**Bài Toán phát hiện té ngã**

1. **Phần Abstract**
2. **Bối cảnh nghiên cứu:**

Phát hiện té ngã dựa trên cảm biến nhúng là một lĩnh vực nghiên cứu đang được quan tâm do tính ứng dụng thực tế cao. Các phương pháp này thường sử dụng các cảm biến vật lý như con quay hồi chuyển và gia tốc kế để nhận diện các chuyển động bất thường có thể dẫn đến té ngã.

1. **Phương pháp truyền thống:**

Các phương pháp trước đây thường dựa vào đặc trưng thủ công (hand-crafted features), tức là các đặc điểm được xác định thủ công từ dữ liệu cảm biến, và sau đó sử dụng các mô hình học máy cơ bản như chuỗi Markov hoặc phương pháp phân loại dựa trên ngưỡng để phát hiện sự kiện té ngã.

1. **Giới thiệu về hệ thống TSFallDetect:**

TSFallDetect là một hệ thống mới được phát triển bao gồm:

Thiết bị nhận dữ liệu dựa trên cảm biến nhúng để thu thập dữ liệu từ người dùng.

Nền tảng triển khai mô hình học sâu di động, giúp triển khai các mô hình học sâu trực tiếp trên thiết bị di động.

Máy chủ đơn giản để thu thập và lưu trữ mô hình, dữ liệu, hỗ trợ việc mở rộng hệ thống trong tương lai.

1. **Phương pháp học sâu theo chuỗi:**

Hệ thống này sử dụng các phương pháp học sâu theo chuỗi (sequential deep-learning methods) để dự đoán chuyển động té ngã. Dữ liệu được thu thập từ các cảm biến quán tính (inertial sensors) và cảm biến áp suất phim (film pressure sensors).

1. **Nghiên cứu thực nghiệm:**

Nghiên cứu thực nghiệm được thực hiện trên cả các tập dữ liệu có sẵn và dữ liệu mới thu thập từ hệ thống TSFallDetect. Kết quả cho thấy mô hình học sâu có tiềm năng ưu thế hơn so với các phương pháp truyền thống.

1. **Phần Introduction**

**1. Giới thiệu**

Ở đây họ đã xây dựng một hệ thống phát hiện té ngã, bao gồm một thiết bị phần cứng cảm biến nhúng với chức năng thu thập và truyền dữ liệu, một ứng dụng di động với chức năng nhận dữ liệu, chạy mô hình phát hiện té ngã và lưu trữ dữ liệu, và một máy chủ thử nghiệm với các chức năng lưu trữ đám mây. Trên cơ sở này, chúng tôi đã thu thập dữ liệu té ngã (như gia tốc áp suất, tốc độ góc chân, v.v. khi té ngã hoặc đi bộ bình thường). Đồng thời, chúng tôi thiết kế một mô hình phát hiện té ngã mang tên FallSeqTCN.Sau đó triển khai nó trên ứng dụng di động.

**2. Các công trình liên quan**

**2.1 Hệ thống phát hiện liên quan hiện có**

1. **Giải pháp sử dụng cảm biến hình ảnh:**

**Cảm biến camera đo chiều sâu (Depth Camera Sensors):** Các cảm biến này, như Kinect RGB, cung cấp thông tin ba chiều về cảnh vật, bao gồm khoảng cách và độ sâu, giúp phân biệt rõ hơn giữa các hoạt động hàng ngày (Activities of Daily Living - ADL) và các sự kiện té ngã. Ví dụ, trong các nghiên cứu [13][14], dữ liệu hình ảnh từ camera được sử dụng để huấn luyện mạng CNN nhằm phát hiện sự biến dạng của hình dạng cơ thể con người và phân loại các hoạt động.

**Lý thuyết đi kèm:** Mạng Nơ-ron Tích chập (CNN) được sử dụng để xử lý và phân tích hình ảnh nhằm phân biệt giữa té ngã và các hoạt động thông thường. Một số mô hình như MyNet1D-D cũng đã được đề xuất để cải thiện hiệu suất trên các hệ thống nhúng.

**Cảm biến radar (Radar Sensors):** Được sử dụng để phát hiện chuyển động và xác định khoảng cách giữa các đối tượng dựa trên sóng radar, cho phép phát hiện té ngã ngay cả khi không có ánh sáng tốt.

**Giải pháp dựa trên tín hiệu âm thanh (Acoustic Signal-Based Solutions):** Sử dụng micrô để ghi lại các âm thanh trong môi trường và phân tích chúng để phát hiện sự kiện té ngã.

**Sàn thông minh với sợi nhạy áp (Smart Flooring Embedded with Pressure-Sensitive Fibers):** Sử dụng sợi nhạy cảm với áp suất tích hợp trong sàn để phát hiện lực tác động do té ngã.

1. **Giải pháp sử dụng cảm biến đeo (Wearable Sensors):**

**Cảm biến eo (Waist Sensor Units):** Thường mang lại độ chính xác cao nhất trong việc phát hiện té ngã do vị trí này ổn định và dễ dàng nhận diện các chuyển động cơ bản của cơ thể.

**Lý thuyết đi kèm:** Cảm biến gia tốc kế và con quay hồi chuyển tại eo giúp ghi lại gia tốc và thay đổi góc nghiêng, từ đó phát hiện các mô hình chuyển động bất thường.

**Cảm biến gắn ở chân (Foot Sensor):** Được sử dụng để theo dõi chuyển động chính xác hơn khi đi bộ, giúp người dùng thoải mái hơn mà không cần các thiết bị truyền phát tín hiệu bên ngoài.

**Lý thuyết đi kèm:** Dữ liệu thu được từ cảm biến gia tốc và áp suất ở chân có thể xác định động lực học của bước đi và phát hiện sự thay đổi đột ngột khi té ngã.

**Cảm biến trên điện thoại thông minh (Smartphone-based Accelerometer):** Sử dụng gia tốc kế trong điện thoại để phát hiện các chuyển động té ngã.

**Đơn vị đo lường quán tính (Inertial Measurement Units - IMU):** Kết hợp cảm biến gia tốc, con quay hồi chuyển để đo gia tốc tuyến tính và tốc độ góc, phục vụ cho việc phát hiện té ngã dựa trên dữ liệu tuần tự.

**Ký hiệu viết tắt và tên cảm biến:**

* **CNN (Convolutional Neural Network):** Mạng Nơ-ron Tích chập, được sử dụng rộng rãi trong xử lý ảnh và phát hiện mẫu.
* **LSTM (Long Short-Term Memory):** Một loại RNN dùng để xử lý dữ liệu chuỗi thời gian, hữu ích trong việc nhận diện các sự kiện té ngã dựa trên dữ liệu cảm biến.
* **RNN (Recurrent Neural Network):** Mạng nơ-ron hồi quy, phù hợp với dữ liệu tuần tự.
* **IMU (Inertial Measurement Units):** Đơn vị đo lường quán tính, gồm cảm biến gia tốc và con quay hồi chuyển.
* **MEMS (Microelectromechanical Systems):** Các hệ thống cơ điện tử vi mô, hỗ trợ phát triển cảm biến nhỏ gọn, hiệu quả cho các thiết bị đeo.

### ****3. Xây dựng hệ thống****

#### ****3.1 Tổng quan về hệ thống****

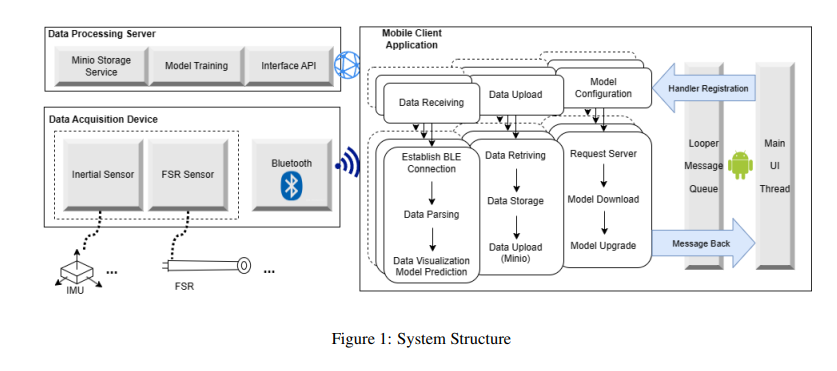
-bao gồm ba phần: thiết bị thu thập dữ liệu, ứng dụng di động và một máy chủ thử nghiệm.

+Thiết bị thu thập dữ liệu cảm biến chủ yếu khởi tạo cảm biến và truyền dữ liệu gốc theo thời gian thực.

+Ứng dụng di động thực hiện chức năng nhận dữ liệu qua Bluetooth Low Energy (BLE), cung cấp giao diện người dùng (UI) để tương tác và điều khiển các thiết bị kết nối.

+Máy chủ Minio được sử dụng để lưu trữ dữ liệu và cập nhật mô hình.

