Visvesvaraya Technological University
Belgaum, Karnataka- 590018

A Project Phase – II Report
On

"Parkinson's Fall Detection System using Machine Learning and IOT"

Submitted in the partial fulfilment of the requirements for the award of the Degree of
BACHELOR OF ENGINEERING
In
INFORMATION SCIENCE AND ENGINEERING

Submitted by

Akhila H	**(1DS19IS012)**
Aneesh Chandra Dixith	**(1DS19IS016)**
Avanthi Anand	**(1DS19IS026)**
Karthik Shivanand	**(1DS19IS050)**

Under the Guidance of
Dr. Vaidehi M

Associate Prof, Dept. of ISE

2022-2023

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
DAYANANDA SAGAR COLLEGE OF ENGINEERING
SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT, BANGALORE-78

DAYANANDA SAGAR COLLEGE OF ENGINEERING

Shavige Malleshwara Hills, Kumaraswamy Layout

Bangalore-560078

Department of Information Science and Engineering

2022-2023

Certificate

This is to certify that the Project Work entitled **"Parkinson's Fall Detection System using Machine Learning and IOT"** is a bonafide work carried out by **Akhila H (1DS19IS012), Aneesh Chandra Dixith (1DS19IS016), Avanthi Anand (1DS19IS026) and Karthik Shivanand (1DS19IS050)**, in partial fulfilment for the 8th semester of Bachelor of Engineering in Information Science & Engineering of the Visvesvaraya Technological University, Belgaum during the year 2022-2023. The Project phase – II report has been approved as it satisfies the academic requirements prescribed for theBachelor of Engineering degree.

Signature of Guide	Signature of HOD	Signature of Principal
[Dr. Vaidehi M]	[Dr. Rajeshwari J]	[Dr. B G Prasad]

Name of the Examiners	Signature with Date
1	
2	

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project.

We take this opportunity to express our sincere gratitude to **Dayananda Sagar College of Engineering** for having provided us with a great opportunity to pursue our Bachelor Degreein this institution.

In particular we would like to thank **Dr. B G Prasad,** Principal, Dayananda SagarCollege of Engineering for his constant encouragement and advice.

Special thanks to **Dr. Rajeshwari J,** Professor and HOD, Department of Information Science & Engineering, Dayananda Sagar College of Engineering for his motivation and invaluable supportwell through the development of this project.

We are highly indebted to our internal guide **Dr. Vaidehi M**, Associate Professor, Department of Information Science & Engineering, Dayananda Sagar College of Engineering for her constant support and guidance. She has been a great source of encouragement throughout the course of this project.

Finally, we gratefully acknowledge the support of our families, friends during the completion of the project.

Akhila H	**(1DS19IS012)**
Aneesh Chandra Dixith	**(1DS19IS016)**
Avanthi Anand	**(1DS19IS026)**
Karthik Shivanand	**(1DS19IS050)**

ABSTRACT

Parkinson's disease (PD) is one of the long-term regressive disorders of the central nervous system that mainly affects the nervous system in humans due to which there is the recurrent occurrence of falls which might even lead to death or put them in a serious situation. For solving this issue, we have proposed an idea that detects patient fall and reports to the caretaker or family member utilizing message/call/Alarm thereby avoiding deaths. Our work focuses on developing and apply the wearable fall-detection system for Parkinson's disease long-suffers formed on a Wi-Fi module. Patient falls were precisely traced on the outcome of an accelerometer, oximeter, and pressure sensors which were sent to the cloud via NodeMCU-ESP32. Furthermore, Node MCU ESP32 is planned for attaining smooth communication between the fall detection device and the system for obtaining and processing the data anytime and anywhere until we have a Wi-Fi connection. In consequence, the cloud will perform the calculations based on SVM which is the Support Vector Machine. In turn, it sends the fall detection outcomes to the MIT application. This WFDS turns out to be the best among the existing models in terms of fall detection sensitivity, specificity, and accuracy. The fall of direction in PD can be performed accurately based on the fall detection algorithm at the receiver end. The preliminary outcomes show that the wearable fall-detection system attains eighty-. eight percent accuracy in identifying the patient's fall.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview:

Falls are the first main reason for injury for elderly people. It is estimated that 4% of the population over the age of 60 are suffering from Parkinson’s disease and were diagnosed before and at the age of 50. Overall, as many as 1 million Americans are living with Parkinson’s disease, and around 60,000 Americans are identifying with Parkinson’s every year. Therefore, a fall detection system for the elderly is a technology that is essential for the aging society. Most people will fall often and repeatedly because of Parkinson’s disease. The consequences of falls can be devastating, far-reaching, and costly. People with Parkinson’s disease most commonly experience a reduction in the body balance and its symptoms are tremors, the rigidity of muscles, pain in limbs, Gait impairment, freezing of gait, slow moments, stumbling walking, depression, and visual and cognitive impairment. Recurrent Falls in Parkinson’s disease (PD) and seizures can cause injuries like rupturing of bones, reducing the quality of life and may lead to death. Some of the fall detection systems which were already introduced are having some constraints and provocation like fall detection accuracy, the direction of fall, and needless alarms. As people start aging, and as humans work constantly the human body starts to decline. This implies that the daily tasks or efforts that people are doing every day can lower their physical condition which leads to unexpected falls.

Parkinson’s disease (PD) is a central nervous system disorder that mainly affects the kinetic system in humans. PD affects thousands of individuals worldwide and is expected to affect about 8.7 to 9.3 million people in 2030. Common early symptoms of PD are related to abnormal movements of the human body. PD begins by way of the damage or reduction of brain cells responsible for the production of dopamine, a neurotransmitter responsible for helping the brain perform its essential tasks such as controlling kinetic functions and possibly other functions related to mood and feelings. About 68% of PD patients fall, with around 50% suffering from repeated falls.

In general, recurrent falls occur when a patient falls more than once in a given period (usually 6–12 months). Allen et al.[2] reported that around 70% of PD patients who experience falls suffer from recurrent falls. A survey of 100 people with PD showed that 13% of recorded fall accidents in PD occurred more than once per week and even multiple times a day [2,11]. Goodwin et al. [6] discussed PD patients who suffered a fall 577 times in 20 weeks, which is approximately equivalent to 1,500 falls in a year. These results indicate the need to monitor the fall events of PD patients continuously at their homes or health centers to notify their member families or caregivers as soon as possible of the occurrence by detecting early signs of fall detection [6]. Consequently, fall-detection systems offer a significant service by decreasing the subsequent negative impacts of risk falls such as sudden mortality due to cardiac arrest; stroke; hip fracture, trauma, or injury.

Falls in PD occur when the production of dopamine in the brain decreases by about 75% to 80%. The lack of dopamine equilibrium in the brain can lead to a brain malfunction. Many symptoms have a direct impact on fall-related events in PD, including tremors (spontaneous shaking in the body such as in the arms and legs), muscle rigidity, gait impairment, gait freezing, slow movement, depression, and fall events as shown in. The symptoms of PD are very episodic and, as such, cannot be wholly analyzed or recorded at the doctor’s clinic.

Normally, most people might think that an unexpected fall is very threatening because elders will be affected very badly. Thailand is one of the aging societies as it is having 36 percent of people who are treated in the emergency room and it's been increasing every year. Parkinson’s disease affects thousands of individuals globally and is anticipated to influence about 8.7 to 9.3 million people in 2030. Parkinson’s disease begins to damage and starts to deplete the brain cells responsible for the production of dopamine, the neurotransmitter responsible for serving the brain carrying out its necessary action such as controlling kinetic functions and possibly other functions related to the frame of mind and feelings.

Around 68 percent of Parkinson’s disease patients fall, with around 50 percent hurting from recurrent falls. When the production of dopamine in brain cells reduces to 75 to 80 percent the patients who are suffering from Parkinson’s disease will start to fall. It leads to brain breakdown due to the deficiency of dopamine content in the brain. Common abnormal movements of the body and the other symptoms are Tremor, Gait impairment, freezing of gait, Slow movement, Stumble walking, Depression, Rigidity muscle, Pain of limbs, visual and cognitive impairments due to which fall accident will be happened and results in injures like a broken bone and sometimes it may lead to death by the Cardiac arrest and stroke are the early symptoms of Parkinson's Patients. Normally, repeated falls count if the patient falls more than once in a given interval usually 6-12 months.

About 13% is recorded from the observation of 100 patients that they suffer one fall per week or even twice or multiple times a week. By research, Patients suffer 577 falls in 20 weeks and which comes to around 1,500 times in a year. Through these records, the information for the caretakers and the family members is to be sent as soon as they come across the early symptoms of Parkinson's disease to prevent the fall. In consequence, fall-detection systems offer an outstanding solution by decreasing the following negative impacts of risk falls, such as Gait impairment, Freezing of gait, Slow movement, stumbling walking, Depression, Rigidity muscle, Pain of limbs, and visual and cognitive impairments.

As mentioned above, fall detection research applied either a single sensor or fusion by multiple sensors. The methods of collecting data are typically divided into four main categories, namely individual wearable sensors, individual visual sensors, individual ambient sensors, and data fusion by sensor networks. Whilst some literature groups visual and ambient sensors together we treat them as two different categories in this report due to visual sensors becoming more prominent as a detection method with the advent of depth cameras (RGBD), A fall-detection system consists of a sensing component and an algorithm that detects falls based on the measured sensor readings. The sensing components can be attached to the human body, such as in the case of accelerometers and gyroscopes, or not—for example, cameras and infrared devices. Two main methods of algorithms are used for detecting fall events: a simple threshold method and the machine learning method.

For instance, three-axial acceleration is utilized to detect a fall. An acceleration exceeding the threshold indicates a fall event. In the machine learning method, cases of different types of fall events and activities of daily living (ADL) are trained and tested by a learning algorithm under an evaluation algorithm. Machine learning methods include support vector machines and hidden

Markov models. Unfortunately, machine learning approaches can be difficult to implement because of their heavy computational and resource requirements.

In addition to fall detection, the direction of the fall accident is also valuable information because it can indicate freezing and gait impairment in patients with PD. Many studies have attempted to introduce fall direction. However, the computation of the direction of fall accidents remains largely inaccurate.

In this report, a wearable fall-detection system (WFDS) for use by elderly PD patients is proposed. This WFDS automatically detects fall events and assists with the immediate communication of such adverse accidents wirelessly to medical staff or family to draw medical assistance as soon as possible by sending alerts. In addition, this assistance is valuable to promote the sense of security of elderly patients, especially those living alone, and reduce their fear of falling.

1.2 Problem Statement

Parkinson’s disease (PD) is a central nervous system disorder that mainly affects the kinetic system in humans due to which there is a recurrent occurrence of falls which might even lead to death or put them in a serious condition. In order to solve this problem, we have proposed a system that detects a patient’s fall and reports it to the caretaker or family member by means of a message/call/alarm thereby avoiding deaths.

1.3 Objectives

To develop an intelligent device for detecting falls using the pattern of Parkinson’s patients. To develop a system which detects falls and also sends real time data such as body temperature, spo2, blood pressure and location of patient to caretakers.

1.4 Motivation

Recurrent falls can cause serious injury and even death due to cardiac arrest and stroke. In general, recurrent falls occur when a patient falls more than once in a given period (usually 6–12 months).

These results indicate the need to monitor the fall events of Parkinson's Disease patients continuously at their homes or health centers to notify their member families or caregivers as soon as possible of the occurrence by detecting early signs of fall detection.

With advancements in technology and the prevalence of audio collecting devices in daily lives, reliable models that can translate this audio data into a diagnostic tool for healthcare professionals would potentially provide diagnoses that are cheaper and more accurate.

CHAPTER 2

LITERATURE SURVEY

The below-given category summarizes the references from various reports with similar goals and objectives. Several previous works have been done in the area of monitoring of the elderly without any aid from humans. Based on the different types of sensors used, there are different techniques used to monitor the fall detection. The techniques used can be broadly classified into 1) Wearable sensors approach; 2) vision-based approaches; 3) acoustic, vibrational, and other ambiance sensor-based approaches. Below, we provide a synopsis of the systems using vision-based and non-vision sensors.

