PHYSICAL REHABILITATION AND CLASSIFICATION OF MOTOR IMPAIRMENTS USING MACHINE LEARNING

BY

OGBONNA CHUKWUDI AJAERO

16CJ020749

A PROJECT REPORT SUBMITTED TO DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING IN PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE OF BACHELOR OF ENGINEERING COMPUTER ENGINEERING.

SUPERVISED BY PROFESSOR SANJAY MISRA

JULY 2021

DECLARATION

I hereby declare that the work reported in this thesis was carried out by me under the supervision of Professor Sanjay Misra in the Department of Electrical and Information Engineering, Covenant University. Also, I declare that to the best of my knowledge, no part of the report has been submitted here or elsewhere in a prior application for the award of a degree. All sources of knowledge used herein have been duly acknowledged.

……………………………………….

(Signature and date)

Ogbonna Chukwudi Ajaero

16CJ020749

CERTIFICATION

This is to certify that the project titled *Physical Rehabilitation and Classification of Motor Impairments using Machine Learning* by Ogbonna Chukwudi Ajaero meets the requirements and regulations governing the award of the Bachelor of Engineering, B.Eng. (Computer Engineering) degree of Covenant University and is approved for its contribution to knowledge and literary presentation.

Supervisor 1 Sign: __________________ Date: _______________

Name: Prof Sanjay Misra

Supervisor 2 Sign: __________________ Date: _______________

Name: Mrs. Comfort Lawal

Head of Department Sign: __________________ Date: _______________

Name: Prof Emmanuel Adetiba

Internal Examiner Sign: ___________________ Date: _______________

Name: ___________________________

DEDICATION

This research project is dedicated to my parents – Engr Okechukwu Ogbonna and Mrs Ngozi Ogbonna, who were both very understanding and supportive all through the time I took to complete this project and provided the resources to make this project possible.

ACKNOWLEDGEMENT

Immense gratitude goes to God Almighty, who has gifted me with life to finalize this project and gave me the grace to travel through this era of research and execution of the project.

I appreciate my parents, Mr. Okechuku Ogbonna and Mrs. Ngozi Ogbonna, who were patient and supportive in helping me financially and morally. I thank my supervisor Prof. Sanjay Misra for his relentless efforts to make me have a more robust understanding of the subject and bring this project to life. I remember starting this project with little to no knowledge of the technologies to be used and that they were patient with my growth and encouraged me at every step. I am more content with each moment I spent functioning on this project. I also want to thank Mrs. Comfort Lawal, who helped and supported me in realizing the objectives of this project.

I also recognize the help and support of my colleagues throughout our journey. It was an honor working with each of them. Lastly, I am thanking the leadership and administration of Covenant University, beginning with the Chancellor Bishop David Oyedepo, the Vice-Chancellor, the Registrar, the HOD EIE and everyone in the school for providing a platform on behalf of myself to further my knowledge.

.

ABSTRACT

In recent times, Physical Rehabilitation has become a requirement for anyone afflicted with motor impairments or anyone with the inability to perform daily motor activities either from accidents or natural aging. This paper entails the advantages and influence Artificial intelligence and Machine Learning have on physical rehabilitation. Classification of motor impairments may come a long way in the field of medicine, it would reduce the need of having a real time physiotherapy session. The project will be able to assist physiotherapists in classifying motor disabilities while also reducing the costs incurred by disabled patients during Physical Rehabilitation. This project aims to create a Machine Learning model that classifies the different types of motor impairments based on some certain features while also producing a smooth and simple user experience. Studies have shown that motor abilities and motor impairments are highly correlated. The use of wearable technology will also be employed in this project. This paper makes use of 4 different classification algorithms namely: Decision Trees, Support Vector Machine, Naïve Bayes and k-NN, the algorithm with the highest performance metrics was chosen. The model built was able to achieve a 92% accuracy and was deployed in a desktop application using Kivy.

TABLE OF CONTENTS

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1 BACKGROUND TO THE STUDY

In Nigeria, physical rehabilitation is required because it may provide those with life-altering disabilities a glimmer of hope. Because of limitations caused by accident or age, the data show that an increasing number of people need physical therapy. Stroke and cerebral paralysis are the most frequent causes of disability in people. In the field of physiotherapy, new technologies are being developed to reduce costs while keeping the highest quality of service. [1]

Physiatrists are recognized to be doctors specializing in the assessment and treatment of individuals with physical disabilities in connection with an accident or disease. Mechanical (massage), electromagnetic (heat/cold, light, ultrasound) and hydrotherapeutic (traction) therapies will be utilized in treating those recuperating from sport injuries and motor vehicle accidents (MVAs) along with those suffering from a musculoskeletal disease with restricted joint mobility.

In recent years physiotherapy has become increasingly important for injury treatment and for improving overall physical mobility, physical strength, and physical function as well as for the treatment of chronic diseases due to the increased use of movement science rather than medicine and surgery. Physiotherapy is growing increasingly popular as people are physically active. In addition, a significant rise in the demand for rehabilitation centers. [2]

The most well-known innovation was "the hook," which was used to restore lower limbs that had been amputated or badly damaged. Renaissance innovation was particularly notable in this area; during this time, the creation of real prostheses was facilitated using plastic, iron, and wood.

Many people seem to confuse Physical Therapy with Physical Rehabilitation; however, there is a subtle and significant difference; physical therapy is the restoration of function, assisting you in regaining your independence in the safest and most effective manner possible; physical Rehabilitation is the restoration of function. A person's strength, mobility, and fitness will improve because of physical therapy, while their recovery from a serious injury will be aided by Rehabilitation.

Physical therapy is most often used in the treatment of mild to moderate injuries, while physical Rehabilitation is most utilized in the treatment of severe injuries or accidents.

Physical Rehabilitation has been frowned upon in recent times, with many of the population citing it as "expensive," "a complete waste of resources," and "time-consuming," and who can blame them? Nigeria's financial state is not one to be encouraged. Recent statistics have shown that only 1% of the population of Nigeria can afford full-time Rehabilitation.

Although seen as 'useless,' physical Rehabilitation is, in fact, the opposite. Listed are the numerous benefits:

1. Eliminates surgery as an option
2. Reduces pain
3. Improves recovery rate from motor-impairments
4. Improves your balance
5. Manage Diabetes and Cardiovascular issues

With recent advancements in technology, a game-changing field has emerged, named Machine Learning,

Machine Learning is training your computer/model to see patterns in your data so that it may be able to make predictions on unseen data accurately. It is difficult to achieve 100% accuracy because data is generally noisy, and getting accurate patterns from that data is termed difficult

Generally, data preprocessing is employed, but this is essentially costly, especially for very large datasets, hence simple encoding patterns are used to minimize the complexity of our model.

With machine learning, physical Rehabilitation becomes much more productive, and it can now accurately predict the specific type of physical disability. It makes use of simple classification algorithms that see patterns in data and generalize them. Directly with the help of a desktop application, patients afflicted with physical disabilities can easily measure their progress as they perform these exercises daily. It can also predict the end date of a person's physical rehabilitation journey, minimizing the need for actual physiotherapists in our use case.

Many algorithms depend on the availability of big data to accurately make predictions that generalize the population of physically disabled patients. But data is relatively costly, and this would hence increase the expenses in providing the people with suitable devices they can use for exercise.

Misclassification is a major problem with machine learning in Health; Health is often seen as a precarious field with little to no margin of error; misclassification of a certain disease or injury can lead to further complications and, in extreme cases, even death. That is why 100% accuracy is sought after, as there is no room for error, but this is just theoretical, as this is impossible in real-life scenarios regardless of the amount of data one has.

This paper will be making use of the supervised learning approach which uses labeled data to train our ML model to be able to accurately make predictions on new test data, the paper also shows comparisons between the popular supervised learning algorithms such as Decision Tree, k-NN, Support Vector Machine and Naïve Bayes to pick the algorithm with the best performance metrics. This project work tries to bridge the gap between physical Rehabilitation and technology.

1.2 SIGNIFICANCE AND MOTIVATION OF STUDY

Physical Rehabilitation in recent times has become quite expensive, and most of the population is unable to afford it or complete the entire procedure. People afflicted with motor impairments must perform live checkups daily to evaluate their performance and this may be uncomfortable for them, as they must travel long distances daily especially without the help of others.

Traditional exercises in rehabilitation programs make it difficult for physiotherapists to objectively track, measure and monitor patients' progress & recovery.

With Physical therapists can observe a lot of movement, but it's hard for them to get all the information they need.

My project helps to address the stated problem

1.3 PROBLEM STATEMENT

This project seeks to address the high costs incurred by many physically disabled patients during a standard physical therapy session, many of these costs include

1. Transportation costs
2. Consultation costs
3. Therapy costs

The ratio of abled physiotherapists to disabled patients raises a cause for alarm. The project also helps to provide load sharing for physiotherapists by reducing the need for physiotherapists to be always present during rehabilitation exercises, hence improving productivity and efficiency of these physiotherapists during patient evaluation

The project additionally helps to fill in the void between technology and physiotherapy, and its various applications by providing ways of translating body motions extracted from wireless sensor technologies into output values that determine the most probable physical impairment along with the seriousness of these physical impairments

1.4 AIM AND OBJECTIVES OF THE STUDY

1.4.1 AIM

This project aims to create a platform where patients can be able to self-diagnose their motor impairment with the use of high accuracy, complex classification algorithms

1.4.2 OBJECTIVES

The objectives of the project are:

1. To create an SVM model that classifies motor impairments.
2. To make a simple bar chart easily interpretable by everyone.
3. To create an app with a simple user interface, to create a smooth user experience
4. To deploy my model using my developed app

1.5 METHODOLOGY

The application would be programmed, making use of the Python programming language for creating a user interface suitable for people non-conversant with newer technologies or rather not computer literate. It uses simple buttons spaced apart with a very natural black color that aids readability and improves transparency. It uses big white fonts for people with cognitive disabilities.

The ML model is a classification model that makes classifications based on a certain number of features. The classification algorithm to be implemented is the Support Vector Machine, the reason for this choice shall be explained in later chapters. In an SVM algorithm, the features represent the number of dimensions, it is an eager learning approach because as data points are fed into the algorithm, it starts introducing hyperplanes to distinguish each class from each other and then waits for a test tuple to properly classify it.

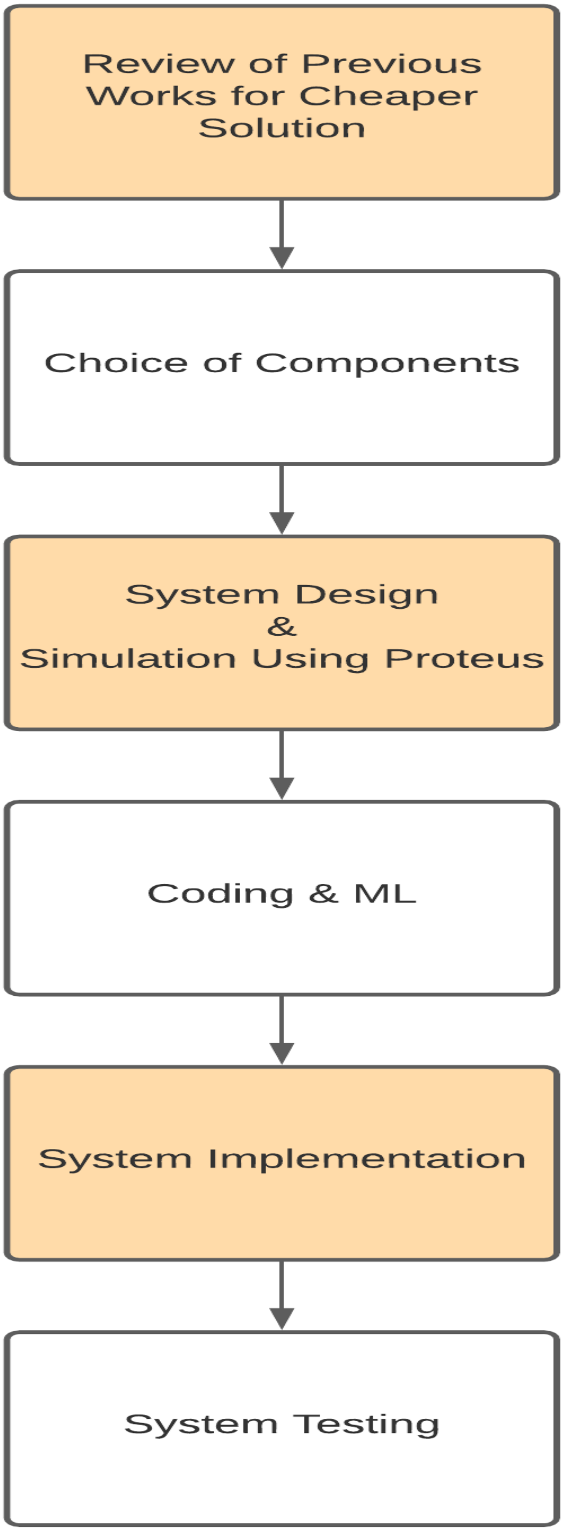

Review of Previous Works for Cheaper Solution
Choice of Components
System Design & Simulation Using Proteus
Coding & ML
System Implementation
System Testing

Figure 1.1 Block Diagram for the System

The figure above represents the flowchart of the project, which starts with review of past and related works to see alternative and cheaper solutions for the general public, we then evaluate and access the reason for our choice of components, then we design the system using tools like Proteus and PyCharm, then the software development and programming is done, along with the production and deployment of our supervised learning model, then the sub systems are combined and implemented, then finally our system is tested for release. The hardware part of this project would be implemented and tested by my colleague Amadi Lawrence.

1.6 SCOPE OF THE STUDY

The scope of this project is to develop an application that categorically classifies motor impairments using data extracted from wireless sensor technology. The mobile application is built using Kivy on the Python library. It utilizes the pre trained SVM model from Scikit learn

1.7 LIMITATIONS

Every system is subject to limitations as no system is 100% efficient. The following are certain limitations associated with the system.

1. Finding datasets proved to be an issue as this was a generally new field in science
2. The Kivy framework was limited to some user gestures and icons
3. The small number of datasets could lead to inaccurate results, and lots of variance across different datasets

4. Training and test data were sampled from the same sampling distribution.
5. Packet loss from Bluetooth model could affect readings.

1.8 PROJECT ORGANIZATION

The project is ordered to contain the following chapters

Chapter 1 contains the background and significance of study, the problem statement, the aim and objectives our project hopes to address, the methodology, scope, limitations and the overall project organization

Chapter 2 contains the introduction, the definition of concepts used in our paper, the review of our related works, and the summary giving a recap of the entire chapter

Chapter 3 contains the System design, system description, modelling, design, analysis, data collection and analysis, software development, system simulation and the implementation of our software

Chapter 4 contains our results and discussions, where we compared the performance metrics of our 4 available classification algorithms and provided reasons for selecting the SVM model.

Chapter 5 contains our results, summary, conclusion, recommendations and limitations of our project.

CHAPTER TWO: LITERATURE REVIEW

2.1 INTRODUCTION

Artificial intelligence is described as the advancement of technology that is utilized to accomplish technological activities that require human intelligence. Artificial intelligence has recently played a significant part in technological growth. Machine learning is a vital component of artificial intelligence that allows us to train our model using both supervised and unsupervised learning methods. Furthermore, machine learning provides a plethora of models that can be employed to improve our system's training and prediction. Today's AI technology can take many forms, including software programs and hardware interfaces, to create a system that can learn from its own datasets. In our project, AI and machine learning are utilized to identify posture and subsequently analyze patients.[1]

Artificial intelligence techniques are being used in a variety of settings, including health care and research. Machine learning is a subfield of artificial intelligence that studies algorithms that improve themselves over time. In the field of Rehabilitation, artificial intelligence (AI) offers a variety of uses. In health care, AI beat cardiovascular risk algorithms and detected skin cancer as well as or better than a dermatologist. ML is being utilized in Rehabilitation for symbiotic neuroprosthetics and myoelectric control, brain computer interface technologies, perioperative medicine, and other applications. In musculoskeletal medicine, machine learning approaches have been applied in diagnostic imaging, patient data assessment, and clinical decision support.[2]

In healthcare, machine learning techniques are used to enhance patient outcomes by utilizing the growing amount of health data offered by the Internet of Things. These strategies have a lot of potential, but they also have a lot of drawbacks. The three main areas machine learning is applied to include medical imaging, natural language processing of medical documents, and genetic information. Many of these fields are concerned with diagnosis, detection, and forecasting. Currently, a massive infrastructure of medical devices creates data, but there is often no supporting infrastructure in place to effectively use that data. Medical information comes in a variety of formats, which can complicate data formatting and increase noise.

This chapter features different literature and discusses their strengths, weaknesses and gaps concerning this study alongside the various technologies that make up the voice chatbot system.

