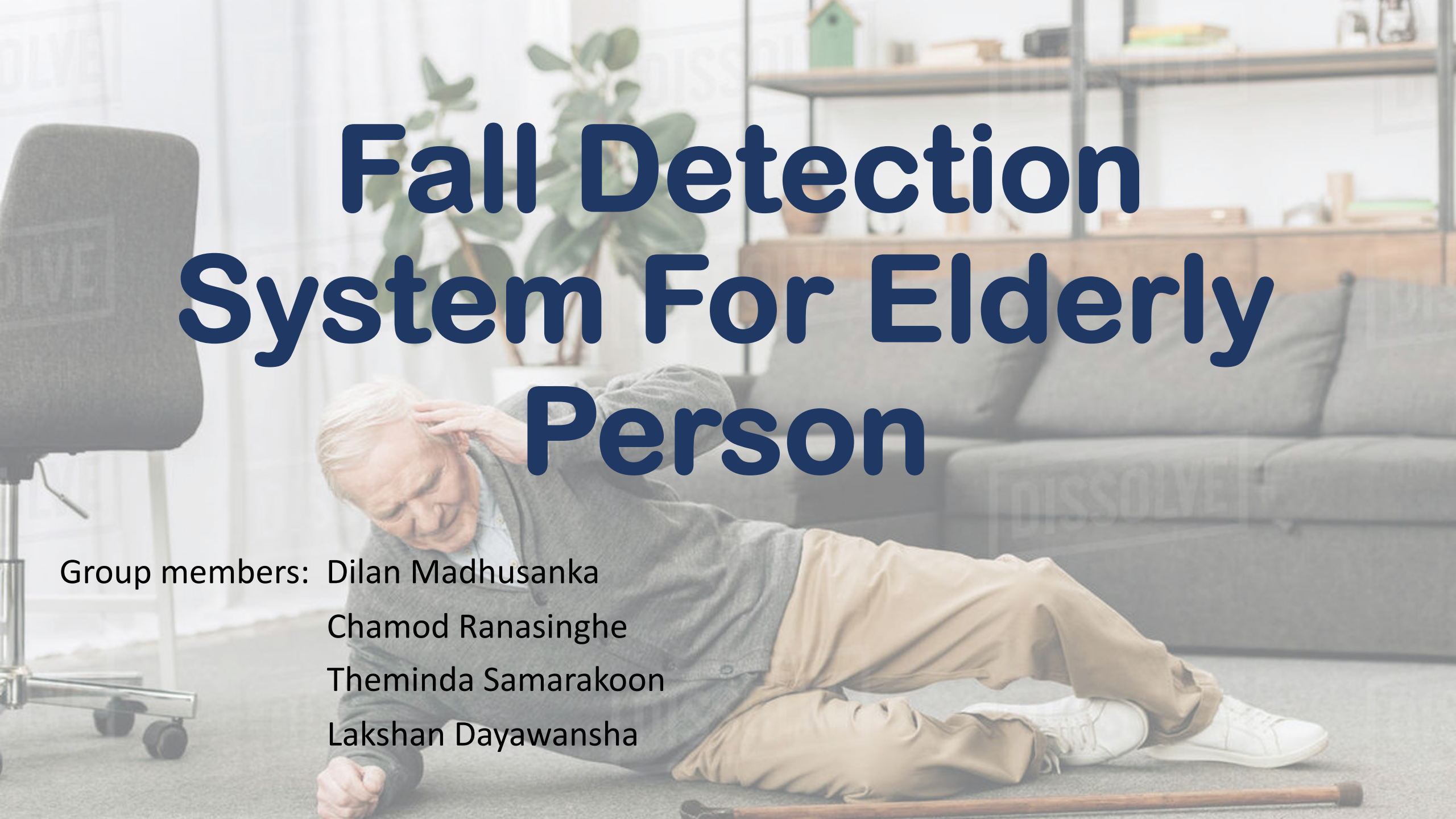


# Fall Detection System For Elderly Person

Group members: Dilan Madhusanka
Chamod Ranasinghe
Theminda Samarakoon
Lakshan Dayawansha

# Background

- Falls are a major cause of injury and death among elderly people
- Detecting falls in a timely manner is crucial for ensuring prompt medical attention and reducing the risk of serious injury or death.