**3.2 Cấu trúc hệ thống bao gồm:**

1. **Thiết bị thu thập dữ liệu (Data Acquisition Device):**
   * Sử dụng các cảm biến vật lý bao gồm gia tốc kế, con quay hồi chuyển và cảm biến áp suất để đo áp lực, gia tốc và hướng chuyển động trong quá trình vận động. Thiết bị cảm biến dữ liệu sử dụng dữ liệu thu thập từ các cảm biến này để gửi đến ứng dụng di động thông qua Bluetooth để giải mã.
2. **Ứng dụng khách di động (Mobile Client Application):**
   * Ứng dụng di động dựa trên nền tảng hệ điều hành Android. Luồng chính chịu trách nhiệm đăng ký các luồng thực thi các chức năng cụ thể vào pool luồng tương ứng (hàng đợi), bao gồm các luồng con như tiếp nhận dữ liệu, trực quan hóa và dự đoán mô hình, tải dữ liệu lên và tải mô hình xuống.
3. **Máy chủ xử lý dữ liệu (Data Processing Server):**
   * Được xem như một nơi lưu trữ dữ liệu, chủ yếu dùng để lưu trữ dữ liệu chuyển động lịch sử do người dùng tạo ra theo thời gian. Máy chủ này giúp lưu trữ an toàn và quản lý các mô hình học sâu, cũng như dữ liệu người dùng để phân tích sau này.

**3.3 Các thiết bị được và hiệu quả đem lại của từng cái**

**Các thiết bị**

1. Data Acquisition Device (Thiết bị thu thập dữ liệu):

+Inertial Sensor (Cảm biến quán tính - IMU)

+FSR Sensor (Force Sensing Resistor Sensor - Cảm biến lực)

+Bluetooth Module

1. Mobile Client Application (Ứng dụng di động)

+Data Receiving (Nhận dữ liệu)

+Data Upload (Tải lên dữ liệu)

+Model Configuration (Cấu hình mô hình)

1. Data Processing Server (Máy chủ xử lý dữ liệu)

+ Minio Storage Service (Dịch vụ lưu trữ Minio)

+Model Training (Huấn luyện mô hình)

+Interface API (Giao diện API)

1. **Quy trình hoạt động của họ**

Thiết bị thu thập dữ liệu sử dụng cảm biến quán tính và cảm biến lực để ghi lại các thông số chuyển động của người dùng. Sau đó Ứng dụng di động nhận và xử lý dữ liệu, thực hiện các dự đoán bằng mô hình học sâu, và trực quan hóa kết quả cho người dùng và cuối cùng máy chủ xử lý dữ liệu lưu trữ dữ liệu lịch sử, huấn luyện các mô hình mới dựa trên dữ liệu thu thập được, và cung cấp API để ứng dụng di động tải xuống hoặc cập nhật mô hình mới.

1. **Chi tiết thiết bị**
2. Inertial Sensor (Cảm biến quán tính - IMU): được cấu tạo từ 2 thành phần

+**Accelerometer (Gia tốc kế):** Đo gia tốc tuyến tính theo các trục X, Y, Z.

+**Gyroscope (Con quay hồi chuyển):** Đo tốc độ quay (tốc độ góc) xung quanh các trục này.

Công dụng:

**+ Accelerometer:** Được sử dụng để phát hiện các chuyển động bất thường như cú ngã, khi gia tốc thay đổi đột ngột. Khi một người ngã, gia tốc của họ thay đổi đột ngột, và cảm biến này ghi lại sự thay đổi đó.

+ **Gyroscope:** Theo dõi hướng và tốc độ xoay của cơ thể. Khi té ngã, hướng của cơ thể và tốc độ quay cũng thay đổi đột ngột, và con quay hồi chuyển giúp nhận diện những thay đổi này.

Tần số thu thập dữ liệu của loại cảm biến này   
+Thường từ **100 Hz** đến **1000 Hz** (100 đến 1000 lần mỗi giây)

Hiệu quả:

Có thể cung cấp dữ liệu rất chi tiết về các chuyển động cơ thể, phù hợp cho các ứng dụng đòi hỏi độ chính xác cao như phát hiện té ngã.

Hạn chế:

+ Nhạy cảm với nhiễu (như rung động từ môi trường) và yêu cầu phải có thuật toán xử lý dữ liệu tốt để loại bỏ nhiễu và giảm thiểu sai lệch.

1. FSR Sensor (Force Sensing Resistor Sensor - Cảm biến lực):

+ Là cảm biến điện trở nhạy áp, thay đổi điện trở của nó khi có lực tác dụng lên bề mặt

+ Cấu trúc bên trong của 1 thiết bị FSR được chế tạo từ 2 lớp vật liệu dẫn điện:   
 - Lớp vật liệu dẫn điện( được phân cách vởi lớp dẫn điện rất nhạt cảm với biến đổi áp xuất).

Công dụng:

+ Đo áp suất hoặc lực tác dụng lên bề mặt nơi đặt cảm biến.

+ Có thể được gắn vào đế giày hoặc các vị trí thích hợp khác trên cơ thể để phát hiện khi lực tác động đột ngột thay đổi.(Áp dụng vào bài toán hiện tại là phát hiện té ngã )

Tần số thu thập dữ liệu:

+ Tần số thu thập của cảm biến FSR thường dao động từ **10 Hz** đến **200 Hz** (10 đến 200 lần mỗi giây)

Lưu ý : con số này có thể thay đổi tùy vào ứng dụng và khả năng xử lý dữ liệu của hệ thống.

Hiệu quả:

+ Có ưu điểm là nhẹ, giá thành rẻ, và dễ tích hợp vào các hệ thống đeo.

Hạn chế

+Độ chính xác của chúng có thể bị ảnh hưởng bởi nhiệt độ và sự mòn của cảm biến theo thời gian.

+ Không thể sử dụng đơn thiết bị mà phải sử dụng với các cảm biến khác để tăng độ chính xác.

1. Bluetooth Module (Mô-đun Bluetooth Low Energy - BLE):

+ Là một chuẩn giao tiếp không dây tiêu thụ năng lượng thấp, được thiết kế đặc biệt để truyền dữ liệu giữa các thiết bị trong khoảng cách ngắn (dưới 100 mét).

Công dụng:

+ Sử dụng để truyền tải dữ liệu thu thập từ các cảm biến (gia tốc kế, con quay hồi chuyển, FSR) đến ứng dụng di động.

+ Có khả năng tiêu thụ ít năng lượng, phù hợp cho các thiết bị đeo cần hoạt động trong thời gian dài mà không cần sạc lại.

Tần số thu thập dữ liệu:

+ Có khả năng truyền dữ liệu với tốc độ khoảng **1 Mbps** (megabit trên giây), và độ trễ truyền dữ liệu thường rất thấp, khoảng **3 ms** (mili giây).

Hiệu quả:

+ Cho phép truyền dữ liệu cảm biến gần như thời gian thực.

+ Rất hiệu quả trong việc duy trì kết nối ổn định và tiêu thụ năng lượng thấp.

Nhược điểm:

+ Có phạm vi giới hạn và có thể bị ảnh hưởng bởi các vật cản hoặc nhiễu từ các thiết bị không dây khác.

1. Minio Storage Server (Máy chủ lưu trữ Minio):

+ Là một nền tảng lưu trữ đối tượng nguồn mở, tương thích với giao thức S3 của Amazon, được thiết kế để lưu trữ dữ liệu quy mô lớn.

* Amazon S3 là gì?  
  +  là dịch vụ lưu trữ đối tượng được xây dựng để lưu trữ và truy xuất bất kỳ lượng dữ liệu nào từ bất cứ nơi nào. Đây là dịch vụ lưu trữ đơn giản có độ bền, độ sẵn có, hiệu suất, tính bảo mật dẫn đầu ngành và khả năng thay đổi quy mô gần như không giới hạn với chi phí cực kỳ thấp.

Công dụng:

+ Dùng để lưu trữ dữ liệu lịch sử được thu thập từ các thiết bị đeo và mô hình học sâu mới được huấn luyện.

+ Cung cấp một nơi an toàn và hiệu quả để lưu trữ dữ liệu và đảm bảo tính sẵn có của dữ liệu cho việc phân tích sau này.

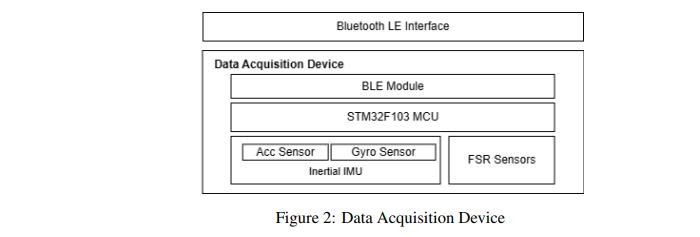
Hiệu quả:

+ Tối ưu hóa cho việc lưu trữ dữ liệu quy mô lớn.

+ Có khả năng bảo mật cao.

+ Hỗ trợ tích hợp dễ dàng với các ứng dụng khác thông qua API, giúp đơn giản hóa quá trình quản lý dữ liệu.