Wearable sensor-based methods depend on accelerometer and gyroscope sensors which are mounted on the subject's body. This system relies on the wearable embedded sensors to detect the motion and location of patients. The advantage of this method includes the ease of installation, cost efficiency, and its simple operation. Disadvantages: Most fall events cannot be detected, and the system will be restricted to limited coverage areas.

Ambient sensor-based methods depend on capturing the vibrational data of the subject such as shaking signals or audio signals, that are generated during fall events. This system involves external sensors such as infrared, acoustic, pressure, and vibrational sensors. These sensors can detect the nature of everything in the sensor arrangement environment and the distance between the subject and sensor location has a direct effect on fall detection accuracy. Thus, the system tends to generate false alarms. The main disadvantage of this method is that it cannot cover all fall direction angles, the detection area is limited to a distance of about 12m only, and the system has a high cost. Vision-based fall-detection systems use a computer or a multi-set of video cameras embedded into the monitoring environment of elderly patients. Data acquisition relies on analyzing body posture changes or three-dimensional head motion However, the performance of the system is reduced when the patient is blocked by an object (e.g., furniture). In this case, the system can not analyze and process the ongoing patient activity, rendering it useless. In addition, cameras or videos are limited to use in specific areas, do not guarantee privacy, and are expensive. Several methods have previously been proposed for the fall detection problem.

Ying-Wen Bai, et al.[3] proposed a threshold-based method to detect falls using a three-axis accelerometer sensor data gained from a smartphone.Their algorithm had five states, each checked by some thresholds on acceleration magnitude, to decide whether a fall has occurred or not.

Stefano Abbate, et al. [4] defined some index metrics such as Impact Duration, Maximum Peak, Peak Duration, etc., and obtained values for them in 3-second windows. Their method detects falls by using a feed-forward neural network with one hidden layer and seven neurons on it. HE Jian and HU Chen [5] detected backward falls only, by using accelerometer and gyroscope sensors and a KNN algorithm. Their features were the sum of 20 samples for acceleration and gyroscope magnitude in each window. Finally, backward falls are classified into 3 other moves (walking–turning–walking, sitting down-standing up, squatting-standing). Ryan M. Gibson, et al. [6] used a multi-comparator classifier for detecting falls. They used ANN, KNN, RBF, PPCA, and LDA classifiers. The acceleration data obtained from the Shimmer device and they used discrete wavelet transform (DWT) was applied to the time domain tri-axial acceleration vector to determine the wavelet approximation coefficients for the equivalent acceleration components.

Bruno Andò et al. [7] presented a novel threshold-based multi-sensor data fusion approach by using combined data from an accelerometer and a gyroscope. To create features, they used maximum cross-correlation between recorded signals. Our method resembles [4] in using a neural network, though the MLF network acts differently from conventional neural networks. Finally, the proposed method uses features such as: mean, standard deviation, min, max of the magnitude of acceleration, and each x, y, and z and another feature that is calculated by some formula, and the considered states include nine daily activity and four fall movements.

Yu et al. [29] proposed a computer-vision fall detection system based on posture recognition by the projection histogram method. The proposed system employs a single camera for monitoring at home. Experimental results showed that the system has a fall detection accuracy of 97.08% and a false detection rate of 0.8%. Disadvantages: Limited detection area preventing the body from being tracked outside the range of vision cameras and the external sensing of the camera entails a high cost.

Chen et al. [32] implemented a video-based human fall-detection system that uses a single camera and a computer This fall-detection system is based on a combination of human shape variations and skeleton features. The sensitivity and false alarm rates of the proposed system are 90.9% and 6.25%, respectively. Disadvantages: Most fall events cannot be detected, the system has a limited coverage area, and only three types of fall events and three ADLs have been tested. Ambient sensor–based methods rely on capturing the vibrational data, such as shaking signals or audio signals, that are generated during fall events. Ambient sensor–based systems involve external sensors, such as infrared, acoustic, pressure, and vibration sensors. However, these sensors can detect the nature of everything in the sensor deployment environment and the distance between the subject and sensor location has a direct effect on fall detection accuracy Thus, the system tends to generate false alarms. Indoor environments should contain multiple sensors to achieve high accuracy, thereby increasing effort and costs. The main shortcomings of this strategy are a low accuracy for fall identification (<95%). Previous researchers such as Ariani et al [34] proposed a dual-technology sensor (DTS) based on wire-less ambient sensors. This modality uses a simulated environment to validate the potential possibility of using wireless ambient sensors. The proposed DTS system consists of the hybrid sensor [pyro-electric infrared (PIR) and microwave] as a motion-detection method and pressure mats. Two DTS detectors are placed at each location; one detector tracks the upper half of the room, and the other tracks the lower half. Experimental results showed that the accuracy, sensitivity, and specificity of the proposed fall algorithm are 89.33%, 100.00%, and 77.14%, respectively. Disadvantages: The system is that it cannot cover all fall direction angles, the detection area is limited to a distance of about 12 m only, and the system has a high cost.

Liu et al. [35] introduced an elderly healthcare system for fall detection. The system consists of three sensitive PIR sensors that are placed at different heights on a wall. The system detects the fall event by responding to temperature changes produced by human motion. The optical flow technique was also applied to capture and analyze the direction of motion caused by the head, lower limbs, and upper limbs, respectively. The system achieved 92.5% sensitivity and 93.7% specificity. Advantages: Low operational and manufacturing costs. Disadvantage: The PIR sensor has a limited coverage area. Giuseppe and Pietro [28] presented a fall-detection system called the ''m-Health system,'' which consists of an accelerometer sensor used to monitor the subject's activity and posture and an integrated device including an ECG sensor with a breathing rate sensor. The accelerometer sensor monitors the patient’s posture and activity and then sends samples wirelessly to the mobile device via Bluetooth. Experimental results showed that the system has 92% accuracy, 86% sensitivity, and 96% specificity. Disadvantages: The system has a complex computational algorithm and the types of test cases of fall events used in evaluating the system were not mentioned.

Panicker et al. [27] developed a fall-detection system by monitoring parameters in real time. The developed system includes a wearable wrist body sensor unit and involves a PC with the Android operating system. A body sensor unit consists of an accelerometer sensor, a blood pressure sensor, a temperature sensor, a pulse rate sensor, and a Bluetooth module. Experimental results showed that this fall-detection system could achieve 80% accuracy, 87.5% sensitivity, and 75% specificity. Advantages: The wrist sensor node is suitable for obese bodies with large arms and the device can be comfortably worn because it does not require one to remove clothing while

obtaining measurements. Disadvantages: Low accuracy, data classification of test cases was obtained only from young healthy subjects, and the authors did not mention the number and types of test cases used in evaluating the system.

Ren et al. [26] designed an energy-efficient prototype of a fall-detection system called Asgard, which consists of a triaxial accelerometer sensor, a microcontroller, and a Zigbee module. Test cases were carried out by 10- to 30-year-old subjects (seven males and three females). Two AA batteries (1500 mAh) are used to power up the sensor node. Experimental results of comprehensive evaluations showed that Asger achieved 93.21% accuracy. Advantages:

Low cost nd optimal placement of the sensor node (waist). Disadvantages: Low accuracy; the system has a long detection time of about 12s, and data calcification was tested only in young subjects. A wearable device using a single tri-axial accelerometer was developed by Wu et al. in [22]. The wearable device is placed on the user’s waist. The device detects falls based on acceleration analysis and sends a short message with the user’s location data to caregivers. The system also gives the user a chance to withdraw an alarm.

No special mounting is required for this system as the algorithm does not need the axes of the accelerometer to be fixed strictly. The proposed system achieved 97.1% sensitivity and 98.3%

specificity. The system may trigger false alarms as the normal activity of resting has a similar rotation as falling.

Lim et al. [23] developed a system in which the fall feature parameters are calculated from a single tri-axial accelerometer. The parameters are first applied to a simple threshold method and then the detected falls are applied to a Hidden Markov Model (HMM) to distinguish between falls and fall-like events. The system achieved 99.5% accuracy, 99.17% sensitivity, and 99.69% specificity when the ASVM = 2.5g and $\theta = 55°$ were the threshold values for the simple threshold method and the parameter θ was applied to the Hidden Markov Model. The system conserves computing effort and resources by applying only the detected fall events to the HMM.

An unobtrusive smartphone-based fall detection system was proposed by Aguiar et al. in [24]. The system continuously screens the accelerometer data of the smartphone when the smartphone is carried in the user's belt or pocket. From the acceleration vector output of the accelerometer of the smartphone, a total of 14 different signal components (x, y, z projections, magnitude value, and angles with x, y, and z axes of the phone) are computed and passed through a Butterworth digital filter by the system. The system then uses a decision tree to retrieve information regarding the most significant features and thresholds. This information is used in a state machine algorithm to detect the fall and send SMS and email notifications containing the user's location to the caregivers and emergency services. The sensitivity and specificity of the system are close to 97% and 99% for both usage positions (belt or pocket). Future works might incorporate the use of gyroscopes for detecting the rotation and barometer for calculating altitude in the system. As the system continuously screens the accelerometer data, battery usage of the smartphone is a concern.

Li et al. [25] developed a waist-mounted device using TelosW mote with an accelerometer as the detector and the Neyman-Pearson detection framework as the classifier. The accelerometer in the systems periodically samples the acceleration of the person and compares the data with a predefined threshold. If the data exceeds the threshold, an alert message is delivered to the base station using an 802.15.4/Zigbee network. The base station can then forward the message to the emergency services and caregivers using mobile communication networks.

Laura et al [26] developed a footwear-based fall detection system employing an accelerometer sensor and Force Sensing Resistors (FSR) situated in the sole of the footwear. The processing device, a Raspberry Pi, encased in a box, is connected to the sensors via long wires and worn at the waist like a belt. The threshold-based algorithm achieved a 97.1% maximum accuracy and 90% accuracy with a FAR (False Alarm Rate) of 0.

The Simbad project [11] uses infrared sensors, making people’s detection and feature extraction easier. Fall detection is performed by using a neural network classifying a vertical velocity descriptor. Nevertheless, the requirement of fast movements recognition may lead to a sensitivity to noise tending to send false alarms.

The Ubisense project [12] proposes to classify human poses by computing the orientation of each detected blob. However, no motion modeling and recognition are proposed for analyzing the pose sequences.

Nait-Charif and McKenna [13] propose a method for automatically extracting motion trajectory and providing human-readable summarization of activity and detection of unusual inactivity. Tracking is performed with an omnidirectional camera by means of a particle filter estimating ellipse parameters describing human posture. Fall is detected as a deviation from usual activity. However, no information about the pose of the person or his motion dynamic is taken into account.

Töreyin et al. [14] suggest a method for fall detection by making use of an HMM using both audio and video. For the vision part of the approach, the aspect ratio of the bounding box of the moving region detected with a standard camera is analyzed by the motion model. More precisely, its wavelet transform is used as an input feature for the HMM. Using conjointly video and audio cues seems to be well-founded. Defining HMM states in the frequency domain is interesting because it makes explicit use of motion features. However, the viewpoint robustness of the bounding box aspect ratio feature for discriminating standing and lying postures is not discussed, and the evaluation is mainly limited to frontal views. It is clear, for example, that the aspect ratio observed in the image corresponding to a standing posture will sensitively vary between a vertically-oriented optical axis and a horizontally oriented one. The problem remains for the wavelet coefficients, making the motion recognition efficiency only limited to some specific viewpoint configurations.

Recently, Cucchiara et al. [15] propose a multi-view solution dedicated to fall detection. They make use of histogram projections to classify the silhouette between the standing and lying poses. Interestingly, warping people’s silhouettes between the different views makes it possible to detect partial occlusions, and compensate for them. An HMM-based approach is proposed for

Overview of the proposed multi-view fall detection system. making the pose recognition more robust. However, the motion is only taken into account at very small time scales, to disambiguate the pose estimation, and no explicit modeling of the motion in terms of pose sequence is proposed.

CHAPTER 3

REQUIREMENTS

3.1 Functional Requirements

1. **Fall Detection:** The system should be able to accurately detect falls or sudden movements indicative of a fall in individuals with Parkinson's disease.

2. **Sensor Integration:** Integration with IOT devices and sensors (such as accelerometers, gyroscopes, or wearable devices) to capture relevant data related to movement, posture, and activity.