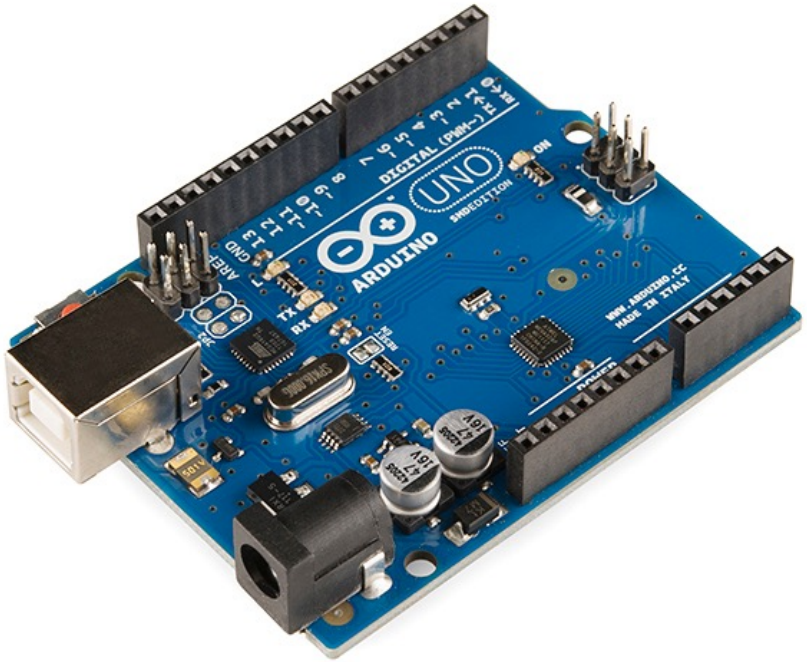

Fall Detection System

- Automated fall detection systems are needed to quickly and accurately identify fall events and alert caregivers or emergency services.
- Wearable sensors such as the MPU6050 have made it possible to detect falls using machine learning algorithms.
- The proposed project aims to develop a real-time fall detection system using Arduino and Python with KNN machine learning algorithm to monitor elderly people and alert caregivers or emergency services in case of a fall event.

# Scope of the project

- Detect accidental falls by physically weak or partially disabled elderly people.
- As the target demographic is not physically proactive, the scope can be narrowed down to distinguishing between falling and other slow movements like walking and sitting/standing up.