2.2 DEFINITION OF CONCEPTS

2.2.1 ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is a computer's ability to believe or create a computer-controlled robot that can normally conduct intelligent human tasks. The phrase is frequently used to create systems with human abilities such as knowledge, discernment of meanings, generalization, learning from previous experiences. Digital computer technology in the 1940s showed that computers can be taught to do very complex tasks, like as proving mathematical theorems and playing chess. No software has yet equal human adaptability, despite continued progress in computer processing speed and memory capacity. Applications that utilize artificial Intelligence in this limited meaning include examples of medical diagnosis, computer search engines and voice or handwriting recognition. [3]

Bridging the AI veil shows a host of difficulties. The primary issue was a lack of computational power to do anything important: computers were not able to store or analyze information fast enough. For example, to communicate one needs know and understand the meanings of numerous words in a number of situations. Hans Moravec, McCarthy PhD student. In the 1980s, two things re-energized AI: extension and rise in financing of the algorithm toolbox John Hopfield and David Rumelhart developed 'deep learning' methods which enabled computers to learn via experience. On the other side, Edward Feigenbaum has pioneered systems of experts which reflect a human expert's decision-making process. Non-experts might get software advice when they asked an expert in a certain area how to answer under a certain situation. [4]

The branches are:

Machine Learning

Machine learning is a technique for learning without being expressly scheduled by computers. It's being used actively in everyday life, including apps that many people don't know about Science enables machines to understand, implement and investigate data in order to solve problems in the real world.[3]

Neural Networks

The neural network is an artificial intelligence area which uses neurology and includes knowledge and machinery to do tasks (a part of biology that concerns the nerve and nervous system of the human brain). The neural network imitates the human intelligent brain, which is full of neurons, and the purpose of the neural network is to code brain neurons into a system or computer. [3]

A neural network is basically a collection of algorithms which serve to discover elementary correlations between huge quantities of data using a technique which imitates the functioning of the human brain. A neural network is a system of neurons either natural or artificial in nature that refer to artificial neurons as perceptrons and understand the whole perceptron model in the neural network. In a neural Network, a new neuron is a mathematical function which collects and categorizes input according to a certain structure (such as activation functions). The network depends largely on statistical methods such as regression analysis to achieve this objective.[3]

Robotics

This has become a center for research in artificial intelligence. The design and building of robots is a fascinating topic of R&D. Robotics is a multidisciplinary field, which encompasses mechanics, electrical engineering, computer science and many other areas.[3]

The science of the design, fabrication, operation, and usage of robots is robotics. It deals with the control, intelligent outcomes and computer systems information transformation.[3]

Expert Systems

Expert systems were considered one of the first AI software models to be successful. They were originally created in the 1970s and subsequently augmented in the 1980s. Under AI technology, a system of experts is a computer system that replicates a human expert's decision-making intelligence. It does so via reasoning and insights rules in response to user enquiries, which draw information from its knowledge store..[3]

The success of the expert system depends completely on the knowledge of the expert stored in a knowledge base. The more information the system gathers, the greater the efficiency.

For example, the expert system offers recommendations for orthography and errors in Google Search Engine.[3]

Fuzzy Logic

In the world today we sometimes find circumstances where it is hard for us to identify whether a condition is true. Its fugitive logic allows for considerable flexibility in thought, which in every case leads to mistake and uncertainty. In other words, the fugitive logic is a technique to depict and alter confusing information by assessing to what extent a hypothesis is correct. Fuzzy logic is used to explain concepts that are confusing. Fuzzy logic is an easy, flexible method to use machine learning techniques and help to replicate human knowledge logically.[3]

Natural Language Processing

NLP is an information search, analysis, understanding, and extraction technique. Programmers utilize NLP libraries to instruct computers on how to extract significant information from text input. Computer algorithms may determine whether an e-mail is a garbage or not by looking at the subject of a row or the text of an NLP e-mail.[3]

A field of computer science and artificial intelligence, NLP enables computers and people to interact naturally. It's a computer-aided language analysis technique. It enables a computer to understand and analyze data by mimicking conversational language..[3]

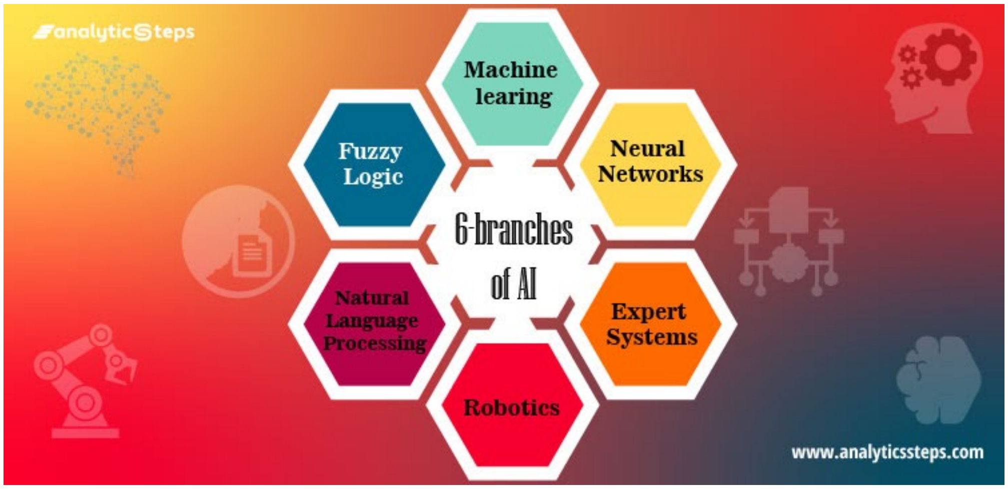


Figure 2.1 6 branches of AI

2.2.2 MACHINE LEARNING

Machine learning is a type of artificial intelligence that enables computers to learn and grow. Computer programs that can learn and access data are called self-learning programs. The learning process begins with observations, facts, examples, or a mix of the two. we may search for trends in the data and make better choices in the future based on our examples. The goal is for computers to learn on their own and alter their behavior accordingly. Semantic analysis replicates a person's capacity to understand the meaning of a document, as opposed to conventional machine learning techniques.[4]

Types of ML Algorithms:

Supervised Algorithms

Supervised machine learning algorithms can utilize labeled examples to apply previous learning to new data and predict future events. Based on a known training dataset, the learning technique infers a function to predict output values. It may set objectives for fresh input after appropriate training. The learning system may also detect errors and update the model as needed. [4]

Unsupervised Algorithms

When the data being learned isn't categorized or labeled, unsupervised machine learning is used. For example, computers can derive a function from unlabeled data to describe a hidden structure. Inferences from datasets may be drawn to explain unlabeled data with hidden structures. [4]

Semi-Supervised Algorithms

Semi-supervised machine learning algorithms train using both labeled and unlabeled data — generally a small amount of labeled data and a large number of unlabeled data. If used, this approach may greatly increase learning accuracy. To train / learn from labeled data, semi-supervised learning is commonly used. Labeled data retrieval requires fewer resources..[4]

Reinforcement Learning Algorithms

It is a type of learning algorithm that generates actions while also sensing their environment, faults, or rewards. Reinforcement learning is based on trial-and-error search and delayed reward. This approach allows machines and software agents to communicate to automatically pick the best behavior in each situation and optimize efficiency. The agent must then learn which action is optimal by getting basic reward feedback known as the reinforcement signal..[4]

With Machine learning, we aim to solve two types of problems namely:

Classification

There are various methods for classifying data, including using a function to help in categorization. It is used to train computer algorithms. then classifies the data based on that training. Goal of the classification algorithm is to find the mapping function from discrete input(x) to discrete output(y).[5]

Examples include:

- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machines
- Kernel SVM
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

Regression

Regression is the study of relationships between dependent and independent variables. It helps forecast continuous variables such as market movements and property values. This technique aims to find a mapping function from a continuous input (x) to a discrete output (y) .[5]

Examples include:

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

2.2.3 APP DEVELOPMENT

When it comes to developing software for tiny, wireless computing devices such as smartphones and other hand-held devices, mobile application development is the collection of processes and procedures that are involved. As with web application development, the origins of mobile application development may be traced back to more conventional software development methods. Mobile applications, on the other hand, are often developed specially to take use of the unique characteristics of a particular mobile device, which is an important distinction. Using an example, a game app might be developed to take use of the accelerometer on an iPhone, while a mobile health app could be written to take advantage of the temperature sensor on a wristwatch. Apple's iOS and Google's Android are the two most widely used mobile systems today, with iOS being the most popular. Apple's smartphones and tablets come preinstalled with a variety of important apps, including a complete web browser and access to the Apple App Store. In addition to iOS devices, Android smartphones come preinstalled with comparable applications, and you can add more by visiting the Google Play Store.[6]

It is usually split into:

Front End Development

Front end development is a type of computer programming that focuses on the coding and building of user-visible components and functionalities of a website. It's about ensuring that a website's aesthetic elements are functioning. Front end is often known as the "client side" of an application.

They consist of elements like:

- Buttons
- Layouts
- Navigation
- Images
- Graphics
- Animations
- Content organization

Back End Development

The portion of the website that visitors don't see is called back-end development. It is what allows a website to be interactive. Let's suppose you're the proprietor of a social networking platform, and the back end of your website is referred to as the "server side." To store all of your users' data, you'll need a single place. A database is a kind of storage facility, with Oracle, SQL Server, and MySQL being prominent examples. A server, which is just a computer that is placed someplace, manages databases. A back-end developer will be in charge of this database and the site material it contains. This ensures that as visitors explore submitted content and other user profiles, your social networking website's front-end features continue to function correctly. They consist of:

- Building code
- Troubleshooting and debugging web applications
- Database management
- Framework utilization

2.3 REVIEW OF RELATED WORKS

The authors in [7] presented a paper that utilized wearable technologies and virtual reality applications to access patients' involvements during physical rehabilitation exercises. The paper failed to account for the lower limbs of Stroke patients, which is also vital in their Rehabilitation. The project also failed to account for patients with cognitive disabilities such as epilepsy or blindness, as people afflicted by this would be unable to play the Virtual Reality games and their progress would be inhibited.

The authors in [1] presented a paper that introduced the use of a mobile app to access physiotherapy activities of users, and then give recommendations based on their individual performances to improve therapy. The project utilized the Open Pose Library to account for the angles between each joint. This type of treatment requires a large amount of data in order to achieve high accuracy, and this may be prohibitively expensive in areas where data is not readily available. The system operates in real time, and there is a significant likelihood of packet loss, which may cause the model's readings to be skewed, resulting in misclassifications or failure to identify the motions of the joints at all.

The authors in [8] presented a paper that used an inertial sensor to test the type of physical exercise performed by an individual during lower limb exercises. 58 people did each of the seven lower-limb exercises 10 times. Three inertial sensor units were attached to the leg being trained: thigh, shin, and foot. The data was quantified using machine learning algorithms. The research did not account for tiredness, which may occur while doing many exercises in a short amount of time, particularly for a recuperating patient. This might lead to bias in the machine learning model, which appears as an exercise not being completed correctly.

The authors in [9] presented a paper that addressed the purpose for both active and passive exercises for individuals afflicted by acute stroke. The model producing torque gives immediate live feedback to individuals wearing them. When the mode is passive, the device forces the ankle safely to properly flex it. The patient is usually advised and motivated to participate actively in the movements during gameplay. Their study however failed to consider poor speed control of the motor, or a faulty motor, which can increase the torque, and hence increase the speed of the robot attached to the ankle, hence causing further complications during exercise.

The authors in [10] presented a paper that utilized socks that had sensors embedded in them, these were then worn by the patients, the project utilized a web interface that displayed data composed of weight distribution, foot movement, foot orientation which was constantly updated in real time for physiotherapists during video sessions. Students undergoing their PHD program evaluated the functions of the lower limb in a 40-hour video consultation. The project utilized a sensor with a small sensing area, and this tends to provide inaccuracies with a large sensing area such as the soles of feet, hence measurements provided may not truly be accurate.

The authors in [11] presented a that proposed an autonomous system built to properly identify physical rehabilitation exercises, and analyze the data using wearable technology. The project utilized a multi-template, multi-match dynamic time warping (MTMM-DTW) technique to identify numerous occurrences of more than one exercise type in the recording of a physical therapy session as a logical extension of DTW.

The authors in [12] presented a paper where Individuals afflicted with Cerebral Palsy were asked to participate in the study, accelerometer and gyroscope motion sensors were used to measure physical activity. To make and test machine learning models for identifying PA types in cerebral palsy children. The paper made use of popular classification algorithms and picked the model that gave the best performance metrics., but a Deep Learning strategy may offer substantially higher accuracy and be more suited for reuse in the case of various physical impairments in children and adolescents, as well as generate better generalization throughout the population.

The authors in [13] presented a paper that extracted symptoms from case sheets and use this data to train a model. 507 patient case files were collected from Sugam Multispecialty Hospital in Kumbakonam, Tamil Nadu, India. The stemmer was utilized to extract common and distinctive features mainly used to classify stroke. Multiple machine learning techniques were used to analyze the data, including ANNs, SVMs. Boosting, Bagging, and random forests were used to improve the performance of the "overfitting models".

The authors in [14] presented a paper that attempted to classify ischemic stroke using the popular classification algorithms like k-NN and Decision trees, based on the results Its classification algorithm is based on a dataset of 400 data points that were gathered from different Sudanese medical centers. It is beneficial for medical specialists in the field to utilize the findings of the decision tree algorithm to classify and possibly diagnose individuals afflicted with Ischemic stroke. Furthermore, the results of the research proved that some features can be utilized to identify the kind of ischemic stroke. These findings will help medical professionals in the classification of ischemic strokes in the immediate future.

The authors in [15] presented a paper that utilized wearable technology based on IoT used for physical therapy, monitoring of these activities and characterization. To capture these movements, they made use of a 3-axis accelerometer and 3 axis gyroscopes. The data collated by these sensors is used to classify movement, enabling continuous monitoring and an initial estimate of the patient's current condition. The system is made up of three major components: a data collation unit, a data processing unit, and a cloud service that enables remote access to data. Each of these sections includes a description and visualization of the implementation of hardware. The function is shown in the context of elbow rehabilitation monitoring. It has been proven that the device can be utilized for accurate and precise monitoring of elbow flexing, enabling for remote rehabilitation via the use of a cloud service, as shown by the results.

The authors in [16] presented a paper that discussed recent additions that aim to aid the rehabilitation process of patients who have been afflicted with upper-limb impairments following a stroke by involving a combination of game-based learning methods.

The authors in [17] presented a paper that utilized smart crutch embedded with sensors, enabled for wireless communications. Utilizing the smart crutch, it is possible to track the development of balance over time and create correlation between these changes and the exercises performed. A mobile application for tabs and phones was created to offer an increased processing, data visualization, and data synchronization with a database stored on the cloud.

The authors in [18] presented a paper that made use of a smart glove to aid physiotherapists in accessing patient’s flexion of fingers performed at home. It is possible to send data from an embedded glove to physiotherapists utilizing a smart phone and an Arduino app, which is connected to an Arduino platform. The data is then sent via a created App. While the flex sensor on the index finger finds and computed movement, it is a Nano that handles the processing and wireless transmission of the data. When it comes to power utilization, the wearable smart glove relies on a Li-ion 3.3V rechargeable 400mAH battery. This Smart Glove also helps in the monitoring of the patient's adverse reaction to certain medicine and/or other motions that are advised. The info collated can be used to properly analyze the patient's condition over time, as well as to help caretakers in adapting to scheduled activities or exercises as required.

The authors in [19] presented a paper that proposed using a wearable armband, machine learning algorithms, and a 3-dimensional printed robot hand were utilized to show a stroke rehab system that was enabled by IoT. Comfort is most critical thing to consider when making wearable technology. The armband was used to connect a low power consumption, small sensing device embedded with electrodes that find, analyze, and wirelessly send potential data, according to the researchers. Very low classification accuracy and poor performance metrics can be reduced by distributing electrodes over the user's forearm. The effectiveness of machine learning algorithms was analyzed using classification accuracy and PCA to analyze and identify features of distinct hand movements.

The authors in [20] presented a paper that utilized a smart glove, embedded with Flex Force sensors, it enables collation and processing of data received from the hand of the individual in a simple and concise manner. Using FPGA chips, we designed a blueprint which includes all possible sensors needed to detect force and rotation of a human limb. For a more precise position and orientation, quaternion-based values are used to combine the sensor data from finger joints and one wrist joint, which is then combined with the resulting information. Besides that, neural networks are used to classify potential hand exercises that can be employed to assist therapy with the aid of exercise sessions by analyzing the results. It may then be possible to assess the efficiency of both traditional and technology-assisted Rehabilitation

The authors in [21] presented a paper that enumerates on the use of a pair of smart gloves to engage with serious games for rehabilitation of the upper limbs. These serious games give individuals afflicted with motor impairments the opportunity to do exercises in an engaging and trivial manner while completely immersed in the various VR setting. Software tools were used to collate and process the data retrieved from the sensors; the system includes a library of immersive VR games that may be used in addition with limb rehabilitation exercises. The data collated allows the physiotherapist to analyze and evaluate the patient's performance throughout the various exercises performed during a session. Piezo-resistive force sensors provide the analog channels, which are linked to an Arduino Nano platform. The data collection is handled by an Arduino Nano platform.

Using the Bluetooth wireless connection protocol, information is exchanged between the smart sensors and the VR platform. These exercises have shown a strong correlation with the quality of life and improvement of individuals afflicted with motor impairments.