3. **Real-time Monitoring:** The ability to continuously monitor the user's movements and provide real-time alerts or notifications to caregivers or emergency contacts in the event of a fall or potential fall.

4. **Machine Learning Algorithms:** Utilization of machine learning techniques to analyze the voice data input and predict the presence of the disease. The system should be trained on a dataset of voice inputs specific to Parkinson's disease.

5. **Personalized Thresholds**: The system should allow customization of fall detection thresholds to accommodate individual variations and account for the specific symptoms and characteristics of each person with Parkinson's disease.

6. **Ambient Intelligence:** Integration with smart home technologies or environmental sensors to provide additional context and insights into the user's activities and surroundings, aiding in the detection and prevention of falls.

7. **Localization:** The system should be able to identify the location of the user accurately, either through GPS or indoor localization techniques, to facilitate timely response and assistance in case of a fall.

8. **Data Privacy and Security:** Implementation of robust security measures to protect the personal health data of individuals with Parkinson's disease, including encryption, access control, and compliance with privacy regulations.

9. **User-Friendly Interface:** A user-friendly interface for both individuals with Parkinson's disease and caregivers, allowing easy setup, configuration, and access to relevant information or reports.

10. **Data Logging and Analysis:** The system should have the capability to store and analyze historical data related to falls and movements, enabling long-term monitoring, trend analysis, and potential insights for healthcare professionals.

11. **Integration with Healthcare Providers:** Provision for integration with existing healthcare systems or electronic health records, facilitating seamless communication and information sharing between the Parkinson's fall detection system and healthcare professionals.

3.2 Non-Functional Requirements

1. **Reliability:** The system should be highly reliable and accurate in detecting falls and distinguishing them from other movements or activities.

2. **Accuracy:** The system should have a high level of accuracy in detecting falls while minimizing false positives and false negatives.

3. **Real-time monitoring**: The system should provide real-time monitoring capabilities to detect falls and trigger timely alerts or notifications.

4. **Scalability:** The system should be scalable to handle a large number of users and devices simultaneously.

5. **Security:** The system should ensure the security and privacy of user data, as it may involve personal health information. It should implement appropriate encryption, access controls, and data protection measures.

6. **Compatibility:** The system should be compatible with a variety of IoT devices, sensors, and wearables commonly used by Parkinson's patients, ensuring seamless integration and data exchange.

7. **Usability:** The system should have a user-friendly interface and be easy to set up and use, considering the potential physical limitations of Parkinson's patients.

8. **Energy efficiency:** The system should optimize energy consumption to prolong the battery life of IOT devices or wearables used in the fall detection system.

9. **Adaptability:** The system should be adaptable to different environments and situations, accounting for variations in user behavior, mobility, and location.

10. **Performance:** The system should have efficient processing capabilities, handling real-time data streams from multiple sensors and delivering prompt fall detection and alerts.

11. **Robustness:** The system should be resilient to network interruptions, device failures, or environmental factors that may affect the detection accuracy.

12. **Maintenance and support:** The system should be designed for ease of maintenance, including regular updates, bug fixes, and technical support for users.

13. **Compliance:** The system should comply with relevant legal and regulatory requirements, such as data protection laws and healthcare industry standards.

14. **Interoperability**: The system should support interoperability with other healthcare systems or platforms, allowing seamless integration of fall detection data into broader health monitoring and management systems.

15. **Ethical considerations:** The system should address ethical considerations, such as transparency in data usage, consent for data collection, and adherence to ethical guidelines for AI and machine learning applications.

3.3 Software Requirements

1. ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a software platform specifically designed for programming and developing projects using Arduino boards. It provides a user-friendly interface and a set of tools that simplify the process of writing, compiling, and uploading code to Arduino microcontrollers.

The Arduino IDE is built on the Processing programming language and is open-source, allowing developers to access and modify its source code. It is available for various operating systems like Windows, mac OS, and Linux, making it widely accessible to users across different platforms. When working with the Arduino IDE, users write their code in the Arduino programming language, which is based on the C and C++ languages. The IDE includes a text editor with syntax highlighting and auto-completion features that aid in writing code efficiently.

It also provides a library manager that offers a vast collection of pre-written code modules, called libraries, to extend the functionality of Arduino boards. The IDE supports a straightforward compilation process, where the user's code is transformed into machine-readable instructions called machine code. It automatically handles the compilation and linking steps required to convert the written code into a format that can be executed by the Arduino microcontroller.

2. FIREBASE CLOUD

Firebase is a backend platform for building Web, Android and IOS applications. It offers real time database, different APIs, multiple authentication types and hosting platform. Firebase can power your app's backend, including data storage, user authentication, static hosting, and more. Focus on creating extraordinary user experiences. We will take care of the rest. Build cross-platform native mobile and web apps with our Android, iOS, and JavaScript SDKs. You can also connect Firebase to your existing backend using our server-side libraries or our REST API.

Firebase Features

i. Real-time Database − Firebase supports JSON data and all users connected to it receive live updates after every change.
ii. Authentication – We can use anonymous, password or different social authentications.
iii. Hosting – The applications can be deployed over secured connections.

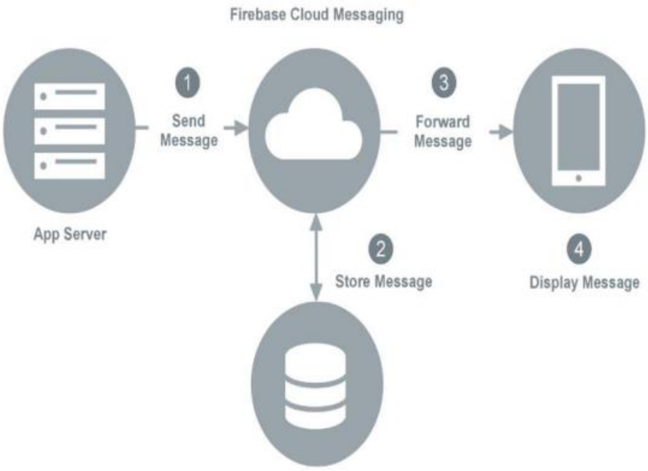

Figure 3.3.2.1. Firebase Cloud

3. MIT APPLICATION

MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the Star Logo, which allows users to drag and drop visual objects to create an application that can run on android devices, while a App-Inventor Companion (The program that allows the app to run and debug on) that works on iOS running devices are still under development. In creating App Inventor, Google drew upon significant prior research in educational computing, and work done within Google on online development environments.

3.4 Hardware Requirements

1. ADXL337

The ADXL337 is a three-axis accelerometer sensor manufactured by Analog Devices. This sensor is designed to measure acceleration in three perpendicular axes: X, Y, and Z. It operates on the principle of microelectromechanical systems (MEMS) technology, where tiny integrated structures sense and convert physical motion into electrical signals. The ADXL337 provides accurate and reliable acceleration data, making it suitable for various applications such as robotics, motion detection, and tilt sensing. The ADXL337 sensor consists of a tiny silicon structure that contains capacitive plates and a suspended proof mass. When subjected to acceleration, the proof mass moves, causing a change in capacitance. This change is converted into electrical signals that can be processed by a microcontroller or other electronic devices. The sensor offers a measurement range of ±3g, meaning it can detect accelerations up to three times the acceleration due to gravity.

One of the key features of the ADXL337 is its small size, which enables easy integration into compact electronic systems. It operates on low power, making it suitable for battery-powered applications. The sensor provides analog output signals proportional to the measured accelerations, which can be easily interfaced with analog-to-digital converters (ADCs) or analog signal processing circuits.

The ADXL337 is commonly used in projects that require accurate acceleration measurements and motion detection. For example, in robotics, it can be employed to monitor the orientation and movement of robotic arms or platforms. In motion detection systems, the sensor can be utilized to trigger events based on specific movements or gestures. It can also be used in tilt sensing applications to detect the inclination of an object or device.

Overall, the ADXL337 three-axis accelerometer offers a reliable and precise solution for measuring acceleration in three dimensions. Its small size, low power consumption, and analog output make it versatile and suitable for a wide range of applications where monitoring and analysis of physical motion are crucial.

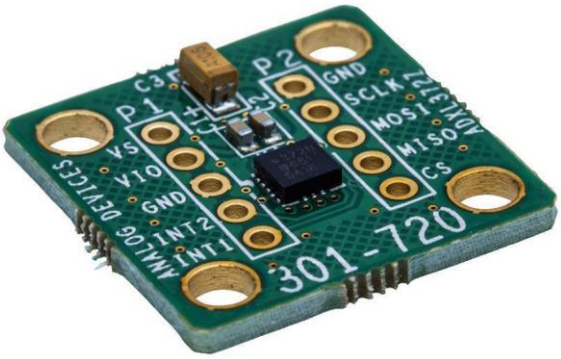

Figure 3.4.1.1 ADXL337

2. ESP32

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSM ultra-low power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios. TheESP32 series of chips includes ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, ESP32- D0WDQ6, ESP32-D2WD, ESP32-S0WD, and ESP32-U4WDH, among which, ESP32-D0WD-V3, ESP32-D0WDQ6-V3, and ESP32-U4WDH are based on ECO V3 wafer. ESP32 is designed for mobile, wearable electronics, and Internet-of-Things (IOT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes and dynamic power scaling. For instance, in a low-power IOT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends. The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption.

Figure 3.4.1.2 ESP32

ESP32 is capable of functioning reliably in industrial environments, with an operating temperature ranging from –40°C to +125°C. Powered by advanced calibration circuitries, ESP32 can dynamically remove external circuit imperfections and adapt to changes in external conditions. Engineered for mobile devices, wearable electronics and IOT applications, ESP32 achieves ultralow power consumption with a combination of several types of proprietary software. ESP32 also includes state-of-the-art features, such as fine-grained clock gating, various power modes and dynamic power scaling. ESP32 is highly integrated with in-built antenna switches, RF antenna, power amplifier, low noise receives amplifier, filters, and power management modules. ESP32 adds priceless functionality and versatility to your applications with minimal Printed Circuit Board(PCB) requirements. ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIOorI2C / UART interfaces.

3. MAX30100

A pulse oximeter is basically a device which can measure your pulse and oxygen saturation in your blood. Usually, this sensor consists of two LEDs emitting light: one in red spectrum(650nm) and the other one in Infrared (950nm). This sensor is placed on your finger or earlobe, essentially anywhere where the skin is not too thick so both light frequencies can easily penetrate the tissue. Once both are shined through your finger for example, the absorption is measure with a photodiode. And depending on the amount of oxygen you have in your blood the ratio between the absorbed red light and IR led will be different. oxygen concentration can be measured by calculating the ratio between absorbed light from IRLED and Red LED.

Figure 3.4.1.3 MAX30100

4. NEO-6M GPS

The u-box NEO-6M GPS engine on these modules is quite a good one, and it also has high sensitivity for indoor applications. Furthermore, there's one MS621FE-compatible rechargeable battery for backup and EEPROM for storing configuration settings. The module works well with a DC input in the 3.3- to 5-V range. The GPS modules are based on the u-box NEO-6M GPS engine. The type number of the NEO-6M is NEO-6M-0-001, and its ROM/FLASH version is ROM 7.0.3 (PCN reference UBX-TN-11047- 1). The NEO-6M module includes one configurable UART interface for serial communication, but the default UART (TTL) baud rate here is 9,600. Because the GPS signal is right-hand circular polarized (RHCP), The NEO-6M module includes one configurable UART interface for serial communication, but the default UART (TTL) baud rate here is 9,600. Because the GPS signal is right-hand circular polarized (RHCP), the style of the GPS antenna will be different from the common whip antennas used for linear polarized signals. The most popular antenna type is the patch antenna.

Figure 3.4.1.4 NEO-6M GPS

5. OLED

OLEDs are simple solid-state devices (more of an LED) comprised of very thin films of organic compounds in the electro-luminescent layer. These organic compounds have a special property of creating light when electricity is applied to it. The organic compounds are designed to be in between two electrodes. Out of these one of three electrodes should be transparent. The result is a very bright and crispy display with power consumption lesser than the usual LCD and LED.

