Fall Estimation System Based On Human Gait

M.Tech Post Graduate Project Thesis

to be submitted by

Rajan Shukla

for the partial fulfillment of the degree

of

MASTER OF TECHNOLOGY IN

Communications and Signal Processing

under the supervision of

DR. SRIKANTH SUGAVANAM

SCHOOL OF COMPUTER AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MANDI

KAMAND-175075, INDIA

MAY, 2023

Contents

List of Tables

List of Figures

Timeline

UP-Till	Tasks
1st evaluation (Sem-3, Aug 2022)	Literature Review: Focus will be on developing the understanding of the Human Gait, IMU and sEMG sensors from the papers published in this field.
2nd evaluation (Sem-3, Nov 2022)	Data pre-processing , feature engineering and system architecture design for training the model and evaluating performance.
1st evaluation (Sem-4, Feb 2023)	Experimentation Phase: At this stage there will be rigorous experimentation for finding the right configuration of the model taking in account that trained model will be making inferences on the resource constrained Edge device.
2nd/Final evaluation (Sem-4, May 2023)	Model Optimization and deployment of Pre-Fall detection system on edge device.

Chapter 1

Introduction

Human balance is the capacity to keep the body in a stable position within predetermined limitations, and it applies to all activities of everyday living. Elderly people frequently lose their balance, and the age-related reduction in stability recovery ability raises the danger of falling [2]. As humans, we have developed the capacity to keep our bodies balanced and have become adept at doing so. Our body posture and gait are synchronized with our state of health and well-being. If we assume that some component of our body was not functioning properly, our entire bodily structure is in danger. Now that our older generation lacks this ability to create equilibrium, it is actually the most prevalent issue relating to fall related accident. Indeed, with an estimated 420,000 [3] fatalities each year, falls rank as the second most common cause of accidental death resulting from avoidable injury. Those over 60 make up the majority of this rate.50 percent of them cause significant problems to the elderly [4], Long-term hospitalisation, loss of independence, incapacity, and early mortality are all serious issues for the elderly.

This leads to open many reason to think it is possible to predict if a person is about to fall based on the Human Gait cycle [2]. There are most earlier researcher are based on the computer visions technique but they have the problem of scalability so there some researches are thinking of making use of the IMU (Inertial Measurement Units) sensors or sEMG sensors [5] to make a prediction based on the movement of the body and the at the same time the electrical pulses generated by these motion. In the next

section, we shall first understand the Human Gait cycle.

1.1 Human Gait cycle

Human gait may be defined as the analysis of our body's joint or moment trajectory with regard to time in numerous cycles. There are mainly two sub-cycle of gait called stance and swing. During motion our either of our legs is in stance and swing cycle.The gait cycle begins for a typical person when they take their first step from a standing posture, also known as the first double limb support. After this we either take left or right leg forward, depending upon the choice of leg the one which is in air goes into the swing phase and the leg which is into the ground is in the stance phase.
Now for the first double limb support both of our leg is in stance phase [1].
In second limb support one of the leg is in swing phase other is in stance. For the second double limb support again both of the leg will be in stance phase and second limb support is same as first limb support but with opposite leg. This phases of gait cycle produces the pattern which can be extract for processing.Gait cycle is the time interval between two successive occurrences of one relative repetitive movement of the events.

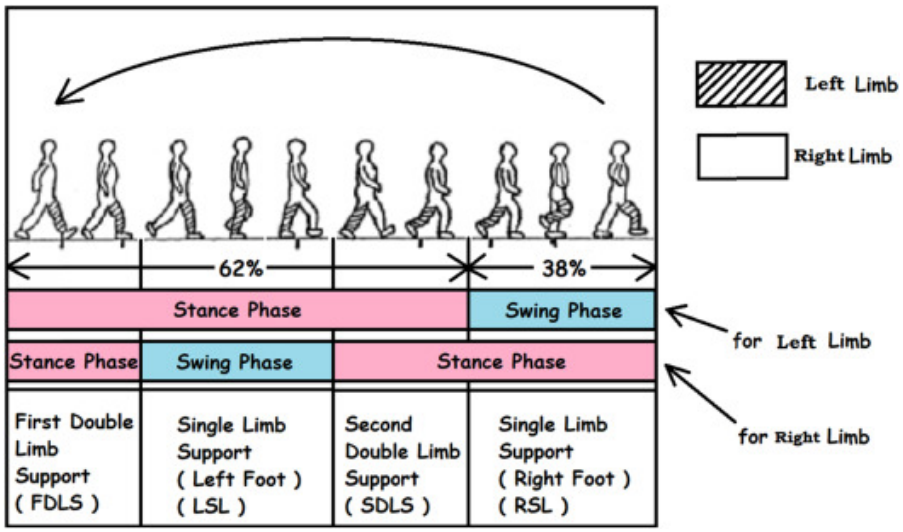


Fig. 1.1: A complete gait cycle showing the swing and stance phase for each left and right leg. Image credits- Modern Methods for Affordable Clinical Gait Analysis[1]

The gait cycle there is the overlapping between the stance phase of the limb and the swing phase of the other limb. With the help of this stance and swing phase we can study the different pattern of the body.

1.2 Behaviour of Muscles

Skeletal muscles are made up of individual muscle fibers. And like muscles themselves, not all muscle fibers are the same. There are two types of skeletal muscle fibers, fast-twitch, and slow-twitch [6], and they each have different functions that are important to understand when it comes to movement. To understand the root cause of muscle dysfunction and the primary cause of stability issues I will be highlighting the biological issue and establishing their relevance in the coming paragraphs First understand that Individual muscle fibers make up skeletal muscles. And, just like muscles, not all muscle fibers are the same. Skeletal muscle fibers are classified into two types: 1)fast-twitch 2)slow-twitch Both of them function differently. In the next subsection focus on how their fibers are different.

1.2.1 Slow twitch muscles fibers Type-1

Less Aerobic fiber, also known as Type 1 fiber, is fatigue-resistant. Mainly contribute to small movements and maintain postural control.
The reason for this behavior is that they contain substantial amounts of mitochondria (an organelle present in most cells also called the powerhouse of the cell responsible for energy generation) and myoglobin (a haem[an iron-containing compound]-containing red protein that transports and stores oxygen in muscle cells)

1.2.2 Fast-twitch muscle fibers Type -2

More aerobic less blood flow fiber muscles can produce more substantial forces for a shorter period of time and they also fatigue faster also known as type II fibers. Now for a person having weak stability can mainly be due to a deficiency of Type 1 fiber and a person who is more prone to falls can because of the significant loss of the Type

1 fiber understand their characteristic was important for making sense of the aging effect on muscles.
Before that we will be highlight the last key to this mystery which is speed of contraction [7].

1.2.3 Speed of Contraction

The main distinction between the Type I and Type II fiber is the rate at which myosin [7] hydrolyzes the ATP (Adenosine Triphosphate Energy carrying molecules found in the cells of all living things).
Type I fiber hydrolyze ATP almost twice as quickly as the slow fibers. Because of this faster hydrolyzing, fast fibers perform cross-bridge cycling i.e drawing the filament closer to the core of sarcomeres (the basic contractile unit of a muscle fiber [myocyte]).
In coming section we will be concluding the cause of muscle weakening due to the defect of Type 1 and type 2 fibers which in turns slow down the speed of contraction

1.2.4 Aging effect on muscles

Aging causes loss in the muscles which results in primary decline in the total number of other type1 and type2 fibers. This condition has a medical term called Sarcopenia [8] which is discussed in the coming section. Atrophy (the medical condition of losing flesh, muscle, strength, etc. in a part of the body because it does not have enough blood) of type II fibers leads to slower contraction in the older muscles.
In addition the loss of alpha motor neurons (are large, multi-polar lower motor neurons of the brainstem and spinal cord.)

The alpha motor neurons innervate (supply [an organ or other body part] with nerves). extrafusal muscle fibers (the standard skeletal muscle fibers that are innervated by alpha motor neurons and generate tension by contracting, thereby allowing for skeletal movement.) and are the primary means of the skeletal muscle contraction. This research helped in establishing the Hypothesis that we started the assumption

that the effect of muscle dysfunction is primarily noticeable in older generations.

1.3 Gait Measurement sensors

In papers on gait prediction, two different types of sensors have been primarily employed to evaluate the human gait pattern. These sensors can be positioned all around our body. The device's data collection in its raw form needs to be further evaluated. IMU and EMG sensors are introduced in the following paragraph.

1.3.1 IMU Sensors

We need a sensor that can convert motion into an understandable signal in order to examine the motion-related anticipation of any phenomena. We have IMU, or inertial measurement unit, sensors, for this.
IMU [9] sensors are nine-axis-senses, which means they combine the magnetometer, gyroscope, and accelerometer sensors. An accelerometer measures acceleration in the x, y, and z directions, a gyroscope measures orientation change, and a A magnetometer is a device that measures magnetic field or magnetic dipole moment that isn't related to our field of research.
An accelerometer sensor will be helpful in deriving details from the gait cycle in order to understand the movements.

1.3.2 sEMG Sensors

Electromayography is a technique for measuring the electrical pulses produced by nerve stimulation. EMG monitors the electrical activity of the pulse during muscle contraction and has been utilised to understand the anomalies of the muscle reacting to brain pulses.
Muscles do not normally produce electrical impulses, but when stressed, they do produce electrical signals that surface electromayography sensors detect. Many articles[10] have been published in order to study a person's electrical reaction when walking. We

will use sensors to identify the trait of the person while they are at that stage because they can fall at any time.

1.4 Sarcopenia

Our muscles perform an important function in our body, helping us to move, keep our posture, and stabilise our joints. However, as early as the age of 30, we all begin to lose muscular mass and strength. Some of us lose it faster than others due to a dangerous disorder known as sarcopenia, which grows more frequent with age and affects 10 to 20% of older persons. Disability, loss of independence, more frequent hospitalizations, the need for long-term care, and even death can result.

We commonly think that our hearing, eyesight, and capacity to move freely will inevitably deteriorate as we age. The fast loss of muscular mass and strength caused by sarcopenia is not unavoidable, but it has been neglected and undertreated. Many elderly people are unaware that their difficulties mounting the stairs or getting out of their chairs is caused by sarcopenia. The good news is that scientists and healthcare practitioners have made significant progress in understanding how the disorder works, how to identify it, and how to effectively treat it in recent years.

What causes sarcopenia is not always evident. Sarcopenia can be caused by a variety of factors. Age-related changes in the body, which we all experience to some extent, can play a role.

Lower levels of specific hormones, reductions in the body's capacity to convert protein into muscle, increased inflammation, interference with signals between the brain and the muscles, and other cellular abnormalities can all contribute to this. Environmental and lifestyle changes, such as inactivity and lack of exercise, extended bed rest, loss of mobility, poor diet, tooth and oral disorders, and obesity, can all contribute to the development of sarcopenias. It is crucial to remember that even those who are overweight or obese, as well as those who are physically active, might be at risk for sarcopenia.

Sarcopenia can also be caused by chronic conditions. These disorders can either pro-

duce some of the molecular alterations that contribute to sarcopenia or interfere with a person's ability to consume a balanced diet and exercise.
Muscle mass loss can have a variety of consequences. It can impact your balance and ability to walk, weaken your bones and make you feeble, increase weariness, increase your risk of disease, aggravate existing diseases, induce weight gain, increase your risk of malnutrition, and more.
All of these factors can contribute to difficulty getting around, difficulty performing normal daily activities, falls and bone fractures, increased disability, diminished quality of life, loss of independence, more and longer hospital visits, a higher risk of post-surgical complications and lower survival rates, and eventually the need for long-term care and institutionalisation.

1.5 Objective

The primary goal of this study is to identify sarcopenia symptoms during the transition period in order to provide adequate health care and prevent the individual from falling. We can notice indications if people examine their muscles on a regular basis, much like they do for other preventive diseases. The successful outcome of this research will be applied to the development of a predictive healthcare application. This can assist the Orthopaedic in providing better care to the individual suffering from this ailment.
The key problem for this study is that the data-set itself is not available for creating any model. There is no low-cost hardware available for gathering the data-set from the sEMG and IMU.
The necessity for research is to obtain a high-quality data collection as well as the necessary gear. Once we get the data set, the Deep Learning model will assist us in predicting the symptoms based on the real-time data set.
With this project, we will aim to tackle this problem from the ground up and create a solution that can be utilised by medical personnel.

1.6 Contribution

My contributions to the thesis by focusing on the detection of anomalous GAIT patterns. Firstly, I developed an innovative method that utilizes an autoencoder network to detect these anomalies, achieving an impressive accuracy of 0.86. This approach provides valuable insights into identifying abnormal gait patterns, which can have substantial implications in healthcare and rehabilitation settings.

Furthermore, I designed and constructed a custom sensor capable of recording synchronized readings from IMU (Inertial Measurement Unit) and sEMG (Surface Electromyography) sensors across multiple sensor nodes simultaneously. This development not only enabled accurate data collection but also facilitated comprehensive analysis and interpretation of gait patterns.

Additionally, I placed great emphasis on optimizing the model size to ensure its efficient deployment on edge devices. By utilizing both Python and C++ programming languages, I was able to develop a compact and efficient codebase, enabling the deployment of the model on resource-constrained devices with less sacrificing performance.

Overall, my contributions to the thesis encompassed the development of an effective anomaly detection method, the design and implementation of a custom sensor system, and the optimization of the model for deployment on edge devices. These advancements hold the potential to enhance gait analysis techniques and contribute to the field of healthcare and rehabilitation.

Chapter 2

Literature Survey

The purpose of the literature review is to better understand the stakes and their influence on the outcome. So the earliest and most obvious beneficiaries of this project's output are older individuals, as it is estimated that thirty to forty percent of [11] will fall at least once, and a falling occurrence can lead to death. The current review summarises demographic and modifiable risk variables [11] and gives a concise description and update of the relevant research. For this project we have curated all of the paper from different perspective of fall related prevention techniques. Below Fig 2.1 gives reader an idea of how we have selected our field of focus. The knowledge for finding the research gap in this field lead to the

The Research institute of Finland VTT [12] was the pioneer in establishing that human gait can be used to identify the recognized the person. Their approach was basic they used concept of signal processing a template matching technique using cross-correlation.They have use an accelerometer fasten on the waist of the 36 test subject they have achieved the (EER) Equal Error rate of 6.4 percent.

Nguyen et al [13] were the first who found out that the convolutional neural network based on architecture can recognize the gait pattern generated form the accelerometer and gyroscope sensors. They have shown that their EER is 10.43 percent.