**3.4 Data Acquisition Device**

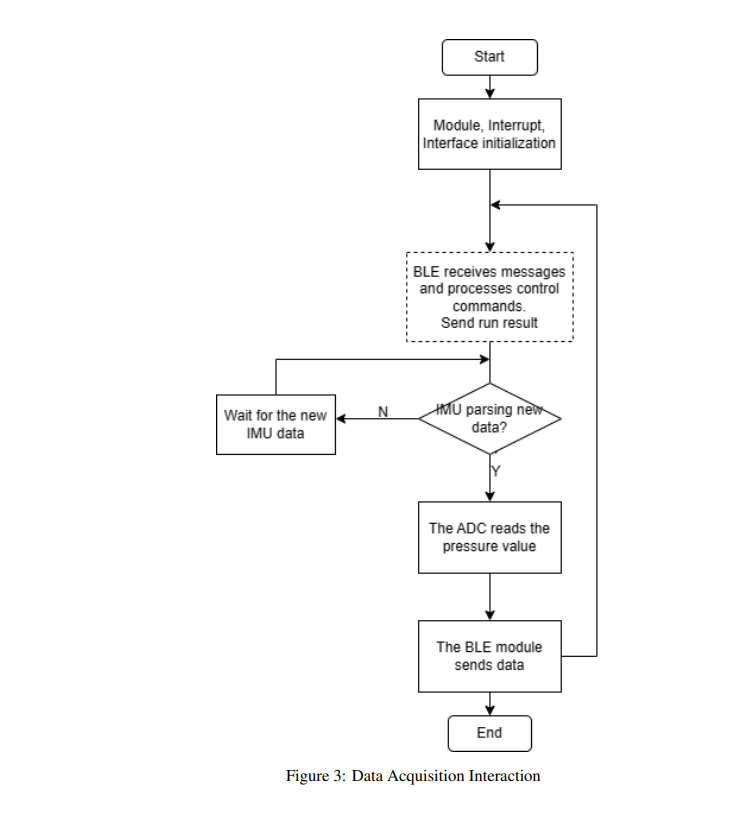
****

**Thông tin chi tiết về các thành phần của hệ thống:**

1. **MCU STM32F103:**
   * **Kiến trúc ARM:** Đây là bộ vi điều khiển sử dụng lõi ARM Cortex-M3, có khả năng xử lý nhanh và hỗ trợ nhiều giao thức truyền thông như UART, I2C, và SPI.
   * **Chức năng:** Chịu trách nhiệm thu thập dữ liệu từ các cảm biến (FSR, IMU901), xử lý sơ bộ dữ liệu, và giao tiếp với mô-đun Bluetooth để truyền dữ liệu đến thiết bị di động.
2. **Cảm biến áp suất (FSR - Force Sensing Resistor):**
   * **Công dụng:** Đo lực tác dụng lên bề mặt để phát hiện các sự kiện như té ngã. Cảm biến FSR được triển khai trên cả hai chân để cải thiện độ chính xác và đồng nhất dữ liệu.
3. **Cảm biến IMU901 (Inertial Measurement Unit):**
   * **Thành phần:**
     + **Accelerometer (Gia tốc kế):** Đo gia tốc tuyến tính trên các trục X, Y, Z.
     + **Gyro Sensor (Con quay hồi chuyển):** Đo tốc độ góc để xác định hướng và vận tốc quay của cơ thể.
   * **Chức năng:** Cung cấp dữ liệu về chuyển động và tư thế của cơ thể người dùng.
4. **Mô-đun Bluetooth (ATK-BLE):**
   * **Công dụng:** Giao tiếp dữ liệu không dây giữa nền tảng cảm biến nhúng và thiết bị di động Android. BLE có ưu điểm tiêu thụ ít năng lượng, phù hợp cho các ứng dụng di động cần kéo dài thời gian sử dụng pin.
5. **Mô-đun chuyển đổi điện áp:**
   * **Công dụng:** Chuyển đổi điện áp để phù hợp với các thành phần khác nhau trên nền tảng, đảm bảo rằng các cảm biến và mô-đun hoạt động ở các mức điện áp thích hợp.

Thiết mới được kể đến   
**STM32F103 MCU (Microcontroller Unit):**

* **Bộ vi điều khiển STM32F103:** Đây là bộ vi điều khiển dựa trên kiến trúc ARM Cortex-M3, được sử dụng để xử lý dữ liệu thu thập từ các cảm biến. STM32F103 hỗ trợ nhiều giao tiếp như I2C, SPI, UART, cho phép kết nối linh hoạt với nhiều loại cảm biến.
* **Đặc tính nổi bật:**
  + **Tốc độ xung nhịp:** Lên đến 72 MHz.
  + **Bộ nhớ Flash:** Tối đa 128 KB, cho phép lưu trữ các chương trình và cấu hình cảm biến.
  + **Bộ nhớ RAM:** 20 KB, hỗ trợ các tính toán cần thiết trong thời gian thực.
  + **Chức năng:** Xử lý dữ liệu thu thập từ các cảm biến, quản lý việc giao tiếp với mô-đun BLE để truyền dữ liệu đến ứng dụng di động.



Quy trình chạy của sơ đồ tư duy

**Quy trình Tương tác Thu thập Dữ liệu:**

1. **Khởi động (Start):**
   * Hệ thống bắt đầu với việc khởi tạo các thành phần cần thiết, bao gồm **Module**, **Ngắt (Interrupt)** và **Khởi tạo Giao diện (Interface Initialization)**. Đây là bước khởi tạo ban đầu để đảm bảo tất cả các mô-đun cảm biến và giao tiếp sẵn sàng hoạt động.
2. **Nhận và xử lý lệnh từ mô-đun Bluetooth (BLE receives messages and processes control commands):**
   * **Mô-đun BLE** nhận các thông điệp và lệnh điều khiển từ ứng dụng di động hoặc PC. Mô-đun này xử lý các lệnh điều khiển và gửi kết quả hoạt động trở lại để xác nhận rằng thiết bị đã sẵn sàng hoặc đang hoạt động đúng chức năng.
3. **Phân tích dữ liệu từ IMU (IMU parsing new data?):**
   * Hệ thống kiểm tra xem liệu dữ liệu mới từ **IMU (Inertial Measurement Unit)** đã được thu thập hay chưa.
   * Nếu chưa có dữ liệu mới ("No" hoặc "N"), hệ thống tiếp tục chờ (Wait for the new IMU data).
   * Nếu có dữ liệu mới ("Yes" hoặc "Y"), tiến trình sẽ tiếp tục với bước tiếp theo.
4. **Đọc giá trị áp suất từ ADC (The ADC reads the pressure value):**
   * **ADC (Analog-to-Digital Converter - Bộ chuyển đổi Tín hiệu Tương tự sang Số)** đọc giá trị áp suất từ **cảm biến FSR (Force Sensing Resistor)**. Đây là bước chuyển đổi tín hiệu từ cảm biến sang dữ liệu số để có thể xử lý tiếp theo.
5. **Gửi dữ liệu qua mô-đun BLE (The BLE module sends data):**
   * Sau khi thu thập và xử lý dữ liệu từ cảm biến IMU và cảm biến áp suất, hệ thống sẽ sử dụng mô-đun **Bluetooth (BLE)** để gửi dữ liệu đã thu thập đến thiết bị di động hoặc máy chủ để lưu trữ và phân tích thêm.
6. **Kết thúc (End):**
   * Chu kỳ thu thập dữ liệu kết thúc và hệ thống quay lại trạng thái chờ cho chu kỳ tiếp theo hoặc sẵn sàng nhận lệnh mới.

**Chi tiết về việc thu thập dữ liệu và các bộ dữ liệu tương tự:**

Các tín hiệu thu được bao gồm giá trị áp suất dưới lòng bàn chân, giá trị gia tốc, giá trị vận tốc góc, và giá trị hướng phương vị  
+ Thiết bị nhúng được đeo ở cả hai chân (trái và phải) của đối tượng và các tín hiệu cảm biến của đối tượng trong trạng thái đi bộ bình thường và trạng thái té ngã, bao gồm té ngã về phía trước và té ngã sang trái, được thu thập.

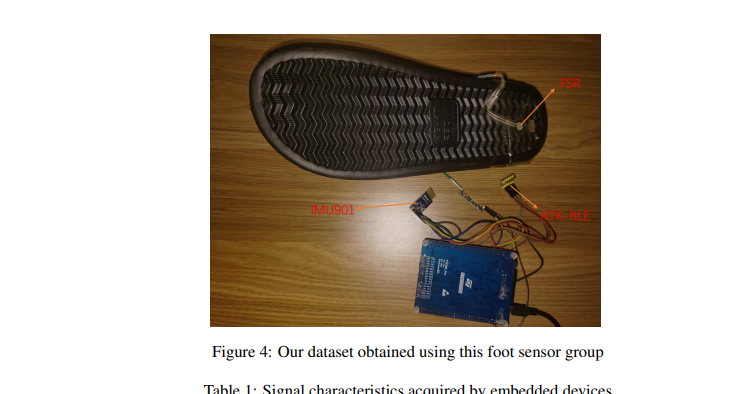
+ Tần số lấy mẫu là **18 Hz(18 lần mỗi giây)** , và tín hiệu thu được mỗi lần bao gồm 20 đặc trưng của giá trị áp suất dưới lòng bàn chân, giá trị gia tốc, giá trị vận tốc góc và giá trị hướng phương vị ở cả chân trái và chân phải, tạo thành bộ dữ liệu chuỗi thời gian ban đầu.