Figure 3.4.1.5 OLED

6. BUZZER

Buzzers can be both fun and useful in electric circuits. We'll use them a lot in Make Crate projects, so let's take a look at what is going on inside a buzzer to produce sound. The buzzer consists of an outside case with two pins to attach it to power and ground. Inside is a piezo element, which consists of a central ceramic disc surrounded by a metal (often bronze) vibration disc. When current is applied to the buzzer it causes the ceramic disk to contract or expand. This then causes the surrounding disc To vibrate. That's the sound that you hear.

Figure 3.4.1.6 BUZZER

7. TILT SENSOR

Tilt sensors allow you to detect orientation or inclination. They are small, inexpensive, low power and easy-to-use. If used properly, they will not wear out. Their simplicity makes them popular for toys, gadgets and appliances. Sometimes they are referred to as "mercury switches", "tilt switches" or "rolling ball sensors" for obvious reasons. They are usually made by a cavity of some sort (cylindrical is popular, although not always) and a conductive free mass inside, such as a blob of mercury or rolling ball. One end of the cavity has two conductive elements (poles). When the sensor is oriented so that that end is downwards, the mass rolls onto the poles and shorts them, acting as a switch throw.

Tilt switches used to be made exclusively of mercury, but are rarer now since they are recognized as being extremely toxic. The benefit of mercury is that the blob is dense enough that it doesn't bounce and so the switch is not susceptible to vibrations. On the other hand, ball type sensors are easy to make, wont shatter, and pose no risk of pollution. While not as precise or flexible as a full accelerometer, tilt switches can detect motion or orientation simply. Another benefit to them is that the big ones can switch power on their own.

Figure 3.4.1.7 TILT SENSOR

8. BMP180

The BMP180 is the next generation of sensors from Bosch, and replaces the BMP085. you can use our BMP085 tutorial and any example code/libraries as a drop-in replacement. The XLCR pin is not physically present on the BMP180 so if you need to know that data is ready you will need to query the I2C bus. This board is 5V compliant - a 3.3V regulator and a i2c level shifter circuit is included so you can use this sensor safely with 5V logic and power. The BMP180 is a piezo resistive sensor that detects pressure. Piezo resistive sensors are made up of a semiconducting material (usually silicon) that changes resistance when a mechanical force like atmospheric pressure is applied. The BMP180 measures both pressure and temperature, because temperature changes the density of gasses like air. At higher temperatures, air is not as dense and heavy, so it applies less pressure on the sensor. At lower temperatures, air is denser and weighs more, so it exerts more pressure on the sensor. The sensor uses real-time temperature measurements to compensate the pressure readings for changes in air density. The BMP180 outputs an uncompensated temperature (UT) value and an uncompensated pressure (UP) value. The temperature measurement is taken first, followed by a pressure measurement.

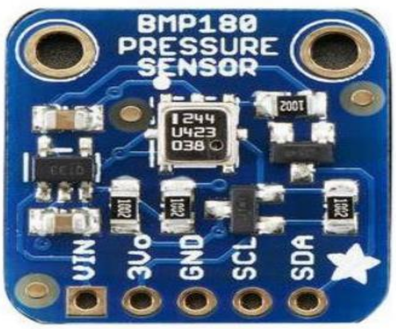

Figure 3.4.1.8 BMP180

CHAPTER 4

SYSTEM ANALYSIS & DESIGN

4.1 Analysis

The Parkinson's fall detection system utilizing the Internet of Things (IoT) and machine learning (ML) is a groundbreaking solution designed to enhance the safety and well-being of individuals affected by Parkinson's disease. By integrating IoT sensors into the environment, such as wearable devices or motion sensors, real-time data can be collected to monitor the movements and activities of patients. Leveraging ML algorithms, this data can be analyzed to identify patterns indicative of a fall event. The system employs advanced ML techniques to distinguish between normal movements and falls, reducing false alarms and ensuring accurate detection. Once a fall is detected, the system triggers automated notifications or alerts to caregivers, enabling prompt assistance and medical intervention. This innovative fusion of IoT and ML holds immense potential in revolutionizing the management and care of Parkinson's patients by providing reliable fall detection, enhancing their safety, and promoting timely support.

4.2 System Design

As IoT-based devices have become more popular in the medical field, many wearable sensors have been developed for the WFDS. Hence, these advancements bring more possibilities to help caretakers in protecting their elders by improving the outcome of the wearable sensor by using ESP32 for the fall detection system. The system consists of an accelerometer node and receiving node. The oximeter sensor is implemented using the MAX30100, it measures the pulse and oxygen saturation levels in the patient's blood. The buzzer which is used acts as an alarm for notifying the caretakers about the fallen patient. OLED displays the decision made. This proposed system has two main parts, the sensor-board, and the monitoring system. The first part is the sensor board or fall detection device. It consists of an accelerometer sensor and the ESP32. Triaxial accelerometer (ADXL337) with a three-axis magnetic field. It detects the kinematic signal from the daily activities of the patient. The ESP32 used acts as a coordinator to send the kinematic signal detected to Firebase Cloud.

Here ESP32 is a microcontroller that calculates the fall detection with risk level measurement by comparing it with the threshold values. ESP32 is a device that is designed for IoT and embedded systems. Low cost, low power, and abilities of Wi-Fi are the main factors for selecting ESP32 in the required prototype. The second major part is the real-time data analytic and data storage system, which is used to record the kinematic signal to Cloud storage. This part alone acts as a controller. It consists of Firebase Cloud, acting as real-time storage which helps in making a prototype, scaling the IOT-based project, and managing data with cloud computing technology. The Firebase Cloud used here acts as a platform to process the kinematic signal received from the ESP32. Firebase Cloud monitors and graphs the data. Hence, the fall detection data will be sent to the web server. The respective hospital or caretakers can monitor the data signal and diagnose medical information related to fall events on their own.

Firebase is the backend platform for building Android, web, and IOS applications. It supports a real-time database, various APIs, many authentication types, and a hosting platform. It can power the app's backend, including user authentication, data storage, and static hosting. It builds cross-platform native mobile, and web apps with IOS, android, and JavaScript SDKs. It also connects Firebase to the existing backend using server-side libraries or REST API authentication. Users can use anonymous, passwords or different social authentications. The Advantages are - It is simple to use and user-friendly. Data is real-time, which means that any change will automatically be updated to connected clients. Firebase Cloud always offers a simple control dashboard.

4.2.1 System Architecture Diagram

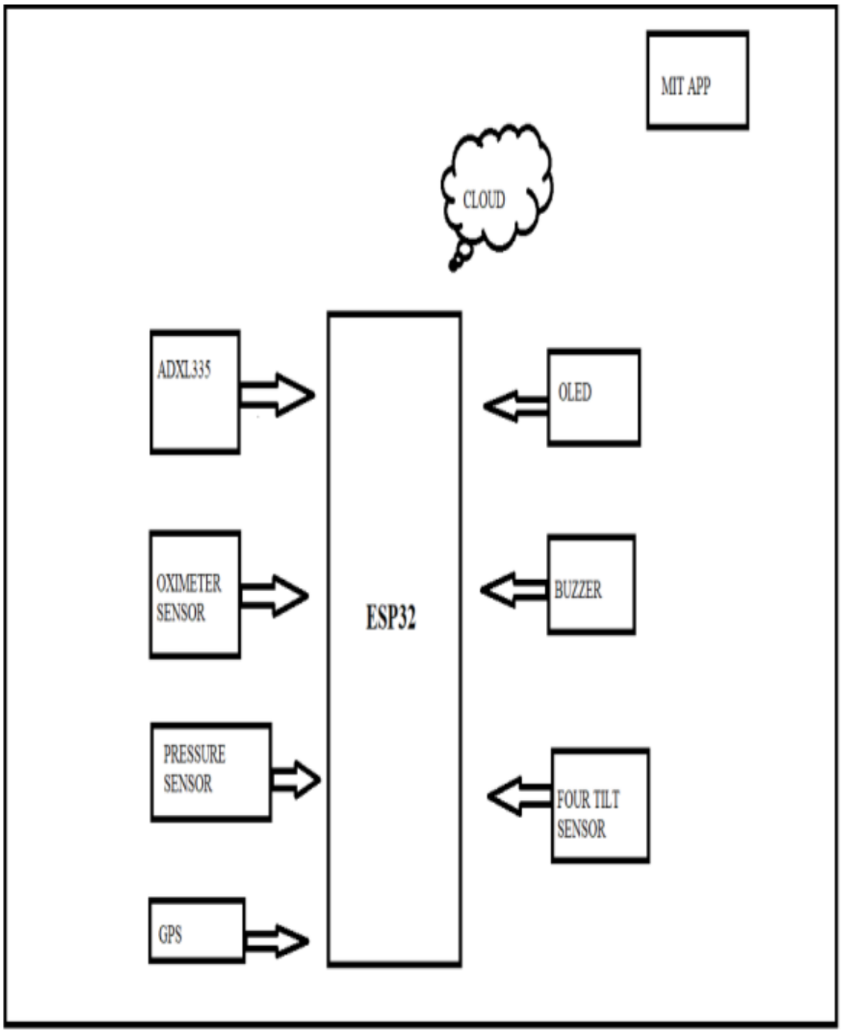

Figure 4.2.1(a) System Architecture Diagram of the IOT Model

The Parkinson's fall detection system using IOT employs a comprehensive architecture that combines various interconnected components to enable efficient monitoring and prompt response to falls. At its core is a network of wearable sensors worn by Parkinson's patients, capturing real-time data related to their movements, balance, and posture. These sensors transmit the collected information to a gateway device, such as a smartphone or a dedicated IoT hub, which acts as a central hub for data processing. Once a fall is detected, an alert is generated and transmitted to a cloud-based server, where it is processed and sent to a caregiver or healthcare professional via a mobile application or web interface. Additionally, the system may integrate with existing home automation devices or emergency services, enabling rapid response and assistance during fall emergencies. This architecture ensures continuous monitoring and timely intervention, enhancing the safety and well-being of Parkinson's patients.

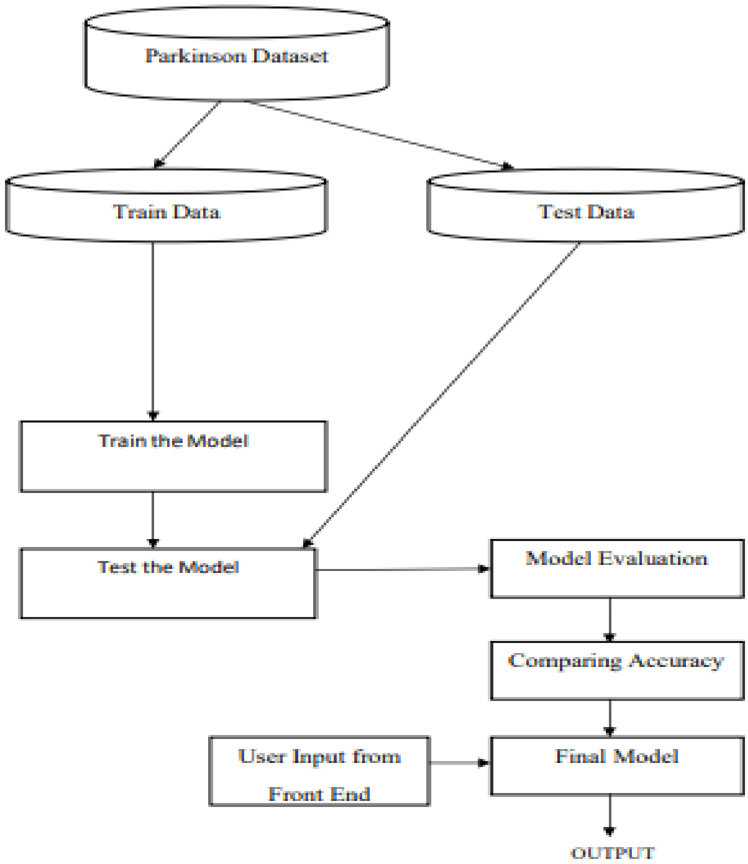

Figure 4.2.1(b) System Architecture Diagram of the ML Model

The Data Collection component involves capturing voice recordings from individuals, particularly focusing on speech patterns, vocal characteristics, and any abnormalities that may be associated with Parkinson's disease. These voice recordings can be obtained through various means, such as dedicated voice recording devices or smartphone applications. Then the pre processed data is classified into training and testing data, Once the model is trained, it can be utilized for Parkinson's Disease Detection using voice data. New voice recordings can be input into the model, and the ML algorithm predicts whether the individual exhibits signs of Parkinson's disease based on the learned patterns and features from the training data.The Detection Result Output component provides the final outcome of the system's analysis. It may present the results as a binary classification (indicative or non-indicative of Parkinson's disease) or provide a probability score reflecting the likelihood of Parkinson's disease presence based on the voice data analysis.