Delgado et al [14] Have found very much significant distinction between weather accerelometer or gyroscope based is good feature for the gait based prediction. Their finding is that when the model is trained on the Accelerometer work better then gy-

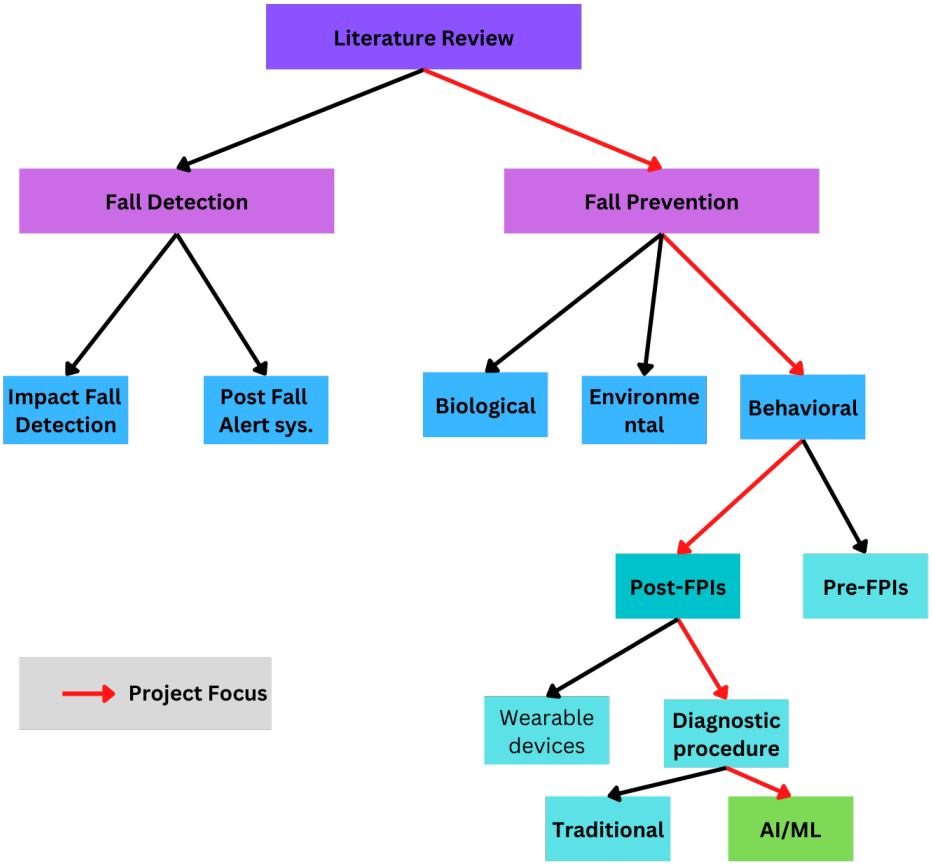


Fig. 2.1: Tree Diagram to show the flow of research over the time

roscope.Furthermore, the system using both sensors performed better in all cases.

Gadalat and Rossi [15] was the first to use deep-learning (CNN) to extract the feature. They used the five consecutive gait cycle. Once the feature are extracted from the CNN the extracted feature are feed to the one class SVM(Support Vector MAchine) classifier.

Rescio et al [16] were the first to use data from sEMG sensors to train a machine learning model. They employed a set of four sEMG sensors on different parts of the test participant. They took a guided approach. They took data from the subject for walking, sitting, bending, and standing. In controlled condition they have predicted the fall of a person before the impact of about 775ms.

Wigran et al [17] suggested a technique for using the EMG and IMU sensors simultaneously. Their findings indicated that IMU-based sensors perform better than EMG-based ones, with accuracy of 77.6 percent compared to EMG's 67.76 percent.
There are mainly two different stratgey for the fall related research:
Fall Detetion
Fall Prevention
The scope of this project lies in the fall prevention in elder maily on the POST-fpi[18]. It is also found out that people who wore fall detector feld more confident and in independent in their regular activities. There are many extrinsic and intrinsic reasons for the fall below the diagram showns the list of reason for fall. The biological reason has been discussed in the introduction section in Behaviour of muscles.In more recent studies the wireless body sensor networks referred as the (BSNs) for the realtime monitoring for the fall preventions. Hence a diagnostic approach would be preferred if developed properly since not research shows that the people tends not to wear sensors for long periods of time in a survey 32% [19] of the users stops wearing in six months and 50% [19] after the a year.Accelerometers, gyrosensors, force sensors, or strain gauges, can assess a variety of human gait parameters. To anticipate a future fall, signal processing algorithms and feature extraction approaches may be created to analyse numerous gait data such as stride speed, stride length, and inter leg spacing [20]. Which is been discussed in the Gait cycle section of Introduction.
Howcroft et al. [21] have done their research on detecting the fall- risk prediction they have used the accelerometer and pressure sensing insoles used to train neural networks aslo the traditional SVM and naive Bayesian. The location of the accelerometer were head, and left shank.
The Post-FPIs (Post Fall prevention systems) are the papers that is most align with the objective of this project. Post-fall prevention intervention systems (Post-FPIs) are commonly employed in the first instance to evaluate patients for fall risks following a fall.In this researches technique peoples are trying their best to estimate the likelihood of future fall based on the both the extrinsic and intrinsic cause as discussed in the paper above. This research include the development of the diagnostic evaluations [22]

function which identifies the reasons for the fall or in terms of industries the Root cause analysis. Beyond this paper the literature review was on the every direction possible now with the papers coming forward has streamline the think and the work for the project. Post-FPIs and technologies that focus on fall prevention by screening and assessing for fall risks may also be viewed in terms of two fall risk factor categories: intrinsic risk factors and extrinsic risk factors [23]. Functional ability deficiencies are the only focus of assessment in for intrinsic risk factors.

Majumder et al. [24] discover irregularities in gait patterns, which are thought to be a prevalent cause of falls in the elderly. Users are warned about the possibility of falling based on data gathered and categorised to evaluate whether or not ADL patterns are abnormal.

Majumder et al [24]. have done the intrinsic fall risk analysis as functional ability deficiency. They studied the goat patterns which they collected over time when the participant were doing their active of daily living (ADL). Then their is a paper by **Staranowicz et al** created a system to track the activities of the elderly persons they do ADLs at their home. IN their research they have focused on the IOT- enabled wearable device that can make the prediction.The contextual information about the patient's behavioural habits and their surroundings is also very crucial for building the predictiove system for fall detection. The Research of **Gravin et al**. gave a detailed on the systematic assessment of the multi-sensor data fusion approaches in Body sensor Networks [25].

In Parkinson's disease the most significant cause is the Freezing of Gait (FOG). The research have detected that the EEG signals and EMG signals pattern can identify the beginning of the FOG. **Handojoseno et al**. used EEG data to train the DNN to identify the FOG in Parkinson's disease patients. Their results has 80% sensitivity and specificity for the beginning of freezing in Parkinson's disease.

The study of Xi et al found that the four sEMG sensors for the fall detection system development is sufficient for reaching the 90% sensitivity and specificity.

Based on sEMG and accelerometers, **Cheng et al**. [26] created a framework for assessing the feasibility of fall detection. Their method achieved an accuracy of over 98%

in activity identification, confirming the practicality of the suggested method in daily activity awareness.
Their procedures, however, were tested on healthy people and did not include aged adults or frequent faller's. [20]

The Research Gap:
According to the preceding literature, our objective of developing the Fall prevention system falls under the post-FPI **Howcroft et el** future work. They indicated that building a preventative system is an effective and practical approach because there is therapy for sracopenia-like symptoms if we notice them early. Our research and approach will be beneficial in this regard.

2.1 Novelty description

The current research in fall risk assessment has primarily focused on detecting the occurrence of a fall event. However, our project aims to revolutionize the diagnostic procedure by replacing the traditional time-based method with a deep learning model. Our approach involves developing a custom sensor that reads IMU and sEMG data from the subject performing a short battery test, which is commonly administered by physiotherapists to assess fall risk. The sensor data is then fed into an edge deep learning model that produces a numerical output indicating the fall risk level of the subject. This novel approach has the potential to enhance the diagnostic procedure by providing a more objective and accurate assessment of fall risk, allowing for timely and appropriate interventions to be implemented. Furthermore, this approach can also minimize the subjectivity associated with traditional methods and can be easily implemented by healthcare practitioners in clinical settings.

Chapter 3

System Model

3.1 TinyML and Edge Computing

TinyML (micro-controller based machine learning approach) is a quickly growing field of ML/DL technologies and applications that consists of hardware, algorithms, and software that can analyse sensor data on-device at very low power—typically in the mW range and below—enabling a variety of always-on use-cases and focusing on battery-powered devices[27]. The main issue with embed devices is that they can have much less RAM than a few hundred kilobytes.

Embedded devices still come with some tough resource constraints, though , they have clock cycle of around 10MHz or less. This configuration make extremely difficult to run ML/DL model on device[28]. Has been overcome by the Tiny-ML approach in fact the Google has launched a TensorFlow-lite-Micro framework for the Tiny-ML application development.

With the help of TinyML we can make the faster inferences where the data is been generated.

3.2 Our Approach

From the literature survey it is evident that research is been moving the direction of making a model which can detect the pre-fall of a person and around this many model

has been developed and they are making to make it a model which can make inferences in the embeded setting.

Though there are several papers who talks about the deployment of the model but our approach is to make the model aware of the deployment. In tinyML we have to make the model aware of the deployment scenario. Our vision is to make the edge device which can make the prediction by capturing the gait pattern through the sensors like IMU or sEMG and start giving the alert even before it happens.

We are implementing on the Edge device which will helps us to make the faster inference. The work of [Rescio et al,6] have shown that they make prediction of fall of a person 775ms before. We are moving with the different approach here we will be developing the diagnostic procedure to prevent the person from falling not the detection of falling. **Training Auto-Encoder Model on ZJU-GaitAcc.**

Over the next couple of month will be spent on finding the right model configuration and training the model of ZJU-Gait Acc data to be able to Distinguish between Normal and Abnormal GAIT. This will be done for through form to training to the deployment to the edge device. We will have to build the code for the prepossessing in c++ for embedded setting.

3.3 About the Data-set

ZJU-GaitAcc[29] is the open source available data-set of gait acceleration time series. They have collected the data from 175 test subjects. and each record contains five gait acceleration series simultaneously recorded at the different location like right wrist, left upper arm, right side of pelvis, left thigh, and right ankle, respectively. The Data-set have about two third are male, one third are female Age is between 16 and 40, Height is between 1.5 and 1.9 m

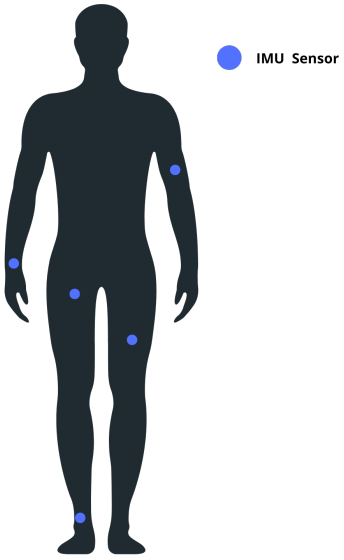


Fig. 3.1: Location of the IMU sensors with blue dots

3.4 Data Prepossessing and Feature Engineering

Reading the ZJU- Gait data-set is difficult since it is not simple to use tfds and run the model. The data-set is the tale that the developer must comprehend in order to make optimal use of it. The paper has helped us comprehend the data collecting process and how it is structured as a result of this. Now, for our development, we have used this data to train models for the identification of normal and pathological gait patterns. For this, we will be employing the unsupervised learning technique, namely the auto encoder.

They developed distinct age groups of participants and separated them into three sessions. In each session, 22 volunteers are made to wear the five sensors in various areas on their bodies. There are six recordings of each individual walking while wearing the sensors.In the same way, we have many recordings of people walking while wearing those sensors in each session.

See the tree diagram below to describe the entire data-set:

The literature found out that type 2 fiber are more responsible for muscle weakness in older age people and most of this fiber are have located at the muscles are forearms and thigh muscles so we have to extract the rows data for training a model.

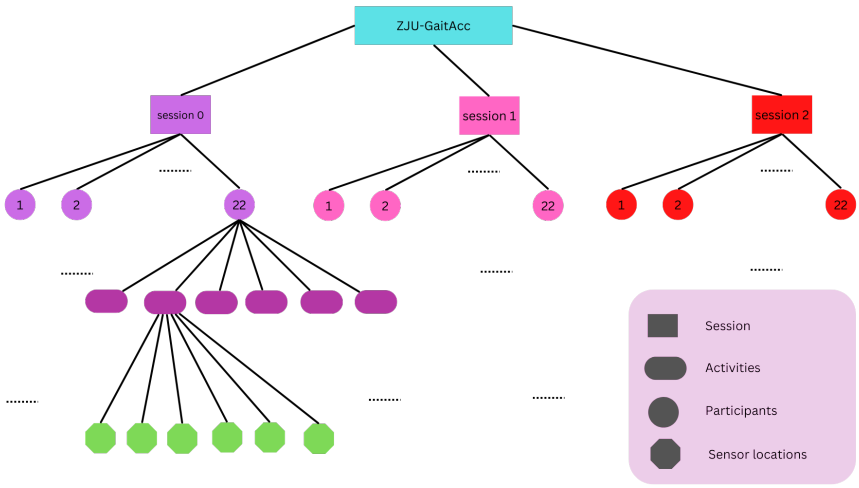


Fig. 3.2: Data Structure of the Gait-ZJU open source dataset

Now for this, I have developed a python script that can automatically extract the key location for the given participant.

From here we can start our data prepossessing pipeline.

The Pipeline as of now contains four different prepossessing

The pipeline and each phase of the data-set, and I want to underline that this pipeline must be repeated for C++ in order to work on the embedded system.

- Selection
- Segmentation
- Transformation
- Augmentation

- **Selection**: Based on the required location the selection of the accelerometer data.Across the session for each participant. The selected part is kept as the NumPy array for the training session. **Segmentation**: As per the paper they have suggested that the starting and terminating part of the signal should be removed as they are readings of a person when they have instructed them to start the recording when they have started the gait activity.

Now we have to eliminate this part for the quality data for that they have provided

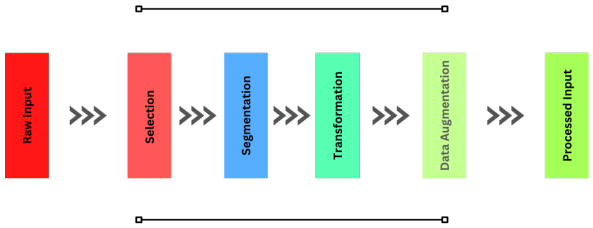


Fig. 3.3: Pipeline - Different stages of data prepossessing

the annotation we have taken their reference indexes.

This process of segmentation is illustrated in the figure below:

Now this segmentation at the time of easy as we have nicely annotation but at the

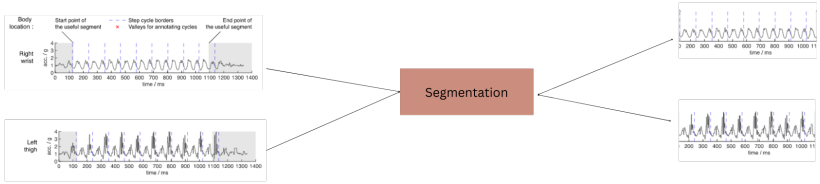


Fig. 3.4: Segmentation

Edge level at the time of inferences we don't have this leverage there we have to use different signal processing approaches like using power spectral density and only taking the part of the signal where this signal has significant power.