The authors in [22] presented a paper that proposed two ways technology can encourage feedback for an affected limb. It can detect movements for the upper limbs while also performing ADL, it can also assess and analyze the quality of in-house rehabilitation exercises.

The authors in [23] presented a paper that introduced a portable rehabilitation monitoring system based on a network of sensors. Locally, Rehabilitation is usually done in hospitals and clinics under supervision of physiotherapists. Remote monitoring, on the other hand, can provide better outcomes, allowing patients to get technology-assisted Rehabilitation in the comfort of their own homes. Mobile technology of nowadays have a very good processing power, a good graphical processing unit, and large storage space. As a result, they're preferred for remote monitoring. Using 2-axis accelerometers, individual sensors are placed on limbs to assess movement. The system tracks specific sensor periods and pressures.

The authors in [24] presented a paper that introduced the use of smart wearables as a solution for upper limb rehabilitation, the paper made use of two gloves and a head. The sensors that have been developed are based on an Arduino Nano that is linked to an IMU module. A set of force and flexion sensors was explored in the instance of smart gloves to give information on finger flexion and force contact. The Bluetooth communication protocol is used to communicate between the wearables and the VR computation platform.

The VR serious games for hand and finger rehabilitation were created as part of a software package and is usually added via a library. During training, an API was created to store and analyze the data collected by the sensors, as well as to extract information about the user's physical abilities and shortcomings. For user electronic records, including training data visualization, a mobile application was created. The paper also includes experimental data collated during the rehabilitation exercises.

The authors in [25] presented a paper that created a wearable technology that aids stroke patients who had just been in a recent accident or inflicted illness. The methods and use of mechanical tools, sensor-embedded Rehabilitation Gloves measuring grasping actions performed by these patients are also discussed in this article. The Gloves will move the gadgets based on the movement of a human fingers and arms, a microcontroller and HyperTerminal are used as the main processing tools. A Rehabilitation Glove is embedded with a set of bend or flex force sensors that collect data from human finger movement. The torque measured by the Rehabilitation Glove, as well as the force controller's performance, have a huge impact of the feedback.

The authors in [26] presented a paper that attempted to use ML algorithms to classify pictures of patients into the two types of stroke: hemorrhage stroke and ischemic stroke, the eight algorithms used in this study are Decision trees, KNN, Logistic regression, Bayes algorithm, MLP, Deep learning methods and SVMs.

The authors in [27] presented a paper that analyzes Deep learning techniques for Parkinson's disease assessment. The paper observes that deep learning has set new boundaries in the realms of voice and picture categorization but has yet to be extensively explored as a method for analyzing data from wearable sensors. The authors used measuring devices to collate data from ten unique individuals with idiopathic Parkinson's disease. Multiple motor tasks were labeled and categorized for detecting bradykinesia. This was done by comparing traditional ML pipelines with convolutional neural network-based deep learning.

The authors in [28] presented a paper that verifies a brilliant and easy-to-use system for training that uses a user specific VR game to give feedback to the patient. The model’s complexity and the quantity of data exchanged for managing the game was reduced while still maintaining efficacy in home rehabilitation systems.

The authors in [29] presented a paper that addresses physical rehabilitation for upper limbs using an Exoskeletal device. The Exoskeletal Device is a small, remotely operated, and effective device used for upper limb rehabilitation usually controlled Attention is paid to reduce the complexity of the system and the amount of data to be exchanged for controlling the game and to be stored for assessing the patient’s progresses, still ensuring the effectiveness of the system to monitor the patient also in home-rehabilitation scenarios. The by signals with smart interfaces for both patients and therapists. Through extremely precise, flexible, and rapid reactions based on muscle strength, brain activity, and motion tracking technology, the SPRED system enables complete range of joint motions and helps handicapped limbs more organically. Patients may carry the device with them wherever they go because of its tiny size and wireless nature, enabling them to prolong the training time naturally and obtain more successful treatment results.

The authors in [30] presented on a paper that showcases Smart Crutches, an instrumented crutch system that can detect the weight that a patient puts on his or her lower extremities and delivering vibratory feedback in response to the measured weight. Smart Crutches have the potential to improve patient outcomes. An automated motion capture system and force plates were used to calibrate the gadget. Calibration was accomplished via the use of linear regression and support vector regression (SVR), and accuracy was estimated using 10-fold cross-validation, which was applied to the system.

2.4 CONCLUSION

In this chapter, various papers on physical rehabilitation using wearable technology and artificial intelligence were reviewed. These papers shed more light on the different approaches by which a person can perform physical rehabilitation using technology and its advantages and disadvantages. The use of wearable technology viewed from the above papers indicate how reliable and effective physical rehabilitation can become.

CHAPTER THREE: SYSTEM DESIGN

3.1 INTRODUCTION

This section delves into the methodology applied to create a total system. Furthermore, the reasons why a specific approach was chosen over others are thoroughly explained. To achieve the intended results, the tools and technique used, as well as the logic and sequential flow design used by the system's control and use case diagrams.

3.2 METHODOLOGY

This chapter delves into the project's functioning model in great depth. The machine learning model is deployed via a Kivy based mobile/desktop app. The user can login into the app, calibrate his IOT wearables, these values would be binary encoded using threshold values based on some research I conducted, these values would then be stored in a test tuple, converted to a NumPy array, reshaped into the correct order, to then be passed into the model and get a classification based on the values from the sensors. The result would then be presented to the front end for the users to see. Several classification models were tested during the implementation. The one with the highest recall and precision was used. SVM was finally chosen with the highest accuracy, precision and recall with several iterations and randomizations.

It is a "collection of techniques, processes, and conventions used by people in a field," according to the Project Management Institute (PMI). Lean methods, Kanban, and Six Sigma are examples of project management approaches that attempt to aid and guide project managers through tasks. As difficulties arise during project execution, different methods offer different solutions. The software development technique utilized in this work is separated into three parts: OOA (Object Oriented Analysis), which focuses on defining what the system should perform in relation to important objects in the issue domain, and overall design. Finally, is the application of OOD (object-oriented design) to system architectural interfaces and classes through OOP (object-oriented programming).

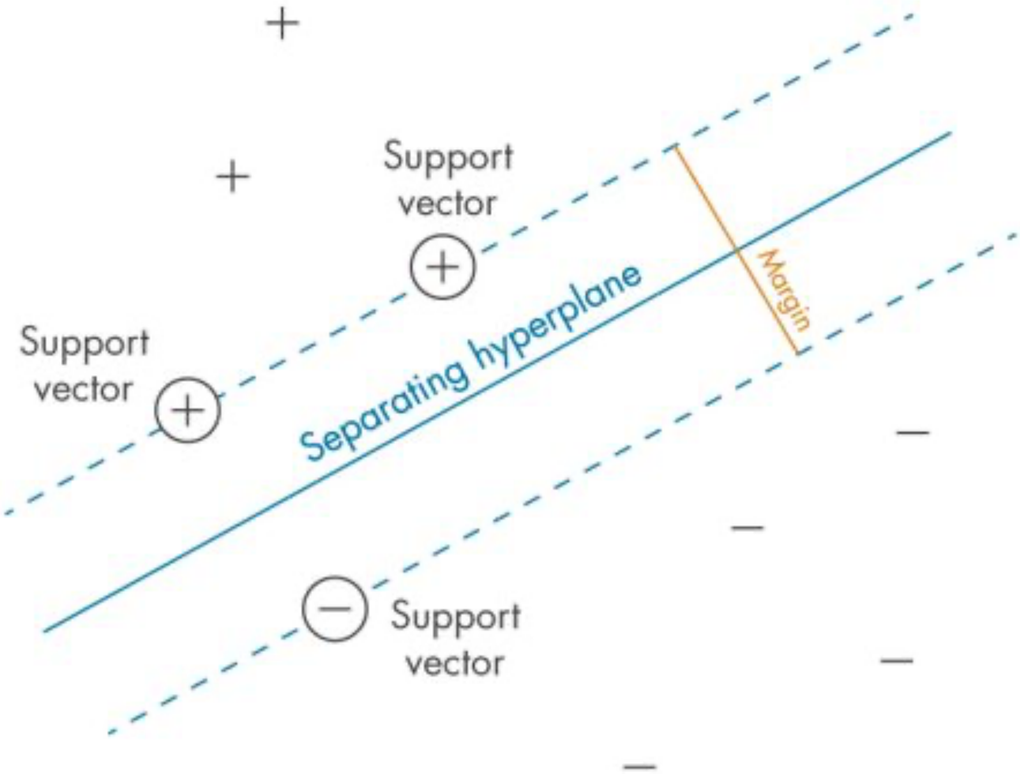


Figure 3.1 SVM hyperplane

Support Vector Machines is an algorithm used to create a hyperplane to distinguish between two or more classes (multiclass classification). It can also be used for regression, but other algorithms perform better, which eradicates its use. It is normally used when you want to reduce training time, and produce fast and accurate models, or on especially low amount of data

3.3 MACHINE LEARNING

This section will be discussing the step-by-step procedures of building the Machine Learning model and deploying for commercial use. The steps are as follows

1. Data collection
2. Data cleaning
3. Data Encoding
4. Training the model
5. Performance Evaluation
6. Deployment

3.3.1 Data Collection

This is the most difficult stage of any Machine Learning related task. Sampling bias must be avoided. Sampling must be random. I conducted a research on a hospital nearby, over the span of 5 days, I asked 150 patients afflicted major motor impairments and asked various questions on their ability to perform certain tasks. Names were removed for non-disclosure purposes. The 5 major impairments are Arthritis, Cerebral Palsy, Parkinson Disease, Left Sided Stroke, Right Sided Stroke. I ensured uniformity to prevent my dataset from being biased towards a particular motor impairment.

3.3.2 Data Cleaning

Data cleaning is another important step, it helps to improve the accuracy of your model, and prevent overfitting caused by irrelevant features in your sample space. I acquired 150 rows of data. I had to remove outliers (values that lie far from other data points). In this case the outliers are people that performed extremely well in the activities for a particular motor impairment. Irregularities had to be removed. In this case people on special medication/ rehabilitation that have already shown improvements and would be useless in the dataset.

3.3.3 Data Encoding

Data is usually encoded to provide a simpler format to pass data into our ML model. Models only understand numerical data, so all features and labels must be encoded appropriately, so that the model can understand. Binary encoding was used for my 10 features and Ordinal encoding was used for my Labels. A binary value of 1 represents the ability to perform the activity, while a binary value of 0 represents the inability to perform that same exercise. My labels were encoded. With the values 0-5 meaning a different motor impairment.

My ten features are listed below:

1. Can move left hand
2. Can move right hand
3. Can apply left hand
4. Can apply right hand
5. Can bend left hand

6. Can bend right hand
7. Can apply left leg
8. Can apply right leg
9. Can move left leg
10. Can move right leg

My labels are listed below:

1. Arthritis
2. Parkinson Disease
3. Cerebral Palsy
4. Right-sided stroke
5. Left-sided stroke
6. Healthy

The encoding threshold is listed below, after conducting various research

The maximum force a finger can exert is 110N, I chose my threshold as 10N, hence if a patient is unable to exert 10N or above, he is deemed unable to apply that finger, averages are taken over all fingers to represent the hand.

The maximum bend a human can flex is 90 degrees, I chose my threshold as 25 degrees, hence if a patient is unable to bend his fingers more than 25 degrees, he is deemed unable to bend his fingers, averages are taken over all fingers to represent the hand

The maximum angular velocity a human can move at is 2rad/s, I chose my threshold as 0.25rad/s, hence if a patient is unable to move at 0.25 rad/s, he is deemed unable to move his arm

The maximum force a human foot can exert is usually his own weight, I chose my threshold as 20N, due to the sensitivity of my FSR not reading forces as high as the weight of a human body, hence if a patient is unable to exert 50N or more, he is deemed unable to apply his foot, averages are taken over the distribution of the foot.

3.3.4 Training the model

My model architecture was built using the Sci-kit Learn framework. It was trained using a Linear kernel. A linear kernel was chosen to prevent overfitting (modelling the noise in your training data, such that your model sees patterns that do not exist, and hence performs badly on unseen data). Overfitting usually occurs on datasets with high number of features and small number of observations. My data set is then split with a ratio 0.8:0.2 for the training data and test data respectively. The Scikit SVC implements the One vs One approach. This produces better results as it considers all classes before making predictions. Although it is the much slower approach.

3.3.5 Performance Evaluation:

The evaluation metrics used on this model were precision, recall, accuracy, and f1 score. Plotting a confusion matrix is relatively easy with Sk-learn. An ideal model would show a high rate of true positives on the graph.

Accuracy can be described as the ratio of the total correctly predicted outcomes to the total outcome/observations. This metric is not completely reliable as it is only best when the dataset is symmetric (the values of false negatives and false positives are almost the same). The mathematical formula is ;

$\text{Accuracy} = TP + TN/TP + FP + FN + TN$

Accuracy is avoided in classification tasks because in imbalanced datasets, our models would still perform very good.

For example: Let us say we have a data set of 1000 people, 950 are terrorists and 50 are non- terrorists, we could simply tune our model to call every single person a terrorist, and our accuracy would be 95%. You can see how it is not really a good measure of performance of our model.

Precision is the ratio of the correctly predicted true positives to the total number of true positives. Simply put a high precision score prevents false accusations, or misclassifications. The mathematical formula is;

$\text{Precision} = TP/TP + FP$

Recall or sensitivity can be described as the ratio of the correctly predicted true positives to all the observations. Simply put a high recall score prevents missing out relevant points The mathematical formula is;

$\text{Recall} = TP/TP + FN$

Usually, Precision and Recall is a tradeoff, the more you predict as True, the higher your chance of misclassifications and the lower your chances of missing relevant points.

Alternatively, the more you predict as False, the lower your chances of misclassifications, but the higher your chances of missing relevant points.

In most cases, depending on the task you want to solve, A high recall could be better, or a high precision could be better

For example: In a terrorist catching model, you'll prefer to have misclassifications than miss the actual terrorists, so in this case, a high recall and low precision is preferred

In my model, a high precision and high recall is preferred, because missing out on classifying motor impairments is just as bad as assigning a wrong motor impairment.

F1 score is the weighted average of recall and precision. This metric considers both the false positives and the false negatives. It is similar to accuracy but is more useful in the sense that it takes into account the cost. In the case of uneven class distribution, this is an important advantage over accuracy. The mathematical formula is;

F1score = 2 *(Recall * Precision)/ (Recall + Precision)

TP = True positives

TN = True Negatives

FP = False positives

FN = False Negatives

These four metrics were used to assess the ML model's performance and verify that it wasn't overfitting to the training data.

3.3.6 Deployment

The model was deployed through a desktop application. Python Kivy was used to create the desktop application. Kivy is a python opensource framework used for building mobile applications. The front-end presents the user with the option to log into the app, he then uses the IOT wearables and data extracted from these sensors are processed and encoded, and then passed into our trained model which outputs a motor impairment. The type of impairment is now updated in the user's database under his username. Based on the type of motor impairment, exercises are provided to him to perform, after each exercise, his data is inserted into the database. The results are extracted to the front end from the database by the push of a button.

The mobile application is built with Python A dynamic programming language preferred by a lot of data scientists and machine learning engineers as well as server-side developers. Python is said to be a good replacement for Java because it combines both functional programming and object-oriented programming. The desktop application running on the laptop can connect to multiple slaves without buffering data.

3.4 SOFTWARE DEVELOPMENT

3.4.1 Python

This is a high-level programming language that may be used for a variety of tasks. Guido Van Rossum, a Dutch computer scientist who worked in the late 1980s, was the one who invented it.[31]. The language consists of three entities.

1. Variable
2. Programs

3. Functions

It has since become popular since its inception, there are a variety of reasons why python is preferred over other languages, listed are the advantages:

1. Its many libraries cater to the unique needs of the user.
2. It can utilize reduced code lengths.
3. When compared to other programming languages, it has a large ecosystem, which includes frameworks and libraries.
4. It is platform-independent
5. It has no problems with memory management.
6. It is open source and user-friendly.

Python Programming Language has multiple features:

1. Easy to use and concise
2. Uses modules, which are blocks of code which can be imported to another program.
3. It utilizes Object Oriented Programming and Functional Programming.
4. It has built in data types
5. It can be used in relation with multiple other programming languages.

3.4.2 KIVY

Kivy is a python framework used to produce Graphical user Interfaces, it is deployed in mobile and desktop apps running Python 3 and above. It is a free and open-source framework. It is compatible with IOS, android, Linux, macOS and Windows. It was developed by the Kivy organization, it was initially designed to be used on all platforms in 2012.it comes with a KV file like how the xml file works in android development environment.

3.4.3 SQLite Database

SQLite is a relational database management system (RDBMS) programmed using the C library. Unlike many other database management systems, SQLite is not a client-server database engine (a database that resides on the server, and clients write to access that server). It uses weakly typed SQL Syntax. It's a popular choice because of its readability, usability and accessibility

The features of SQLite are:

1. Supports Cross Platform
2. Requires no external dependencies
3. Simple
4. Reading and writing is fast
5. Little code footprint
6. Transactions are atomic and consistent
7. No configuration needed- Plug and Play

The reason I chose SQLite for my project is its simplicity, efficiency and integration with the Internet of Things (IOT).