+ Trong cùng môi trường thử nghiệm, các đối tượng thực hiện 11 lần đi bộ bình thường, 10 lần té ngã về phía trước, và 5 lần té ngã sang trái, tổng cộng 26 lần thử nghiệm

+ Tất cả được sắp xếp thành 26 tệp TXT. 20 giá trị đặc trưng thu thập được trong mỗi thử nghiệm được hiển thị trong Bảng 1.

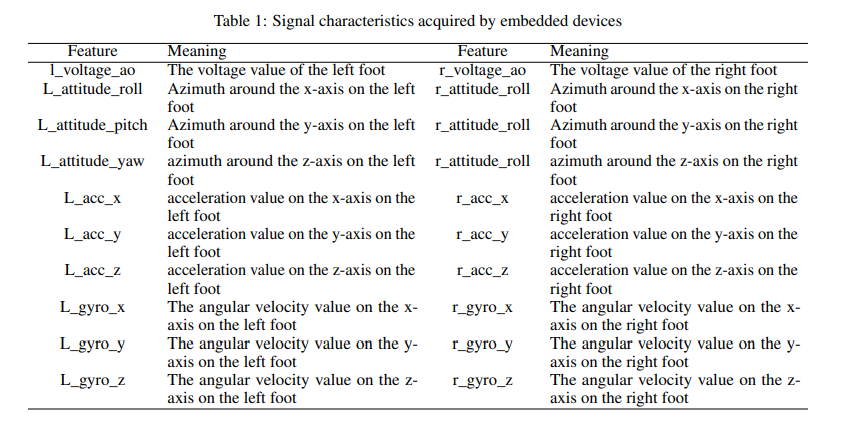
**Bộ dữ liệu công khai UMAFall:**

* **Bộ dữ liệu UMAFall:** Đây là bộ dữ liệu chứa dấu vết chuyển động được thu thập từ 5 điểm cảm biến đeo trên cơ thể, giúp cung cấp dữ liệu chi tiết hơn so với các bộ dữ liệu chỉ sử dụng 1 hoặc 2 điểm cảm biến.
* **Ứng dụng trong huấn luyện và kiểm tra:** UMAFall được sử dụng để huấn luyện và kiểm tra mô hình phát hiện té ngã, cung cấp một cơ sở dữ liệu đa dạng và phong phú cho các thuật toán học sâu.



**Mô tả chi tiết về nhóm cảm biến bàn chân:**

1. **FSR (Force Sensing Resistor Sensor - Cảm biến lực):**
   * **Vị trí:** Được gắn vào đế giày để đo lực tác động từ lòng bàn chân.
   * **Công dụng:**
     + FSR đo áp lực khi chân chạm đất trong các hoạt động đi bộ hoặc khi té ngã. Thông tin này hữu ích trong việc xác định mức độ tiếp xúc và lực tác động tại các điểm khác nhau trên lòng bàn chân, giúp phân biệt giữa các loại hoạt động như đi bộ bình thường và té ngã.
   * **Ưu điểm:** Nhẹ, giá thành rẻ, dễ tích hợp vào các thiết kế giày hoặc đế lót.
2. **IMU901 (Inertial Measurement Unit - Đơn vị đo lường quán tính):**
   * **Vị trí:** Được gắn trên đế giày gần cảm biến FSR.
   * **Thành phần:** Tích hợp gia tốc kế (Accelerometer) và con quay hồi chuyển (Gyroscope) để đo gia tốc và vận tốc góc theo các trục X, Y, Z.
   * **Công dụng:**
     + Thu thập dữ liệu về chuyển động của chân trong thời gian thực, bao gồm gia tốc tuyến tính và tốc độ quay. Dữ liệu này giúp phát hiện các chuyển động bất thường và phân tích các sự kiện té ngã.
3. **ATK-BLE (Bluetooth Low Energy Module):**
   * **Vị trí:** Gắn gần IMU901 và kết nối với vi điều khiển.
   * **Công dụng:**
     + Mô-đun BLE được sử dụng để truyền dữ liệu cảm biến từ thiết bị đeo (trên chân) tới thiết bị di động hoặc máy chủ qua giao thức Bluetooth Low Energy. BLE giúp tiết kiệm năng lượng, đảm bảo thiết bị đeo có thể hoạt động liên tục mà không cần sạc lại thường xuyên.
4. **STM32 Microcontroller (Vi điều khiển STM32):**
   * **Vị trí:** Kết nối với các cảm biến và mô-đun BLE để xử lý và truyền dữ liệu.
   * **Công dụng:**
     + Vi điều khiển này xử lý dữ liệu từ cảm biến FSR và IMU901, sau đó truyền dữ liệu qua mô-đun BLE tới thiết bị di động để phân tích thêm.



Các đặc trưng được nêu trong table 1:

 **Voltage (l\_voltage\_ao, r\_voltage\_ao):** Giá trị điện áp được đo ở chân trái và chân phải. Điện áp này có thể liên quan đến cảm biến áp suất hoặc một số thành phần điện tử khác được sử dụng trong thiết bị.

 **Attitude (L\_attitude\_roll, L\_attitude\_pitch, L\_attitude\_yaw):** Các giá trị này đại diện cho các góc định hướng (roll, pitch, yaw) quanh các trục x, y, z trên chân trái và phải. Đây là các giá trị phương vị (azimuth) giúp xác định hướng di chuyển của bàn chân trong không gian ba chiều.

 **Acceleration (L\_acc\_x, L\_acc\_y, L\_acc\_z):** Giá trị gia tốc trên các trục x, y, z của chân trái và chân phải. Các giá trị này cho biết mức độ thay đổi tốc độ của bàn chân trong các hướng khác nhau, giúp phát hiện các chuyển động bất thường hoặc té ngã.

 **Angular Velocity (L\_gyro\_x, L\_gyro\_y, L\_gyro\_z):** Giá trị vận tốc góc trên các trục x, y, z của chân trái và chân phải. Các giá trị này cho biết tốc độ quay của bàn chân, hỗ trợ trong việc phân tích chuyển động phức tạp như xoay hoặc ngã.

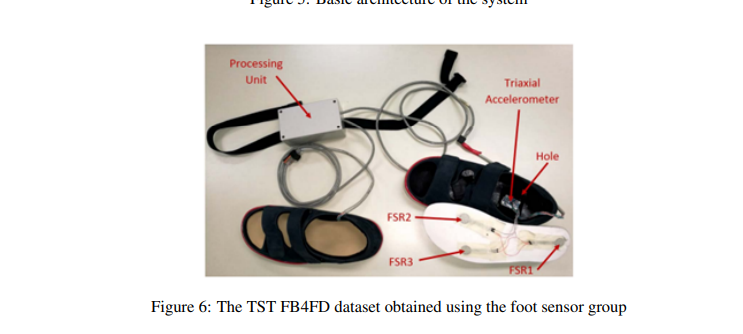
### ****Thông tin chi tiết về bộ dữ liệu TST FB4FD:****

Bộ dữ liệu **TST FB4FD** được đo đạc bằng một chiếc giày thông minh (smart shoe), như minh họa trong Hình 6. Giày thông minh này được trang bị ba cảm biến lực (**Force Sensing Resistors - FSR**) và một cảm biến gia tốc ba trục (**three-axis accelerometer**), cùng với một bo mạch xử lý.

* **Cảm biến lực FSR:** Có ba cảm biến FSR được sử dụng để đo áp lực tác động lên các điểm cụ thể trên lòng bàn chân. Thông tin này giúp nhận diện các thay đổi trong chuyển động và áp lực khi chân tiếp đất, cũng như các hoạt động như đi bộ hoặc đứng yên.
* **Gia tốc kế ba trục:** Đo lường gia tốc theo ba trục (X, Y, Z) để ghi nhận các chuyển động tuyến tính của người dùng, đặc biệt là các thay đổi đột ngột trong vận tốc, như khi người dùng bị ngã.
* **Bo mạch xử lý:** Đơn vị xử lý này phân tích các pha của chu kỳ bước đi (**gait cycle phase**), phân biệt giữa các trường hợp té ngã và không té ngã, và truyền dữ liệu từ xa. Điều này cho phép theo dõi chuyển động trong thời gian thực và gửi thông tin đến một thiết bị khác hoặc một máy chủ để phân tích và phản hồi nhanh chóng.