Data collection and processing

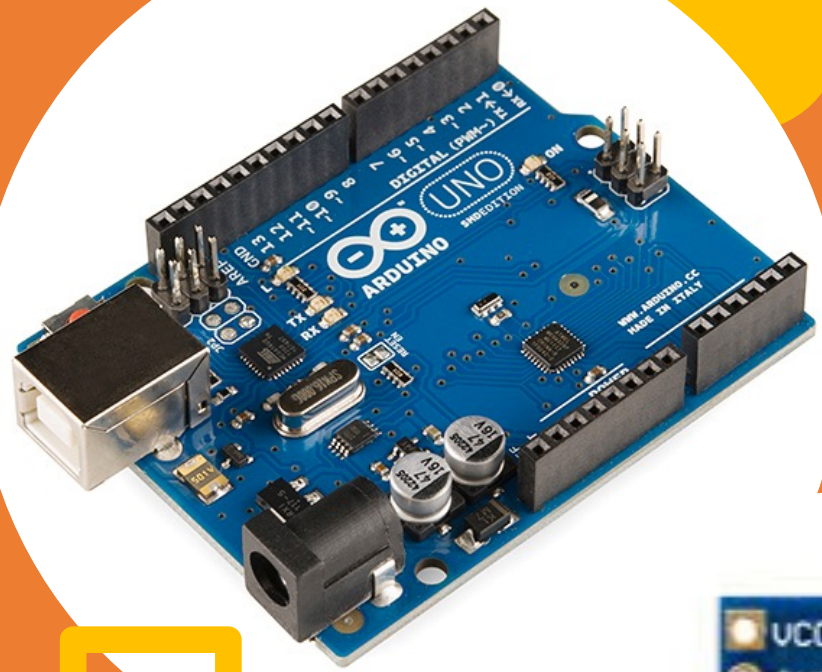



# Tools

Mpu 6050

Arduino uno

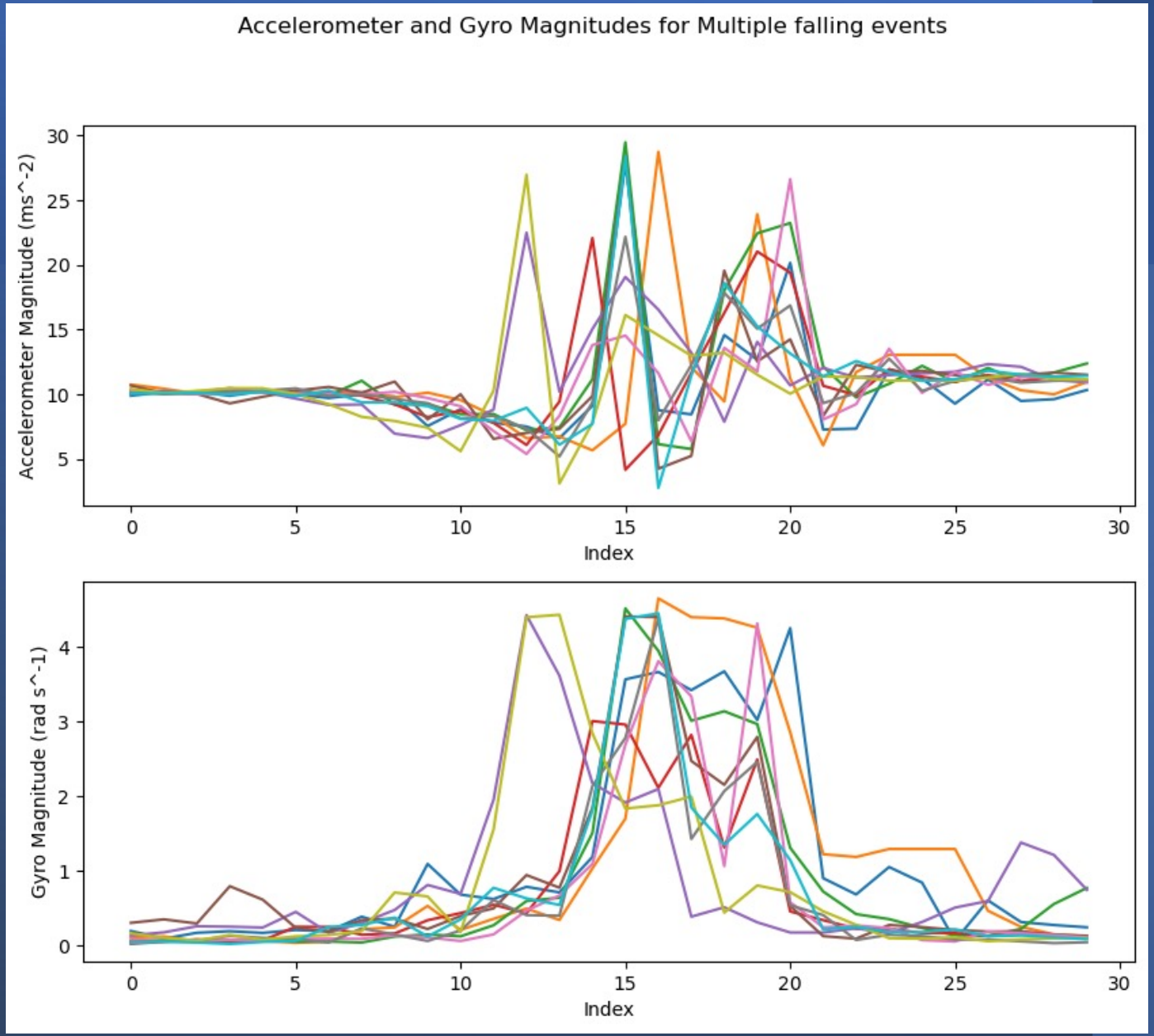
Python





# Collecting data for events

1. Falling
2. Walking
3. Sitting


Accelerometer and Gyro Magnitudes for Multiple falling events
Accelerometer Magnitude (ms^-2)
30
25
20
15
10
5
0
5
10
15
20
25
30
Index
Gyro Magnitude (rad s^-1)
4
3
2
1
0
0
5
10
15
20
25
30
Index

# Processing data for training the algorithm

- Average Magnitudes of accelometer data
- Weighted average magnitudes of gyroscope data
- Standard deviation