3.4.4 PySerial Communication

The PySerial library is a module used by python for serial communication with external hardware devices, it accesses your device’s Bluetooth resources to initiate connections with other devices. It makes the host system act as the Master device, the device it connects to acts as the Slave, it enables opening and closing of the Ports at any time using the serial.open() and serial.close() commands respectively.

Features of PySerial:

1. Capable of wired communication
2. Capable of Bluetooth communication
3. Capable of Wireless communication

3.4.5 Database Design

A database is a catalog or a well-structured file system that can be accessed in several ways. All the information from the wearables is being stored, read and updated from the tables in the database. The database that is being used in this project is SQLite which has been installed using the command line interface. It is being accessed by the local host. The tables contained in the database which are in use for the evaluation module are listed and explained below:

This table stores all the available users of the system along with their predicted impairments

Table 3.1 Users table

Field	Datatype	Function
Username	TEXT	This field serves as the unique identifier for each record, it serves as the primary key
Password	TEXT	This field stores the passwords of each user
Impairment	TEXT	This field stores the impairment predicted by our ML model for each user

Table 3.2 Left hand table

Field	Datatype	Function
Username	TEXT	This field stores the username of the user.
Force	REAL	This field stores the force gotten from the wearables during exercise
Flex	REAL	This field stores the flex angles gotten from the wearables during exercise

Acceleration	REAL	This field stores the acceleration gotten from the wearables during exercise
Date	TEXT	This field Stores the date the exercise is performed

Table 3.3 Right hand table

Field	Datatype	Function
Username	TEXT	This field stores the username of the user.
Force	REAL	This field stores the force gotten from the wearables during exercise
Flex	REAL	This field stores the flex angles gotten from the wearables during exercise
Acceleration	REAL	This field stores the acceleration gotten from the wearables during exercise
Date	TEXT	This field Stores the date the exercise is performed

Table 3.4: Right leg table

Field	Datatype	Function
Username	TEXT	This field stores the username of the user.
Force	REAL	This field stores the force gotten from the wearables during exercise
Acceleration	REAL	This field stores the acceleration gotten from the wearables during exercise
Date	TEXT	This field Stores the date the exercise is performed

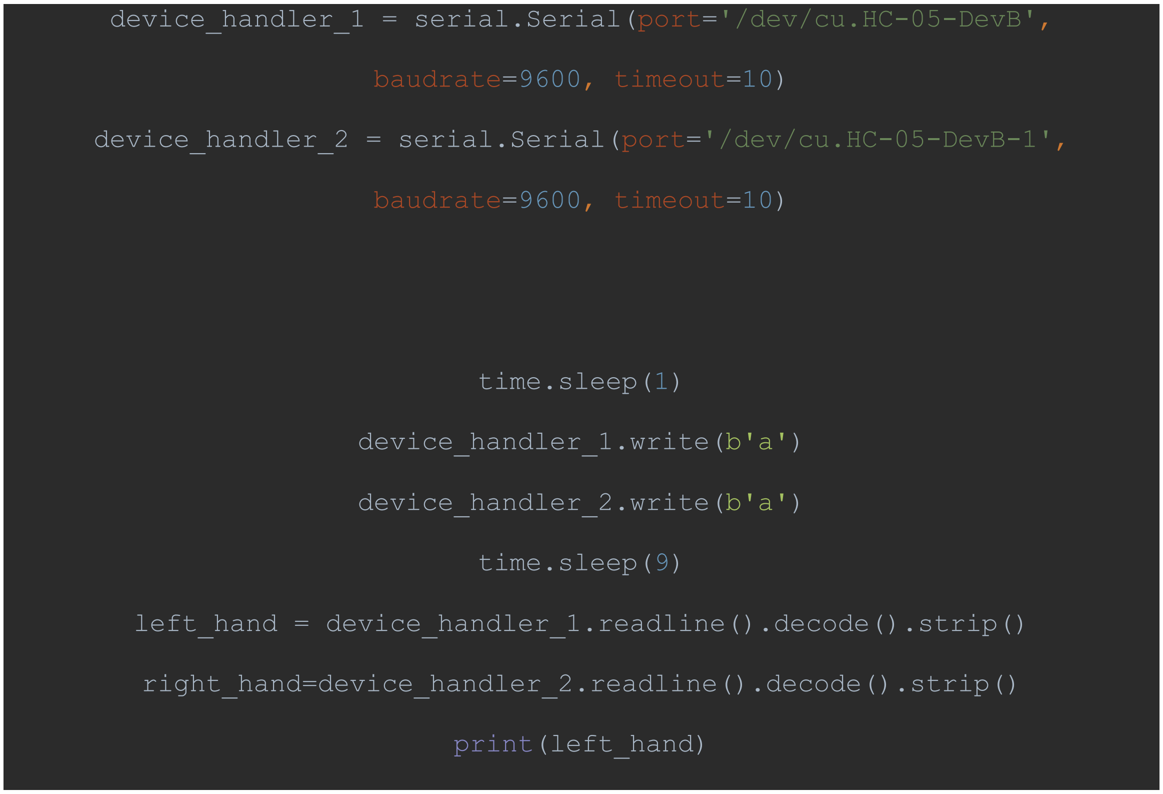
Table 3.5: Left leg table

Field	Datatype	Function
Username	TEXT	This field stores the username of the user.
Force	REAL	This field stores the force gotten from the wearables during exercise

Acceleration	REAL	This field stores the acceleration gotten from the wearables during exercise
Date	TEXT	This field Stores the date the exercise is performed

3.4.6 Bluetooth Communication Design

The Bluetooth communication is designed using PySerial module that allows opening, closing, writing to and reading from Bluetooth ports to enable a fast and reliable serial communication between master and slave devices



```
device_handler_1 = serial.Serial(port='/dev/cu.HC-05-DevB',
baudrate=9600, timeout=10)
device_handler_2 = serial.Serial(port='/dev/cu.HC-05-DevB-1',
baudrate=9600, timeout=10)



time.sleep(1)
device_handler_1.write(b'a')
device_handler_2.write(b'a')
time.sleep(9)
left_hand = device_handler_1.readline().decode().strip()
right_hand=device_handler_2.readline().decode().strip()
print(left_hand)
```

Figure 3.2: Code for connecting Bluetooth device

3.4.7 Unified Modeling Language

Software architecture has been developed to break complicated, huge software systems into smaller components in order to reduce software design complexity. [32]. According to Kruchten [33], software architecture can be viewed from various angles:

1. Logical
2. Process
3. Development
4. Physical

A software system is divided into components and their connections from a logical standpoint. The interaction of the system's components is the primary emphasis from a process standpoint. The emphasis of development is on implementation problems. The hardware and how it is linked are the primary emphasis from a physical standpoint. Different modelling languages, such as Network Description Language (NDL) and Unified Modelling Language (UML), may be used to describe software from various perspectives (UML). The UML, on the other hand, is the most widely used software language.

The Unified Modeling Language (UML) is a 3rd generation modeling language. It was first published in November 1997.[34]. The UML diagrams help with representing complex system software in visual forms. They are often divided into two groups. They are static and dynamic. Static diagrams usually take into consideration the structure of the system. Examples are deployment diagram, object diagram, class diagram and profile diagram. A dynamic diagram takes into consideration how the system changes with the environment. Examples are timing diagrams, use case diagrams and activity diagrams.[34]

The four types of UML diagrams are as follows:

1. Structural Diagrams: these diagrams focus mainly on the purpose, they include structure diagrams, object diagrams, class diagrams, component diagrams, package diagrams, deployment diagrams.
2. Behavioral diagrams: these diagrams focus mainly on the behavior of the system; they include activity diagrams and state machines.
3. Interaction diagrams: These diagrams focus mainly on the relationship between the elements of the system. They include collaboration, sequence, and timing diagrams
4. Functional diagrams: These diagrams focus mainly on the functionality, not the structure or behavior, they include case diagram and information flow graphs.

Complex systems can generally be modelled with three diagrams These are Class diagram, State diagram and Sequence diagram [34]. This project makes use of Class diagrams, Activity diagrams and Sequence diagrams.

Advantages of using UML include:

1. It is a well explained language
2. It is easy for people without technical knowledge to understand
3. It has an extensive support
4. It is very easy to learn

3.4.8 Software Development Life Cycle Model

The proposed Software Development Life Cycle Model makes use of the waterfall model. The waterfall model is chosen because the system requirements are fully known, and user requirements are hardly needed. The system also does not change much with a change in environment. Hence a waterfall model would perform well.

3.4.9 User or Desktop Interface Design

A desktop application is a program that is usually integrated in computers or laptop systems, they are optimized to run on these systems, and may produce unfavorable and different results for other

For my portion of this project, I used the PyCharm Editor with the Kivy Framework to produce my system. The programming language that was used is Python, for the designing of the user interface and for the back-end programming. The diagram below shows how the individual pages are connected.

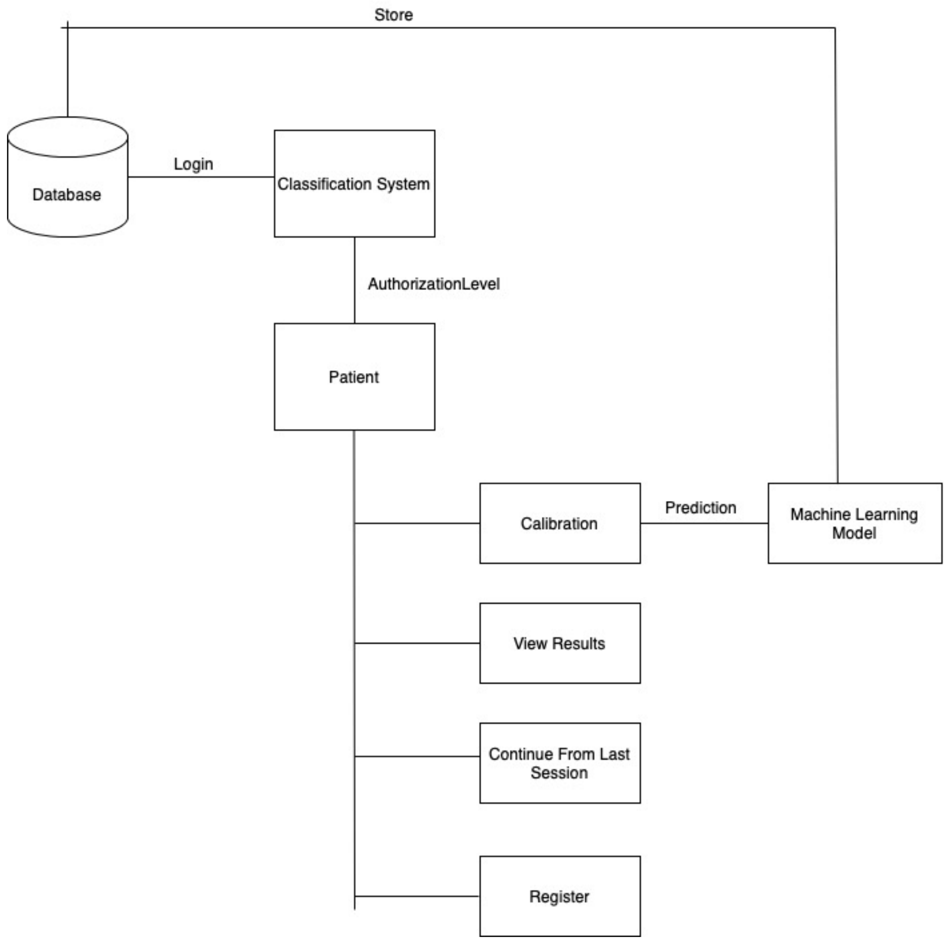


Figure 3.3: Interface design

As shown in the diagram above, the Classification System is accessed after the user has logged in. Based on the authorization level; the patient has 4 options, he can initiate calibration to get his most likely motor impairment, he can view his results after his exercise, he can continue from his last session, and he can register as a new user to enable another motor impairment classification.

The Calibration page uses machine learning to predict motor impairments, the value is then stored in the users table with the username as the primary key.

3.4.10 Use Case Diagram

A use case diagram produced using the Unified Modeling Language (UML) is used to illustrate the purpose of a system and how it works from the point of view of the users, in other words, it depicts the scenarios and how they are linked to accomplish the objective.[36]

The use case diagram shows the system or application and the basic flow of events that happen in the system. The lines show the interaction between the actors and the system; the actors are the users; they initiate the use case [35], which is always represented with an oval shape.

In the use case diagram, there is one actor:

1. The primary actor initiates the use of the system, and they are always positioned at the left-hand side of the system. The primary actors in this system are those who make use of the application in this case, the patients.

The actor interacts with the use case as indicated in figure 3.9. From the use case diagram in figure below, the primary actor can log in or register, view the results, calibrate his wearables or continue from his last session

