and Fig.7.10 so that the GRU network model was more accurate and efficient. The output is verified using various scenarios given in Chapter 3 and the result is examined using the anticipated path as shown in Fig. using a GRU network model and user input using a keypad.

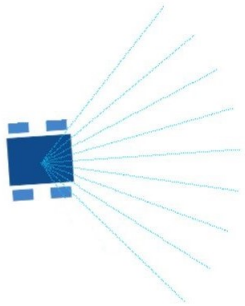

Figure 7.17: Resultant PW direction as forward

The first scenario considered is when the powered wheelchair starts moving in the with nothing in its path. The result obtained is that as the powered wheelchair model starts moving with nothing in range of the sensors, all the distances were set to faraway. The resultant direction of GRU model and user input for this case is to move in forward direction, shown in Figure 7.17.

In the second scenario, when the powered wheelchair moved toward the front, it was assumed that it would notice an object to its right. The outcome demonstrated that as the sensor detects

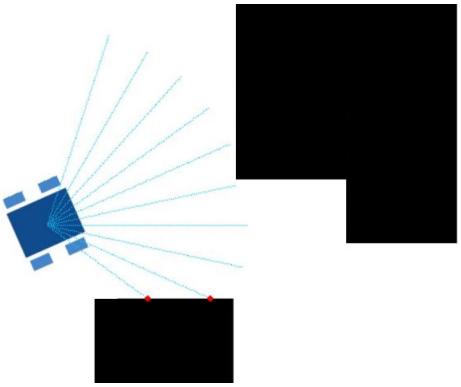

Figure 7.18: Resultant PW direction as left turn

things in its vicinity, the distances are set to close by or adjacent, and the GRU network model's and the user's suggested direction is to proceed forward while making a left turn. Projected course is depicted in Figure 7.18.

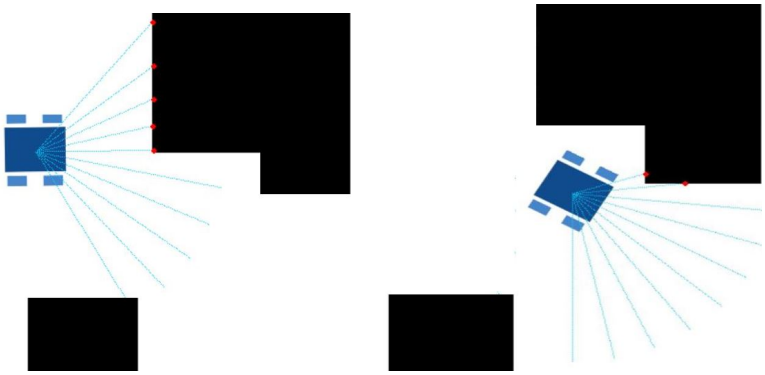

Figure 7.19: Resultant PW direction as right turn

The third and final case is when the powered wheelchair notices something blocking its left and frontal lanes. The simulation's findings demonstrate that when the powered wheelchair model moves forward in the second scenario and detects objects to its left and in front of it, the necessary distance command is set, and the GRU network model, along with the user's input, predicts the direction of the subsequent movement. The combined direction order is to turn right and proceed to the destination, shown in Figure 7.19.

7.7 Comparison Study

Long Short Term Memory (LSTM), Bidirectional LSTM (BLSTM) and Gated Recurrent Unit (GRU) results are taken into account for comparison purposes, as the network models for each

Table 7.1: Comparison of various deep learning techniques

DL model	Metrics	Forward	Right turn	Left turn	Right spin	Left spin	Stop
LSTM	Precision	0.96	0.94	0.98	0.95	0.97	0.95
	Recall	0.96	0.96	0.97	0.94	0.96	0.95
	f1-score	0.97	0.95	0.94	0.95	0.97	0.96
	Accuracy	95.17%					
BLSTM	Precision	0.97	0.95	0.97	0.96	0.97	0.96
	Recall	0.96	0.97	0.96	0.96	0.98	0.96
	f1-score	0.97	0.96	0.95	0.96	0.98	0.97
	Accuracy	95.89%					
GRU	Precision	0.98	1.00	0.98	0.97	0.98	1.00
	Recall	0.98	0.98	0.97	0.97	0.99	0.99
	f1-score	0.98	0.97	0.97	0.98	1.00	1.00
	Accuracy	97.38%					