4.2.1.1 Data Flow Diagram

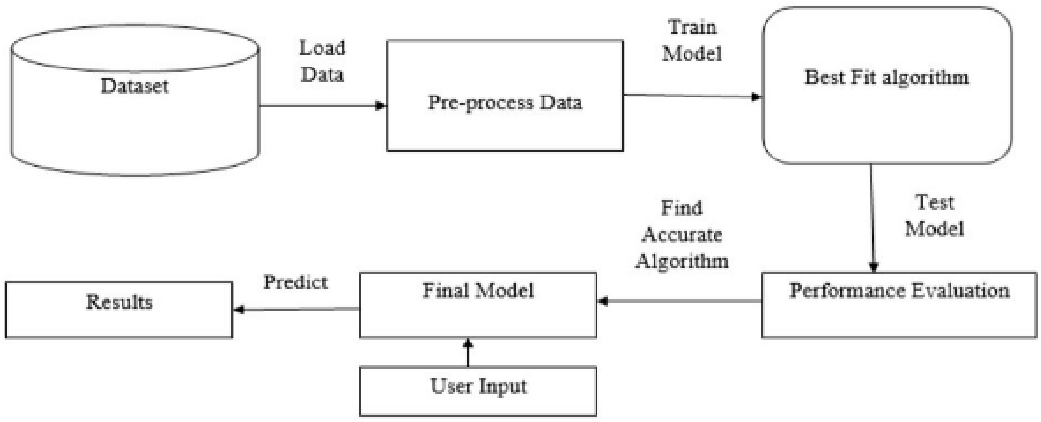

Figure 4.2.1.1(a). Data Flow Diagram of the ML Model.

i. Parkinson dataset is taken and loaded.
ii. The data is preprocessed in order to fill the missing values and convert strings to numeric data etc. in the dataset etc to increase the accuracy.
iii. The model is built using machine learning algorithms like Linear Regression, XGB Regressor and Random Forest Regressor.
iv. The model is trained with the preprocessed data.
v. The model is tested and accuracy is calculated for different ML algorithms.
vi. The algorithm with best accuracy is finalized and that model will predict Parkinson disease based on user given new data from front end.

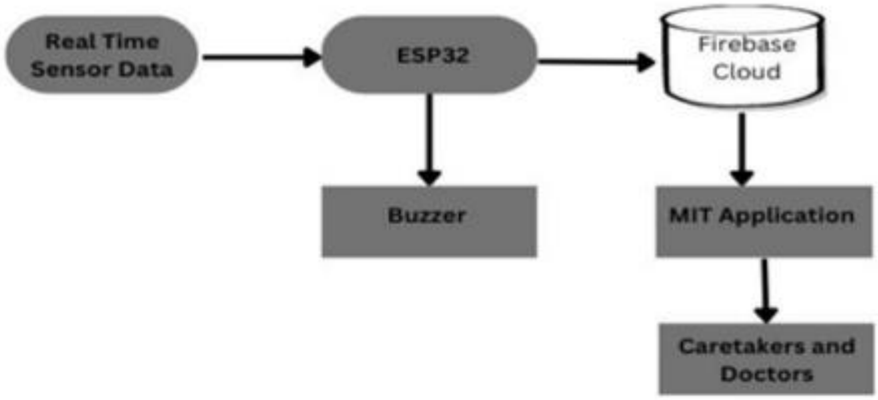

Figure 4.2.1.1(b) Data Flow Diagram of the IOT Model.

1. **External Entities:**
   i. Patients: Individuals with Parkinson's disease who wear IoT devices for fall detection.
   ii. Caregivers: Individuals responsible for monitoring the patients and receiving fall notifications.
2. **IoT Devices:**
   i. Fall Detection Sensor: An IOT device worn by the patient that captures motion data and detects falls.
   ii. Data Transmitter: Transmits the sensor data wirelessly to the system.
3. **Data Collection:**
   i. Sensor Data Receiver: Receives the sensor data from the IoT devices.
4. **Fall Detection and Notification:**
   i. Fall Detection Module: Utilizes the ML model to analyze preprocessed data in real-time and detect falls.
   ii. Fall Notification System: Alerts the caregivers about fall incidents through various means.
5. **Caregiver Response:**
   i. Caregiver Monitoring Interface: An interface accessible by caregivers to view fall notifications and patient status.
   ii. Caregiver Acknowledgement: Caregivers can acknowledge the fall notification, ensuring they are aware of the incident.

4.2.1.2 Flow Chart

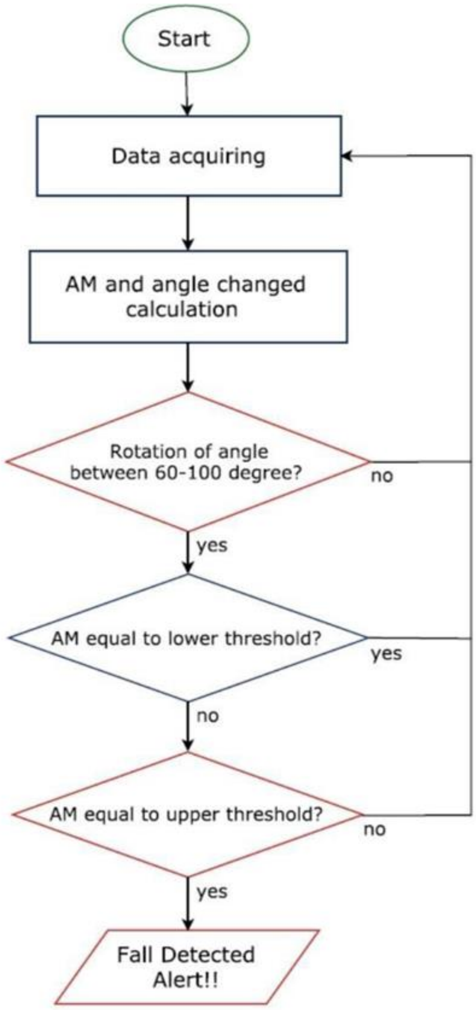

Figure 4.2.1.2(a) Flow Chart of the IOT Model.

The total acceleration as well as the rotation angle value are calculated by the threshold. The generated results are compared to the thresholds that were set using the collected data As a result, the multidimensional fall detection technique is used to provide effective threshold values. The system's fall detection mechanism is depicted as a flowchart. This method creates three triggers for collecting events that occur throughout the system's operation. The angle of human movement is calculated using the first trigger, and the angle of human movement is calculated using additional triggers. The flowchart illustrated in Fig depicts the system's fall detection method. This technique establishes three triggers for gathering events that occur throughout the system's operation. The first trigger is used to calculate the human movement angle, while the remaining triggers are utilized to determine the amplitude of the accelerated movement. The acceleration magnitude is given by,

$$\boldsymbol{AM} = (\boldsymbol{Roll}^2 + \boldsymbol{Heading}^2 + \boldsymbol{Pitch}^2)^{1/2} \ldots \textit{equation 1}$$

The Roll is the angle rotation around the x-axis that describes the human body's sideslip angle to the right and left. The pitch indicates the angle of rotation around the y-axis, which depicts the forward and backward motion of the body; the heading indicates the angle of rotation around the z-axis, which depicts the right and left corners of the human body. The rotational angle, or angular Kinematics of movement, is given by the equation as

$$\boldsymbol{a} * \boldsymbol{b} = |\boldsymbol{a}|.|\boldsymbol{b}| \cos \boldsymbol{\theta} \ldots \textit{ equation 2}$$

The arranged angle is calculated by equation as illustrated below,

$$\boldsymbol{\theta} = \boldsymbol{cos}^{-1}|\boldsymbol{a} * \boldsymbol{b}||\boldsymbol{a}||\boldsymbol{b}| \ldots \textit{ equation 3}$$

Where a (ax, ay, az) represents the device's rapid value, b(bx, by, bz) represents the elderly's reference position when using the device vertically, and is the angle rotation between the vectors a and b. The system will calculate the rotation of the angle altered value once it obtains the accelerated magnitude (AM) value. The fall detection algorithm compares the generated values to the preset degree of rotation angle and threshold values to check the fall detection result. If the degree of rotation angle exceeds 60 - 100, it means the body has lost its balance, resulting in an unintentional fall.

The estimated AM value will be compared to a predetermined threshold value in this algorithm. If the computed accelerated magnitude exceeds the lower threshold, the algorithm will activate trigger 1 to signify that no fall event has been detected because the danger of a fall is too low, or the elderly are engaged in activities that are not likely to result in a fall. The method will trigger if the calculated accelerated magnitude exceeds the upper threshold, indicating that a fall event has been identified. It shows that a genuine fall event has occurred. It will then send a notification to the caretaker, indicating that a rescue is required. It shows that a genuine fall event has occurred. It will then send a notification to the caretaker, indicating that a rescue is required.

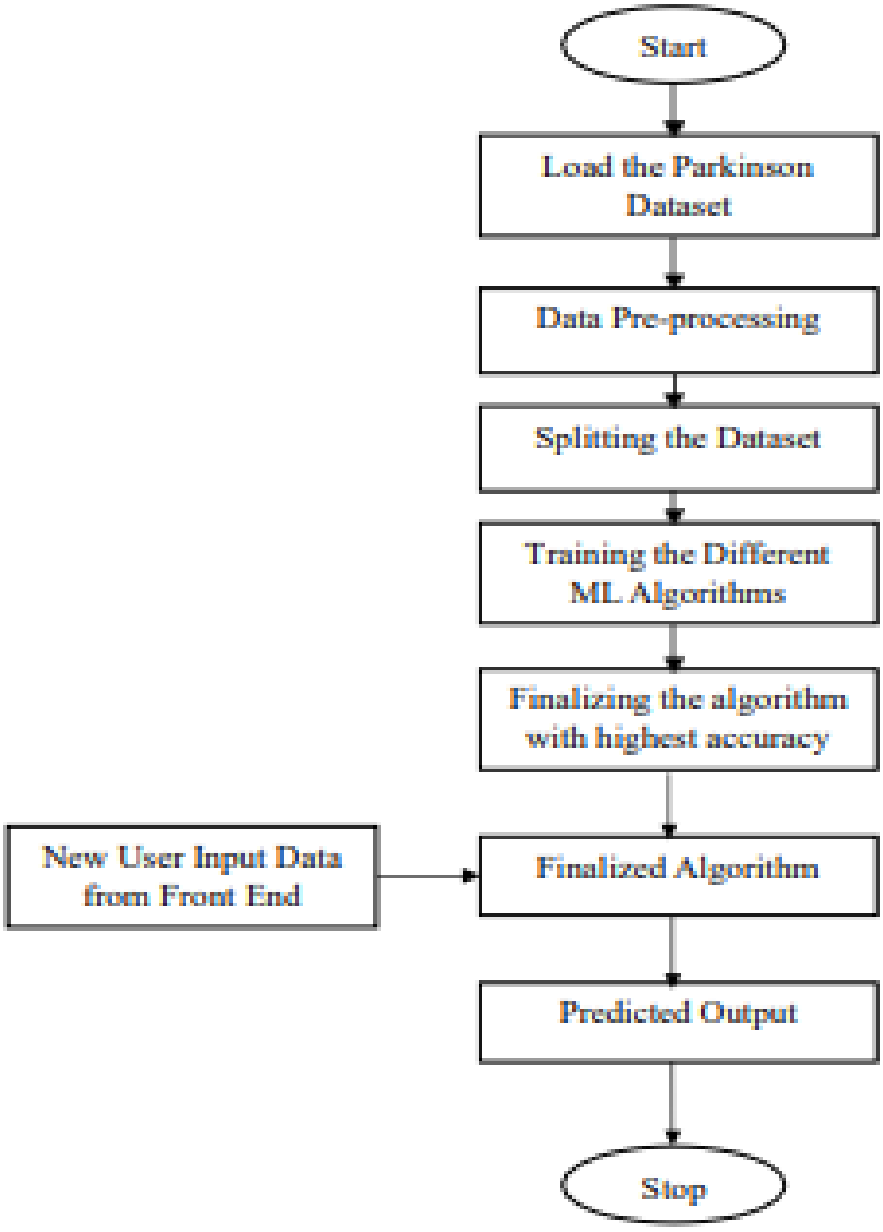

Figure 4.2.1.2(b) Flow Chart of the ML Model.