### ****Ý nghĩa và lợi ích:****

* **Phân tích chu kỳ bước đi (Gait Cycle Phase):** Việc phân tích chu kỳ bước đi giúp xác định các mẫu chuyển động cụ thể, như bước chân bình thường, bước dài hoặc bước ngắn. Điều này rất quan trọng để phân biệt các hoạt động hàng ngày với các sự kiện bất thường như té ngã.
* **Phân biệt giữa té ngã và không té ngã:** Dữ liệu từ FSR và gia tốc kế được xử lý để xác định khi nào một sự kiện té ngã xảy ra, dựa trên các thay đổi đột ngột về áp lực hoặc gia tốc. Điều này giúp giảm thiểu cảnh báo sai và tăng độ chính xác trong việc phát hiện té ngã.
* **Truyền dữ liệu từ xa:** Bo mạch xử lý có khả năng truyền dữ liệu từ xa, cho phép giám sát và phản hồi tức thời khi có sự cố. Điều này rất hữu ích trong các ứng dụng giám sát sức khỏe từ xa, đặc biệt là đối với người cao tuổi hoặc những người có nguy cơ té ngã cao



**3.5 Mobile Client Application**

Trong quá trình thiết kế phần mềm cho nền tảng điện thoại thì nó lại có vấn đề lớn đó là việc xử lý đa thiết bị đồng thời dẫn đến dữ liệu có thể bị sai sót. Từ đó họ đã đề xuất với một cơ chế gọi là **Message-Handler**.

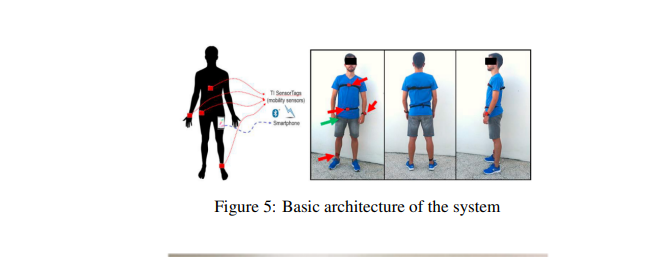
Việc sử dụng cơ chế **Message-Handler** giúp ứng dụng xử lý nhiều tác vụ đồng thời một cách mượt mà và hiệu quả. Khi ứng dụng cần kết nối với nhiều thiết bị cùng lúc, các thông điệp được tạo ra để quản lý dữ liệu từ các thiết bị này có thể được xử lý theo đúng thứ tự và thời gian thông qua **Handler** và **Looper**. Điều này giúp tránh xung đột dữ liệu, ngăn ngừa các tình huống deadlock (tắc nghẽn), và giảm thiểu rủi ro về lỗi phần mềm.

Ngoài ra, bằng cách sử dụng **Handler**, chúng ta có thể tối ưu hóa việc sử dụng tài nguyên và đảm bảo rằng ứng dụng luôn phản hồi nhanh chóng mà không làm giảm hiệu suất khi xử lý các tác vụ nền. Cơ chế này rất phù hợp cho các ứng dụng đòi hỏi phải xử lý lượng lớn dữ liệu thời gian thực hoặc tương tác với nhiều thiết bị ngoại vi.

**Kiến trúc cơ bản của hệ thống:**

Hình ảnh minh họa cho thấy hệ thống phát hiện té ngã sử dụng một cấu trúc bao gồm nhiều cảm biến gắn trên các bộ phận khác nhau của cơ thể, cụ thể như sau:

1. **Cảm biến Từ xa (Sensor/Tag):**
   * Các cảm biến được gắn trên những vị trí chiến lược của cơ thể bao gồm cổ tay, cánh tay, thắt lưng, và mắt cá chân.
   * Mỗi cảm biến này có thể đo được chuyển động và vị trí của từng phần cơ thể thông qua các dữ liệu gia tốc kế (accelerometer) và con quay hồi chuyển (gyroscope).
2. **Smartphone (Điện thoại thông minh):**
   * Điện thoại thông minh đóng vai trò như một trung tâm thu thập dữ liệu từ các cảm biến gắn trên cơ thể người dùng.
   * Dữ liệu từ các cảm biến được truyền qua Bluetooth đến điện thoại để xử lý và phân tích trong thời gian thực, giúp phát hiện các sự kiện té ngã hoặc các chuyển động bất thường khác.
3. **Cấu trúc hệ thống:**
   * Trong hình minh họa, có ba hình ảnh của một người tham gia thử nghiệm:
     + **Hình bên trái:** Người đeo các cảm biến trên cổ tay, cánh tay, lưng, và chân. Các mũi tên màu đỏ chỉ ra vị trí gắn các cảm biến này.
     + **Hình giữa và phải:** Hiển thị người tham gia ở các góc độ khác nhau, cho thấy rõ vị trí gắn cảm biến và hướng di chuyển.



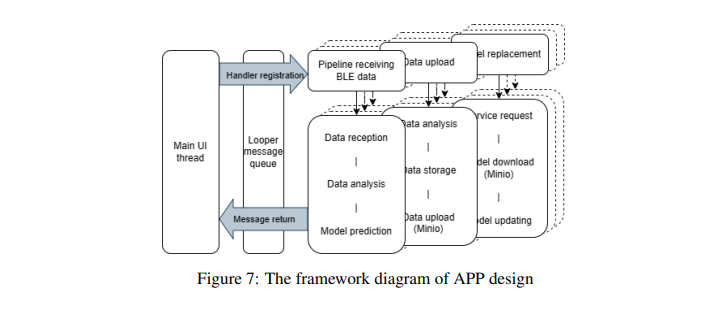
**Mô tả chi tiết về kiến trúc ứng dụng di động và quá trình kết nối thiết bị:**

**Hình 7: Kiến trúc khung của thiết kế ứng dụng (The framework diagram of APP design)**

Hình 7 cho thấy kiến trúc quy trình của thiết kế này. Ứng dụng di động được thiết kế dựa trên nền tảng hệ thống Android, với luồng chính (main thread) chịu trách nhiệm đăng ký các luồng thực thi các chức năng cụ thể vào hàng đợi luồng tương ứng. Các thành phần chính bao gồm:

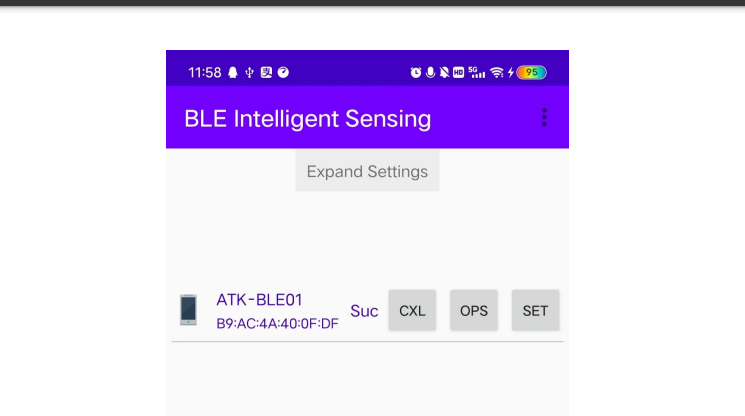
* **Pipeline nhận dữ liệu BLE:** Nhận dữ liệu từ thiết bị qua Bluetooth Low Energy (BLE).
* **Phân tích dữ liệu, dự đoán mô hình và hiển thị:** Sau khi nhận dữ liệu, các bước tiếp theo bao gồm phân tích dữ liệu, dự đoán mô hình (sử dụng thư viện TensorFlow Lite), và hiển thị kết quả.
* **Tải lên dữ liệu và thay thế mô hình:** Các dữ liệu đã phân tích có thể được tải lên một máy chủ (Minio) và mô hình phát hiện té ngã có thể được tải về hoặc cập nhật từ xa khi cần thiết.

Khi các tác vụ tương ứng được hoàn thành, luồng sẽ đóng gói kết quả vào một **Message** và trả về luồng chính thông qua **Handler**.



**Hình 8: Kết nối thiết bị (Device connection)**

Hình 8 minh họa giao diện ứng dụng khi kết nối với một thiết bị BLE (như ATK-BLE01). Người dùng chỉ cần nhấn nút "Connect" để bắt đầu quá trình kết nối với thiết bị. Sau khi thư viện kết nối BLE được triển khai, hệ thống thực hiện một cuộc gọi lại (callback) để quét và đăng ký kết nối, và các thiết bị quét được sẽ được đưa vào danh sách để người dùng lựa chọn.