Once we have clean data stored in the NumPy array we can proceed with the trans-

formation.

Once we have clean data stored in the NumPy array we can proceed with the trans-

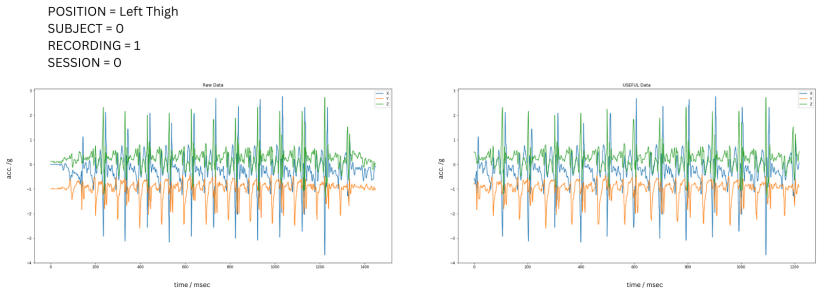


fig. Data preprocessing for eliminating the not useful data point at the starting and the ending of the gait session

Fig. 3.5:

formation.

Transformation: Since the literature review I have found out that the convolution neural network would be giving better results for the accelerometer data.

So that we need to convert the signal into some sort of plot which then can be fed into the network. In some papers, they have identified that the Spectrogram plot is been suggested For this task.

However again a problem is their accelerometer we have three plots on how can we use this plot to feed into the network is been answered by this paper.

This paper has suggested that using two different ways of them is equally effective during their study.

- **Concatenation**: Here we have conceded the x,y z spectrogram side by side and then feed them into the network.The below illustration can help us understand the process.

- **Stacking**: Using these three plot as the channel of the plot and stacking them

into the network.

The below illustration can help us understand the process.

Augmentation: Now the deep learning model tends to over-fit usually onto the data

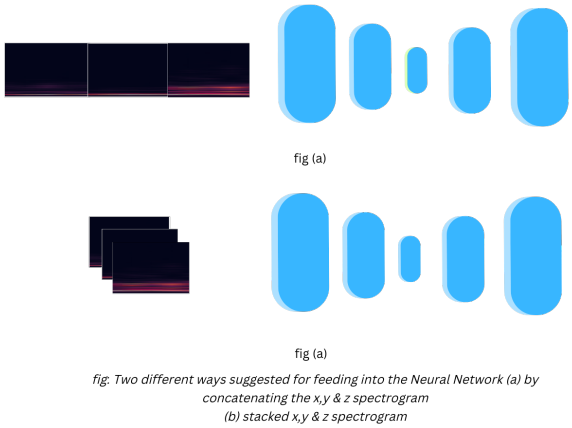


Fig. 3.6:

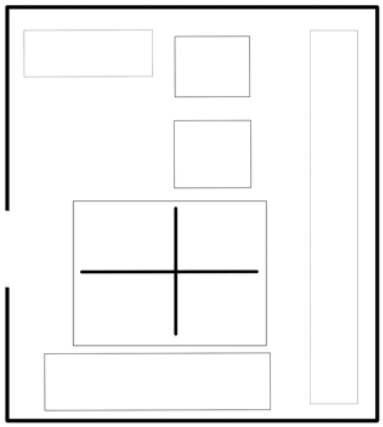
so we will internationally add small noised onto the data-set on the fly as regularization. All of the pre-processing except the augmentation will be first used for the training when the project progress to the deployment stage all the code will be converted into c++ for making inferences on the embed setting.

3.5 Data Collection

We are designing the sensor network architecture and framework to gather data from the IMU and sEMG. Despite the fact that the specifications and designs are still being worked on. According to our preliminary study, there is presently no market product available for the sort of data we intend to collect from the test participant.The only close product available in market are IMU sensors used by professional sports temp to analyze the performance of athlete's and their product and quite and we have developed the sensor network which is explained in the next section.

3.5.1 Data collection protocol.

The validity of the study's findings might be significantly impacted by the reliability and correctness of the data that was gathered. We will describe the special data collection procedure we devised for our research project in this report. In accordance with our policy, participants are invited to a room where they will be asked to complete five GAIT tests and have their medical history taken. Medical History: The participant's medical history is entered into a database as the first stage in our data collection process. It is essential to have this information in order to assess the participant's general health and rule out any conditions that would impair their ability to accurately complete the GAIT tests. We will ask them a series of questions regarding their present health condition, any prior injuries or operations, and any drugs they are now taking in order to record their medical history. After obtaining the participant's



0.5

Fig. 3.7: Test protocol layout.

medical background information, we will ask them to complete five GAIT tests. These assessments are made to gauge the participant's gait and balance in a variety of ways. Let's examine each of these tests in more detail:

1. The 30-second chair stand test, or 30CST, gauges the stamina and strength of the lower limbs. As many times as they can in 30 seconds, participants must get

up from a chair and seat down (Fig 3.8).

2. Balance and gait are assessed using the Performance-Oriented Mobility Assessment (POMA). A series of exercises, including heel-toe walking, standing with their feet together, and turning around, will be required of the participants. For further information on the test site, see (Fig. 3.8).

3. The Four Square Step Test (FSST) gauges one's dynamic mobility and balance. Players must move as swiftly and precisely as they can in and out of the four squares that have been set up on the floor (Fig 3.8).

4. GAIT Speed Test: This test gauges how quickly you can walk. Participants will be required to stroll for a brief distance at their typical pace, and their walking speed will be recorded (Fig 3.8).

5. VISUAL GAIT Pattern: The Visual GAIT Pattern exam uses visual analysis to assess the participant's gait. While being videotaped, the subject will be requested to walk straight. we will review the footage to assess the participant's walking pattern (Fig 3.8).

These examinations are made to gauge the participant's gait and balance, two factors that are crucial to assessing their general health and mobility. Our approach guarantees the reliability and quality of the data collected, which is essential for producing legitimate study findings.

To assess various gait and balance factors. The 30-second chair stand test (30CST) evaluates the strength and endurance of the lower limbs, which is crucial for actions like walking and getting out of a chair. Both the Four Square Step Test (FSST) and the Performance-Oriented Mobility Assessment (POMA) are mobility and balancing tests that are crucial for total functional mobility and fall prevention. The GAIT speed test measures walking speed, a crucial sign of general mobility and functionality. The Visual GAIT Pattern test, which offers a visual study of the participant's gait pattern, allows researchers to spot any gait irregularities. Overall, these five assessments offer a thorough assessment of several gait-related factors.

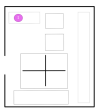


Fig. 3.8: Location for Medial Screening

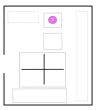


Fig. 3.9: Location for 30CST Test

3.5.2 Statement of the design problem

The design problem for this study is to create a sensor system that can accurately capture both IMU and sEMG data during the GAIT test. The system must be capable of wirelessly transmitting the collected data to a computer for analysis. The primary objective is to gain a better understanding of how mechanical and neuromuscular systems interact during the complicated process of gait. The sensors must be designed to align with body segments and record the motion and orientation of various parts of the body, including stride length, step breadth, and foot clearance. They must also record the electrical impulses produced by contracting muscles to provide information about the timing and pattern of muscle contractions. The design of the sensor system must overcome any technical challenges to ensure accurate and reliable data collection for analysis.

3.5.3 Sensor Considerations

The use of EMG and IMU sensors in the design of the testing procedure is crucial for the study of gait characteristics. These sensors help to capture important information about the body's neuromuscular and mechanical systems during walking. The IMU sensors record the motion and orientation of the body's various parts during walking, providing essential details on gait characteristics such as stride length, step breadth, and foot clearance. On the other hand, the sEMG sensors record the electrical impulses produced by the muscles during walking, giving information about the timing

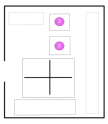


Fig. 3.10: Location for POMA

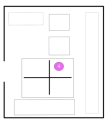


Fig. 3.11: Location for FSST

and pattern of muscle contractions.

By combining the data from both sensors, a more comprehensive understanding of the interaction between the mechanical and neuromuscular systems during walking can be obtained. This data can be used to analyze the effectiveness of therapeutic interventions, evaluate gait abnormalities, and develop more effective rehabilitation strategies. Furthermore, the use of wireless transmission and micro-controller technology ensures accurate and real-time data recording, which is essential for the effectiveness of the testing procedure.

3.5.3.1 IMU Consideration

Basic Principles An IMU (Inertial Measurement Unit) sensor is a device that measures the motion, orientation, and acceleration of an object in three-dimensional space. The basic principle of an IMU sensor is to use the laws of physics and the properties of inertial forces to determine the object's motion.

The IMU sensor consists of three main components: Accelerometers: These sensors measure the linear acceleration of an object in three axes (x, y, z). They use the principle of Newton's second law of motion, which states that the force acting on an object is proportional to its mass and acceleration.

Gyroscopes: These sensors measure the angular velocity of an object in three axes (x, y, z). They use the principle of conservation of angular momentum, which states

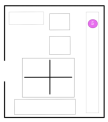


Fig. 3.12: Location for GAIT Speed Test

that the angular momentum of an object remains constant unless acted upon by an external torque.

Magnetometers: These sensors measure the magnetic field strength and direction. They use the principle of the Lorentz force, which states that a magnetic field will exert a force on a charged particle moving through it.

Figure example of an IMU sensor:

The output of an IMU sensor is typically a set of raw data measurements from each

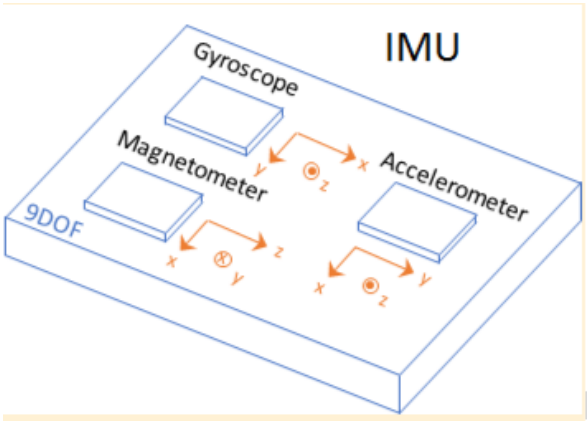


Fig. 3.13: IMU sensor schematic diagram.

of the three components (accelerometers, gyroscopes, magnetometers). The measurements are then processed and fused together using complex algorithms to provide a more accurate estimate of the object's motion and orientation. The mathematical expressions for an IMU sensor can be quite complex, but here are some basic equations that are commonly used: Accelerometer measurement: a = F / m

where a is the linear acceleration, F is the force acting on the object, and m is the mass of the object.

Gyroscope measurement: = / t where is the angular velocity, is the change in angle over a certain period of time, and t is the time interval.

Magnetometer measurement: B = H + b

where B is the magnetic field strength, is the magnetic permeability of free space, H is the magnetic field intensity, and b is the magnetic field bias. Data format and specifications The Arduino BLE Sense is a development board based on the Nordic Semiconductor nRF52840 chipset, with built-in Bluetooth Low Energy (BLE) connectivity and a variety of sensors including an IMU (Inertial Measurement Unit). The IMU on the board includes a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. The data from these sensors can be accessed and processed using the Arduino software environment and various libraries.

The data from the IMU sensors on the Arduino BLE Sense is output in a specific data format and specification. Here are some details about the data format and specification:

Data Format: The data from the IMU sensors is output in a binary format, where each measurement value is represented by a certain number of bits. The specific format depends on the sensor and the range of values it can measure. For example, the accelerometer and gyroscope on the board can measure values from -32,768 to +32,767, and are represented as 16-bit signed integers.

Data Specification: The data specification for the IMU on the Arduino BLE Sense includes the following parameters:

Table 3.1: Sensor Specifications

Sensor	Parameter	Value
Accelerometer	Measurement range	+/- 2g, 4g, 8g, or 16g
	Resolution	16 bits per axis
	Output data rate	13Hz to 104Hz
Gyroscope	Measurement range	+/- 250, 500, 1000, or 2000 degrees per second (dps)
	Resolution	16 bits per axis
	Output data rate	13Hz to 104Hz
Magnetometer	Measurement range	+/- 4900 microtesla (uT)
	Resolution	16 bits per axis
	Output data rate	13Hz to 104Hz

Data Acquisition:

The data from the IMU sensors on the Arduino BLE Sense can be acquired using the built-in sensors library in the Arduino software environment. The library provides a set of functions that allow the user to access the sensor data and configure various settings such as the measurement range and output data rate. The acquired data can then be processed and used for various applications, such as motion tracking and gesture recognition.

Overall, the data format and specification for the IMU on the Arduino BLE Sense provide a detailed set of parameters that allow the user to configure and acquire the sensor data for various applications. The binary format of the data enables efficient transmission and processing of the sensor data, while the various parameters provide flexibility in selecting the appropriate settings for the specific application.

3.5.3.2 sEMG Consideration

Basic Principles Electromyography (EMG) is a technique used to measure the electrical activity produced by skeletal muscles. It is based on the principle that when a muscle contracts, it generates electrical signals called action potentials. These action potentials can be detected by placing electrodes on the skin above the muscle or by inserting electrodes into the muscle itself.

Mathematically, the EMG signal can be represented as a time-varying voltage or potential difference, V(t), measured in millivolts (mV), which is a function of time, t. The signal is typically acquired at a high sampling rate, such as 1000 Hz, and is processed using various signal processing techniques to extract features such as the amplitude, duration, and frequency of the signal.

The basic principle of operation of an EMG sensor is to detect the electrical activity of muscles using surface or needle electrodes. Surface electrodes are non-invasive and are placed on the skin over the muscle of interest, whereas needle electrodes are invasive and are inserted directly into the muscle tissue.

Surface EMG sensors typically consist of two or more electrodes that are placed on the skin over the muscle of interest. The electrodes are connected to an amplifier

that amplifies the electrical signals produced by the muscle. The amplified signal is then filtered to remove noise and other unwanted signals, and is digitized using an analog-to-digital converter. The digitized signal can then be processed using various signal processing techniques to extract features such as the amplitude, duration, and frequency of the signal.

Needle EMG sensors, on the other hand, consist of a fine needle electrode that is inserted directly into the muscle tissue. The electrode is connected to an amplifier that amplifies the electrical signals produced by the muscle. The amplified signal is then filtered to remove noise and other unwanted signals, and is digitized using an analog-to-digital converter. The digitized signal can then be processed using various signal processing techniques to extract features such as the amplitude, duration, and frequency of the signal.

One common application of EMG sensors is in the field of rehabilitation medicine, where they are used to assess muscle function and to monitor the progress of rehabilitation. EMG sensors are also used in sports medicine to assess muscle function and to monitor muscle fatigue during exercise.