4.2.1.3 Use Case Diagram

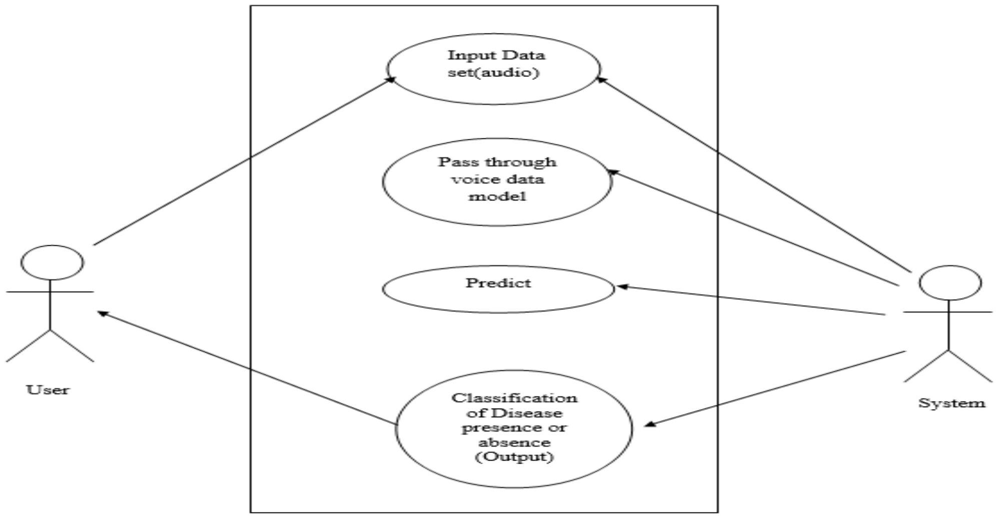

Figure 4.2.1.3(a) Use Case diagram of ML model.

- The system will collect the data from the user.
- The system will analyze the collected data.
- The system will do pre-processing and model building.
- The built model is evaluated and tested for accuracy.
- The user will finalize the model with best accuracy.
- The user will give new data to the system and the model will predict results and display.

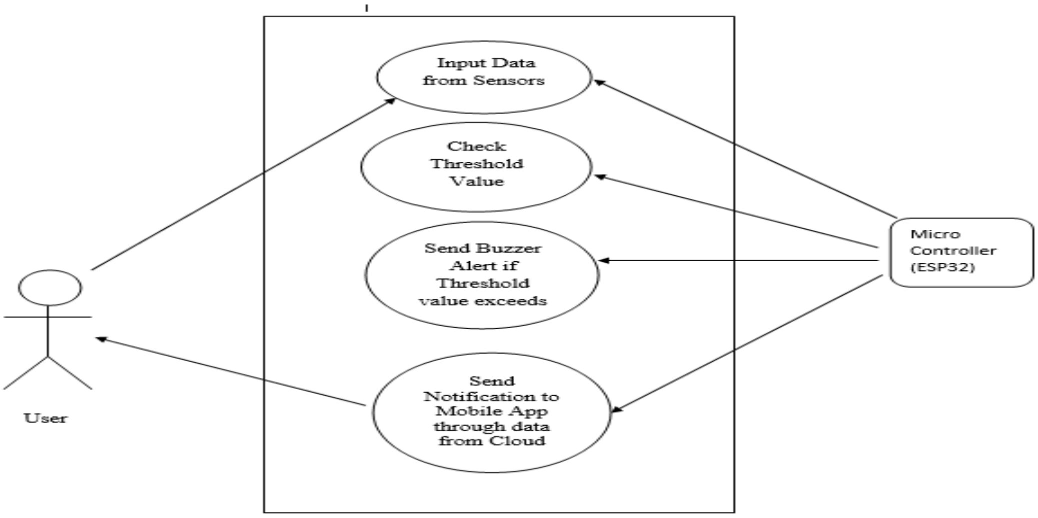

Figure 4.2.1.3(b) Use Case diagram of IOT model.

1. Users: The diagram would typically include three main types of users:
   i. Parkinson's Patients: Individuals who have Parkinson's disease and wear the IoT devices that detect falls and collect data.
2. IOT Devices: These are physical devices worn by Parkinson's patients that capture data related to falls. These devices usually include sensors (e.g., accelerometers, gyroscopes) to detect motion and changes in orientation. The devices may also have wireless communication capabilities to transmit data to the system.
3. Fall Detection System: This represents the core of the solution and incorporates the IoT devices, ML algorithms, and data processing capabilities. It consists of the following components:
   i. Data Collection: Responsible for collecting data from the IoT devices, including movement patterns and sensor readings.
   ii. Fall Detection Algorithm: Utilizes ML techniques to analyze the collected data and identify patterns associated with falls. This component classifies the data and triggers alerts when a fall is detected.
   iii. Alert Generation: Once a fall is detected, this component generates alerts and notifications to inform the appropriate users (caregivers and healthcare professionals) about the event.
   iv. Data Storage: This component is responsible for securely storing the collected data for future analysis and reference.
   v. Reporting: Provides functionality for generating reports and statistics based on the fall-related data. Healthcare professionals can utilize this feature to gain insights and make informed decisions.
4. External Systems: These represent external entities or systems that interact with the Parkinson's fall detection system. For example:
   i. Mobile Applications: Mobile apps can be used by caregivers to receive alerts and view the status of Parkinson's patients. These apps may also provide additional features such as communication with healthcare professionals and access to historical data.
   ii. Electronic Health Record (EHR) Systems: Integration with existing healthcare systems allows healthcare professionals to access patient data and synchronize the fall detection.

4.2.1.4 Sequence Diagram

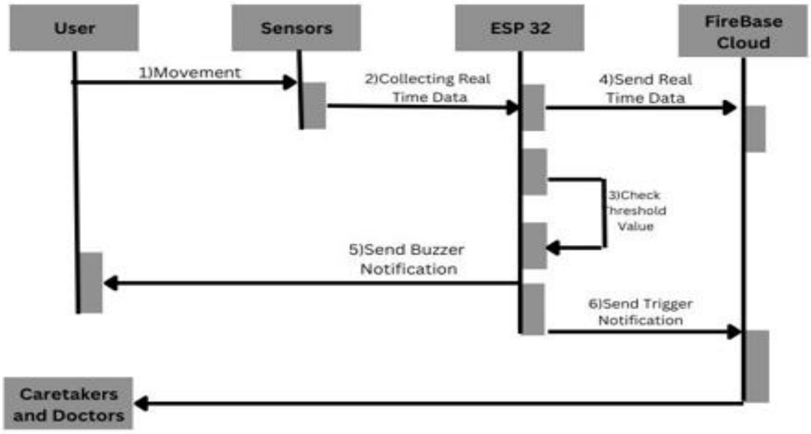

Figure 4.2.1.4(a) Sequence diagram of IOT Model.

User Interaction:

The sequence diagram starts with the user interacting with a wearable IOT device, such as a smartwatch or a sensor-embedded garment. The user may trigger an action or simply wear the device, initiating the monitoring process.

Sensor Data Collection:

The wearable IoT device collects sensor data related to the user's movements, such as accelerometer and gyroscope readings. The device continuously captures and transmits this data to the central system for analysis.

Data Transmission:

The wearable IoT device transmits the collected sensor data to the cloud or a central server for processing and analysis. This transmission typically occurs through wireless communication protocols, such as Wi-Fi or Bluetooth.

Data Preprocessing:

The central system receives the transmitted sensor data and performs preprocessing tasks, such as filtering noise, normalizing values, or segmenting data into time intervals.

Fall Detection Algorithm:

The preprocessed sensor data is then fed into a machine learning algorithm designed to detect fall incidents. This algorithm analyzes the patterns and characteristics of the data to determine if a fall event has occurred.

Fall Detection Result:

Based on the analysis, the fall detection algorithm produces a result indicating whether a fall has been detected or not. If a fall is detected, the system proceeds to the next steps for appropriate actions.

Alert Generation:

In case of a fall detection, the system generates an alert or notification to inform the relevant parties, such as caregivers, family members, or emergency services. This alert may be sent through various means, including mobile notifications, emails, or phone calls.

Emergency Response:

Upon receiving the alert, the appropriate entities, such as emergency services or caregivers, initiate the necessary response actions. These actions may involve contacting the user, dispatching medical assistance, or notifying designated contacts.

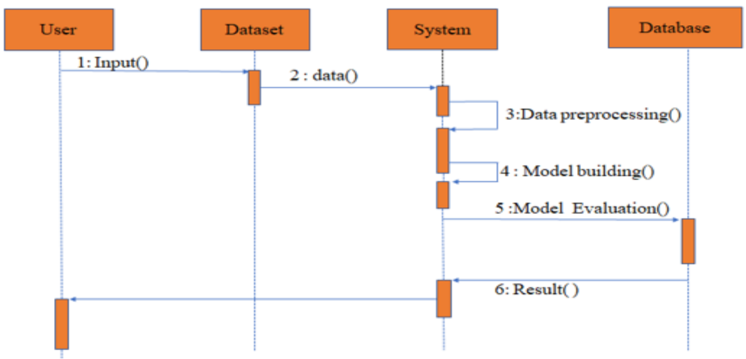

Figure 4.2.1.4(b) Sequence diagram of ML Model.

- The user will give dataset as input to the system.
- The system will store the dataset given by the user in its database.
- The system will do pre-processing of the data stored.
- The model is built using various ML algorithms and trained using preprocessed data.
- The model is evaluated and the algorithm with best accuracy is finalized.
- The finalized model will predict the results.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

The implementation of a Parkinson's fall detection system using IOT (Internet of Things) technology aims to address the safety concerns and challenges faced by individuals with Parkinson's disease. The Parkinson's fall detection system utilizes IOT devices and sensors to monitor and detect falls in real-time. By deploying various sensors, the system can collect data on movement, posture, and other relevant parameters. This data is then analyzed and processed using algorithms to identify instances of falls and distinguish them from other activities like sitting or lying down.

5.2 Implementation

Since the IOT in medical devices has become more popular, many wearable sensors have been developed for fall detection systems. But nowadays the sensors that fuel the IoT are getting cheaper, smaller, and have higher performances. Therefore, these developments bring more possibilities to help caregivers protect their elder by improving on the wearable sensor using ADXL337 and NodeMCU-ESP32 for elderly fall detection system This proposed system has two main parts, the sensor board, and the monitoring system The sensor board consists of Microcontroller (ESP32), Tilt sensor, Buzzer, OLED, Max30100, Bmp180, Accelerometer sensor (ADXL337), Global Positioning System (GPS) neo6m. The main component of this system is the accelerometer sensor (ADXL337). ESP32 microcontroller is used which consists of an inbuilt Wi-Fi.

Generally, the proposed system consists of three main parts which are the sensor board and monitoring system. The first main part is the sensor board or fall detection device. It consists of a triaxial accelerometer and the NodeMCU-ESP32. It is used to detect the kinematics signal from the daily living of the elderly. Consequently, the NodeMCU-ESP32 acts as a coordinator to send the kinematics signal detected from the ADXL337 to Firebase Cloud. Moreover, the NodeMCU-ESP32 acts as a microcontroller to calculate the fall detection result with risk level measurement by comparing it with the threshold values. Further, the NodeMCU-ESP32 is a product designed for the IoT and embedded systems. The low cost, low power, and capabilities of Wi-Fi and Bluetooth are the main factors for selecting this device for the proposed prototype.