technique are identical procedures as explained in the methodology with the exception of model layer implementations. Each model's output is assessed, and the relevant conclusions are drawn. Different parameters are considered and corresponding difference in value of the same is shown in Table 7.1. The accuracy of the LSTM model is 95.17%, while the output of the BLSTM technique was 95.89%. There was a significant difference between the two methods when testing with various validation data inputs. Additionally, these models and user input did not accurately manage wheelchair navigation. Among all the three deep learning techniques GRU neural network proved to be efficient and accurate method with an accuracy of 97.38% and other parameters in both model implementation and powered wheelchair model simulation

7.8 Concluding Remarks

This chapter dealt with the outputs of the proposed deep learning models which include the LSTM, BLSTM and GRU neural networks. The comparison of the confusion matrices produced suggested the most effective approach. The chapter verifies the predicted direction output and the PW simulation model with GRU. Next chapter deals with the conclusion.

Chapter 8

CONCLUSION AND FUTURE SCOPE

The work presented would benefit the majority of people with disabilities in our society and enhances their independent movement as well as their quality of life by boosting their self-assurance and self-reliance. For some Powered wheelchairs (PW) users, navigation and collision avoidance both require steering assistance, or the ability to manage speed and direction. In order to increase the safety and mobility of persons who use PWs, a wheelchair and assistive technology work together to create a Smart Wheelchair (SW). Here rule-based deep neural networks are applied to aid with PW steering. The proposed system suggests a resultant direction while avoiding objects on collision, and efficiently classifies and predicts a direction for wheelchairs, and this is mixed up with user input from joystick, chin tubes, keypad etc. If the user wants to improve navigation performance, he or she can override the system. The system is neural network based and can be integrated to any types of powered wheelchairs or existing powered wheelchairs, making it faster and more dynamic in response to sudden appearance of objects. Therefore, the safest and most secure route with object avoidance is predicted using neural networks. LSTM, BLSTM and GRU are the key models considered in this work. This study provides PW simulation model and comparison of these frameworks, among all the three GRU model showed the best performance in direction prediction and during PW model simulation with an accuracy of 97.38%. The future scope of the project includes considering more advanced and computationally cheap AI techniques.

References

[1] Malik J Haddad and David A Sanders, "Deep Learning to Assist with Steering a Powerd wheelchair," *IEEE Transactions on Neural systems and Rehabilitation Engineering*, vol 28, no. 12, pp. 2987-2994, 2020.

[2] Malik J Haddad and David A Sanders, "Selecting a best compromise direction for a powered wheelchair using PROMETHEE," *IEEE Transactions on Neural systems Rehabilitation*, vol 27, no. 2, pp. 228-235, 2019.

[3] David A Sanders, "Using self-reliance factors to decide how to share control between human powered wheelchair drivers and ultrasonic sensors," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol 25, no. 8, pp. 1221–1229, 2016.

[4] Sooyong Lee, "Use of infrared light reflecting landmarks for localization," *Industrial Robot: An International Journal*, vol. 36, 2009.

[5] David A Sanders, Alexander Gegov, Giles E Tewkesbury and Rinat Khusainov, "Rule-based system to assist a powered wheelchair driver," *2017 Intelligent Systems Conference (IntelliSys)*, pp. 558–565, 2017.

[6] David A Sanders, Martin Langner and Giles E Tewkesbury , "Improving wheelchair-driving using a sensor system to control wheelchair-veer and variable-switches as an alternative to digital-switches or joysticks," *Industrial Robot: An International Journal*, 2010.

[7] David A Sanders and Ian Stott, "A new prototype intelligent mobility system to assist powered wheelchair users," *Industrial Robot: An International Journal*, 1999.