Data format and specifications EMG sensor is a bio-signal acquisition device that measures the electrical signals generated by the muscles. The sensor outputs an analog signal that is proportional to the electrical activity of the muscles, which can be read by an analog-to-digital converter or microcontroller.

The output of Seedstudio EMG sensor is a raw analog signal, which can be acquired

Fig. 3.14: SeedStudio EMG sensor.

using an analog-to-digital converter (ADC) or microcontroller. The sensor has an output voltage range of 0-3.3V and can measure electrical activity in the frequency range of 20Hz-1kHz.

The data format of Seedstudio EMG sensor is analog voltage, which is proportional to the electrical activity of the muscles. The voltage output can be read using an ADC or microcontroller and converted to digital data.
The data specification of Seedstudio EMG sensor is as follows:

Table 3.2: Sensor Specifications

Parameter	Value
Output voltage range	0-3.3V
Frequency range	20Hz-1kHz
Noise level	<3mV
Input impedance	>1MΩ
Output impedance	<10kΩ

The sensor requires a supply voltage of 3.3V and has a low power consumption of ¡1mA. It has two electrodes that can be attached to the skin surface of the muscle of interest. The sensor is compatible with the Arduino platform and can be used with various software libraries and tools for signal processing and analysis.

3.5.4 Sensor Network Requirements

The sensors used in the testing procedure are specifically designed to record the body's IMU and sEMG data. Four sensors from each set of sensors are used to record information about motion and muscle activity during the test. Several operating principles are used by the sensors to gather data. Body segment alignment and motion are determined by the IMU sensors using an accelerometer and gyroscope. Electrodes used by the sEMG sensors to measure the electrical impulses produced by contracting muscles. Connected to a micro-controller that stores and processes data will be both sorts of sensors. The data is wirelessly transmitted by the micro-controller to a computer for analysis.
IMU and sEMG data are specifically intended to be captured by the sensors utilised in the testing process. They utilise several operating philosophies and are linked to a micro-controller for the transmission and processing of data.

For a more thorough examination of the subject's gait during the short battery GAIT test, synchronous data from IMU and sEMG sensors must be recorded. The use of these kinds of sensors can help to better understand how these systems interact throughout the complicated process of gait, which involves both mechanical and neuromuscular systems. During walking, the IMU sensors record the motion and orientation of the body's various parts, which can provide crucial details on gait characteristics including stride length, step breadth, and foot clearance. During walking, the sEMG sensors record the electrical impulses produced by the muscles, giving information about the timing and pattern of muscle contractions. It is possible to combine the data from both kinds of sensors [30].

With the help of Arduino BLE sensing nodes, we have created a proprietary data

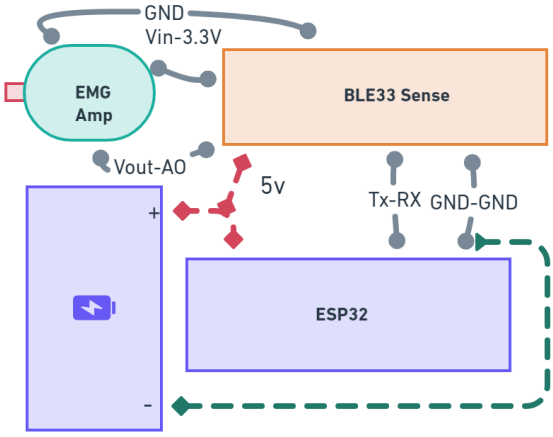


Fig. 3.15: a diagram for single node representation.

collecting protocol that gathers information from both internal and exterior sensors. A Raspberry Pi controls the protocol and gathers data from each node (see fig. 3.13), saving it to a database. Although this protocol has several drawbacks, we have modified the network to overcome these problems. Several applications that call for the simultaneous data collection from multiple nodes can benefit from this protocol. Fig (3.14)

I'll go over a unique data collection protocol that I created. To gather IMU data from internal sensors and an external sensor known as the Seed Grove EMG sensor, the protocol makes use of four Arduino BLE sensing nodes. There are 4 separate

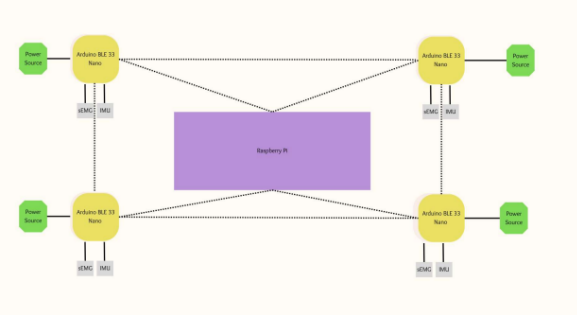


Fig. 3.16: BLE Based sensor network.

nodes in this system, which is referred to as a single node. A Raspberry Pi serves as the system's central coordinator, coordinating the data collection process across all nodes. This system's goal is to simultaneously gather data from each node. The methodology for gathering data entails multiple steps. In Fig. 3.15, the working is summarised as a flow chart, the Raspberry Pi sends a "START" message using its BLE in order to start things off. Each node begins sampling data while listening to this message. The Raspberry Pi broadcasts a "STOP" signal to all nodes after a predefined amount of time. Each node temporarily stores the data. The Raspberry Pi then creates a BLE connection with each node individually and gathers data from each of them. The information is kept in a database. This protocol has significant

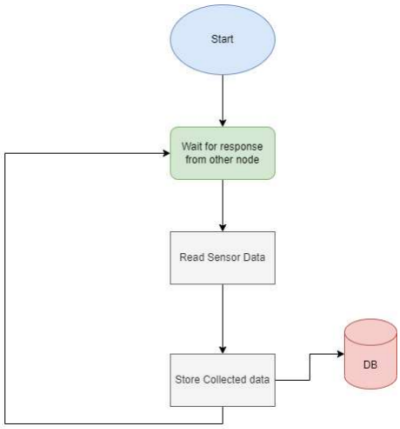


Fig. 3.17: Single node data reading algorithm

drawbacks despite being useful. The Raspberry Pi BLE's unreliability in moving data from node to node is a significant issue. We have adjusted the network in response to

this problem. Bluetooth was used in its place because it offers more dependable data transmission between nodes.

We have upgraded the network by switching from data transfer via Bluetooth to a Wi-

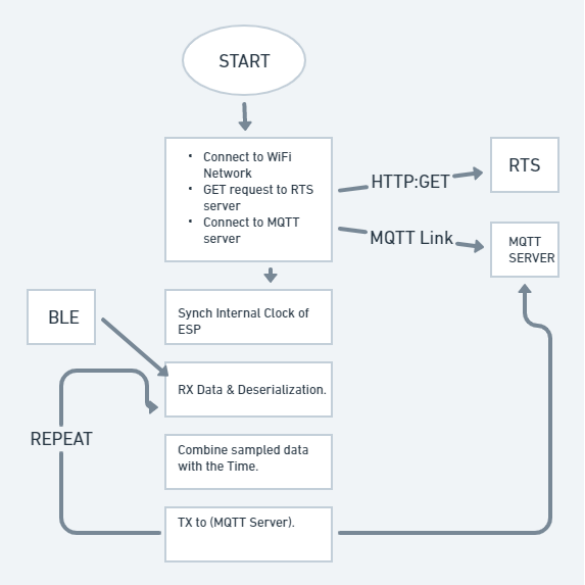


Fig. 3.18: Modified MQTT Based algorithm.

Fi network because the BLE connection was unreliable. The MQTT protocol is now being used to control data transport. A central Raspberry Pi serving as the MQTT server is connected to each of the ESP32 nodes in the enhanced network through a Wi-Fi antenna. A Python script on the Raspberry Pi subscribes to many topics where each node publishes its data. When an ESP32 node starts up, it connects to the Wi-Fi network first before requesting the NTP server to obtain precise time information (see Fig. 3.15). We could have utilised the hardware external clock since the ESP32's internal clock may be reset; but, to save hardware complexity, we used an NTP server that can give us the necessary time data to manage between the nodes. When we sample data, each node resets its clock at the same moment, and this allows us to manage the synchrony between the nodes by attaching the time information to each sample. Each MQTT node establishes a wireless connection to the server as soon as the ESP32's internal clock is reset. The node then creates a UART connection to BLE in the following phase, which is transferring a JSON serialised object comprising the data for I, Am, You, and EMG. After being received, the data is deserialized. At

the moment of deserialization, the data is integrated with the time information and published to the appropriate topic. I've included an algorithmic chart below Fig 3.17 to help you understand the process, and the figure also shows how the sensor node functions as a whole. We added Apache Kafka as a component to this sensor network to improve its functionality even more. As a streaming device, each node streams data to an Apache Kafka client, which then publishes it to a Kafka broker. This enables us to remotely record the data from any location. Real-time data pipelines and streaming applications can be created using Apache Kafka, a distributed streaming platform. It can manage fault-tolerant, scalable, high-throughput data streaming. We are able to stream data from each node of our sensor network to a central location using Apache Kafka, which makes it simpler to manage and analyse the data.

Receiving the data stream from each sensor node and publishing it to a Kafka broker is the responsibility of the Apache Kafka client. The data is subsequently stored as topics by the Kafka broker, which are used to separate and arrange the data. This makes it simple to retrieve and analyse the data. We are able to store and analyse

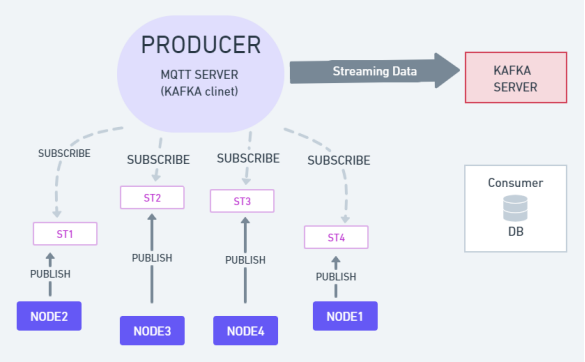


Fig. 3.19: Overall flow of data via node to the MQTT Broker

data in real-time using Apache Kafka in our sensor network, which makes it simpler to spot trends and anomalies in the data. This makes it a potent tool for sensor node applications since it permits remote monitoring and data processing.

3.5.5 High Level Design of the Training architecture.

The High-Level Design (HLD) for a data collection, processing, and modelling system that will be installed on an Arduino BLE sensor is described in this paper. A YAML-configured Model is used to represent the data after it has been processed using a combination of MQTT, Kafka, MongoDB, pandas, and numpy. The system is built to collect data from a bespoke sensor.

For ease of understanding, the HLD is shown below in a point-by-point style and in Fig. 3.18 as a wireframe:

- A custom data collection sensor is developed to collect data.
- The collected data is published to an MQTT broker.
- A Raspberry Pi subscribes to the MQTT broker and collects the data.
- The Raspberry Pi acts as a Kafka producer and transmits the data to the Kafka Broker (Confluent).
- A consumer script running on IIT Mandi HPC consumes the data and saves it onto the MongoDB database.
- The schema of the data is predefined, and schema validation is performed during data ingestion.
- The data is loaded from the MongoDB and converted into a pandas DataFrame by the Data Ingestion script.
- Separate files for the train and test data are created and saved as numpy object files.
- The Model is selected and configured using the model.yaml file.
- Once selected, the Model is optimized for deployment onto an Arduino BLE sensor.

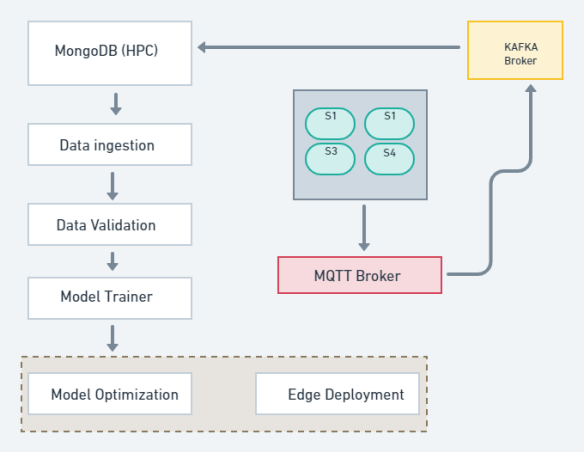


Fig. 3.20: High level design

Chapter 4

Model Training Results.

4.1 The need of deep learning

Traditional methods for fall risk [31] assessment using electromyography (EMG) and inertial measurement unit (IMU) sensors often rely on simple threshold-based rules that do not take into account the complexity and variability of human movement patterns. These methods are typically based on static and simplistic features such as muscle activation and joint angles, which may not capture the subtle changes and dynamic interactions that are indicative of fall risk. On the other hand, deep learning methods can automatically learn complex patterns and features from EMG and IMU sensor data. They can extract hidden and non-linear relationships between variables, and adapt to individual differences, making them more personalized and accurate. Moreover, deep learning models can handle missing or noisy data and can automatically adjust their parameters during training to optimize performance. Therefore, the need for deep learning methods in fall risk assessment using EMG and IMU sensors lies in their ability to capture complex and dynamic patterns of muscle activation and movement, and to provide accurate and personalized predictions based on multiple sources of sensor data. These methods have the potential to improve the accuracy and reliability of fall risk assessment, leading to better prevention and management of falls in the elderly and other populations at risk [32]. Some examples of state-of-the-art deep learning models for fall risk assessment using IMU and EMG sensors: The recent

studies have shown that DL approaches can achieve very good outcomes.

4.1.1 Exploring the IMU sensor data to be feed into various and ML and DL algorithm

This section contains the results of the data pre-processing and contains plots for each stage. The plots shown are for user id = 2, recording session = 3, and each sensor placement from the ZJU-dataset. The plot below (Fig 4.1) shows the segmentation stage for the unprocessed 3 channel input. As it is evident from the starting and ending parts of the data, there is not significant change. Therefore, after clipping these portions, we have the processed signal in Fig 4.1 (b). Fig 4.1 (c) shows the cumulative acceleration 3D plot, which indicates that there is significant movement in only the x direction, which makes sense as we move our hand to and fro while we are walking.

In Fig 4.2, after segmentation, the accelerometer data is converted into a spectrogram, where the y-axis represents frequency in Hz and the x-axis represents time in seconds. Fig 4.2 (a), (b), and (c) are the spectrogram plots for the processed signal. These plots will either be stacked or concatenated and fed into the neural network.

Fig 4.3 and Fig 4.4 show the same signal processing process, but for the sensor placed on the left uppermost arm. Fig 4.5 and Fig 4.6 show the same signal processing process, but for the sensor placed on the right side of the pelvis. Similarly, the plots from Fig 4.7 to Fig 4.10 are for the body locations of the left thigh and right ankle.