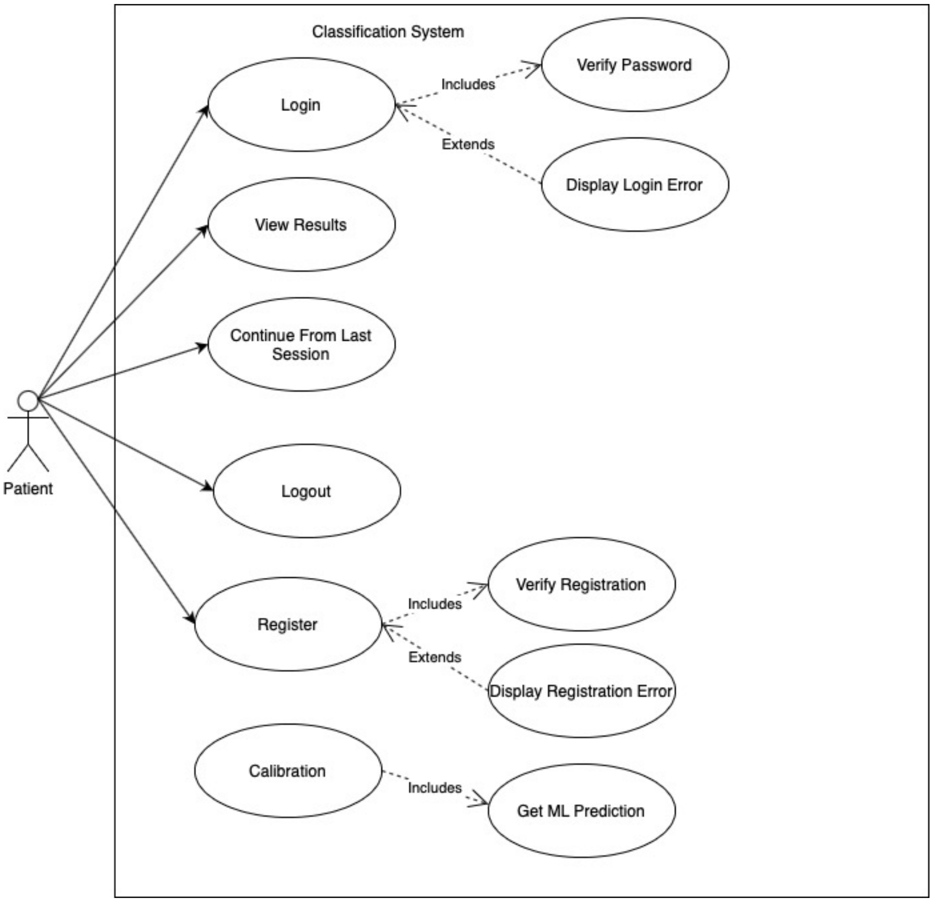


Figure 3.4 Use case diagram

3.4.11 Flowchart

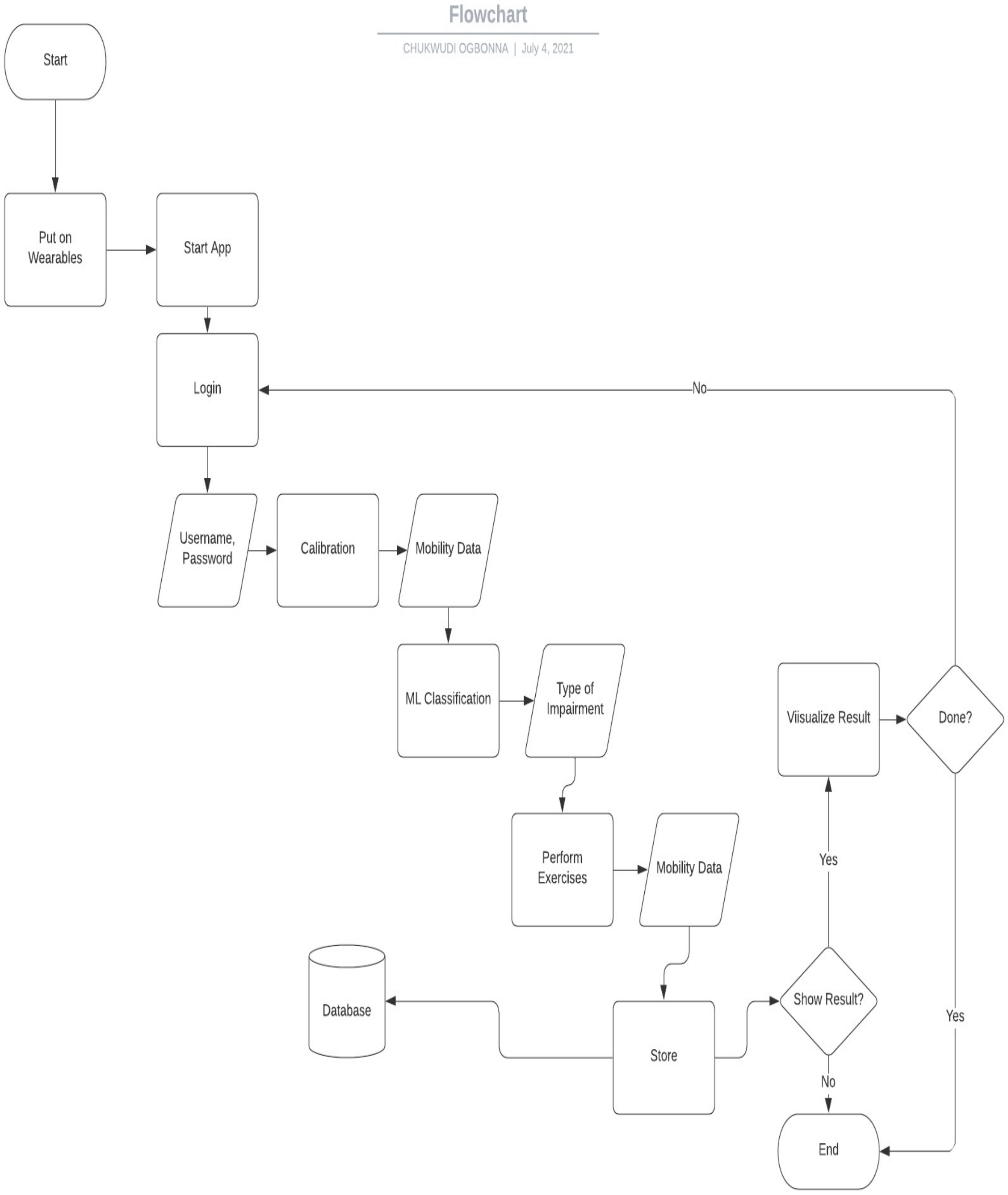


Figure 3.5: Flowchart for classification system

3.4.12 Activity Diagram

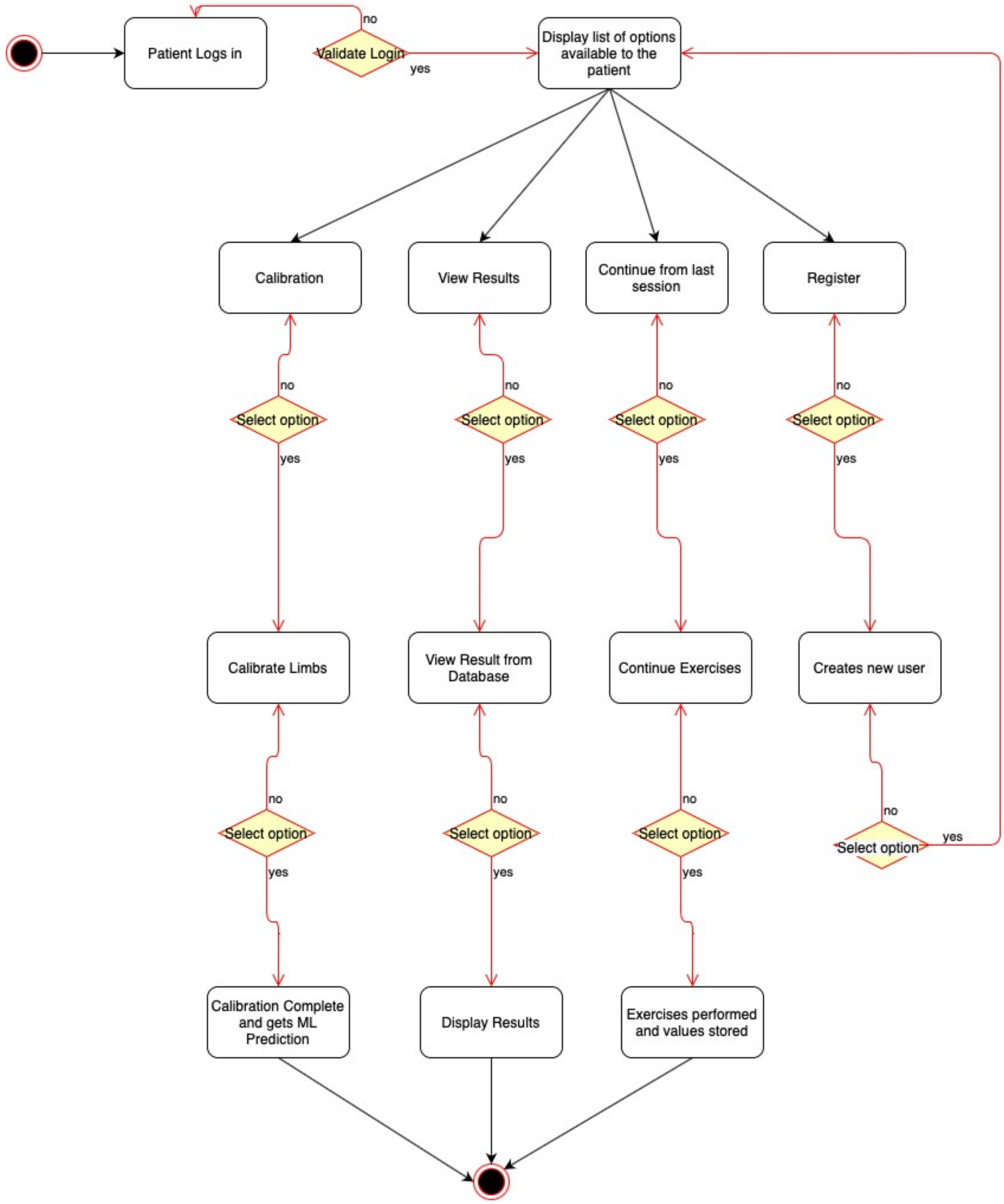


Figure 3.6 Activity diagram

From the diagram above, the user logs in and is presented with 4 options, depending on his options he is taken to various other pages ,if he chooses the calibration, he is then given the option to totally calibrate his limbs, if he then chooses that option, he gets a prediction for the most likely motor impairment he has, if he chooses the view results option, a visual results from the database is shown, and he then has the option to terminate the program with the escape key, if continue from last session is chosen, he is asked to continue his previous exercises that corresponds to the type of motor impairment assigned to him, he then performs exercises and the values are stored in the database, if the new user option was chosen, he is taken to a registration page and his data is stored in the database as a new user

3.7 CONCLUSION

The design concept and specifications of the system have been stated out in this chapter and also depicted with the use of block diagrams, flowcharts and UML diagrams All the components and software to be used were also talked about in detail.

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this section, the details of the implementation of the physical rehabilitation system would be discussed., the performance metrics and evaluation of each algorithm is also discussed along with the reason a specific algorithm was chosen. The results obtained on how the entire system is functioning is discussed in this chapter

4.2 MATERIALS AND SYSTEM REQUIREMENTS

This section of this report contains the system requirements for building the beta version of the application. The hardware requirements for building the application includes a high-end CPU capable of processing a large amount of numerical data at a fast rate to train the model. The software requirements include:

1. PyCharm for writing Python code to develop the desktop application
2. Anaconda Navigator and Spyder for writing ML code to display performance metrics
3. Excel to store our training and test dataset used to create our model
4. Python, Python Kivy, NumPy, pandas and Scikit learn.

4.3 PYTHON MODULES

A module is a single or multiple-routine software component or portion of a program. The concept originates from modular programming which proposes that software should be divided into several independent and interchangeable modules. Python is known to have over 10,000 modules. Some of python modules used in this work have been listed above, these and more modules will be discussed in detail;

Table 4.1: Python Modules

s/n	**Packages**	**Functions**
1	Pandas	Pandas is an acronym for “Python Data Analysis Library”. It's a Python data analysis tool that's open-source and BSD-licensed, with simple data structures for Python. .
2	Matplotlib	Matplotlib is a Python library for creating static, animated, and interactive visualizations .
3	Numpy	NumPy is a Python library that adds support for huge, multi-dimensional arrays and matrices, as well as a vast set of high-level mathematical functions for manipulating them.
4	Scikit Learn	It is a Python-based machine learning package (also known as sklearn) that is accessible for free. It is meant to interact

		with the Python numerical and scientific libraries NumPy and SciPy, and features support vector machines, random forests, gradient boosting, k-means, and DBSCAN, among other classification, regression, and clustering techniques.
5	Kivy	Kivy is a python open source framework used for building mobile applications
6	Concurrent	The Concurrent module is used for asynchronous callbacks, usually called for system intense processes that use a lot of system resources
7	sqlite3	The sqlite3 module enables communication between my python app and my SQLite databases.
8	serial	The serial module enables serial communication between

		my system and the IOT wearables
9	time	The time module is used to access the current time and date.
10	Error	The Error module allows me to handle exceptions and errors that can potentially crash the application.

4.4 IMPLEMENTATION AND RESULTS

This step of the Software Engineering life cycle, known as the Implementation stage, is very important. In software engineering, system implementation refers to the manifestation of an algorithm or design as a full Software component, application, or other piece of software. The implementation of the system design described earlier in chapter three occurs at this step. System testing may be described as a comprehensive examination of a completed software product that verifies the functionality of a fully integrated system. This portion of the report is split into two sections: the deployment of the ML model, testing of the model and our Python System software. The deployment of the ML model is the first section of this report.

The purpose of this work is to make accurate diagnosis and prediction of motor impairments using data gotten from our IOT wearables commercially available to all kinds of users, not just physiotherapists. Research has shown that making such technologies accessible on desktops is the most effective method to make this happen. A significant proportion of the world's population makes use of these gadgets. It is written in Python and delivered to the Desktop platform using PyCharm, which is a Python-based development environment. The ML model is also made accessible to users on other platforms via the desktop application, which is written in Kivy. The methods of testing include Evaluation metrics;

i. Accuracy

ii. Precision

iii. Recall

iv. F1 score

These metrics have been explained in detail in the previous chapter. A confusion matrix showing all these scores was derived using a built-in module imported from Scikit-Learn's library. In the ML field, a confusion matrix is a table structure that enables the visualization of performance of an algorithm, commonly a supervised learning algorithm.

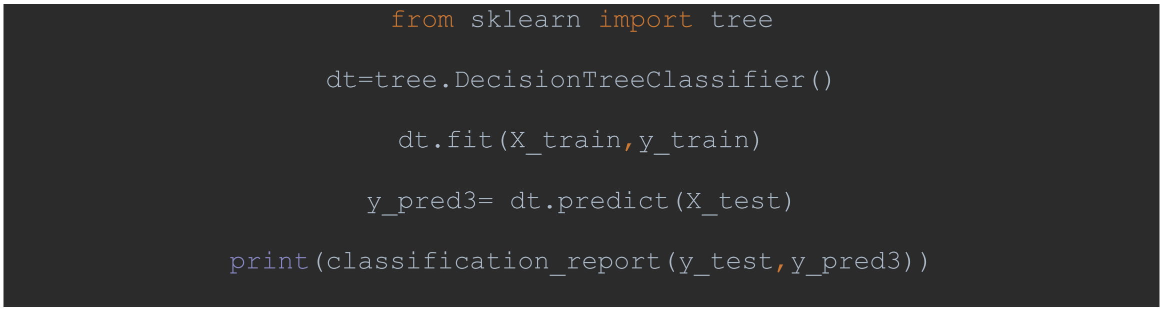
4.4.1 Machine Learning Implementation and Results

For this sub chapter, I will explain why the choice for my ML algorithm is SVM, below the classification reports for the various classification models would be displayed, the performance metrics would also be shown, I will explain the usefulness of each metric.

Decision Tree

This is a non-parametric supervised machine learning algorithm that can be used for both classification and regression. It works by creating a model that predicts the value of a target variable by learning decision rules based on a subset of our dataset's features. They are simple to use and interpret. They can also produce complex trees with a lot of depth that lead to overfitting,

A very deep tree would have a small number of samples, and hence our predictions would be too reliant on the exact subset of training data used.



```
from sklearn import tree

dt=tree.DecisionTreeClassifier()

dt.fit(X_train,y_train)

y_pred3= dt.predict(X_test)

print(classification_report(y_test,y_pred3))
```

Figure 4.1: Decision Tree Code

```
                      precision    recall   f1-score   support


Arthritis              1.00       0.78      0.88         9

Cerebral Palsy        1.00       1.00      1.00         4

Left Side Stroke       0.67       0.67      0.67         3

None                   1.00       1.00      1.00         2

Parkinson Disease         0.67       1.00      0.80         2

Right side stroke         0.80       1.00      0.89         4


accuracy                                      0.88         24

macro avg                 0.86       0.91      0.87         24
```

```
weighted avg              0.90      0.88      0.88         24
```

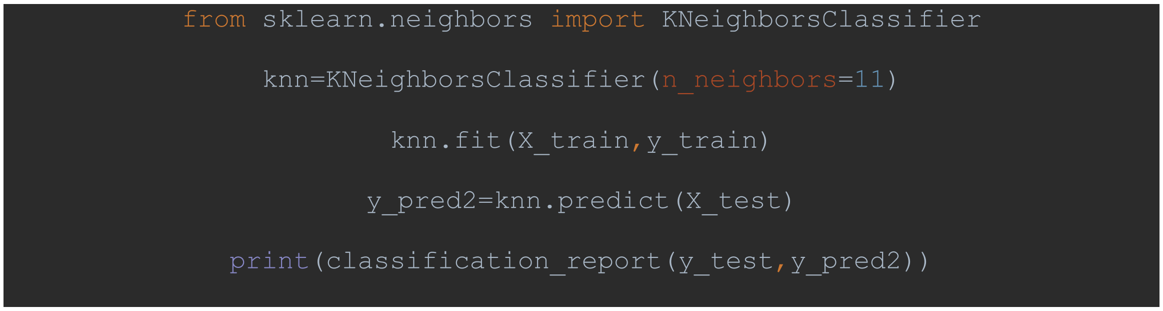
The values above show an average precision value of 90% and an average recall value of 88%, the f1 score likewise is 88%

K-Nearest Neighbors

K nearest neighbors is a non parametric, lazy learning approach supervised learning machine learning algorithm that can be used to solve both regression and classification tasks, it assumes that similar items exist in close proximity. It works by calculating the distances between all data points closest to it and picking the numbers dependent on the number of neighbors we have selected.

A small number of neighbors is prone to overfitting because predictions are made on a small subset of data, so modelling the noise of our training data is likely.

My choice of k is sqrt(n) where n =120, so k is approximately 11



```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=11)
knn.fit(X_train,y_train)
y_pred2=knn.predict(X_test)
print(classification_report(y_test,y_pred2))
```

Figure 4.2: k-NN code

```
                  precision     recall  f1-score   support


Arthritis        `      1.00      0.67      0.80         9

Cerebral Palsy          0.80      1.00      0.89         4

Left Side Stroke        0.75      1.00      0.86         3
```

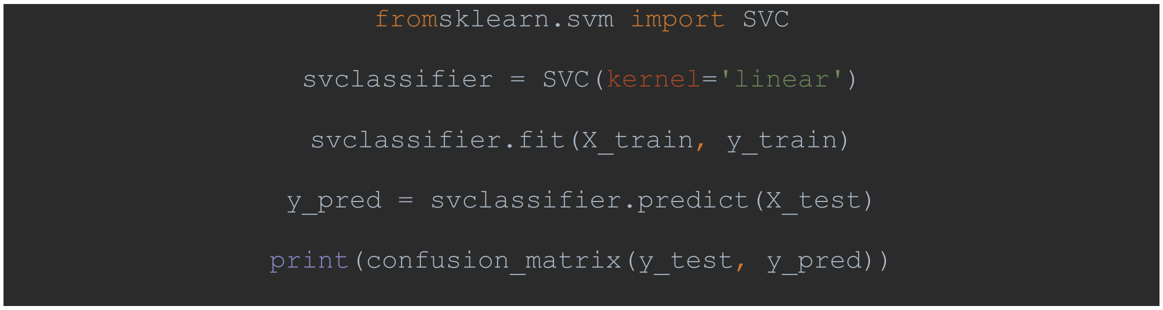
```
None                      1.00       1.00       1.00          2

Parkinson Disease           0.67       1.00       0.80          2

Right side stroke           1.00       1.00       1.00          4


accuracy                                       0.88          24

macro avg               0.87       0.94       0.89          24

weighted avg              0.91       0.88       0.87          24
```

The values above show an average precision value of 91% and an average recall value of 88%, the f1 score likewise is 87%

Support Vector Machine

Support Vector machine is a parametric, eager learning approach supervised machine learning model that can be used for both regression and classification tasks. It works by creating a hyper plane between training data plotted in the n dimensional space where n is the number of features of our dataset, the hyper plane is to make a good enough margin of error to prevent overfitting our model.