**Chi tiết kỹ thuật và cách thức hoạt động:**

1. **Đơn giản hóa kết nối thiết bị:**
   * Để tạo điều kiện thuận lợi cho người dùng thao tác, phần mềm đảm bảo đơn giản hóa tối đa quy trình kết nối, như đã mô tả trong Hình 8. Người dùng chỉ cần nhấp vào nút kết nối và ứng dụng sẽ tự động quét và liệt kê các thiết bị BLE có sẵn.
2. **Xử lý và phân tích dữ liệu:**
   * **Dạng dữ liệu:** Dữ liệu được thu thập từ các cảm biến cần phải thuận tiện cho việc huấn luyện mô hình sau này và đồng thời mô tả rõ trạng thái chuyển động hiện tại của người dùng.
   * Mặc dù việc nhận dữ liệu thời gian thực được thực hiện qua thư viện Bluetooth, dữ liệu thường bị phân đoạn thành các mảnh byte nhỏ do giới hạn của BLE. Do đó, phân tích và tuần tự hóa dữ liệu trở thành ưu tiên hàng đầu.
   * **Lớp xử lý dữ liệu (Data Sink Class):** Lớp này duy trì một hàng đợi chuỗi nội bộ an toàn cho việc xử lý đồng thời, gọi là lines\_queue, để lưu trữ các dòng dữ liệu đã được phân tích. Phương thức receiveData sẽ nối các mảnh byte nhận được vào bộ đệm chuỗi và phân tách khi gặp ký tự xuống dòng.
3. **Quy trình xử lý không đồng bộ:**
   * Phương thức parseData sử dụng một lược đồ gọi lại không đồng bộ. Khi lines\_queue không trống, dòng dữ liệu đầu tiên sẽ được lấy ra, phân tích và đánh dấu là đã phân tích (isParsed), sau đó chuyển tiếp đến một hàm gọi lại được khởi tạo trong ứng dụng thực tế để tiếp tục các thao tác tiếp theo.
4. **Lưu trữ và quản lý dữ liệu:**
   * Sau khi dữ liệu được phân tích xong, lớp xử lý dữ liệu cung cấp một hàm gọi lại không đồng bộ để tiếp tục xử lý. Phần trình phân tích dữ liệu được khởi tạo trong hàm onCreate của giao diện ứng dụng và các cuộc gọi receiveData và parseData được đưa vào hàng đợi luồng chính sau khi thiết lập thông báo thành công để nhận các byte dữ liệu thô mới. Cuối cùng, ứng dụng sẽ thiết lập đường dẫn lưu trữ tệp và lưu dữ liệu vào tệp.
5. **Tần số thu thập dữ liệu:**
   * Sau khi thử nghiệm, tần số thu thập dữ liệu của hệ thống có thể đạt từ **12 Hz đến 80 Hz**, tùy thuộc vào cấu hình và điều kiện hoạt động.

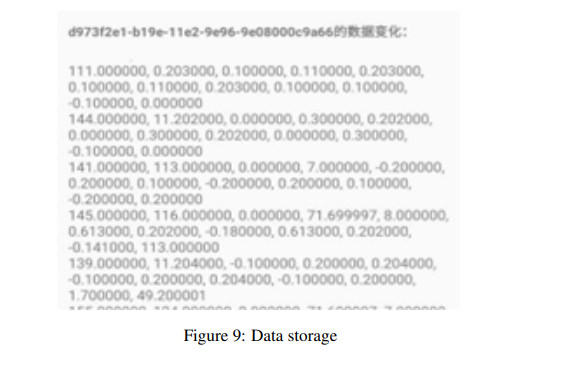
**Tối giản giao diện và trực quan hóa dữ liệu:**

Để giữ cho giao diện đơn giản, nhóm phát triển đã sử dụng thư viện mã nguồn mở **ECharts** cho việc trực quan hóa dữ liệu. Khi dữ liệu được gọi lại trả về và nhận được đối tượng dữ liệu đã phân tích, dữ liệu này sẽ được hiển thị dưới dạng chuỗi (String) trên giao diện. Đồng thời, tensor dữ liệu đã phân tích được sử dụng làm đầu vào để chạy mô hình học sâu, từ đó thu được kết quả trạng thái chuyển động tương ứng với dữ liệu chuyển động. Những kết quả này có thể được trực quan hóa thông qua **EChartsView** hoặc được sử dụng cho cảnh báo trạng thái chuyển động.

**Tối ưu hóa mô hình học máy trên thiết bị hạn chế tài nguyên:**

Ngoài ra, để đảm bảo mô hình chạy hiệu quả trên các thiết bị có tài nguyên tính toán và bộ nhớ hạn chế, đồng thời cho phép các thiết bị chạy mô hình học máy ngoại tuyến, nhóm sử dụng **TensorFlow Lite** như một bộ công cụ để triển khai quy trình công việc end-to-end.

**TensorFlow Lite** có thể được coi là bao gồm hai phần chính:

1. **Bộ chuyển đổi (Converter):** Nén và tối ưu hóa mô hình, chuyển đổi nó sang định dạng .tflite, giúp giảm kích thước và tăng tốc độ thực thi của mô hình.
2. **Bộ thông dịch (Interpreter):** Hỗ trợ nhiều môi trường runtime khác nhau, giúp mô hình chạy hiệu quả trên các thiết bị với khả năng tính toán hạn chế.

 **Mã định danh duy nhất (Unique Identifier):**

* Dòng đầu tiên trong hình ảnh chứa một chuỗi ký tự dài ("d973f2e1-b19e-11e2-9e96-9e08000c9a66"), đây có thể là một mã định danh duy nhất (UUID) được sử dụng để gán nhãn hoặc định danh một phiên ghi dữ liệu cụ thể. UUID này đảm bảo rằng mỗi tập dữ liệu có thể được truy xuất một cách chính xác.

 **Các giá trị đo lường:**

* Các giá trị phía dưới UUID là các dữ liệu số, bao gồm các giá trị như:
  + Các số thập phân (ví dụ: 111.000000, 0.203000), có thể đại diện cho các thông số như gia tốc, vận tốc góc, hoặc giá trị áp lực từ các cảm biến trên cơ thể người dùng.
  + Các giá trị dương, âm và zero, thể hiện các hướng khác nhau của chuyển động hoặc trạng thái của cảm biến trong quá trình đo lường.

 **Cấu trúc dữ liệu:**

* Dữ liệu được lưu trữ thành từng hàng và mỗi hàng chứa nhiều thông số đo lường được tách biệt bằng dấu phẩy. Điều này cho thấy dữ liệu được tổ chức dưới dạng ma trận hoặc bảng, với mỗi hàng tương ứng với một mốc thời gian cụ thể hoặc một mẫu dữ liệu từ các cảm biến.

**3.5 Data Processing Server**

**Cách triển khai và sử dụng MinIO:**

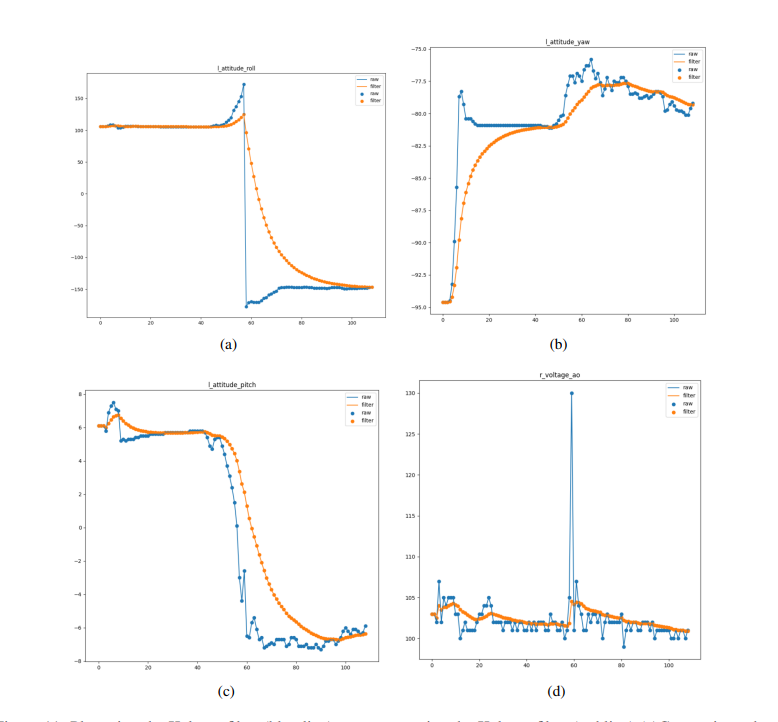
* **Sử dụng công cụ Docker để triển khai MinIO Server:**
  + Chúng tôi đã sử dụng các công cụ Docker để xây dựng máy chủ MinIO. Sau khi lấy hình ảnh MinIO từ trang web chính thức, chúng tôi kích hoạt nó theo các hướng dẫn được chỉ định, sau đó thiết lập thông tin tài khoản người dùng và tạo thư mục chứa mô hình và dữ liệu.
  + Trong quá trình vận hành tiếp theo, dữ liệu động được thu thập từ thiết bị sẽ được tải lên đây. Đồng thời, với các cập nhật tiếp theo từ nhóm chúng tôi, sẽ có nhiều mô hình phát hiện được cung cấp cho mọi người sử dụng.

**Lợi ích và tính năng của việc sử dụng MinIO:**

1. **Lưu trữ dữ liệu linh hoạt và quy mô lớn:**
   * MinIO cho phép lưu trữ và quản lý hiệu quả lượng lớn dữ liệu không cấu trúc, phù hợp với các ứng dụng cần xử lý và lưu trữ nhiều loại tệp khác nhau.
2. **Tương thích với nhiều hệ thống khác:**
   * Do tương thích với giao diện Amazon S3, MinIO có thể dễ dàng tích hợp với các dịch vụ đám mây khác và các hệ thống lưu trữ hiện có.
3. **Dễ dàng triển khai và tích hợp:**
   * Nhờ sự nhẹ nhàng và linh hoạt, MinIO có thể được triển khai nhanh chóng và dễ dàng kết hợp với nhiều ứng dụng, giúp quản lý dữ liệu trở nên dễ dàng và hiệu quả hơn.