4.2 Synthesising Raw Data

For the analysis dicussed in this section we don't have the data for the abnormal gait patients. So we have to use the existing method of adding noise to the raw dataset so that we can train our model. Here is the mathematical explanation for each of the noise that i have used to synthesise raw IMU data. Gaussian noise [33]: The mathematical expression for adding Gaussian noise to the IMU data can be expressed as:

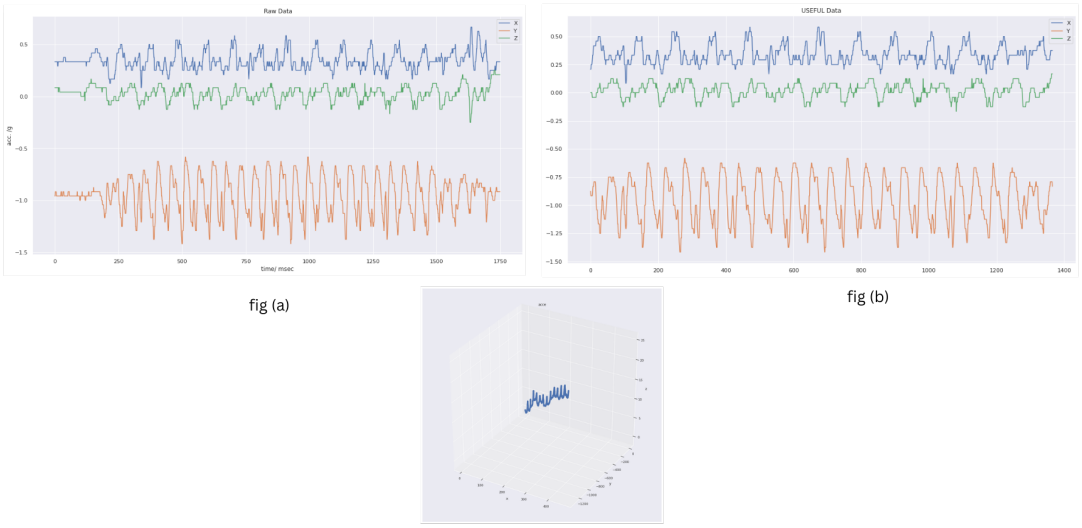


fig (a) fig (b)

fig (c)

*fig. For **the Right waist** - (a) Raw input accelerometer reading*
(b) Removed unwanted part, (c) cumulative acceleration x,y & z direction 3d plot

Fig. 4.1:

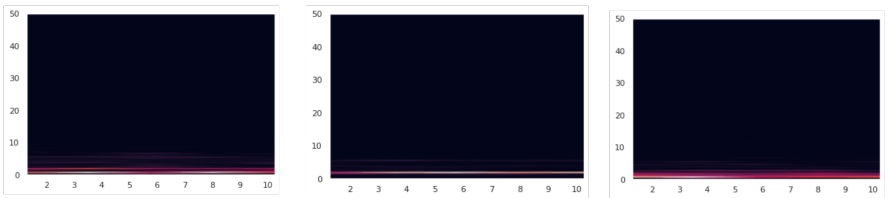


fig (a) fig (b) fig (c)

*fig. For **the Right waist** - (a) ,(b) & (c) are the spectrogram of the processed signal*
in x, y & z respectively

Fig. 4.2:

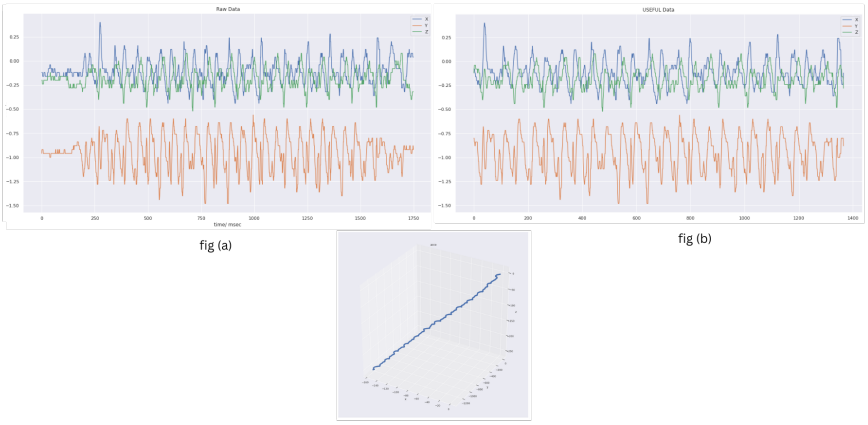


fig (a) fig (b)

fig (c)

*fig: For **the left upper arm** - (a) Raw input accelerometer reading*
(b) Removed unwanted part, (c) cumulative acceleration x,y & z direction 3d plot

Fig. 4.3:

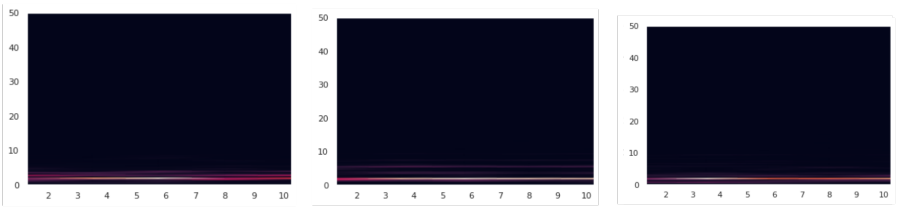


fig (a) fig (b) fig (c)

*fig: For **the left upper arm** - (a) ,(b) & (c) are the spectrogram of the processed signal in x, y & z respectively*

Fig. 4.4:

$$Y = X + \epsilon$$

Where:

X is the original IMU data Y is the new data with added noise ϵ is the noise generated from a Gaussian distribution with a mean of zero and a standard deviation of σ Thus, Y is obtained by adding random values drawn from a Gaussian distribution with a mean of 0 and a standard deviation of σ to the original data X.

Random noise [34]: The mathematical expression for adding random noise to the IMU data can be expressed as:

$$Y = X + \epsilon$$

Where:

X is the original IMU data Y is the new data with added noise ϵ is the noise generated from a uniform distribution between the values of a and b Thus, Y is obtained by adding random values drawn from a uniform distribution between the values of a and b to the original data X.

Outliers [35]: The mathematical expression for adding outliers to the IMU data can be expressed as:

$$Y = X + \epsilon$$

Where:

X is the original IMU data Y is the new data with added outliers ϵ is the noise generated from a normal distribution with a mean of zero and a standard deviation of σ multiplied by a factor k Thus, Y is obtained by adding random values drawn from a normal distribution with a mean of 0 and a standard deviation of σ multiplied by a factor k to the original data X.

signal interference[36]: The mathematical expression for adding signal interference to the IMU data can be expressed as:

$$Y = X + \beta$$

Where:

X is the original IMU data Y is the new data with added interference β is the interference data multiplied by a factor k Thus, Y is obtained by adding the interference data β multiplied by a factor k to the original data X.

Signal dropouts [37]: The mathematical expression for adding signal dropouts to the IMU data can be expressed as:

$$Y = X \cdot m$$

Where: X is the original IMU data Y is the new data with added dropouts m is a mask generated from a Bernoulli distribution with probability p Thus, Y is obtained by multiplying the original data X with a mask m generated from a Bernoulli distribution with probability p. The mask m is a binary vector that has the same length as the original data X and contains 1s and 0s, where 1s indicate that the corresponding data point should be retained and 0s indicate that it should be dropped out. (See Fig 4.5) for the raw input and output and the Synthetic data.

4.3 Exploratory Data Anaylsis on IMU Data

The dataset includes information gathered from a smartphone's accelerometers and gyroscopes while a person engaged in six different physical activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying) [38]. Below plot shows the activities counts.

Number of Data-points per activities.

For these plots it is evident that we have pretty much good distribution per activity. We visualised and examined IMU data using t-SNE (using the tsne dimensional reduction we would be are to reduce the data size and make less complex model which would otherwise require more computation). We may better grasp the underlying structure

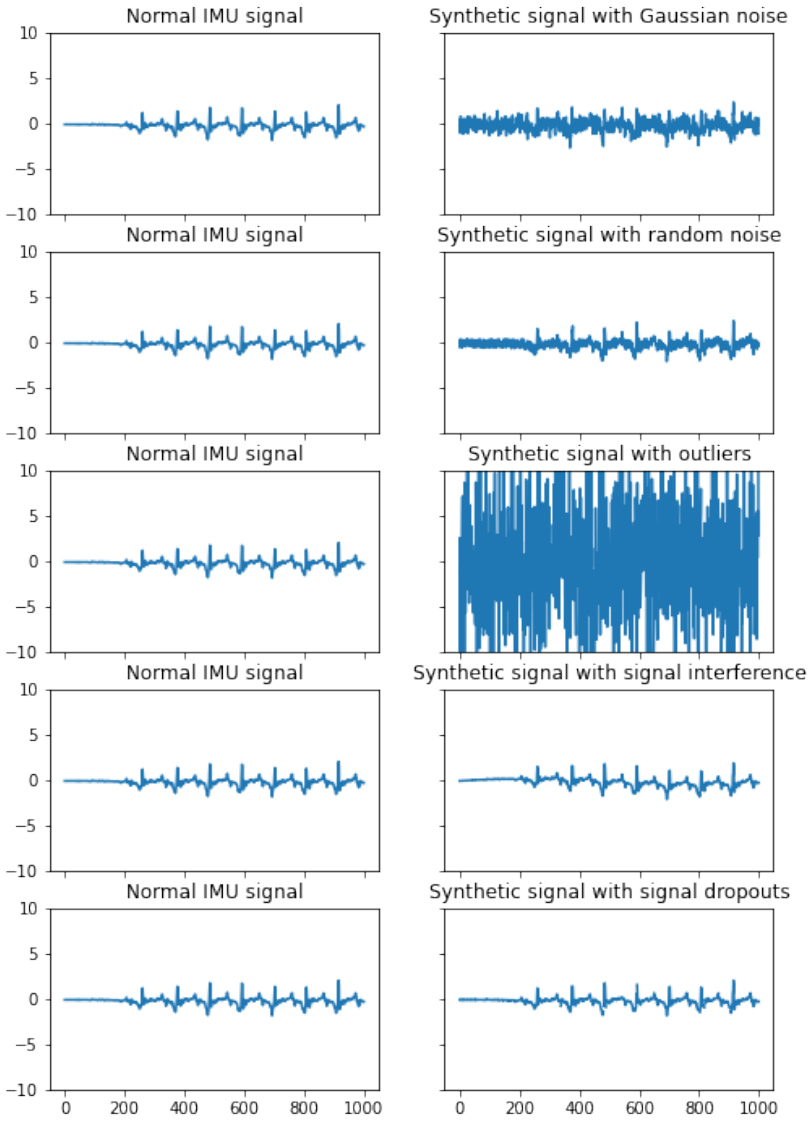


Fig. 4.5: The above plots are taken from the same healthy individual. Only the x-axis data is Synthesis for the easy representation.

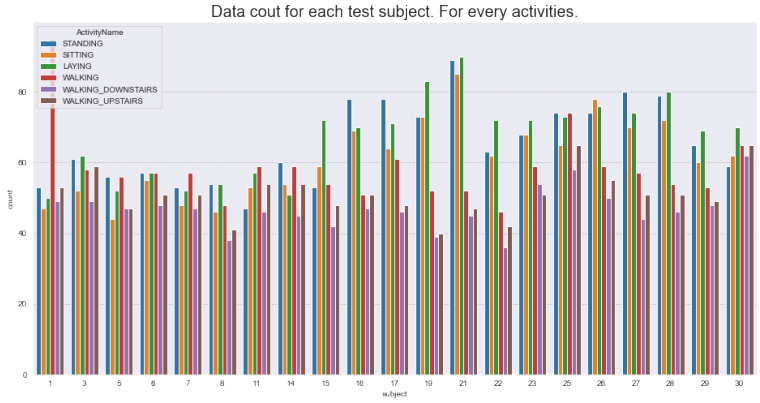


Fig. 4.6: Actvity counts by the people.

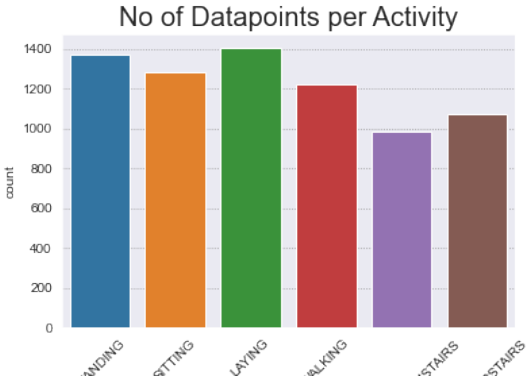


Fig. 4.7: Activity counts

of the data and spot patterns and clusters in the data by translating high-dimensional IMU data into a lower-dimensional space using t-SNE. These patterns and clusters might not be visible in higher dimensions. To better understand the context, I have adjusted the perplexity value. When trying to generate a map of all the data points, the method uses a parameter called perplexity to determine how many neighbours to consider. It may include too many neighbours and provide a confused map if the perplexity is too high. Yet, if it is too low, certain significant relationships between data points may be missed. So finding the right perplexity value is important to make a good map that shows the relationships between all the data points.

Fig. 4.8: with the low perplexity the clustest are very much close.

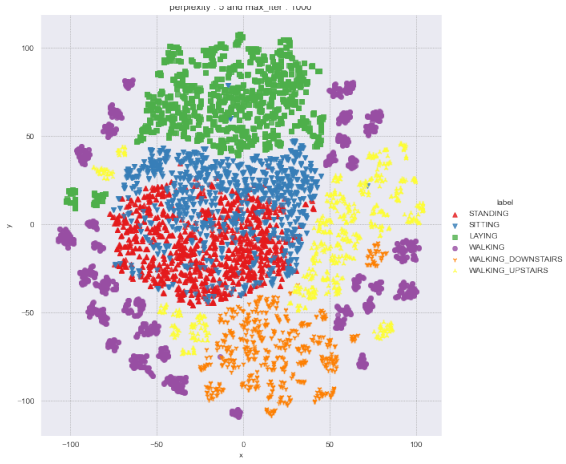


Fig. 4.9: The formation of similar clusters needs to be enhanced or augmented.

Fig. 4.10:

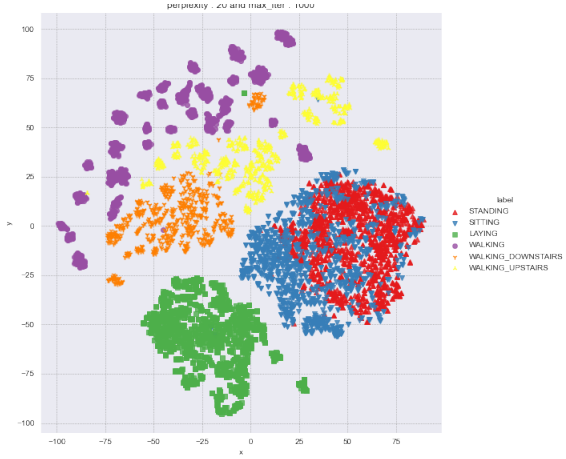


Fig. 4.11: Now, the non-static and static activities are becoming distinct and separating.