5.3 Overview of System Implementation

The Parkinson's fall detection system utilizing IoT and ML involves wearable sensors that capture motion data from individuals with Parkinson's disease. This data is processed and transmitted to an IoT gateway, which securely sends it to a cloud platform. In the cloud, machine learning models analyze the data in real-time, detecting fall events by comparing sensor data with predefined patterns. Alerts are generated and sent to caregivers or healthcare professionals through a mobile or web application. The system provides insights into long-term movement patterns and can be integrated with existing healthcare systems for scalability and improved care management.

5.3.1 System Implementation

1. Hardware Components:

   i. Wearable Sensors: The system utilizes wearable sensors, such as accelerometers and gyroscopes, to collect motion data from the user. These sensors are typically embedded in devices like smartwatches or specialized wearable devices.

   ii. IOT Gateway: An IOT gateway acts as a bridge between the wearable sensors and the cloud platform. It collects the sensor data and transmits it securely to the cloud for further processing.

2. Data Collection and Transmission:

   i. Sensor Data Acquisition: The wearable sensors continuously capture motion data from the user's movements. This data includes acceleration, orientation, and angular velocity information.

   ii. Preprocessing and Filtering: The raw sensor data may contain noise or irrelevant information. Preprocessing techniques, such as noise filtering and feature extraction, are applied to clean and extract relevant features from the data.

iii. Data Transmission: The filtered and processed data is sent to the IoT gateway through wireless communication protocols like Bluetooth or Wi-Fi. The gateway securely transmits the data to the cloud platform.

iv. Dataset features:

a) MDVP:Fo(Hz) - Average vocal fundamental frequency

b) MDVP:Fhi(Hz) - Maximum vocal fundamental frequency

c) MDVP:Flo(Hz) - Minimum vocal fundamental frequency

d) MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP - Several measures of variation in fundamental frequency

e) MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Several measures of variation in amplitude

f) NHR,HNR - Two measures of ratio of noise to tonal components in the voice

g) status - Health status of the subject (one) - Parkinson's, (zero) - healthy

h) RPDE,D2 - Two nonlinear dynamical complexity measures

i) DFA - Signal fractal scaling exponent

j) spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation

3. Cloud Platform:

i. Data Storage: The cloud platform receives and stores the sensor data in a database for further analysis and long-term storage.

ii. Machine Learning Models: ML models are developed and deployed on the cloud platform to analyze the collected data. These models are trained to recognize patterns and detect falls based on the sensor data.

iii. Real-time Analysis: The ML models process the incoming data in real-time to identify fall events. They can distinguish between normal activities and fall-related movements by comparing the sensor data with predefined fall patterns.

iv. Alert Generation: When a fall event is detected, the system generates alerts or notifications, which can be sent to caregivers, healthcare professionals, or emergency services, depending on the severity of the situation.

4. User Interface and Monitoring:

   i. Mobile or Web Application: The system provides a user interface through a mobile or web application. Users, caregivers, or healthcare professionals can access the application to monitor the user's activities and receive fall notifications.

   ii. Historical Data Analysis: The system can also provide insights into the user's overall movement patterns and fall occurrences over time. This information can be used for long-term monitoring and analysis of the user's condition.

5. Integration:

   i. Integration with Existing Systems: The fall detection system can be integrated with existing healthcare systems, electronic health records (EHR), or remote patient monitoring platforms to enhance the overall care management process.

5.3.1.1 Programming Languages

i. C

The C programming language is widely used in IoT (Internet of Things) development due to its efficiency, low-level control, and compatibility with various hardware platforms. C provides direct access to memory, allowing developers to optimize code for resource-constrained IoT devices.C is often used to write firmware for embedded systems and microcontrollers. These small computing devices, found in IoT devices, require lightweight and efficient code to operate with limited resources such as memory and processing power. C's low-level control allows developers to directly manipulate hardware registers and efficiently utilize system resources, making it well-suited for these resource-constrained environments. C provides a wide range of libraries and APIs (Application Programming Interfaces) that facilitate communication protocols commonly used in IoT, such as MQTT (Message Queuing Telemetry Transport) or CoAP (Constrained Application Protocol). These libraries enable developers to establish connections, exchange data, and interact with IoT platforms, cloud services, or other devices on the network.

ii. Python

Python is a widely used programming language in the field of Machine Learning (ML) due to its versatility, simplicity, and rich ecosystem of libraries and frameworks. Python provides a conducive environment for ML tasks, offering a balance between ease of use and powerful capabilities.

Python's ML ecosystem also includes popular deep learning frameworks like Tensorflow, keras and scikit-learn . These frameworks offer high-level abstractions and APIs for building and training deep neural networks, enabling complex tasks like image recognition, natural language processing, and sequence modeling. They provide efficient computation on both CPUs and GPUs, allowing for scalable and accelerated ML model training.

5.3.2 Libraries Used

i. **Flask:** Flask is a popular web framework for building web applications in Python. It provides a lightweight and flexible way to create web services and APIs. One of the essential features of Flask is its ability to handle HTTP requests and responses efficiently.

ii. **Flask_restful:** It extends Flask and simplifies the development of RESTful APIs in Python by providing a convenient way to define resources and handle HTTP methods

iii. **Numpy:** NumPy is a widely-used Python library for numerical computing that provides efficient data structures and functions for working with arrays and matrices, enabling high-performance mathematical and scientific computations. It is a fundamental tool in data analysis, machine learning, and scientific research.

iv. **Keras.models**: The keras.modules library is a part of the Keras deep learning framework, which provides a modular and user-friendly interface for building and training neural networks. This library offers a collection of high-level neural network building blocks called modules, which can be combined to create complex network architectures

v. **Flask_cors:** The flask_cors library is a Flask extension that simplifies Cross-Origin Resource Sharing (CORS) handling in Flask applications. CORS is a security mechanism implemented by web browsers to control cross-origin requests, which are requests made from a different domain, port, or protocol

vi. **Pickle:** The pickle library in Python provides a convenient way to serialize and deserialize Python objects. Serialization refers to the process of converting an object into a format that can be stored or transmitted, while deserialization is the reverse process of reconstructing the object from the serialized form.

vii. **Joblib:** The joblib library in Python provides utilities for efficiently serializing and deserializing Python objects, especially for objects that contain large amounts of data.

CHAPTER 6

PSEUDO CODE

1. Data Preparation:
   i. Load the voice dataset containing features extracted from voice recordings.
   ii. Separate the features (X) and corresponding labels (Y) for training and testing.
2. Feature Scaling:
   i. Apply feature scaling to normalize the input data.
   ii. Common techniques include standardization or normalization.
3. Model Selection and Training:
   i. Choose a suitable machine learning algorithm for classification, such as Decision Tree, Gaussian Naïve Bayes Support Vector Machines (SVM, Logistic Regression and Random Forest.
   ii. Split the data into training and testing sets.
   iii. Train the chosen model using the training data.
4. Model Evaluation:
   i. Evaluate the trained model's performance using appropriate metrics like accuracy, precision, recall, or F1 score.
   ii. Use the testing dataset to assess the model's generalization ability.
5. Hyperparameter Tuning (Optional):
   i. Perform hyperparameter tuning to optimize the model's performance.
   ii. Use techniques like grid search or random search to find the best combination of hyperparameters.
6. Prediction:
   i. Once the model is trained and evaluated, use it to make predictions on new, unseen data.
   ii. Preprocess the voice data by extracting the required features.

iii. Apply the same feature scaling techniques used during training.

iv. Feed the preprocessed data to the trained model for prediction.

7. Interpretation and Reporting:

i. Analyze the model's predictions and interpret the results.

ii. The results are compared and best one is picked

CHAPTER 7

TESTING

7.1 Test Cases

1. Normal Activity Test:

i. Simulate normal daily activities of a person without Parkinson's disease.

ii. Ensure the system does not trigger false alarms or detect falls during regular movements, such as walking, sitting, or lying down.

2. Fall Event Test:

i. Simulate a fall event by intentionally creating a controlled fall scenario.

ii. Verify that the system accurately detects the fall and triggers an appropriate alarm or alert.

3. Non-Fall Movement Test:

i. Perform movements that may mimic falls but are not actual falls, such as sudden gestures or quick movements.

ii. Ensure the system does not mistakenly identify these non-fall movements as falls.

4. Multiple Persons Test:

i. Test the system's ability to differentiate between falls of the user with Parkinson's disease and falls of other individuals in the vicinity.

ii. Verify that the system accurately detects falls specific to the user and disregards falls of others.

5. Environmental Variation Test:

i. Introduce variations in the environment, such as changes in lighting conditions or background noise levels.

ii. Evaluate the system's robustness to environmental factors and ensure accurate fall detection despite these variations.

6. False Negative Test:

i. Intentionally perform falls or simulate fall events that are not detected by the system.

ii. Check the system's sensitivity and identify any instances where falls are missed or not detected accurately.

7. False Positive Test:

i. Perform normal activities that may trigger false alarms, such as abrupt movements or sudden posture changes.

ii. Verify that the system maintains a low rate of false alarms and does not unnecessarily raise alerts.

8. Real-World Scenarios Test:

i. Conduct extensive field tests with individuals having Parkinson's disease in real-world settings, such as their homes or care facilities.

ii. Evaluate the system's performance, accuracy, and reliability in detecting falls and providing timely alerts.

9. Emergency Response Test:

i. Integrate the system with emergency response services or caregivers.

ii. Validate the system's ability to promptly alert and notify the appropriate parties during a fall event.

10. Long-Term Monitoring Test:

i. Test the system's stability and performance over an extended period, such as several weeks or months.

ii. Assess the system's ability to consistently detect falls and maintain accurate performance over time.

11. Test Case: Normal Activity

i. Description: Simulate normal activities, such as walking, sitting, and lying down.

ii. Expected Outcome: The system should not detect a fall and remain in the normal activity state.

12. Test Case: Simulated Fall Event

i. Description: Simulate a fall event by triggering the sensors to indicate a sudden change in acceleration and orientation.

ii. Expected Outcome: The system should accurately detect the fall event and trigger an alarm or alert.

13. Test Case: False Positive Detection

i. Description: Perform activities that may mimic fall-like movements, such as sudden bending or quick movements.

ii. Expected Outcome: The system should not incorrectly detect these activities as falls and should remain in the normal activity state.

14. Test Case: Fall Detection Accuracy

i. Description: Simulate various fall scenarios with different intensities and impact forces.

ii. Expected Outcome: The system should consistently and accurately detect falls of varying magnitudes and trigger appropriate alarms or alerts.

15. Test Case: Environmental Variations

i. Description: Test the system's performance under different environmental conditions, such as varying lighting or background noise levels.

ii. Expected Outcome: The system should maintain accurate fall detection irrespective of environmental variations that may impact sensor readings.

16. Test Case: Network Disconnection and Reconnection

i. Description: Temporarily disconnect the IoT network and then reconnect it.

ii. Expected Outcome: The system should handle network disruptions gracefully and resume normal operation once the connection is restored.

17. Test Case: User Interface Interaction

i. Description: Interact with the user interface, such as checking real-time fall detection status, reviewing historical fall events, and adjusting system settings.

ii. Expected Outcome: The user interface should provide accurate and up-to-date information, allow easy navigation, and respond correctly to user interactions.

18. Test Case: Integration with Emergency Services

i. Description: Trigger a fall event and verify if the system correctly notifies emergency services with relevant information.

ii. Expected Outcome: The system should effectively communicate with emergency services, providing accurate location details and necessary information to facilitate timely assistance.

19. Test Case: User Feedback and Usability Testing

i. Description: Collect feedback from users, caregivers, or healthcare professionals regarding the system's usability, effectiveness, and any potential improvements.

ii. Expected Outcome: The system should be intuitive, user-friendly, and meet the needs and expectations of the intended users.

These test cases cover various aspects of the Parkinson's fall detection system, including accurate fall detection, handling false positives, robustness under different conditions, user interface functionality, integration with emergency services, and overall usability. Conducting comprehensive testing helps ensure the system's reliability, accuracy, and effectiveness in real-world scenarios.