****

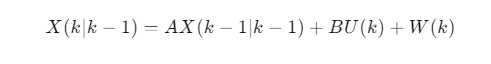
**4 Models**

* 1. **Fall Data acquizition**

Trong nghiên cứu này, nhóm phát triển đã sử dụng bộ lọc **Kalman** để làm sạch nhiễu cho dữ liệu chuỗi thời gian. Bộ lọc Kalman được thiết lập với ước tính ban đầu là giá trị hiện tại, các tham số được đặt như sau:

* **A = 1**,
* **W(k) = 0** (vì đo lường lỗi của thiết bị khá phức tạp, nhóm bỏ qua phần này và thực hiện dự đoán theo xu hướng chung của toàn bộ dữ liệu),
* **H = 1**.

Phương trình Kalman cơ bản được sử dụng là:

****

Hình 11 bao gồm các biểu đồ so sánh về hướng ba chiều và áp lực của bàn chân trái. Các phần của hình minh họa như sau:

* **(a) Biểu đồ so sánh trước và sau xử lý của góc lăn (roll angle):**
  + Ở nhóm 57, có một thay đổi đột ngột về góc phương vị, nhưng sau khi sử dụng bộ lọc Kalman, nó được làm giảm dần thành một giá trị đệm giảm.
* **(b) Biểu đồ so sánh góc pitch trước và sau xử lý:**
  + Trong khoảng từ nhóm 50 đến nhóm 80, giá trị giảm chịu ảnh hưởng của nhiễu và dao động, nhưng sau khi xử lý, sự thay đổi giá trị trở nên mượt mà hơn, phù hợp với các quy luật vật lý.
* **(c) Biểu đồ so sánh góc yaw trước và sau xử lý:**
  + Hiển thị tương tự như các phần trên, bộ lọc Kalman giúp làm mượt sự dao động dữ liệu.
* **(d) Biểu đồ so sánh áp lực của bàn chân trái trước và sau xử lý:**
  + Các dữ liệu được làm mượt bằng cách sử dụng bộ lọc Kalman, giúp phân phối dữ liệu theo xu hướng chung của toàn bộ dữ liệu.

**Nhận xét về Bộ lọc Kalman:**

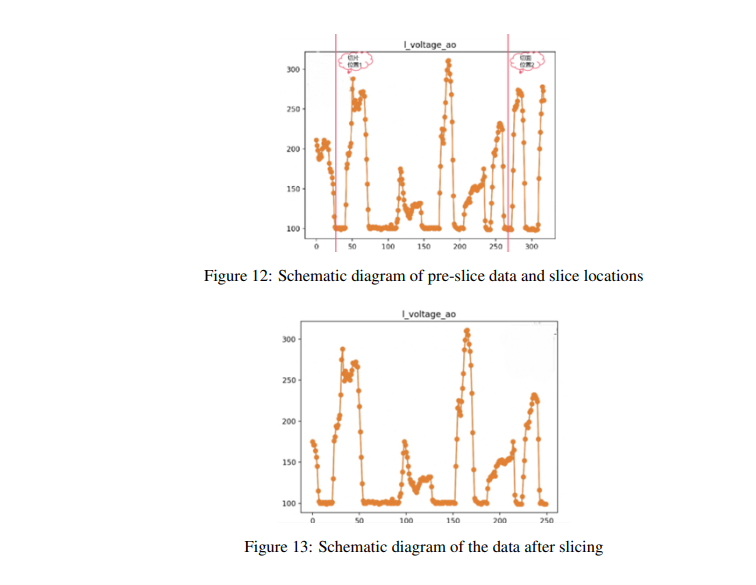
Bộ lọc Kalman được sử dụng để làm mượt các dữ liệu bất thường và đảm bảo rằng dữ liệu tuân theo xu hướng chung. Kết quả cho thấy rằng bộ lọc Kalman có hiệu quả rất tốt trong việc làm sạch dữ liệu và làm giảm sự dao động bất thường.

**Các phương pháp tăng cường dữ liệu:**

Do mẫu dữ liệu huấn luyện tương đối nhỏ, để cải thiện độ chính xác của mô hình, các phương pháp tăng cường dữ liệu như làm nhiễu (dithering), lật (flipping), phóng to hoặc thu nhỏ (zooming in or out), uốn cong (bending), sắp xếp lại (arranging), và cửa sổ trượt (sliding windows) được sử dụng. Tuy nhiên, vì các hoạt động rơi hoặc đi bộ bình thường phải tuân theo quy luật vật lý, các phương pháp như làm nhiễu và lật sẽ không phù hợp, vì chúng sẽ phá hủy tính toàn vẹn và quy luật của chuỗi dữ liệu.

**Phương pháp chia nhỏ (Slicing):**

Phương pháp chính được áp dụng ở đây là chia nhỏ dữ liệu (**slicing**), có nghĩa là một chuỗi dữ liệu dài của hoạt động đi bộ bình thường được chia thành nhiều đoạn ngắn hơn. Cụ thể:

* Điểm chia đầu tiên được đặt tại nhóm 25.
* Điểm chia thứ hai được đặt tại nhóm 275.
* ****Một đoạn dữ liệu mới dài 250 được trích xuất để sử dụng cho việc huấn luyện.

**Hình 12: Sơ đồ vị trí dữ liệu trước khi cắt (Pre-slice data)**

* **Mô tả:** Hình 12 hiển thị một sơ đồ minh họa vị trí dữ liệu trước khi tiến hành cắt (slicing). Trong sơ đồ, hai vị trí cắt đã được xác định (được đánh dấu bằng hai đường màu đỏ tại các điểm từ nhóm 25 đến nhóm 275).
* **Ý nghĩa:**
  + Các đường cắt này đánh dấu khoảng thời gian trong chuỗi dữ liệu đại diện cho hoạt động đi bộ bình thường. Việc lựa chọn các vị trí cắt này giúp loại bỏ các phần không cần thiết của dữ liệu hoặc tạo ra các đoạn dữ liệu ngắn hơn từ một chuỗi dữ liệu dài.
  + Mục đích chính của việc cắt dữ liệu (slicing) này là để chuẩn bị dữ liệu cho quá trình huấn luyện mô hình học máy bằng cách tạo ra các tập dữ liệu nhỏ hơn và đồng nhất.

**Hình 13: Sơ đồ dữ liệu sau khi cắt (Post-slice data)**

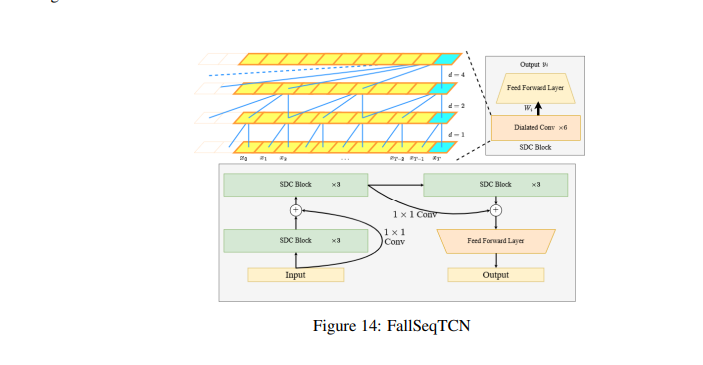
* **Mô tả:** Hình 13 cho thấy một sơ đồ của dữ liệu sau khi đã thực hiện quá trình cắt. Dữ liệu bây giờ được biểu diễn từ nhóm 25 đến nhóm 275, và chỉ còn lại các điểm dữ liệu nằm trong đoạn được chọn.
  1. **FallSeqTCN**

**FallSeqTCN** là một mô hình phát hiện té ngã nhị phân dựa trên mạng **Temporal Convolutional Network (TCN)** [10]. TCN là một mạng dự đoán chuỗi thời gian sử dụng phép tích chập giãn cách (dilated convolution), lấy cảm hứng từ **WaveNet** và **TCN**. Mô hình này có khả năng xử lý nhiều chuỗi thời gian đồng thời và yêu cầu ít thời gian huấn luyện hơn. Giống như TCN, cấu trúc kết nối dư thừa (residual-connected structure) được sử dụng trong FallSeqTCN, và các phép tích chập giãn cách và tích chập nhân quả (causal convolution) được đưa vào từng khối.

**Cấu trúc của SDC Block:**

* **SDC Block (Stacked Dilated Convolution Block):** Được kết hợp từ nhiều mạng tích chập 1 chiều có giãn cách và độ dài bằng nhau với các lớp đệm không (zero-padding convolution networks) và một mạng lan truyền thẳng tuyến tính (linear feed-forward network).
* **Kết nối dư thừa (Residual Connection):** Được thiết lập giữa mỗi 3 khối SDC liên tiếp.
* **Không sử dụng kỹ thuật chuẩn hóa (Normalization):** Vì việc chuẩn hóa có thể làm giảm khả năng đại diện vật lý của dữ liệu gốc. Tuy nhiên, nhóm nghiên cứu vẫn đang tìm cách để tối ưu hóa hơn nữa ý nghĩa vật lý của dữ liệu và tiềm năng của một mô hình tổng quát hơn cho tất cả các dữ liệu cảm biến.