Fig. 4.12: Clear cluster have been formed with the perplexity of 50

4.4 Model Training Results

From the previous section where we have seen that the non static activities are forming separate cluster. It would be better to remove the non static activities for training the network and ML models. Five classification algorithms were applied to the data: Logistic Regression, Linear SVC, rbf SVM classifier, DecisionTree, Random Forest, and GradientBoosting DT. The accuracy and error rates of each algorithm were calculated using 3-fold cross-validation.

With ML Models

It is likely that the performance of the Conv-Auto Encoder and RBF-SVM models

Table 4.1: Accuracy Results

Model	Train Accuracy	Test Accuracy
Logistic Regression	86.3	71.699
Linear SVC	86.5	73.495
rbf SVM classifier	86.27	76.733
DecisionTree	76.39	61.61
Random Forest	81.08	70.924
GradientBoosting DT	81.08	72.924
DL MODEL		
CONVOLITION AUTO-ENCODER	86.31	76.41

will change as we continue to train them. The effectiveness of a machine learning model is heavily influenced by the quality of the training data and the model's hyper-parameters.

Moving forward with the convolutional autoencoder.Here is the list of parmater that it has been tuned.

Table 4.2: Trainable Parameters

Parameter list	
Filter size	[28, 32, 42]
Kernel size	[3, 5, 7]
Dropout rate	[0.45, 0.7]
Pool size	2
Optimizer	Adam & RMSprop
Learning rate	[0.00065, 0.004]
Epochs	25 to 35
Batch size	[16, 32, 64]

4.5 Autoencoder based model for fall risk assessment

An autoencoder is a type of neural network that has proven to be useful in dimensionality reduction, unsupervised learning, anomaly detection, and generative modeling. In this article, we provide a comprehensive introduction to autoencoders, including their architecture, training, applications, and advantages over traditional machine learning techniques. We also highlight some of the latest research on autoencoders and their potential for future developments.

The architecture of an autoencoder is composed of two main parts: the encoder and the decoder. The encoder maps the input data to a compressed representation, while the decoder maps the compressed representation back to the original data. The compressed representation is usually called the "latent space," "code," or "bottleneck" layer.

4.5.1 Training of Autoencoder

The training of an autoencoder is an unsupervised learning process. In other words, the model learns to reconstruct the input data without any labels or supervision. The objective of training is to minimize the difference between the input data and the reconstructed data, also known as the reconstruction error.

The reconstruction error is commonly measured using mean squared error (MSE) or binary cross-entropy (BCE) loss functions. The loss function measures the difference between the input data and the output of the decoder. The training process adjusts the parameters of the encoder and decoder using backpropagation and gradient descent algorithms to minimize the loss function.

4.6 Why Empasise on the Autoencode

While using autoencoder models for fall risk assessment or GAIT classification has shown promising results, it is important to consider other techniques as well. Machine learning (ML) approaches such as support vector machines, decision trees, or random forests can also be used for the same tasks. These ML techniques can provide interpretable models and can handle missing data more effectively. Furthermore, deep learning models like convolutional neural networks (CNNs) can also be used for feature extraction from raw sensor data, which is similar to autoencoder models.

However, one major advantage of autoencoder models over other techniques is that they can learn a compressed representation of the input data, which can be more robust and less prone to noise or missing data. In addition, autoencoder models do not require any prior knowledge of the data distribution or feature engineering, which makes them more objective and efficient. The learned features in autoencoder models can also be used for transfer learning to other related tasks, which can save time and resources.

In conclusion, while there are other techniques available for fall risk assessment or GAIT classification, the use of autoencoder models has several advantages over them. It is important to consider the specific requirements and limitations of the task at hand before deciding on the appropriate technique. Since there have been studies shown that using an autoencoder model can be deployed onto the edge device hence the autoencoder model is preferred.

4.7 Normal and Abnormal Gait pattern based on reconstruction error.

4.7.1 The Mobinet and MobiFall dataset

The Mobinet and MobiFall datasets [39] are valuable resources in the field of fall detection and human activity recognition. It includes accelerometer data captured from

waist-mounted smartphones during controlled experiments involving various simulated falls. The dataset encompasses different types of falls, non-fall activities, and post-fall motions, allowing for comprehensive fall detection algorithm evaluation. These datasets have become widely used benchmarks for developing and evaluating fall detection algorithms and models. Researchers leverage the labeled sensor data from both datasets to train and validate their machine learning models, enabling the development of accurate and robust fall detection systems. The availability of such datasets contributes significantly to advancing the field of fall detection, facilitating the development of innovative approaches and improving the overall performance of fall detection systems.

4.7.2 Data Pre processing.

Extracting data from open-source datasets like MobiFall and MobiAct can be a challenging task due to various factors. These datasets are valuable resources for studying human activity recognition and fall detection, but they often require significant effort and time to obtain the desired data. The MobiFall dataset, for example, contains accelerometer and gyroscope sensor readings captured during simulated falls and activities of daily living. Accessing and processing this raw sensor data requires careful data extraction techniques to ensure accurate and meaningful results.
Here is the preprocessing steps for the dataset in the provided code involve the following:

1. Iterate over the files in the "Normal" directory using the `os.listdir()` function.
2. Filter the files based on the presence of "acc" in the filename.
3. Create an empty list to store the chunk DataFrames.
4. Open each file and read its lines.
5. Find the index of the line containing "@DATA".
6. Extract the lines after the "@DATA" line.

7. Remove leading and trailing whitespace from each line.

8. Parse and store the accelerometer data from each line into the `normal_data` list.

9. Define the sampling rate for the accelerometer data (which was 50 Hz).

10. Specify the duration of each chunk in seconds (which was 10 seconds).

11. Calculate the number of samples in each chunk based on the sampling rate and chunk duration.

12. Determine the total number of chunks by dividing the total number of samples by the number of samples per chunk.

13. Extract each chunk of data from the `normal_data` list and append it to the `chunk_dfs` list.

14. Serialize and save the `chunk_dfs` list into a binary file named "positive.pkl" using the `pickle.dump()` function.

Fig 4.13 plots showcases the variations in accelerometer readings along different axes (X, Y, Z) over time for the Normal and Fall incidents. It allows us to observe the periodic nature of the walking motion and the synchronization between the three axes. The plot provides a valuable visual reference for understanding the sensor data distribution and identifying any potential outliers or abnormalities in the dataset.

4.7.3 Model training

The data-set have 6586 instances of normal GAIT accelerometer reading and 69 instances of Abnormal GAIT reading. For training the auto-encoder model the time series data has been feed into the network with the tensor shape of (1,500,3). First of all the dataset is normalized by subtracting the mean and dividing by the standard deviation. It is then split into training and validation data using a test size of 20% and a random state is set for the reproducibility.

The model architecture consists of several Conv1D layers, each with different numbers of filters and activation functions. The input shape is set to (500, 3), indicating

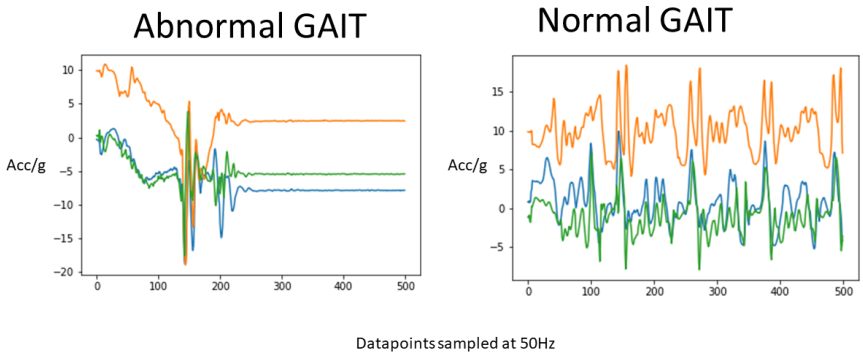


Fig. 4.13: Example plots from two GAIT cycle. Using MobiAct and Mobifall dataset.

the dimensions of the input data. Conv1DTranspose layers are also included for decoding the input data. The model is compiled with the Adam optimizer and mean squared error (MSE) loss function.Below is shown the Model architecture. After train-

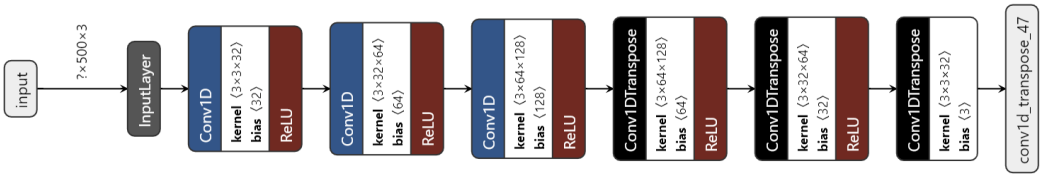


Fig. 4.14: Model Architecture

ing the model for 100 epochs with a batch size of 32, the loss history is collected. The training and validation loss curves are then plotted using matplotlib. This allows visualizing the model's performance over the training process, indicating if the model is learning and if there is any overfitting or underfitting. The plot provides insights into the convergence and generalization of the model. The loss curve serves as a useful tool for monitoring the training progress and assessing the model's performance. It helps identify if the model is improving or if adjustments need to be made, such as modifying the model architecture, changing hyperparameters, or increasing the number of training epochs. The trained model is not optimized for deploying onto the Microcontroller as it memory footprint is 12 MegaBytes and we only have 256KB RAM for

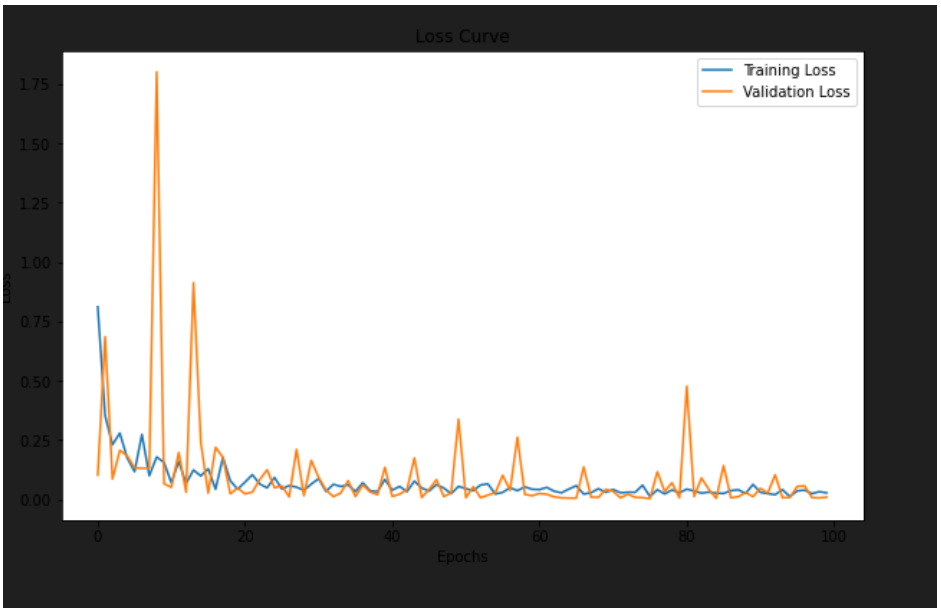


Fig. 4.15: Loss Curve

miking the inferences. So further down this section there is the detailed explanation for optimization of the model.

4.7.4 Estimating the Number of bit for the Quantization- First step of TinyML toward the model optimization.

Estimating the number of bits for quantization is a crucial step in optimizing models for TinyML. By reducing the number of bits used to represent the weights and biases of a model, we can significantly reduce the model's size and memory requirements, making it more suitable for deployment on resource-constrained devices.

Quantization refers to the process of representing numerical values with a limited number of bits. In the context of model optimization, we can quantize the weights and biases of a neural network by reducing their precision. Instead of using floating-point values, we can represent them using fixed-point or integer values with fewer bits.

Mathematically, let's consider a weight or bias value, w, originally represented by n bits. By quantizing it to m bits (m is less then n), we can reduce its memory footprint. The quantize value, q, can be expressed as:

$q = \text{round}(w \cdot 2^b)_{\frac{}{2^b}}$

Here, b represents the number of fractional bits, which determines the precision of the quantized value. By reducing the number of fractional bits, we can further decrease the memory requirements. However, it's important to strike a balance between reducing the model size and maintaining an acceptable level of accuracy and performance. The process of estimating the number of bits for quantization involves evaluating the sensitivity of the model's performance to the reduced precision. By systematically quantizing the weights and biases using different numbers of bits and evaluating the resulting model's accuracy, we can determine the minimum number of bits that can be used without significantly impacting the performance.

Overall, estimating the number of bits for quantization is a fundamental step in TinyML model optimization. By reducing the size of weights and biases through quantization, we can effectively reduce the memory requirements and make models more suitable for deployment on edge devices with limited resources. By analyzing the weight and biases of the trained model it is found that 8 bit can represent the model weight and biases.See the histogram plot of the model weight and biases below.

After this step the model size is shrunk down to 2.56MB which was around 17.8 MB.

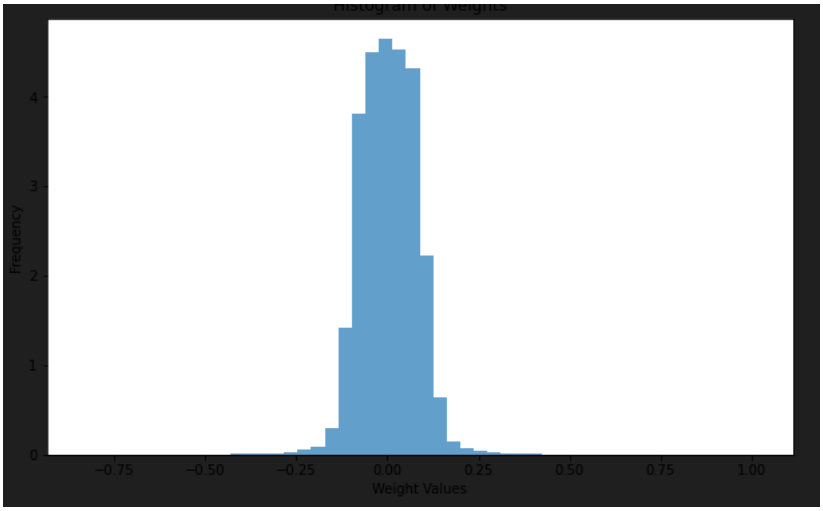


Fig. 4.16: Weight Histogram plot

Further using the Pruning we will try to reduce the model size.