A linear kernel is used (simple model) to prevent overfitting our model



```
fromsklearn.svm import SVC

svclassifier = SVC(kernel='linear')

svclassifier.fit(X_train, y_train)

y_pred = svclassifier.predict(X_test)

print(confusion_matrix(y_test, y_pred))
```

Figure 4.3: SVM code

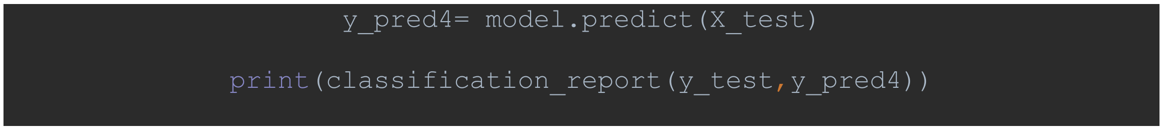
```
                      precision    recall  f1-score   support


Arthritis                1.00       0.78      0.88          9

Cerebral Palsy           1.00       1.00      1.00          4

Left Side Stroke         0.75       1.00      0.86          3

None                      1.00       1.00      1.00          2

Parkinson Disease         1.00       1.00      1.00          2

Right side stroke         0.80       1.00      0.89          4


accuracy                                       0.92        24

macro avg                0.92       0.96      0.94        24

weighted avg             0.94       0.92      0.92        24
```

The values above show an average precision value of 94% and an average recall value of 92%, the f1 score likewise is 92%

Naïve-Bayes Model

The Naïve-Bayed model is a parametric eager learning approach supervised machine learning algorithm that can also be used for regression and classification tasks. It works on the foundation of the Bayes Theorem of conditional probability. Conditional probability is the probability of something happening, given another event has happened

```
from sklearn.naive_bayes import GaussianNB

model = GaussianNB()

model.fit(X_train, y_train)
```



```
y_pred4= model.predict(X_test)

print(classification_report(y_test,y_pred4))
```

Figure 4.4: Naïve-Bayes code

```
                     precision    recall  f1-score   support


Arthritis              1.00       0.78      0.88         9

Cerebral Palsy         1.00       0.75      0.86         4

Left Side Stroke       0.75       1.00      0.86         3

None                   1.00       1.00      1.00         2

Parkinson Disease      0.50       1.00      0.67         2

Right side stroke      0.75       0.75      0.75         4


accuracy                                    0.83        24

macro avg              0.83       0.88      0.83        24

weighted avg           0.89       0.83      0.84        24
```

From the above models, it can be easy to see why the SVM was chosen, it displayed the highest recall and precision values, along with the highest accuracy available.

4.4.2 Confusion Matrix

A confusion matrix is a particular kind of matrix that displays the relationship between true positives, true negatives, false positives, and false negatives in a given set of circumstances. On the test data set, the table displays the performance of a classification model that was developed (in this case images). It is very beneficial to be able to comprehend how the classifier operates on the dataset. The confusion matrix for the SVM model is shown in the illustration below.

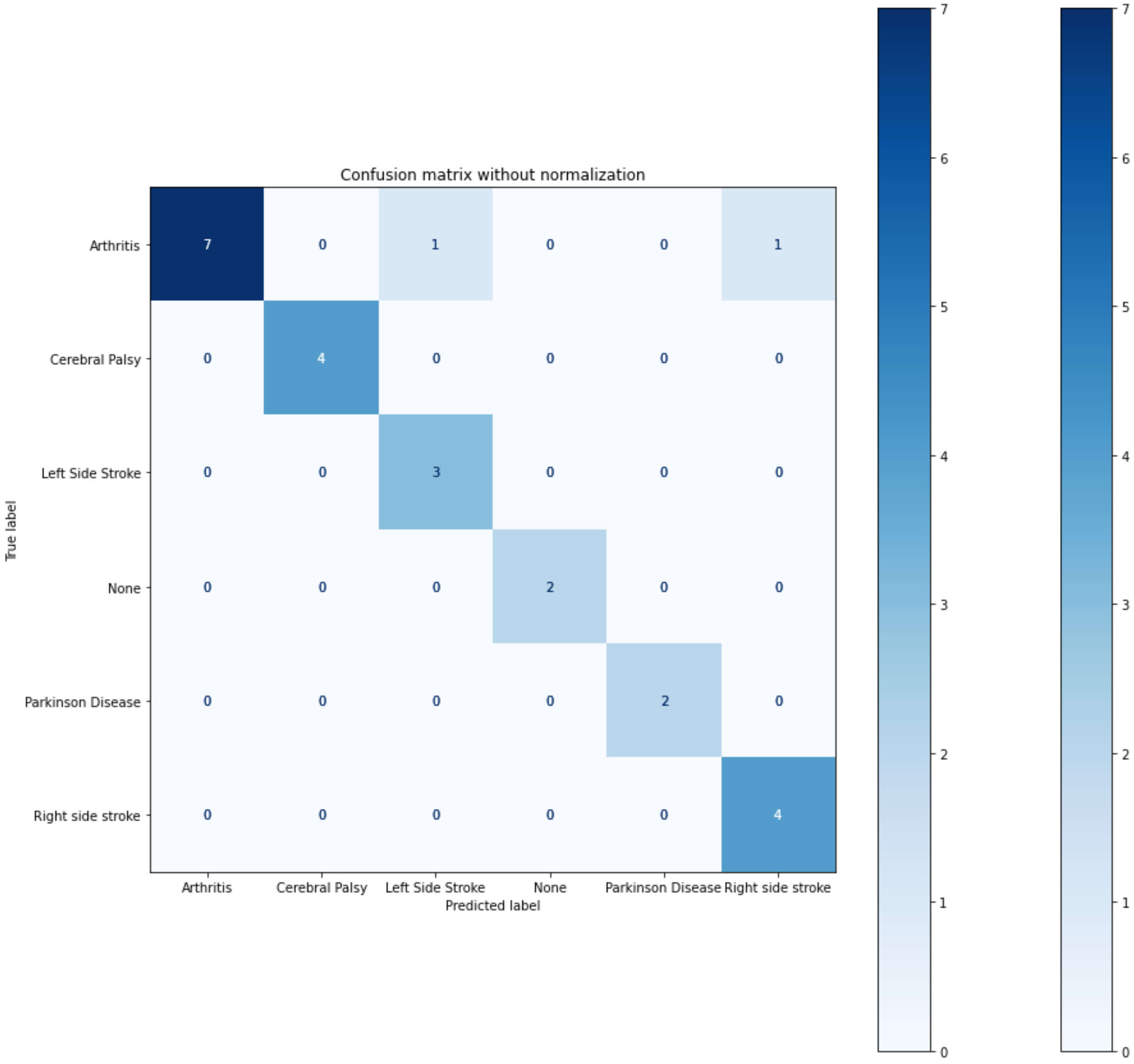


Figure 4.5: Confusion Matrix without Normalization

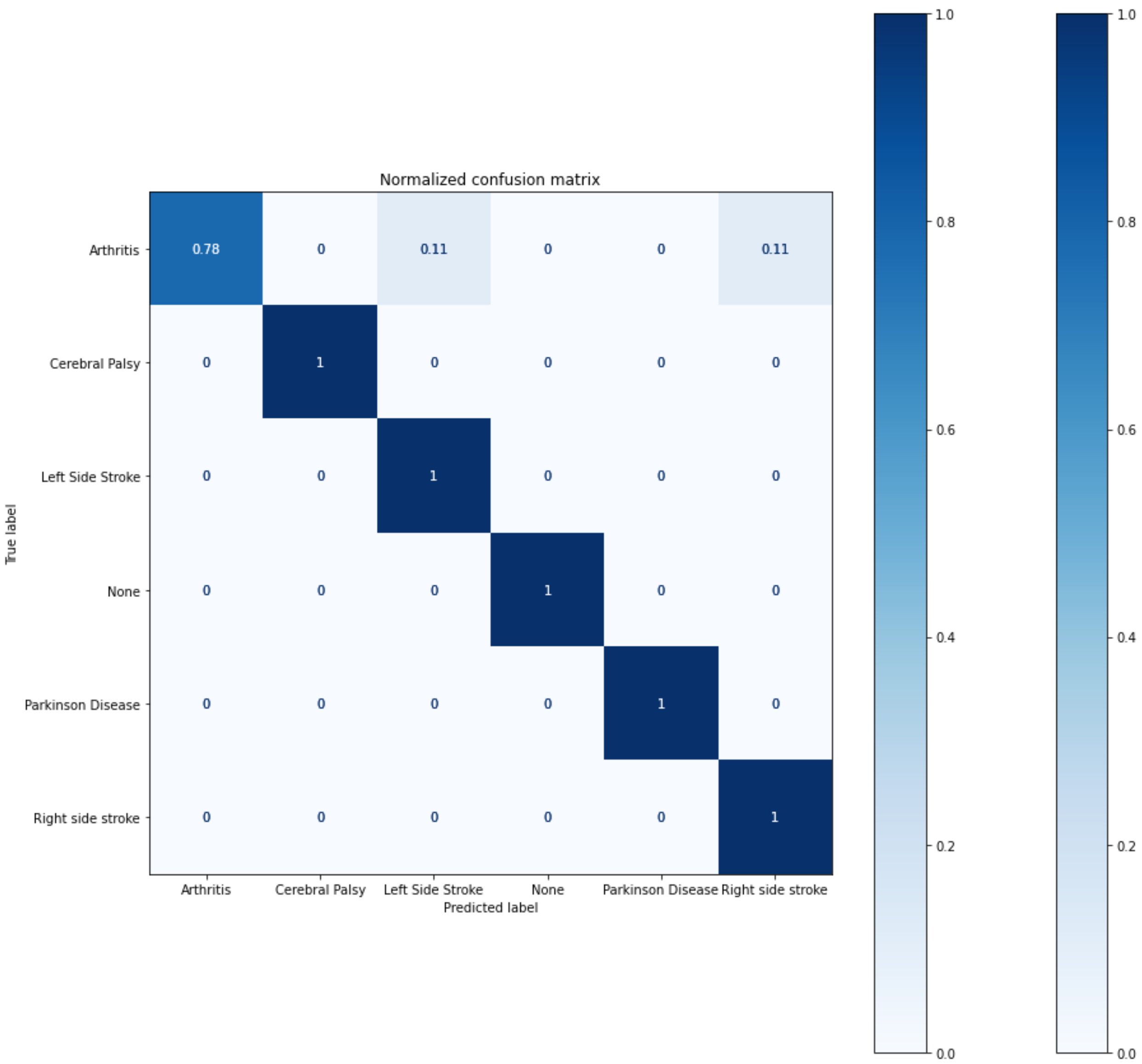


Figure 4.6: Confusion Matrix with Normalization

The interpretation of the information displayed by the confusion matrix based on the performance of the model on the test dataset is as follows;

For the Arthritis class, the model identified 7 of the arthritis class as arthritis class, 1 belonging to the arthritis class as left sided stroke, and 1 belonging to the arthritis class as right side stroke.

For the Cerebral palsy class, the model identified all 4 of the cerebral palsy class as cerebral palsy class.

For the left side stroke class, the model identified all 3 of the left side stroke class as left side stroke class.

For the None class, the model identified one of the None class as None class.

For the Parkinson disease class, the model identified all 2 of the Parkinson disease class as Parkinson disease class.

For the right-side stroke class, the model identified all 4 of the right-side stroke class s right side stroke class.

The uneven distribution of test data is caused by the train test split method that randomizes the way our dataset is split.

The normalized graph just shows the ratio with 1 being all classes identified correctly and 0 being all classes identified wrongly.

4.4.3 Python software

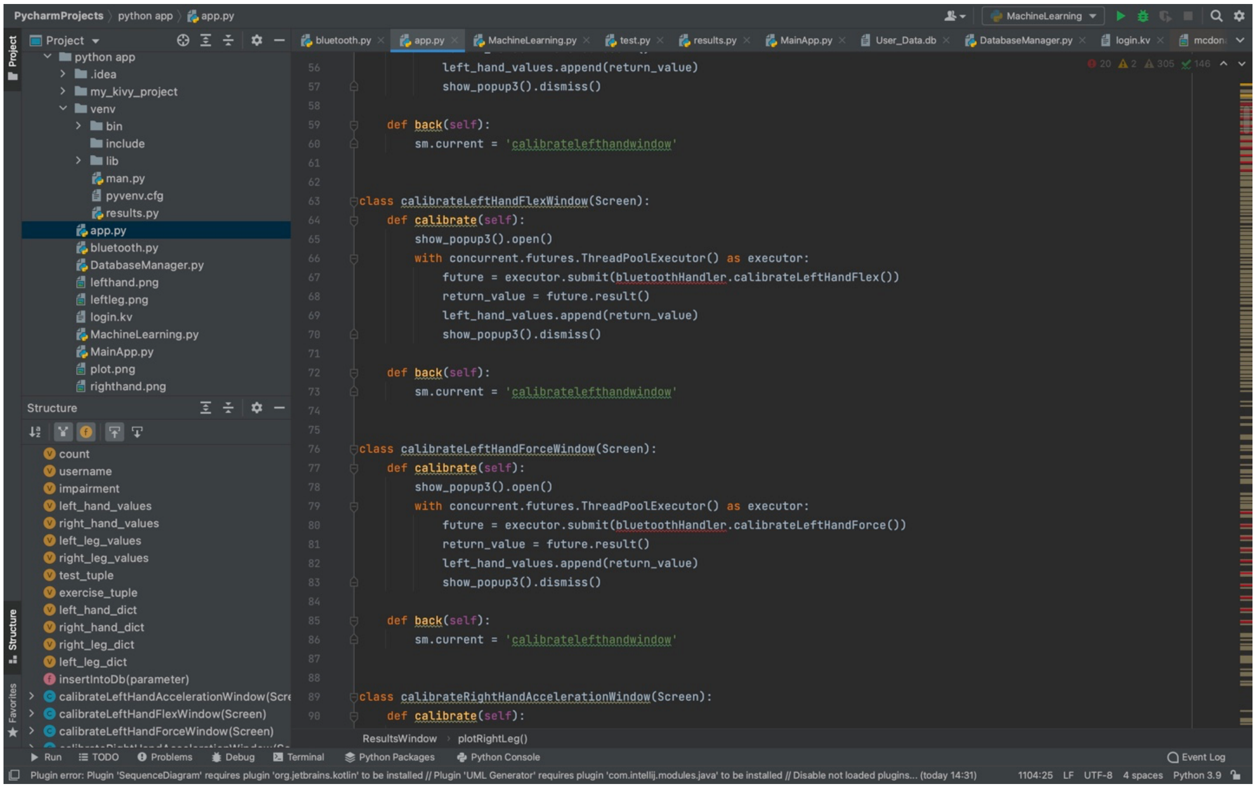
Python with the Kivy framework was used for the creation of the desktop application. The following were installed on the system:

1. Python

ii. PyCharm text editor

iii. Kivy

Kivy packages were installed via the command' *pip install Kivy* on the command line. Then the *kivy_admin* was used to find the sub-commands available. The new project was created by selecting the start project sub-command, i.e. *kivy_admin* start project.



```
            left_hand_values.append(return_value)
            show_popup3().dismiss()

    def back(self):
        sm.current = 'calibratelefthandwindow'


class calibrateLeftHandFlexWindow(Screen):
    def calibrate(self):
        show_popup3().open()
        with concurrent.futures.ThreadPoolExecutor() as executor:
            future = executor.submit(bluetoothHandler.calibrateLeftHandFlex())
            return_value = future.result()
            left_hand_values.append(return_value)
            show_popup3().dismiss()

    def back(self):
        sm.current = 'calibratelefthandwindow'


class calibrateLeftHandForceWindow(Screen):
    def calibrate(self):
        show_popup3().open()
        with concurrent.futures.ThreadPoolExecutor() as executor:
            future = executor.submit(bluetoothHandler.calibrateLeftHandForce())
            return_value = future.result()
            left_hand_values.append(return_value)
            show_popup3().dismiss()

    def back(self):
        sm.current = 'calibratelefthandwindow'


class calibrateRightHandAccelerationWindow(Screen):
    def calibrate(self):
```

Plugin error: Plugin 'SequenceDiagram' requires plugin 'org.jetbrains.kotlin' to be installed // Plugin 'UML Generator' requires plugin 'com.intellij.modules.java' to be installed // Disable not loaded plugins... (today 14:31)

Figure 4.7: Code snippet of Kivy

4.4.4 Patient Section of the Classification System Application

The first page of this section of the web portal is the login page in Figure 4.4, followed by the sign-up page for a new user in Figure 4.5. After the user logs in, the first view page is shown in Figure 4.6. The user can navigate and select the required action from the dashboard. The figure shows the contents of the dashboard. Figures 4.7, 4.8, 4.9 and 4.10 show the corresponding agricultural parameter pages (light, temperature, humidity, and soil moisture). Figure 4.11 shows the user profile page.