CHAPTER 8

RESULTS

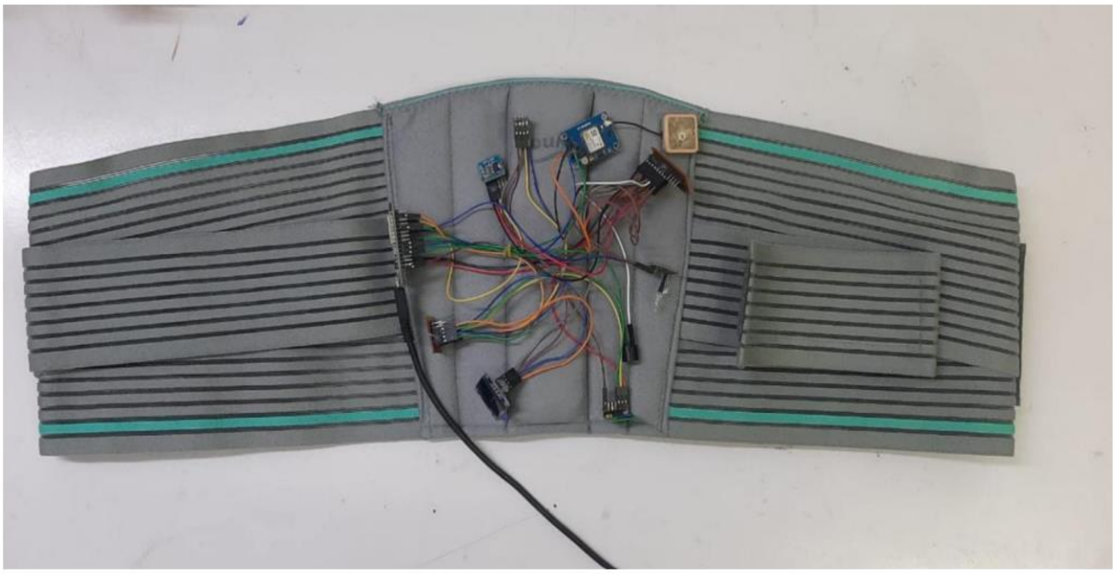

Figure 8.1 Parkinson Belt

Figure 8.2 Display from Firebase Cloud

Figure 8.3 Display from MIT Application Screen

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

print(classification_report(Ytest,predicted_values))
```

```
DecisionTrees's Accuracy is:  74.35897435897436
              precision    recall  f1-score   support

           0       0.44      0.88      0.58         8
           1       0.96      0.71      0.81        31

    accuracy                           0.74        39
   macro avg       0.70      0.79      0.70        39
weighted avg       0.85      0.74      0.77        39
```

Figure 8.4 Decision Tree Accuracy

XGBoost

```
import xgboost as xgb
XB = xgb.XGBClassifier()
XB.fit(Xtrain,Ytrain)

predicted_values = XB.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('XGBoost')
print("XGBoost's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
[15:17:17] WARNING: ../src/learner.cc:1115: Starting in XGBoost 1.3.0,
XGBoost's Accuracy is:  0.8205128205128205
              precision    recall  f1-score   support

           0       0.55      0.75      0.63         8
           1       0.93      0.84      0.88        31

    accuracy                           0.82        39
   macro avg       0.74      0.79      0.76        39
weighted avg       0.85      0.82      0.83        39
```

Figure 8.5 XG Boost Accuracy

Logistic Regression

```python
from sklearn.linear_model import LogisticRegression

LogReg = LogisticRegression(random_state=2)

LogReg.fit(Xtrain,Ytrain)

predicted_values = LogReg.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
Logistic Regression's Accuracy is:  0.8717948717948718
              precision    recall  f1-score   support

           0       0.80      0.50      0.62         8
           1       0.88      0.97      0.92        31

    accuracy                           0.87        39
   macro avg       0.84      0.73      0.77        39
weighted avg       0.87      0.87      0.86        39
```

Figure 8.6 Logistic Regression Accuracy

Support Vector Machine (SVM)

```python
from sklearn.svm import SVC
# data normalization with sklearn
from sklearn.preprocessing import MinMaxScaler
# fit scaler on training data
norm = MinMaxScaler().fit(Xtrain)
X_train_norm = norm.transform(Xtrain)
# transform testing dataabs
X_test_norm = norm.transform(Xtest)
SVM = SVC(kernel='poly', degree=3, C=1)
SVM.fit(X_train_norm,Ytrain)
predicted_values = SVM.predict(X_test_norm)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
SVM's Accuracy is:  0.9230769230769231
              precision    recall  f1-score   support

           0       1.00      0.62      0.77         8
           1       0.91      1.00      0.95        31

    accuracy                           0.92        39
   macro avg       0.96      0.81      0.86        39
weighted avg       0.93      0.92      0.92        39
```

Figure 8.7 SVM Accuracy

Guassian Naive Bayes

```
from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
Naive Bayes's Accuracy is:  0.5897435897435898
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         8
           1       1.00      0.48      0.65        31

    accuracy                           0.59        39
   macro avg       0.67      0.74      0.58        39
weighted avg       0.86      0.59      0.62        39
```

Figure 8.8 Guassian Naïve Bayes Accuracy

Random Forest

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
RF's Accuracy is:  0.8205128205128205
              precision    recall  f1-score   support

           0       0.55      0.75      0.63         8
           1       0.93      0.84      0.88        31

    accuracy                           0.82        39
   macro avg       0.74      0.79      0.76        39
weighted avg       0.85      0.82      0.83        39
```

Figure 8.9 Random Forest Accuracy

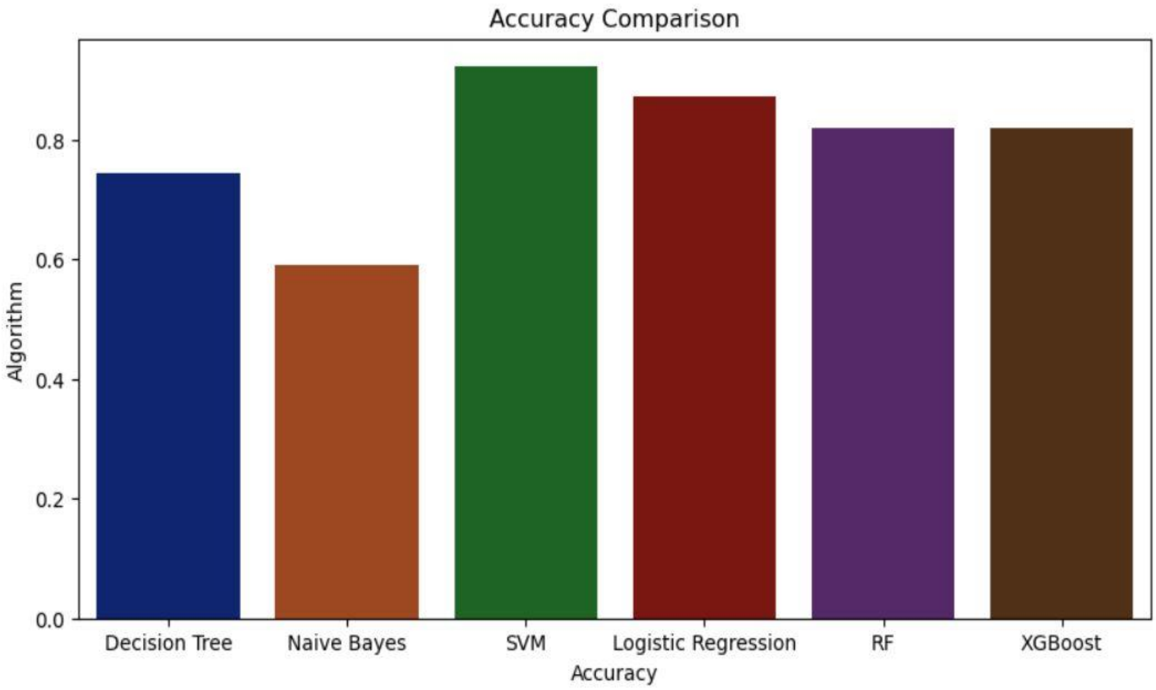

Figure 8.10 Accuracy Comparison of the Algorithms

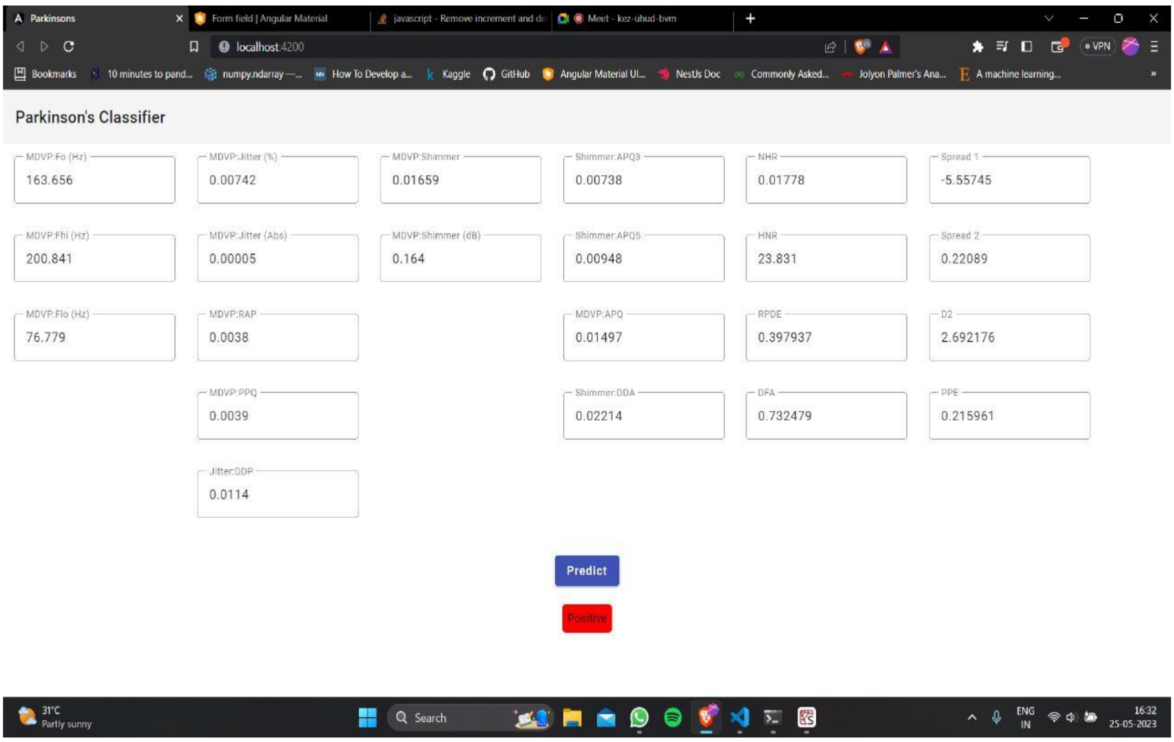

Figure 8.11 User Front End UI

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

The main aim of our project was to develop an intelligent device for detecting falls using the IoT pattern of Parkinson’s patients and we achieved that using this proposed system. The different directions of patients were detected using an accelerometer sensor (ADXL335) and ESP32 provides real-time monitoring of the Firebase cloud. The model was tested with sensor values from different scenarios and for different patients. A better accurate model is selected for the prediction of falls. Our system has around 82.5% precision and 95% of specification compared to other existing systems.

The data uploaded to the Firebase server can be used for medical research. The real-time location update of patients and the direction of fall is the most important facility of our system which helps caretakers and family to reach the patients immediately. We are planning to test our system with more patients to compare the values for better prediction.

The prototype of the proposed system was implemented using low-cost devices, but provide high quality and performance as the high-cost devices. The developers have used GY-85 and NodeMCUESP32 with the real-time monitoring in the Firebase cloud for this purpose. According to the results of testing, as shown in Table 1, the testing results indicate that the proposed system has an accuracy of around 82.50 percent and a specificity of up to 99.99 percent regarding the improvement in signal processing and accuracy in term of analyzing the real accidental fall events from activities of daily living to reduce false alarms.

WFDS can distinguish between falls and ADLs with high accuracy in real time and without any false alarm. The optimization of the accuracy measurements is based on the proposed accurate algorithm DFE. Consequently, the proposed fall detection algorithm achieved 100% accuracy, 100% sensitivity, and 100% specificity. The proposed WFDS outperformed methods proposed by other studies in terms of accuracy and specificity.