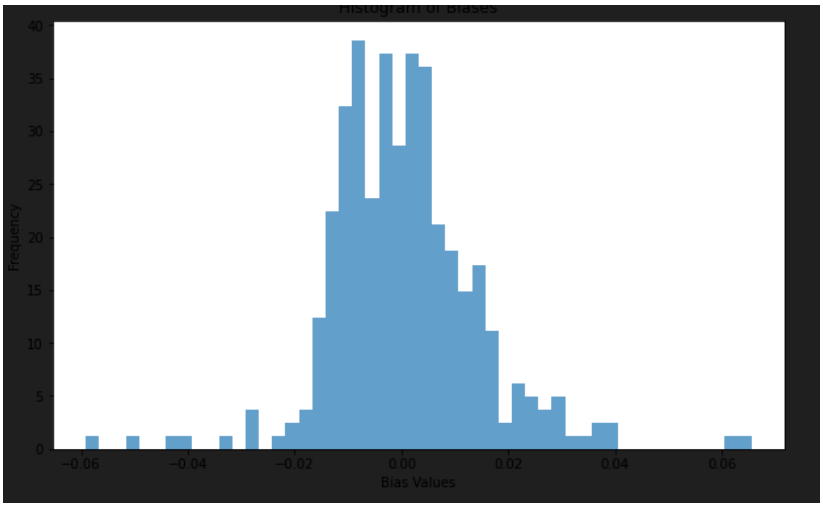


Fig. 4.17: Bias Histogram plot

4.7.5 Pruning - removing the redundant connection

Pruning is a process of eliminating unnecessary connections or parameters from the model. By identifying and removing redundant or less influential connections, the model's size is reduced without significant loss in performance. Pruning helps in further compressing the model, resulting in lower memory usage and faster inference times.

Implementation wise this step was quit easy as tensorflow lite micro provides the API

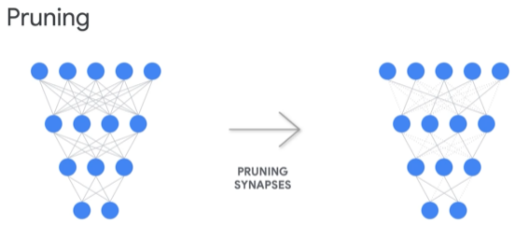


Fig. 4.18: Pruning of Deep Learning Model

for pruning model. Here is the detailed sequence steps for pruning the model.

1. Train the model on a larger dataset to ensure good accuracy and performance.

2. Use the trained model as a starting point for pruning.

3. Apply a pruning algorithm, such as magnitude-based pruning, to identify and remove the least important weights and biases in the model.

4. Fine-tune the pruned model to recover any loss in accuracy caused by pruning.

5. Quantize the pruned model to further reduce the model size and improve inference speed. This involves representing the weights and biases with lower precision, such as 8-bit or even fewer bits.

6. Convert the pruned and quantized model to the TensorFlow Lite Micro format, which is optimized for running on micro controllers and other resource-constrained devices.

7. Deploy the pruned and quantize model on the target device, ensuring that it meets the memory and computational constraints.

8. Validate the performance of the pruned and quantize model on the target device and fine-tune further if necessary.

This is iterative process and can be done until the results are not improving or the desired result are achieved.

4.7.6 Quantize aware Training

. Quantized aware training is a training technique that enables the model to handle low-precision representations during the training process. It involves simulating the quantization process during training, allowing the model to adapt and maintain performance even with reduced precision. This step ensures that the model can effectively learn and generalize using quantized representations, which is crucial for accurate fall risk assessment on edge devices.
For this also TensorFlow lite micro provide the library support.
Here is the plot for error plot during the QAT.
Here are some plots depicting the reconstruction of normal and abnormal gait pat-

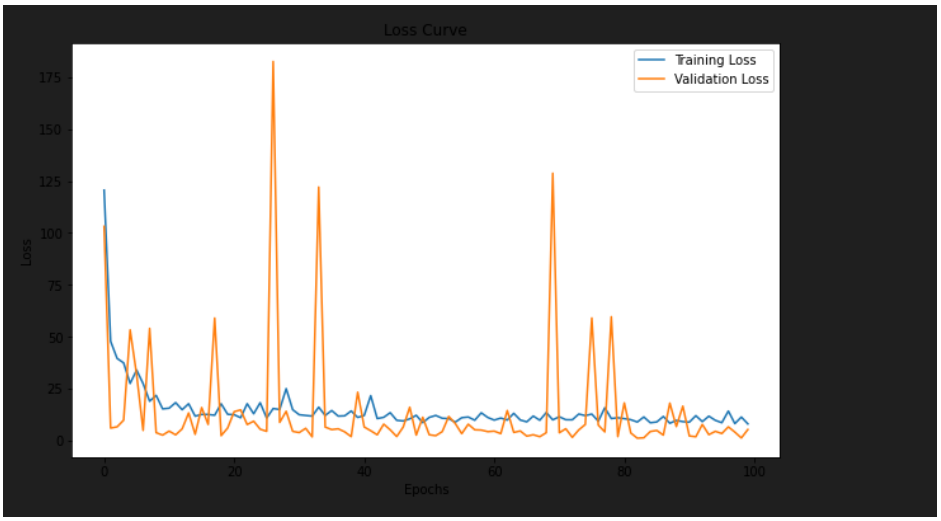


Fig. 4.19: QAT with the embedded emulated tf engine.

terns. It is evident from the plots that the abnormal pattern exhibits a significantly larger difference between the original and reconstructed data compared to the normal pattern. This observation aligns with the initial hypothesis made at the beginning of the study. The hypothesis stated that by training an autoencoder solely on normal data, we would be able to discern and differentiate abnormal gait patterns.

The plots visually demonstrate the effectiveness of the trained autoencoder in capturing the distinguishing characteristics of normal gait patterns. When presented with abnormal gait patterns during the reconstruction process, the autoencoder struggles to reproduce the data accurately, resulting in a noticeable discrepancy between the original and reconstructed samples.

These findings indicate that the auto-encoder has learned to model the normal gait pattern effectively, enabling it to identify deviations from this norm. Therefore, it holds promise for detecting abnormal gait patterns based on the discrepancies observed in the reconstructed data.

The plots serve as evidence supporting the hypothesis that training the auto-encoder exclusively on normal data enables it to differentiate abnormal gait patterns effectively. Further analysis and evaluation can be conducted to assess the model's performance and its potential application in gait anomaly detection and monitoring.

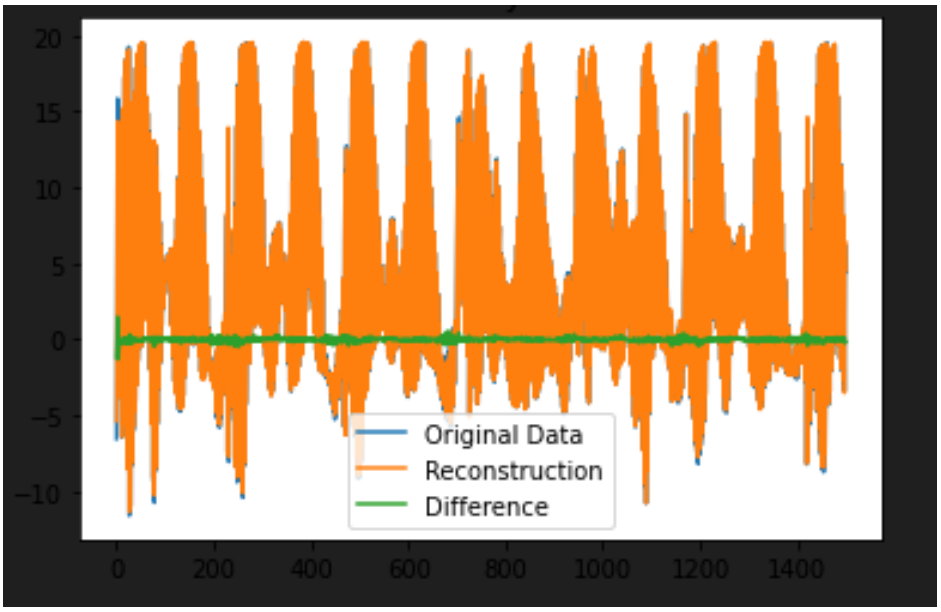


Fig. 4.20: Normal GAIT Pattern reconstruction

4.7.7 The Threshold - How to take the decision.

The statistical way (Current approach) intuition behind setting the threshold for anomaly detection using mean squared error (MSE) involves utilizing the distribution characteristics of the MSE values.

The first step is to calculate the MSE for each data point in the test set. MSE represents the average of the squared differences between the original data and their reconstructions. By squaring the differences, larger deviations are emphasized and smaller ones are penalized.

Next, the mean of the MSE values is computed. The mean is a measure of central tendency that provides an estimate of the "typical" reconstruction error. It represents the average discrepancy between the original and reconstructed data.

To determine the threshold, the standard deviation of the MSE values is added to the mean. The standard deviation measures the dispersion or spread of the MSE values around the mean. By adding the standard deviation to the mean, the threshold is expanded to encompass a range of reconstruction errors that may be considered anomalous.

The threshold acts as a decision boundary. Any MSE value above the threshold is

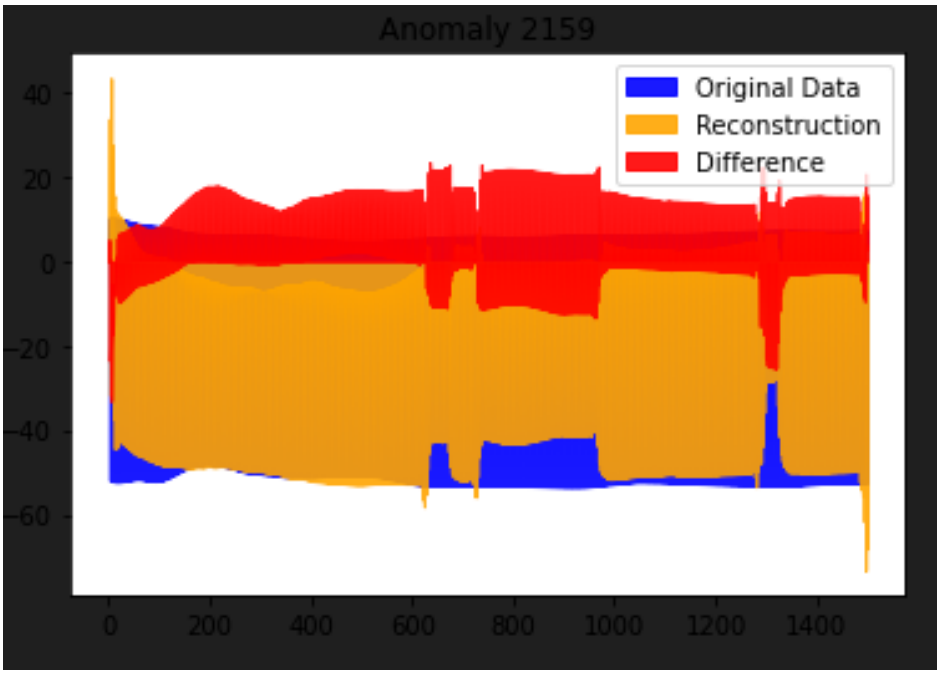


Fig. 4.21: Abnormal GAIT pattern reconstruction.(Falling backward case)

considered an anomaly, indicating a significant deviation between the original and reconstructed data. By setting the threshold based on the mean and standard deviation, a statistical criterion is established to classify anomalies.

This statistical approach leverages the properties of the MSE distribution to identify

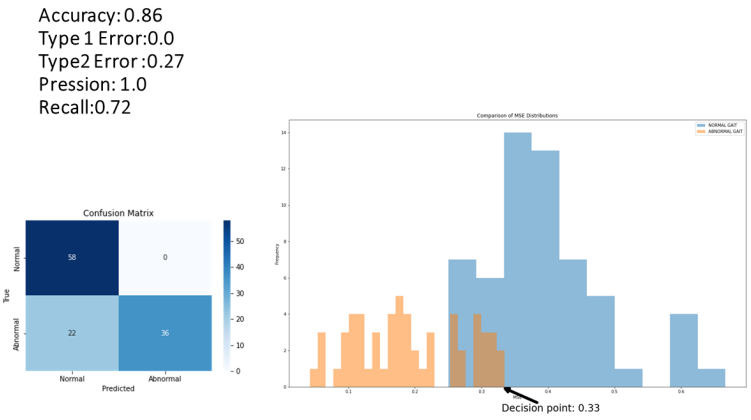


Fig. 4.22: Error distribution of normal and Anomalous GAIT pattern. And classification is based on the threshold level = 0.33

anomalies in the data. It takes into account both the average reconstruction error and the variability of the errors. By comparing the MSE values to the threshold, instances

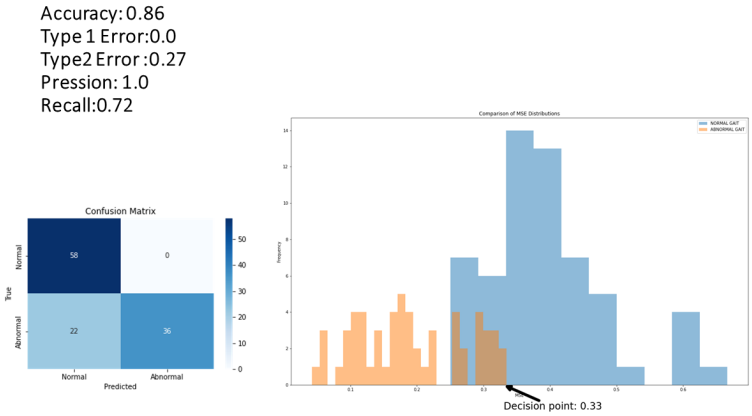


Fig. 4.23: Error distribution of normal and Anomalous GAIT pattern. And classification is based on the threshold level = 0.29

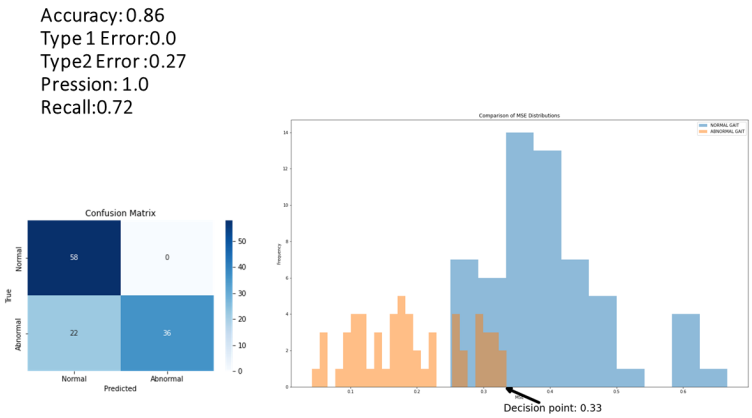


Fig. 4.24: Error distribution of normal and Anomalous GAIT pattern. And classification is based on the threshold level = 0.24

where the reconstruction error exceeds the expected range can be detected, indicating the presence of anomalies.

4.8 What has not been achieved and how to achieve them

In this section, I will attempt to provide an explanation for not being able to achieve the objective step by step. It is important to note that this introductory section was

primarily written in the first person perspective in order to establish a connection with the reader and to Shed light on the challenges I have encountered. Despite these difficulties, I am humbly continuing to advance the field by making the most of the available resources.