[8] D.R Parhi and M.K Singh, "Heuristic-rule-based hybrid neural network for navigation of a mobile robot," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 224, no. 7, pp. 1103–1118, 2010.

[9] David A Sanders, "Non-model-based control of a wheeled vehicle pulling two trailers to provide early powered mobility and driving experiences," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 1, pp. 96–104, 2017.

[10] Edmund F Lopresti and Vinod Sharma,"Performance testing of collision-avoidance system for power wheelchairs," *Journal of Rehabilitation research and Development*, vol. 48, no. 5, pp. 529–44, 2011.

[11] Paul Nisbet, John Craig and Odor, "Smart wheelchairs for mobility training," *Elsevier science: Technology and Disability,*, vol. 5, no. 1, pp. 49–62, 1996.

[12] Ya-chun chang, Yoshio Yamamoto, "On-line path planning strategy integrated with collision and dead-lock avoidance schemes for wheeled mobile robot in indoor environments," *Industrial Robot: An International Journal*, pp. 100–118, 2008.

[13] Yeontack jung, Yoonjae Kin *et.al*, "Path planning algorithm for an autonomous electric wheelchair in hospitals," *IEEE Access*, vol. 8, pp. 208199–208213, 2020.

[14] Chao Wang, Alexey S Matveev and Savkin *et.al*, "A collision avoidance strategy for safe autonomous navigation of an intelligent electric-powered wheelchair in dynamic uncertain environments with moving obstacles," *2013 European Control Conference (ECC*, pp. 4382–4387, 2013.

[15] Tarek Taha, Jaime Valls Mir, and Gamini Dissanayake, "Wheelchair driver assistance and intention prediction using POMDPs," *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pp. 449–454, 2007.

[16] Alexander Hintemann, Eric Demeester and Vander Poorten, "Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model," *2013 IEEE International Conference on Robotics and Automation*, pp. 4376–4383, 2013.

[17] E. Demester, M. Nuttin, D. Varhooyclonck, "A model-based, probabilistic framework for plan recognition in shared wheelchair control: Experiments and evaluation," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2, pp. 1456–1461, 2003.

[18] Vinh The Nguyen, Chandimal Jayawardena and Iman Ardekani, "A navigation model for side-by-side robotic wheelchairs for optimizing social comfort in crossing situations," *Jounal in Robotics and Autonomous Systems*, vol. 100, pp. 27–40, 2018.

[19] Haruna Kokubo, Takaaki Shibanoki and Toshio Tsuji, "Obstacle Avoidance Method for Electric Wheelchairs Based on a Multi-Layered Non-Contact Impedance Model," *Journal of Robotics Networking and Artificial life*, vol. 4, no. 1, pp. 45–48, 2017.

[20] Yi-Tseng Lin and Chung-Hsien Kuo, "Development of SSVEP-based intelligent wheelchair brain computer interface assisted by reactive obstacle avoidance," *2016 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1572–1577, 2016.

[21] Raulcezar MF Alves, Carlos R Lopes, "Obstacle avoidance for mobile robots: A hybrid intelligent system based on fuzzy logic and artificial neural network," *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1038–1043, 2016.

[22] Makoto Itoh, Toshiyuki Inagaki and Hiroto Tanaka, "Haptic steering direction guidance for pedestrian-vehicle collision avoidance," *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3327–3332, 2012.

[23] Martin Langner, "Effort reduction and collision avoidance for powered wheelchairs: SCAD assistive mobility system," *Diss. Department of Mechanical and Design Engineering, University of Portsmouth, Portsmouth,UK,*, 2012.

[24] Klaus Greff and Rupesh K Srivastava, *et al.*, "LSTM: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

List of Publications

[1] Bijeesh B, Sumayya Jaleel, "Steering Control Assistance for a Powered Wheelchair using Deep Neural Frameworks,"*International Congress on Internet of Things, 2022-* Accepted.

[2] Bijeesh B, Sumayya Jaleel, "Electric Wheelchair Navigational Assistance using Deep Learning Techniques," communicated to *IEEE 19th India Council International Conference (INDICON), 2022.*