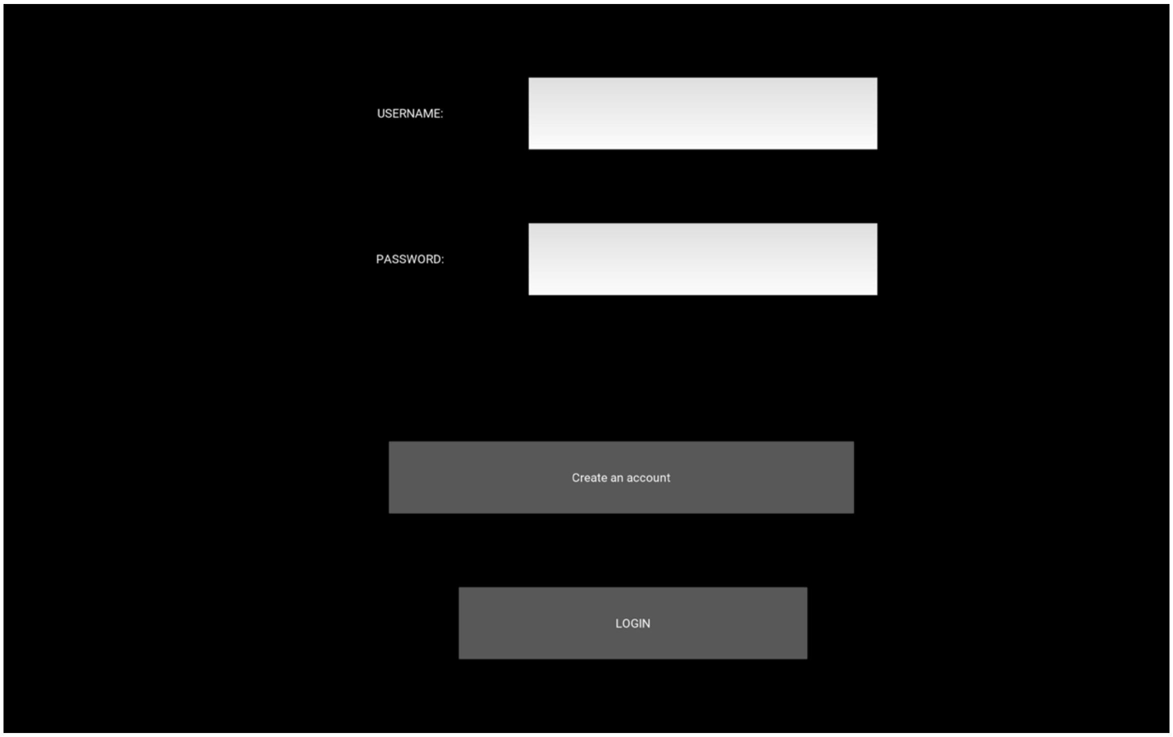


Figure 4.8: Login Page of our system

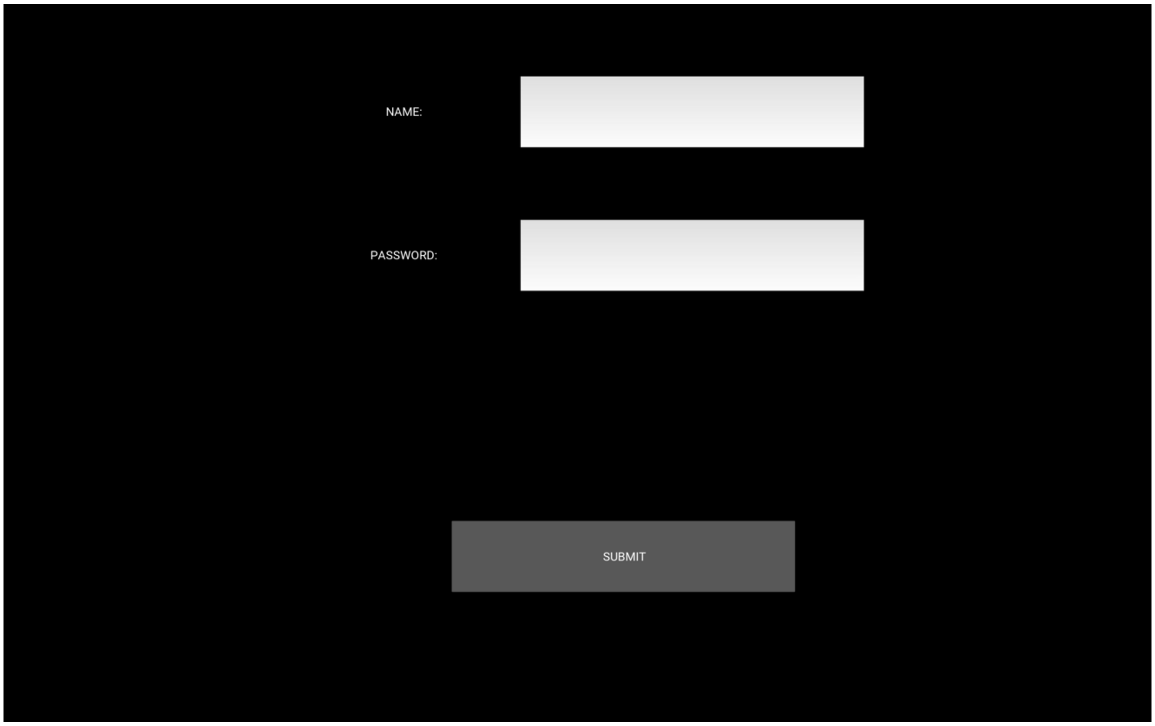


Figure 4.9: Registration Page of our system

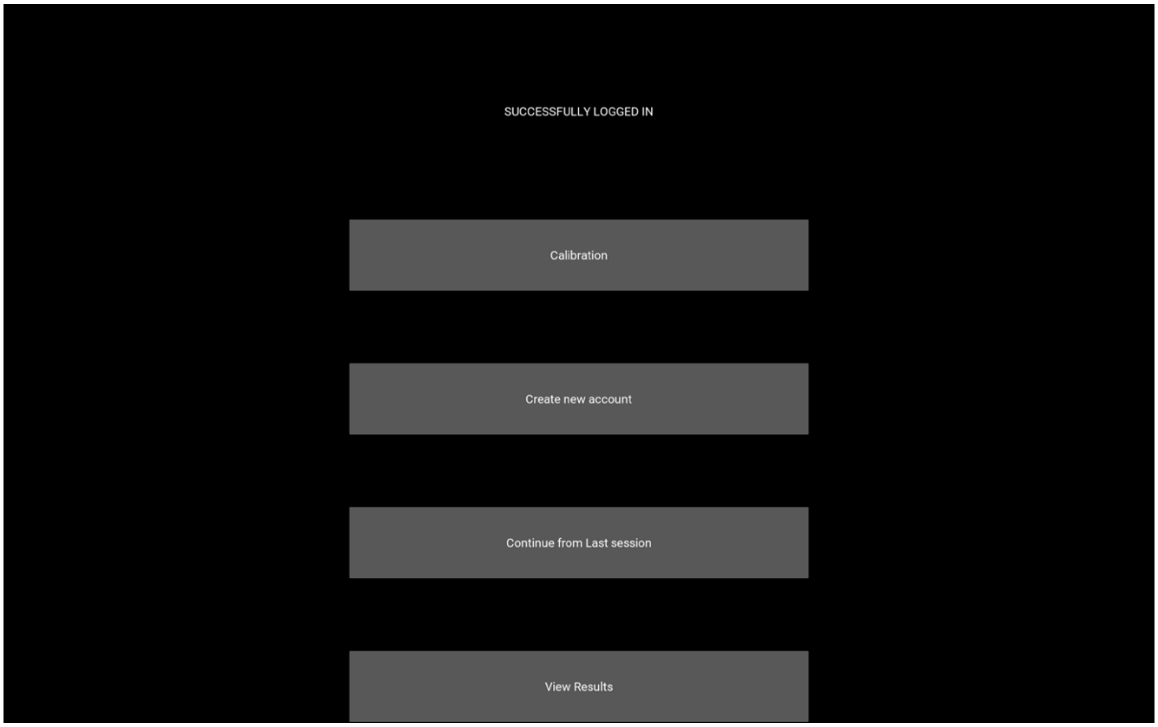

SUCCESSFULLY LOGGED IN

Figure 4.10: Main Page of our system

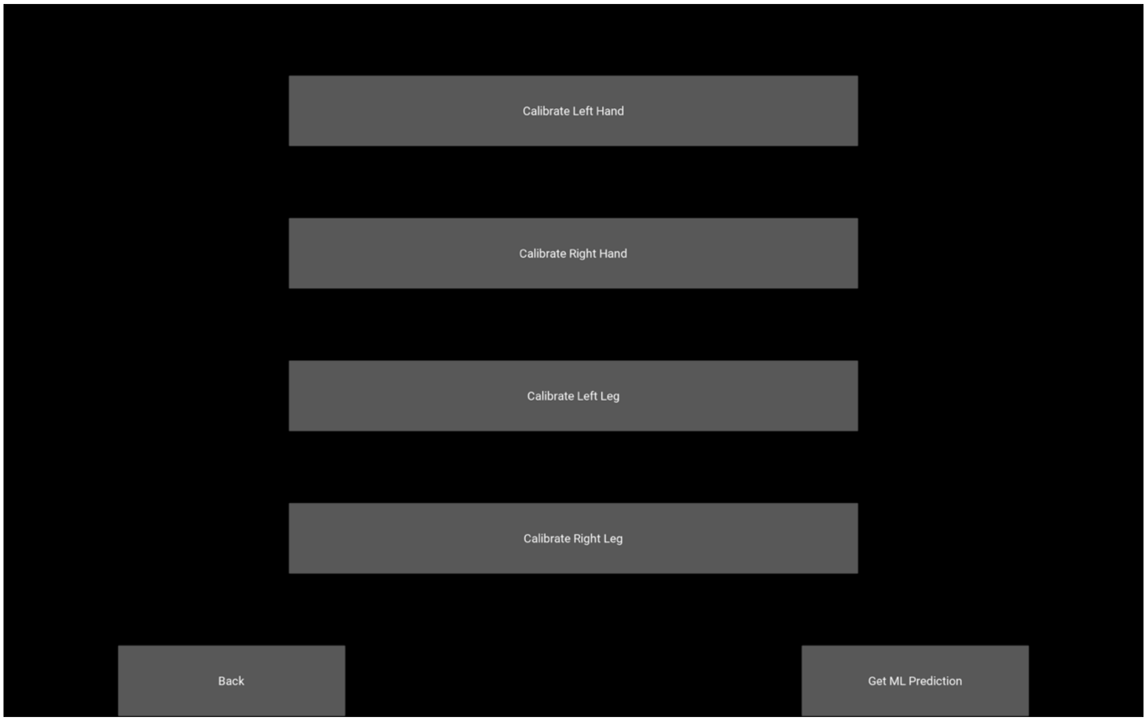
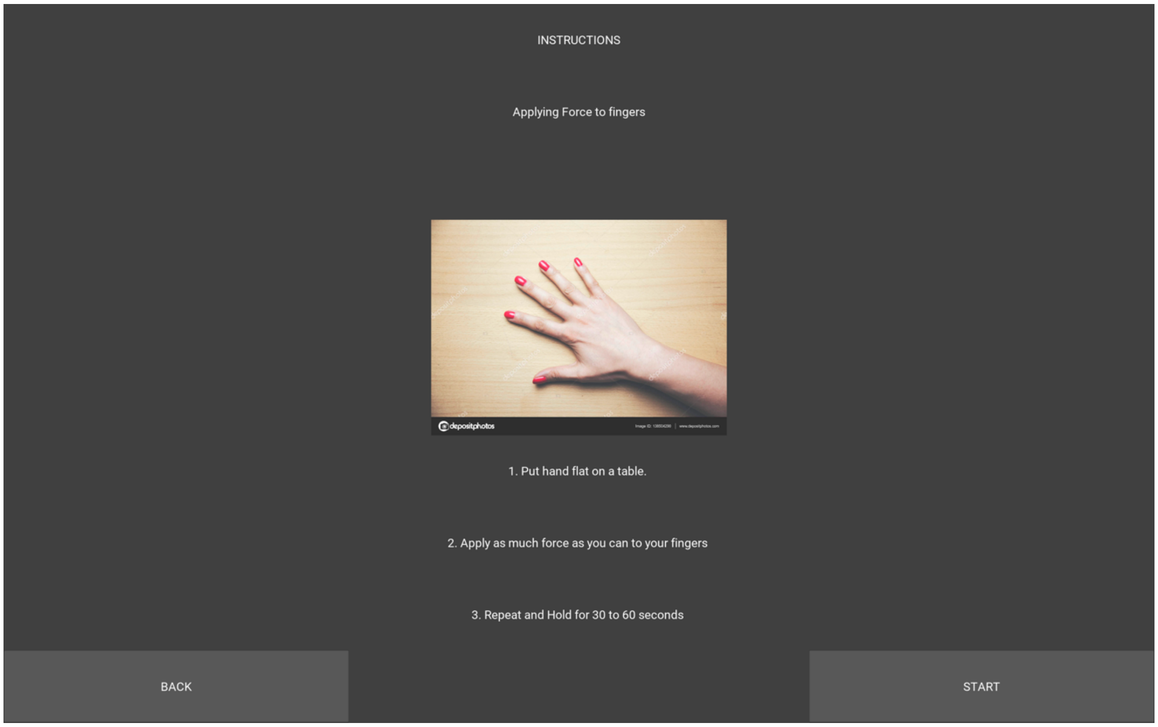


Figure 4.11: Calibration Page of our system



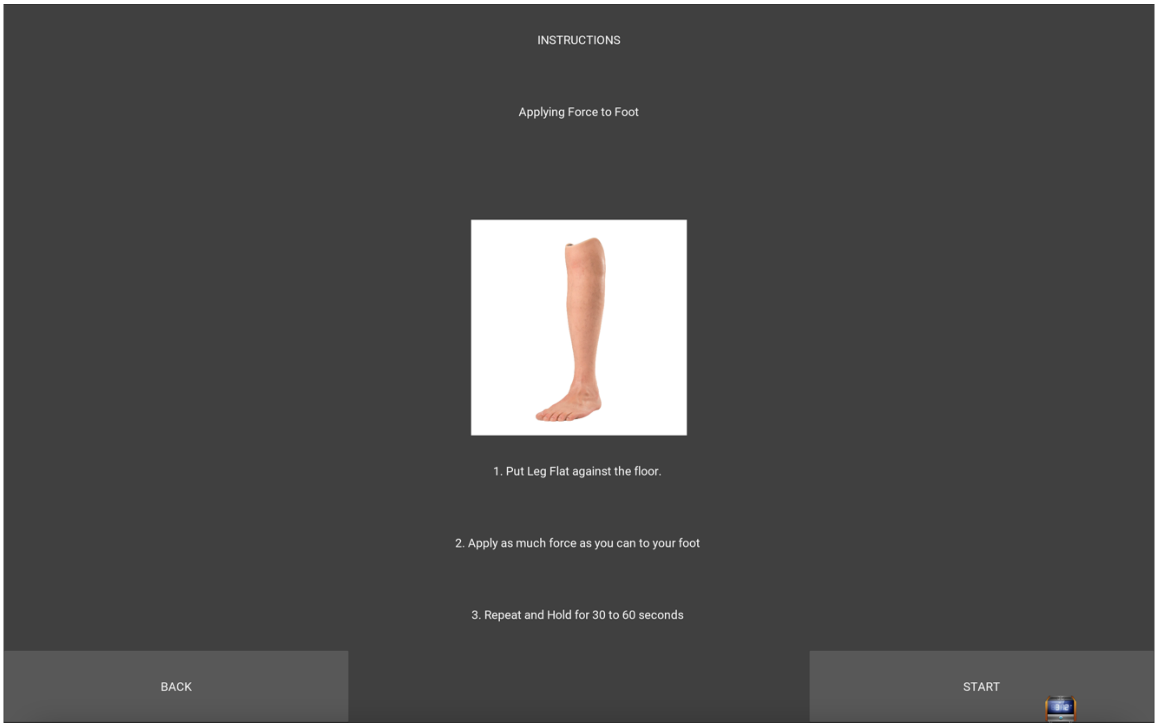
Applying Force to fingers

1. Put hand flat on a table.

2. Apply as much force as you can to your fingers

3. Repeat and Hold for 30 to 60 seconds

Figure 4.12: Tutorial for applying force to the fingers


INSTRUCTIONS

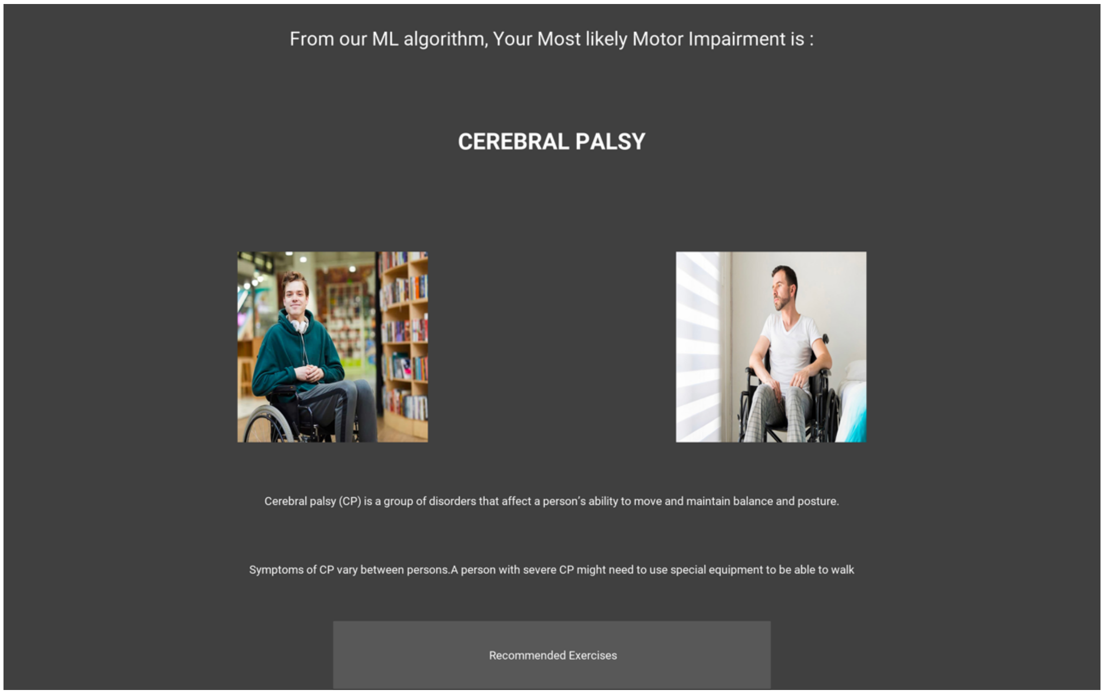
Applying Force to Foot

1. Put Leg Flat against the floor.

2. Apply as much force as you can to your foot

3. Repeat and Hold for 30 to 60 seconds

Figure 4.13: Tutorial for applying force to the leg


From our ML algorithm, Your Most likely Motor Impairment is :

CEREBRAL PALSY

Cerebral palsy (CP) is a group of disorders that affect a person's ability to move and maintain balance and posture.