To achieve ultimate objective of this project as discussed in the introduction chapter, I planned to create a custom sensor that could gather sEMG and IMU (inertial measurement unit) data at the same time. However, due to time and resource limitations, I couldn't complete the sensor, particularly the printed circuit board (PCB) part. Instead, I decided to use an open-source data-set that contains information on both normal and abnormal gait patterns. With this data-set, I trained a model to predict normal and abnormal gait patterns.

The importance of this work lies in understanding and analyzing human movement patterns, specifically in relation to gait abnormalities. Gait analysis is essential in areas like biomechanics, rehabilitation, and clinical diagnostics. By studying sEMG signals, which give insights into muscle activity, along with IMU data that captures motion dynamics, we can gain a comprehensive understanding of gait patterns and identify abnormalities.

Although it would have been ideal to collect real-time data using the custom sensor, using an open-source dataset provides a practical alternative that allows me to continue the research without compromising its validity. By utilizing this dataset, I can train a model to predict and classify normal and abnormal gait patterns based on the data from the sEMG and IMU sensors. This approach ensures that the work maintains scientific rigor and contributes to the advancement of the field.

It's important to note that while the absence of a fully functional custom sensor may be seen as a limitation, my ability to adapt and find alternative solutions demonstrates resourcefulness and flexibility in dealing with constraints. Additionally, using an open-source dataset encourages collaboration and enables the validation of findings by different research groups, fostering a collective effort to enhance gait analysis

methodologies.

The significance of this work lies in understanding and analyzing gait abnormalities through the examination of sEMG and IMU data. Despite not being able to collect real-time data due to the incomplete custom sensor, the use of an open-source dataset ensures the continuation of the research and the development of a predictive model for normal and abnormal gait patterns. By focusing on these objectives, this project contributes to the broader field of gait analysis and its potential impact on various areas such as biomechanics, rehabilitation, and clinical diagnostics.

4.8.1 Collecting the Real world data of true positive

Collecting a comprehensive data set from the older generation of people poses several challenges, yet their data remains crucial for detecting muscular abnormalities using an eEMG sensor.

Moreover, the older population may encounter physical limitations that affect their ability to wear or operate the eEMG sensor effectively. Arthritis, limited dexterity, or other age-related conditions can hinder their cooperation during the data collection process. Researchers and technicians need to adapt and accommodate these limitations by designing user-friendly interfaces, simplifying instructions, or even providing physical assistance to ensure accurate data acquisition. And that is just making the sensor does not mean that we have everything to collect the data from older generation.

In addition, the older generation may have concerns regarding the purpose and potential benefits of sharing their data. It is crucial to establish trust and transparency by explaining the significance of their data in advancing medical research and improving the detection and treatment of muscular abnormalities. Assurances regarding data privacy, confidentiality, and anonymization are vital to alleviate their reservations. This infrastructure is not available for this project even till now.

Despite these challenges, collecting data from the older generation is of paramount importance for detecting muscular abnormalities using eEMG sensors. The human body undergoes various changes as it ages, making the data from older individuals indispensable for understanding age-related muscle degeneration, identifying patterns

of abnormalities, and refining diagnostic algorithms. The older population's data provides valuable insights into the effects of aging on muscle function, aiding in the development of targeted interventions, rehabilitation strategies, and preventive measures. Furthermore, the prevalence of muscular abnormalities tends to increase with age, making the older generation a high-risk group for such conditions. By including their data in research studies, scientists can enhance the accuracy and efficacy of diagnostic models, enabling early detection and timely intervention. This, in turn, can improve the quality of life for older individuals by facilitating personalized treatment plans and reducing the burden of progressive muscle disorders.

While collecting data from the older generation for detecting muscular abnormalities using eEMG sensors presents its share of challenges, their data is indispensable for comprehensive research and accurate diagnosis. By addressing concerns, adapting to physical limitations, and establishing trust, one only can successfully overcome these hurdles and harness the wealth of knowledge hidden within the older population's data. Ultimately, this collective effort will contribute to advancements in medical science, benefiting both the present and future generations.

Even though the collecting the data from young health individual will not ensure that the model is reliable. Thus a strong motivation well a dedicated team for achieving these paramount task is required this is not a work of single person as per my abilities.

4.8.2 Re-training the model using transfer learning.

In order to achieve the ultimate objective of utilizing the available data that includes both eEMG and IMU data, this section will provide a step-by-step guide. The first step is to access the code required for retraining the pre-existing model, which is saved as "model.h5" in a specific GitHub repository. The code can be obtained by visiting the provided GitHub link(i will provide the link before the submission).

Before utilizing the code, it is important to ensure that the data is structured according to a specific schema. If the data does not adhere to this schema, modifications will need to be made to the "datareader()" function. The normal GAIT and abnormal

GAIT data should be stored in separate folders, with each subject's data in a separate file. These files should be in CSV format, where the first column represents the timestamp, followed by the accelerometer readings (X, Y, Z), gyroscope readings (X, Y, Z), and finally, the eEMG reading.

Once the data is properly configured, the next step is to run the "main.py" file, which will initiate the retraining process of the "model.h5" model. This process will update the model's weights and biases based on the newly provided data. Additionally, the code includes a pipeline to convert the retrained model into a TFLite micro model, which will be saved as "model.tflite" in the same folder.

The resulting "model.tflite" file is a flatbuffer that can be used in embedded systems such as Arduino BLE 33 Sense for making inferences. However, to use the model effectively, it is necessary to implement a data preprocessing pipeline, as discussed in Chapter 3, in C/C++. This preprocessing pipeline is crucial for processing the data in real-life scenarios and detecting abnormalities accurately.

By following this guide, future contributor can utilize the available data containing both eEMG and IMU data. The retraining process and conversion to a TFLite micro model enable the integration of the model into embedded systems, expanding its usability beyond traditional computing platforms. The implementation of a data preprocessing pipeline in C/C++ ensures compatibility and efficiency in real-life scenarios, making it possible to detect abnormalities effectively using the provided model.

4.9 Conclusion

In conclusion, this thesis project encountered challenges in achieving the objective step by step, particularly in creating a custom sensor for collecting sEMG and IMU data. However, by adapting and finding alternative solutions, the research continued using an open-source dataset. The importance of this work lies in understanding and analyzing gait abnormalities through the examination of sEMG and IMU data. Despite the absence of a fully functional custom sensor, utilizing an open-source dataset allowed for the development of a predictive model for normal and abnormal gait pat-

terns. While the limitations of this approach are acknowledged, the ability to adapt and find alternative solutions demonstrates resourcefulness and flexibility in dealing with constraints. Additionally, using an open-source dataset encourages collaboration and validation of findings, fostering a collective effort to enhance gait analysis methodologies. The significance of this work extends to areas such as biomechanics, rehabilitation, and clinical diagnostics, where gait analysis plays a crucial role in improving patient outcomes. Moving forward, addressing challenges in collecting data from the older generation and retraining the model using transfer learning can further enhance the research's impact and contribute to advancements in the field of fall risk assessment and prevention. By overcoming these hurdles and harnessing the wealth of knowledge hidden within the data, the research endeavors to improve the detection and treatment of muscular abnormalities, benefiting both the present and future generations.

Bibliography

[1] A. Nandy, S. Chakraborty, J. Chakraborty, and G. Venture, *Modern methods for affordable clinical gait analysis: theories and applications in healthcare systems*. Academic Press, 2021.

[2] J. Nogas, S. S. Khan, and A. Mihailidis, "Deepfall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders," *Journal of Healthcare Informatics Research*, vol. 4, no. 1, pp. 50–70, 2020.

[3] T. E. Lockhart, J. L. Smith, and J. C. Woldstad, "Effects of aging on the biomechanics of slips and falls," *Human factors*, vol. 47, no. 4, pp. 708–729, 2005.

[4] S. Sadigh, A. Reimers, R. Andersson, and L. Laflamme, "Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a swedish municipality," *Journal of community health*, vol. 29, no. 2, pp. 129–140, 2004.

[5] G. Rescio, A. Leone, and P. Siciliano, "Supervised machine learning scheme for electromyography-based pre-fall detection system," *Expert Systems with Applications*, vol. 100, pp. 95–105, 2018.

[6] R. Martinek, M. Ladrova, M. Sidikova, R. Jaros, K. Behbehani, R. Kahankova, and A. Kawala-Sterniuk, "Advanced bioelectrical signal processing methods: Past, present, and future approach—part iii: Other biosignals," *Sensors*, vol. 21, no. 18, p. 6064, 2021.

[7] A. P. Marsh, W. J. Rejeski, M. A. Espeland, M. E. Miller, T. S. Church, R. A. Fielding, T. M. Gill, J. M. Guralnik, A. B. Newman, and M. Pahor, "Muscle strength and bmi as predictors of major mobility disability in the lifestyle interventions and independence for elders pilot (life-p)," *Journals of Gerontology Series A: Biomedical Sciences and Medical Sciences*, vol. 66, no. 12, pp. 1376–1383, 2011.

[8] T. Snijders, L. B. Verdijk, and L. J. van Loon, "The impact of sarcopenia and exercise training on skeletal muscle satellite cells," *Ageing research reviews*, vol. 8, no. 4, pp. 328–338, 2009.

[9] V. Rajasekaran, J. Aranda, A. Casals, and J. L. Pons, "An adaptive control strategy for postural stability using a wearable robot," *Robotics and Autonomous Systems*, vol. 73, pp. 16–23, 2015.

[10] J. M. Weiss, L. D. Weiss, and J. K. Silver, *Easy EMG-E-Book: A Guide to Performing Nerve Conduction Studies and Electromyography.* Elsevier Health Sciences, 2021.

[11] D. Hamacher, D. Liebl, C. Hödl, V. Heßler, C. K. Kniewasser, T. Thönnessen, and A. Zech, "Gait stability and its influencing factors in older adults," *Frontiers in physiology*, vol. 9, p. 1955, 2019.

[12] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 2. IEEE, 2005, pp. ii–973.

[13] H. Lu, J. Huang, T. Saha, and L. Nachman, "Unobtrusive gait verification for mobile phones," in *Proceedings of the 2014 ACM international symposium on wearable computers*, 2014, pp. 91–98.

[14] R. Delgado-Escano, F. M. Castro, J. R. Cózar, M. J. Marín-Jiménez, and N. Guil, "An end-to-end multi-task and fusion cnn for inertial-based gait recognition," *IEEE Access*, vol. 7, pp. 1897–1908, 2018.

[15] M. Gadaleta and M. Rossi, "Idnet: Smartphone-based gait recognition with convolutional neural networks," *Pattern Recognition*, vol. 74, pp. 25–37, 2018.

[16] Y. Zhang, G. Pan, K. Jia, M. Lu, Y. Wang, and Z. Wu, "Accelerometer-based gait recognition by sparse representation of signature points with clusters," *IEEE transactions on cybernetics*, vol. 45, no. 9, pp. 1864–1875, 2014.

[17] W. Siwadamrongpong, J. Chinrungrueng, S. Hasegawa, and E. Nantajeewarawat, "Fall detection and prediction based on imu and emg sensors for elders," in *2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE).* IEEE, 2022, pp. 1–6.

[18] A. Ahmadiahangar, Y. Javadian, M. Babaei, B. Heidari, S. Hosseini, and M. Aminzadeh, "The role of quadriceps muscle strength in the development of falls in the elderly people, a cross-sectional study," *Chiropractic & manual therapies*, vol. 26, no. 1, pp. 1–6, 2018.

[19] P. J. López-Soto, M. H. Smolensky, L. L. Sackett-Lundeen, A. De Giorgi, M. A. Rodríguez-Borrego, R. Manfredini, C. Pelati, and F. Fabbian, "Temporal patterns of in-hospital falls of elderly patients," *Nursing Research*, vol. 65, no. 6, pp. 435–445, 2016.

[20] A. F. Ambrose, G. Paul, and J. M. Hausdorff, "Risk factors for falls among older adults: a review of the literature," *Maturitas*, vol. 75, no. 1, pp. 51–61, 2013.

[21] J. Howcroft, J. Kofman, and E. D. Lemaire, "Prospective fall-risk prediction models for older adults based on wearable sensors," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 25, no. 10, pp. 1812–1820, 2017.

[22] A. Zia, S. B. Kamaruzzaman, and M. P. Tan, "Polypharmacy and falls in older people: balancing evidence-based medicine against falls risk," *Postgraduate medicine*, vol. 127, no. 3, pp. 330–337, 2015.

[23] C. A. Chase, K. Mann, S. Wasek, and M. Arbesman, "Systematic review of the effect of home modification and fall prevention programs on falls and the performance of community-dwelling older adults," *The American Journal of Occupational Therapy*, vol. 66, no. 3, pp. 284–291, 2012.

[24] A. J. A. Majumder, F. Rahman, I. Zerin, W. Ebel Jr, and S. I. Ahamed, "iprevention: Towards a novel real-time smartphone-based fall prevention system," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 513–518.

[25] R. Rucco, A. Sorriso, M. Liparoti, G. Ferraioli, P. Sorrentino, M. Ambrosanio, and F. Baselice, "Type and location of wearable sensors for monitoring falls during static and dynamic tasks in healthy elderly: a review," *Sensors*, vol. 18, no. 5, p. 1613, 2018.

[26] K. M. Gerling, J. Schild, and M. Masuch, "Exergame design for elderly users: the case study of silverbalance," in *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*, 2010, pp. 66–69.

[27] V. Rajapakse, I. Karunanayake, and N. Ahmed, "Intelligence at the extreme edge: A survey on reformable tinyml," *arXiv preprint arXiv:2204.00827*, 2022.

[28] C. Banbury, C. Zhou, I. Fedorov, R. Matas, U. Thakker, D. Gope, V. Janapa Reddi, M. Mattina, and P. Whatmough, "Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 517–532, 2021.

[29] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3197–3212, 2020.

[30] E. L'opez-Larraz, J. Minguez, L. Montesano, A. Gil-Agudo, and J. L. Pons, "Wearable lower-limb robotics exoskeleton for gait rehabilitation of motor disorders: A review," in *Wearable Robotics: Challenges and Trends*. Springer, Cham, 2016, pp. 169–189.

[31] J. H. LeLaurin and R. I. Shorr, "Preventing falls in hospitalized patients: State of the science," *Clin Geriatr Med*, vol. 35, no. 2, pp. 273–283, May 2019, epub 2019 Mar 1.

[32] R. N. Ferreira, N. F. Ribeiro, and C. P. Santos, "Fall risk assessment using wearable sensors: A narrative review," *Sensors (Basel)*, vol. 22, no. 3, p. 984, Jan 2022.

[33] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, 2019.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016, vol. 1.

[35] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[36] S. Mallat, *A wavelet tour of signal processing: The sparse way*, 3rd ed. Academic Press, 2008.

[37] S. Haykin, *Neural networks and learning machines*, 3rd ed. Prentice Hall, 2009.

[38] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," UCI Machine Learning Repository, 2013. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

[39] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The mobiact dataset: Recognition of activities of daily living using smartphones." in *ICT4AgeingWell.* Rome, 2016, pp. 143–151.