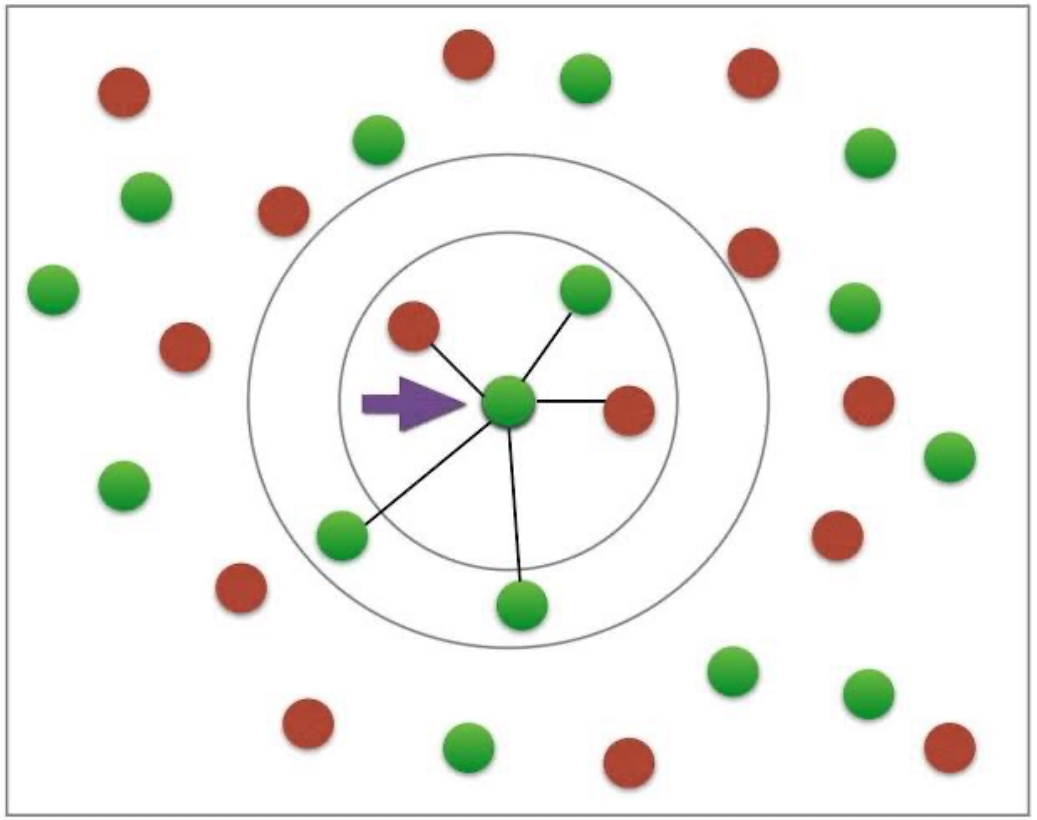

# Code Explanation And Real Time Implementation

# How KNN algorithm works

- KNN algorithm is a type of machine learning algorithm used for classification (and regression).
- It works by finding the k nearest neighbors of a given data point and assigns a label to it based on the most common class among its k nearest neighbors.
- The algorithm calculates the distance between the given data point and all other data points in the training set to find the nearest neighbors.
- The value of k is chosen by the user and is typically a small odd number.

# How KNN algorithm works

- The value of k is chosen by the user and is typically a small odd number.



# Why KNN?

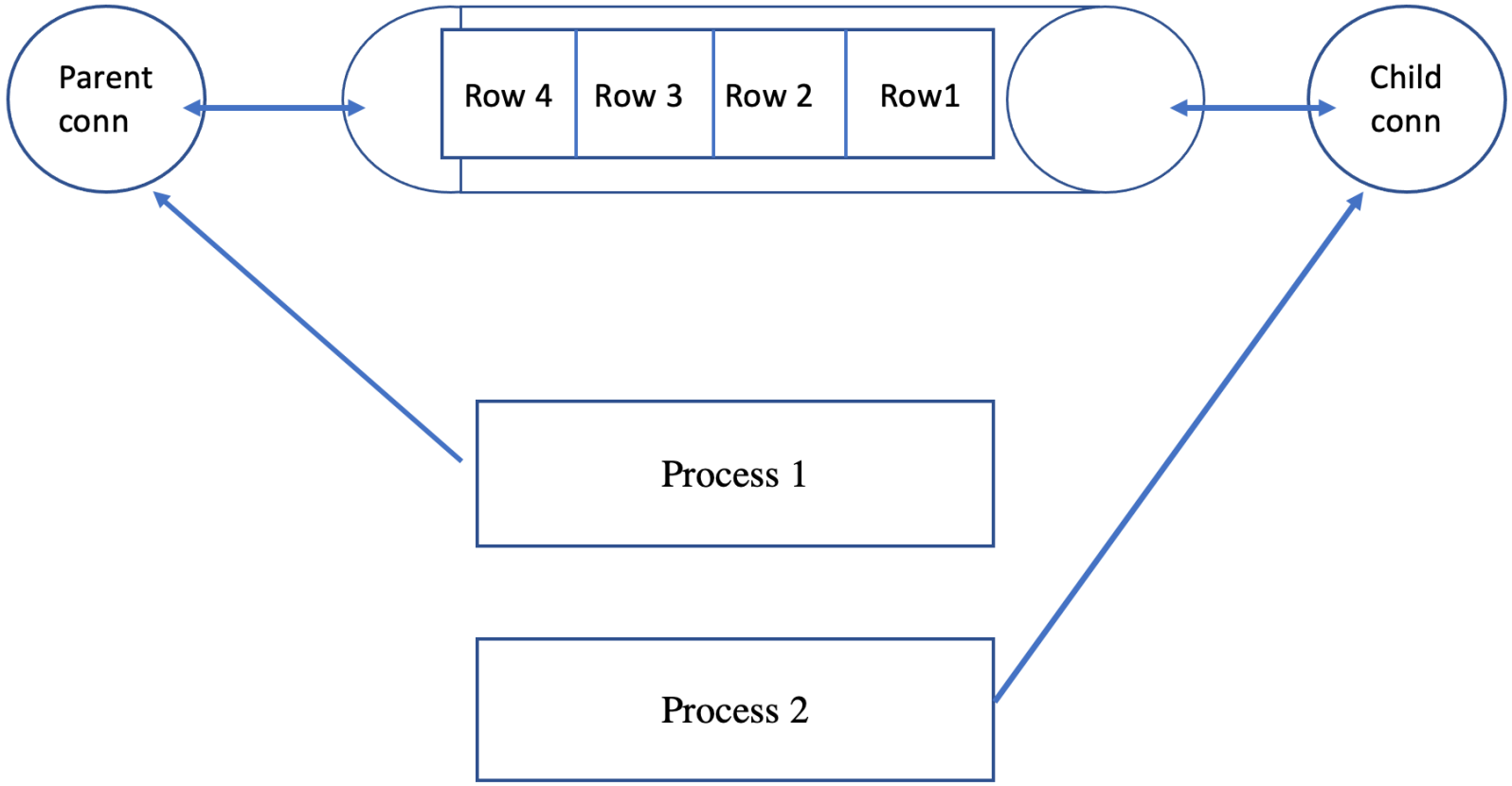
- Simplicity
- Easy to implement
- Fairly accurate for classification purposes