Symptoms of CP vary between persons.A person with severe CP might need to use special equipment to be able to walk

Figure 4.14: Prediction page showing Cerebral Palsy

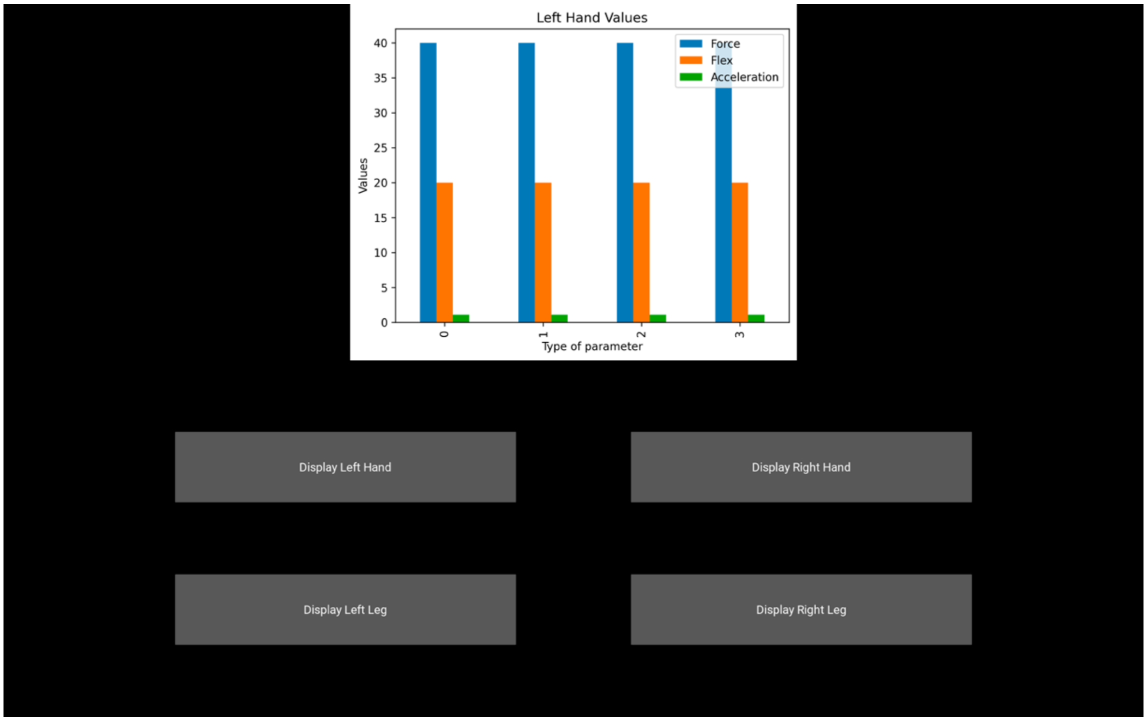


Figure 4.15: Bar charts showing the three different parameters

4.5 TESTING

To test the desktop application, SQLite was used to manage my databases, PyCharm was my development environment to handle the programming aspect of the project, the ML model was developed using Scikit Learn, a unit test was made by validating the speed and user interface of the software, as well as testing the accuracy of our prediction model, the sub system was later integrated with the hardware sub system to create the total system After system integration, a system test was carried out using a group of my friends who mimicked the symptoms experienced by physically disabled patients, and the test proved that the ML model was accurately predicting the type of motor impairment.

4.6 CONCLUSION

The testing of this project is currently being done by a group of my friends in Covenant University, as getting actual motor impaired patients is a difficult feature. When this is completed, the application would be revamped will be fully implemented. The final outcomes of the project are included in this chapter. Images of the software was put in this chapter as well as an overview of the code development, were presented. After testing, the system's findings were satisfactory for all requirements, and the project's goal and objectives were met effectively.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

This chapter contains the summary, achievements, limitations and conclusions of our project, and the recommendations to improve the efficiency of our project

5.2 SUMMARY

Following the successful completion of the whole project and testing, it may be concluded that the project achieves the goals and objectives that were set out to be achieved, with the possibility to do much more.

In this report, a short introduction was provided, as well as the purpose and goals of the project. The project's component components, as well as a study of relevant and existing related literature, were also discussed. Also covered in as much depth was the software development lifecycle and the architecture of the system under consideration. After that, the outcome of the testing and implementation process was discussed

5.3 RECOMMENDATIONS

The project was able to meet its objectives. There is a need for improvement on the functionality of the system. Some additional features that were not implemented in this project that could be added to later works are:

1. A physiotherapy aspect of the application
2. More motor impairments would be accounted for
3. Video therapy would be included

4. The Interface for the system would be improved
5. The application would be ported to Android and IOS

5.4 ACHIEVEMENTS

This project implemented the design of calibration system with the aim of physically impaired patients utilizing the application for physical rehabilitation. The hardware system comprises of sensors and a microcontroller; it also contains a Bluetooth module that sends data acquired from the sensors to the application system before encoding and storing that data.

5.5 CONCLUSION

The installation of a calibration system will be beneficial in a variety of settings, including residential, industrial, and commercial settings. If the techniques used in this research are put into practice, they will significantly alleviate some of the difficulties that physiotherapists are now experiencing. It would also be beneficial in terms of time and resource conservation.

REFERENCES

[1] S. P. Godse, “Musculoskeletal Physiotherapy using Artificial Intelligence and Machine Learning,” *Int. J. Innov. Sci. Res. Technol.*, vol. 4, no. 11, pp. 592–598, 2019, [Online]. Available: www.ijisrt.com.

[2] C. Tack, “Artificial intelligence and machine learning | applications in musculoskeletal physiotherapy,” *Musculoskelet. Sci. Pract.*, vol. 39, pp. 164–169, 2019, doi: 10.1016/j.msksp.2018.11.012.

[3] AnalyticSteps, “6 Major Branches of Artificial Intelligence (AI) | Analytics Steps.” https://www.analyticssteps.com/blogs/6-major-branches-artificial-intelligence-ai (accessed Jul. 13, 2021).

[4] Expert.ai Team, “What is the Definition of Machine Learning? | Expert.ai | Expert.ai.” https://www.expert.ai/blog/machine-learning-definition/ (accessed Jul. 13, 2021).

[5] “Regression vs Classification in Machine Learning - Javatpoint.” https://www.javatpoint.com/regression-vs-classification-in-machine-learning (accessed Jul. 13, 2021).

[6] Matthew David, Amy Novotny, and James Denman, “What is mobile application development? Summary and app development process.” https://searchapparchitecture.techtarget.com/definition/mobile-application-development (accessed Jul. 13, 2021).

[7] O. Postolache, D. J. Hemanth, R. Alexandre, D. Gupta, O. Geman, and A. Khanna, “Remote Monitoring of Physical Rehabilitation of Stroke Patients Using IoT and Virtual Reality,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 562–573, 2021, doi: 10.1109/JSAC.2020.3020600.

[8] G. Oonagh M., S. Kevin T., and C. Brian, “Rehabilitation exercise assessment using inertial sensors: A cross-sectional analytical study,” *J. Neuroeng. Rehabil.*, vol. 11, no. 158, pp. 1–10, 2014, [Online]. Available: http://www.embase.com/search/results?subaction=viewrecord&from=export&id=L609131738%0Ahttp://dx.doi.org/10.1186/1743-0003-11-158.

[9] Y. Ren, Y. N. Wu, C. Y. Yang, T. Xu, R. L. Harvey, and L. Q. Zhang, “Developing a Wearable Ankle Rehabilitation Robotic Device for in-Bed Acute Stroke Rehabilitation,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 6, pp. 589–596, 2017, doi: 10.1109/TNSRE.2016.2584003.

[10] D. Aggarwal, W. Zhang, and T. Hoang, “SoPhy : A Wearable Technology for Lower Limb Assessment in Video SoPhy : A Wearable Technology for Lower Limb Assessment in Video Consultations of Physiotherapy,” no. May, 2017, doi: 10.1145/3025453.3025489.

[11] A. Yurtman and B. Barshan, “Automated evaluation of physical therapy exercises

using multi-template dynamic time warping on wearable sensor signals," *Comput. Methods Programs Biomed.*, vol. 117, no. 2, pp. 189–207, 2014, doi: 10.1016/j.cmpb.2014.07.003.

[12] M. Ahmadi, M. O. Neil, M. Fragala-pinkham, N. Lennon, and S. Trost, "Machine learning algorithms for activity recognition in ambulant children and adolescents with cerebral palsy," pp. 1–9, 2018.

[13] P. Govindarajan, R. Kattur, and S. Amir, "Classification of stroke disease using machine learning algorithms," *Neural Comput. Appl.*, vol. 0123456789, 2019, doi: 10.1007/s00521-019-04041-y.

[14] S. Yahiya, A. Yousif, and M. Bakri, "Classification of Ischemic Stroke using Machine Learning Algorithms," *Int. J. Comput. Appl.*, vol. 149, no. 10, pp. 26–31, 2016, doi: 10.5120/ijca2016911607.

[15] D. Bilic and E. Golubovic, "Internet of Things-based System for Physical Rehabilitation Monitoring," no. October, 2017, doi: 10.1109/ICAT.2017.8171641.

[16] A. Al-Mahmood and M. O. Agyeman, "On wearable devices for motivating patients with upper limb disability via gaming and home rehabilitation," *2018 3rd Int. Conf. Fog Mob. Edge Comput. FMEC 2018*, pp. 155–162, 2018, doi: 10.1109/FMEC.2018.8364058.

[17] P. Martim, V. L. Frango, O. Postolache, and Y. Yang, "Smart Object for Physical Rehabilitation Assessment," *EPE 2018 - Proc. 2018 10th Int. Conf. Expo. Electr. Power Eng.*, pp. 678–682, 2018, doi: 10.1109/ICEPE.2018.8559647.

[18] M. Raad *et al.*, "An IOT based wearable smart glove for remote monitoring of rheumatoid arthritis patients," *BIOSIGNALS 2019 - 12th Int. Conf. Bio-Inspired Syst. Signal Process. Proceedings; Part 12th Int. Jt. Conf. Biomed. Eng. Syst. Technol. BIOSTEC 2019*, no. Biostec, pp. 224–228, 2019, doi: 10.5220/0007573302240228.

[19] G. Yang *et al.*, "An IoT-Enabled Stroke Rehabilitation System Based on Smart Wearable Armband and Machine Learning," *IEEE J. Transl. Eng. Heal. Med.*, vol. 6, May 2018, doi: 10.1109/JTEHM.2018.2822681.

[20] L. E. Emokpae, R. N. Emokpae, and B. Emokpae, "Flex Force Smart Glove Prototype for Physical Therapy Rehabilitation," *2018 IEEE Biomed. Circuits Syst. Conf. BioCAS 2018 - Proc.*, no. 1, pp. 1–4, 2018, doi: 10.1109/BIOCAS.2018.8584774.

[21] R. Alexandre and O. Postolache, "Wearable and IoT Technologies Application for Physical Rehabilitation," *2018 Int. Symp. Sens. Instrum. IoT Era, ISSI 2018*, pp. 1–6, 2018, doi: 10.1109/ISSI.2018.8538058.

[22] S. I. Lee *et al.*, "Enabling stroke rehabilitation in home and community settings: A wearable sensor-based approach for upper-limb motor training," *IEEE J. Transl. Eng. Heal. Med.*, vol. 6, no. c, pp. 1–11, 2018, doi: 10.1109/JTEHM.2018.2829208.

[23] M. Milenkovic, E. Jovanov, J. Chapman, D. Raskovic, and J. Price, “An accelerometer-based physical rehabilitation system,” *Proc. Annu. Southeast. Symp. Syst. Theory*, vol. 2002-January, pp. 57–60, 2002, doi: 10.1109/SSST.2002.1027005.

[24] R. Alexandre, O. Postolache, and P. S. Girao, “Physical rehabilitation based on smart wearable and virtual reality serious game,” *I2MTC 2019 - 2019 IEEE Int. Instrum. Meas. Technol. Conf. Proc.*, vol. 2019-May, 2019, doi: 10.1109/I2MTC.2019.8826947.

[25] A. K. Kushsairy, M. A. A. Malik, M. Y. Zulkhairi, H. Nasir, and S. Khan, “Real time monitoring system for upper arms rehabilitation exercise,” *2015 IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2015*, no. 2, pp. 2–6, 2016, doi: 10.1109/ICSIMA.2015.7559002.

[26] T. Badriyah, N. Sakinah, I. Syarif, and D. R. Syarif, “Machine Learning Algorithm for Stroke Disease Classification,” *2nd Int. Conf. Electr. Commun. Comput. Eng. ICECCE 2020*, no. June, pp. 12–13, 2020, doi: 10.1109/ICECCE49384.2020.9179307.

[27] B. M. Eskofier *et al.*, “Recent machine learning advancements in sensor-based mobility analysis: Deep learning for Parkinson’s disease assessment,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2016-October, pp. 655–658, 2016, doi: 10.1109/EMBC.2016.7590787.

[28] M. Lapresa *et al.*, “A Smart Solution for Proprioceptive Rehabilitation through M-IMU Sensors,” *2020 IEEE Int. Work. Metrol. Ind. 4.0 IoT, MetroInd 4.0 IoT 2020 - Proc.*, pp. 591–595, 2020, doi: 10.1109/MetroInd4.0IoT48571.2020.9138193.

[29] D. Y. Kim, J.-H. Kim, M. Prabakar, and Y. Jung, “Design of Smart Portable Rehabilitation Exoskeletal Device for Upper Limb,” pp. 134–134, 2016, doi: 10.1109/sbec.2016.97.

[30] Y. F. Chen, D. Napoli, S. K. Agrawal, and D. Zanotto, “Smart Crutches: Towards Instrumented Crutches for Rehabilitation and Exoskeletons-Assisted Walking,” *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, vol. 2018-August, pp. 193–198, 2018, doi: 10.1109/BIOROB.2018.8487662.

[31] H. Alyuruk, “Chapter 1 - Introduction to R and Python,” in *R and Python for Oceanographers*, H. Alyuruk, Ed., ed: Elsevier, 2019, pp. 1-21.

[32] M. Ozkaya and F. Erata, “A survey on the practical use of UML for different software architecture viewpoints,” *Information and Software Technology,* vol. 121, p. 106275, 2020/05/01/ 2020.

[33] P. B. Kruchten, "The 4+1 View Model of architecture," *IEEE Software,* vol. 12, pp. 42-50, 1995.

[34] B. P. Douglass, “Chapter 1 - Introduction to UML,” in *Real-Time UML Workshop for Embedded Systems (Second Edition)*, B. P. Douglass, Ed., ed Oxford: Newnes, 2014, pp. 1-32.

[35] M. J. Chonoles, “Chapter 12 - Use Cases,” in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 207-225.

[36] M. Fowler, C. Kobryn, and K. Scott, UML distilled: a brief guide to the standard object modeling language: Addison-Wesley Professional, 2004.