(A potentially better alternative:  Random Forest algorithm)

# Real-time data collection (sliding window technique)

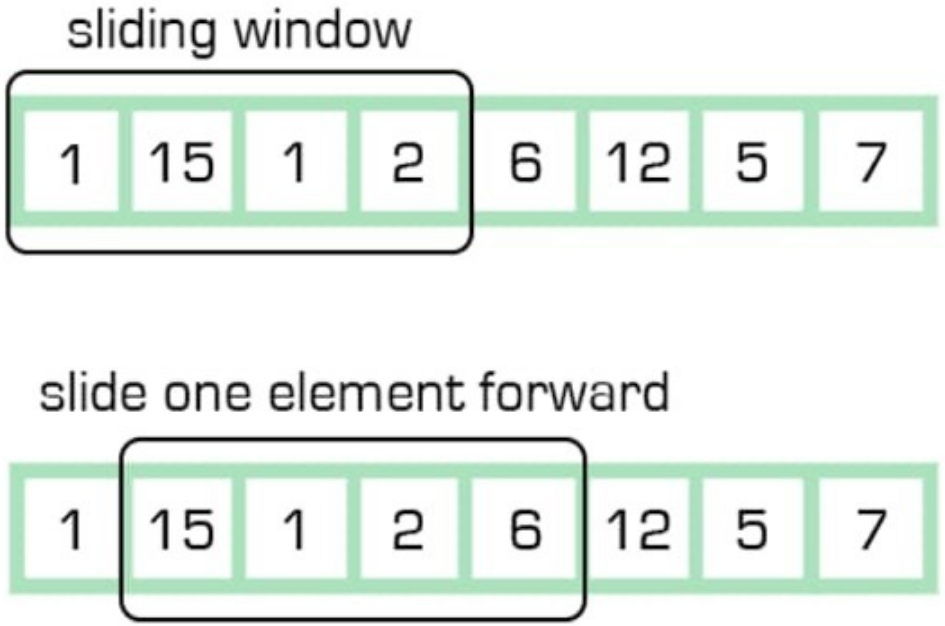
- Real-time data collection involves continuously collecting data as it is generated.
- The technique is useful for detecting falls quickly and providing timely alerts or notifications.
- Real-time implementation requires consideration of factors such as data sampling rates and processing speed.

# Python Multiprocessing.process() method and pipe()



# Sliding Window Technique

- Sliding window technique involves dividing the data into overlapping windows of a fixed length.



# Results

- The system can distinguish between falling, walking, and sitting.
- The system was tested using two individuals. During testing, accuracy was over 90%.

# Future work

- Integration with Smart Home Systems
- Wearable Fall Detection Devices
- Cloud-based Analytics
- Mobile Applications
- Integration with Emergency Services