**Đặc điểm và quá trình huấn luyện của mô hình FallSeqTCN:**

* **Chuỗi đầu vào:** Mỗi chuỗi đầu vào gồm 20 tín hiệu gia tốc trong miền thời gian và 2 tín hiệu áp lực chân được lấy mẫu với tần số 18 Hz.
* **Chuyển đổi dữ liệu chuỗi:** Giống như lọc theo cửa sổ (windowing filtering), dữ liệu chuỗi được chuyển thành các lô cửa sổ (window batches) với bước là 1 và độ dài là 64, sau đó các lô này được đưa vào mô hình để huấn luyện và kiểm tra.
* **Tỷ lệ chia dữ liệu huấn luyện và kiểm tra:** 7:3.
* **Kết quả dự đoán:** Xác suất dự đoán được lấy từ lớp **softmax** và cuối cùng là kết quả phát hiện té ngã nhị phân.
* **Giảm thiểu hiện tượng quá khớp (Overfitting):** Để giải quyết vấn đề quá khớp trong quá trình huấn luyện, nhóm đã thêm tùy chọn **Dropout**.

**5. Thí nghiệm (Experiments)**

Trong phần này, chúng tôi trình bày chi tiết về cài đặt thí nghiệm và kết quả tương ứng. Chúng tôi thực hiện các phân tích và điều tra toàn diện để minh họa hiệu quả của mô hình **FallSeqTCN**. Dữ liệu và mã nguồn của **FallSeqTCN** đã được cung cấp kèm theo bản nộp này.

**5.1 Bộ dữ liệu (Datasets)**

Chúng tôi sử dụng hai bộ dữ liệu để đánh giá hiệu suất của mô hình **FallSeqTCN**:

1. **UMAFall:**
   * Được thu thập thông qua mô phỏng có hệ thống một tập hợp các hoạt động hàng ngày định sẵn (**Activities of Daily Life - ADLs**) và các trường hợp té ngã vào năm 2016.
2. **Dữ liệu của chúng tôi:**
   * Được thu thập từ thiết bị cảm nhận dữ liệu gắn ở chân trái và chân phải của người tham gia. Trong cùng một môi trường thí nghiệm, các đối tượng đã thực hiện 11 lần đi bộ bình thường, 10 lần té ngã về phía trước, và 5 lần té ngã về phía bên trái.

**Mục tiêu của Thí nghiệm:**

* **Đánh giá hiệu suất:** So sánh và đánh giá hiệu suất của mô hình **FallSeqTCN** dựa trên hai bộ dữ liệu khác nhau, một bộ từ một nghiên cứu trước đó và một bộ được thu thập từ thí nghiệm hiện tại.
* **Khẳng định hiệu quả của mô hình:** Phân tích kết quả để khẳng định hiệu quả của mô hình trong việc phát hiện té ngã từ dữ liệu cảm biến.

Những bộ dữ liệu này cung cấp một nền tảng đa dạng và toàn diện để kiểm tra khả năng của mô hình **FallSeqTCN** trong các tình huống thực tế khác nhau.

**5.2 Các Mô Hình Đối Chiếu (Baselines)**

Chúng tôi so sánh **FallSeqTCN** với các mô hình sau để đánh giá hiệu quả của phương pháp đề xuất:

1. **SVM (Support Vector Machine):**
   * Sử dụng thuật toán SVM để tính toán kết hợp gia tốc, tốc độ gia tốc và ngưỡng góc độ thái độ của các trường hợp té ngã và các hành vi hàng ngày. Sau đó, thuật toán dự đoán được tái cấu trúc trên máy tính vi mạch để thực hiện dự đoán té ngã thời gian thực.
2. **Decision Tree (Cây Quyết Định):**
   * Sử dụng cây quyết định để xây dựng mối quan hệ ánh xạ giữa thuộc tính của đối tượng và giá trị đối tượng, sau đó thực hiện dự đoán té ngã.
3. **LSTM (Long Short-Term Memory):**
   * Dựa trên một mạng bộ nhớ ngắn hạn dài một lớp để thực hiện dự đoán, sử dụng bộ nhớ lớn để lưu trữ các kết quả đầu ra một phần từ nhiều cổng tế bào của chúng.

**5.3 Cài Đặt Thí Nghiệm (Experimental Setup)**

Là một mô hình phân loại nhị phân cho các trường hợp bình thường và té ngã, mô hình này đánh giá kết quả phân loại bằng cách xây dựng ma trận nhầm lẫn (confusion matrix) từ kết quả phân loại của tập kiểm tra. Bốn giá trị trong ma trận nhầm lẫn được định nghĩa như sau:

* **T (True):** Đúng
* **F (False):** Sai
* **P (Positive):** Kết quả 1
* **N (Negative):** Kết quả 0

Các giá trị cụ thể:

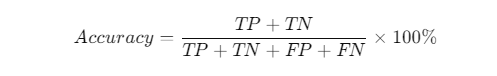
* **TP (True Positive):** Giá trị dự đoán là 1, giá trị thực tế là 1, và dự đoán là đúng.
* **FP (False Positive):** Giá trị dự đoán là 1, nhưng giá trị thực tế là 0, và dự đoán là sai.
* **FN (False Negative):** Giá trị dự đoán là 0, nhưng giá trị thực tế là 1, và dự đoán là sai.
* **TN (True Negative):** Giá trị dự đoán là 0, giá trị thực tế là 0, và dự đoán là đúng.

Dựa trên bốn chỉ số này, chúng tôi sử dụng các công thức để tính ba chỉ số hiệu suất:

1. **Accuracy (Độ Chính Xác):** Tỷ lệ phần trăm của các kết quả dự đoán đúng trong tổng mẫu.
2. **Precision (Độ Chính Xác của Mẫu Dương):** Xác suất mà tất cả các mẫu được dự đoán là dương thực sự là dương.
3. **Recall (Độ Nhạy):** Xác suất một mẫu được dự đoán là dương trong một mẫu thực sự là dương.

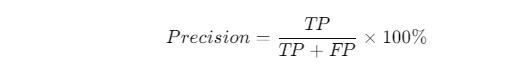
Đây là các công thức tính toán cho các chỉ số hiệu suất của mô hình phân loại nhị phân được sử dụng để đánh giá mô hình phát hiện té ngã **FallSeqTCN**:

1. **Accuracy (Độ chính xác):**
   * Công thức:



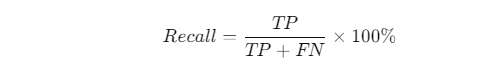
* + Độ chính xác biểu thị tỷ lệ phần trăm các dự đoán đúng (cả đúng dương và đúng âm) trên tổng số mẫu.

1. **Precision (Độ chính xác của mẫu dương):**
   * Công thức:



* + Độ chính xác biểu thị xác suất mà các mẫu được dự đoán là dương (1) thực sự là dương.

1. **Recall (Độ nhạy):**
   * Công thức:



* + Độ nhạy biểu thị xác suất mà mẫu thực sự dương (1) được dự đoán đúng là dương.

**Giải thích các thành phần trong công thức:**

* **TP (True Positive):** Số lượng dự đoán đúng là 1 trong khi giá trị thực tế cũng là 1.
* **TN (True Negative):** Số lượng dự đoán đúng là 0 trong khi giá trị thực tế cũng là 0.
* **FP (False Positive):** Số lượng dự đoán sai là 1 trong khi giá trị thực tế là 0.
* **FN (False Negative):** Số lượng dự đoán sai là 0 trong khi giá trị thực tế là 1.

**5.4 Kết quả (Results)**

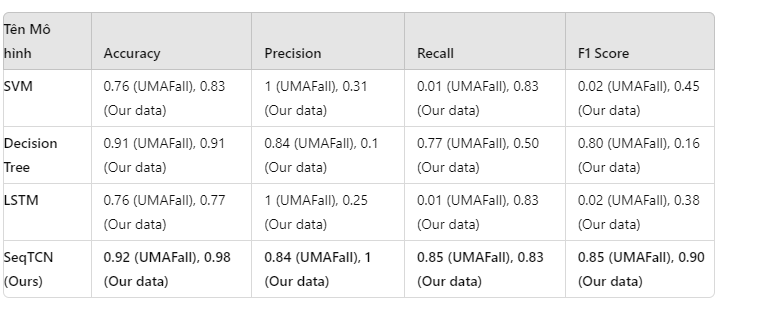
Thông qua huấn luyện so sánh, có thể thấy rằng mạng thời gian tích chập (time convolutional network) có khả năng mạnh mẽ hơn trong việc nắm bắt các đặc điểm của chuỗi thời gian hiệu quả dài hạn. **Recall** của mô hình **SeqTCN** trong bộ dữ liệu tự kiểm tra và bộ dữ liệu **UMAFall** lần lượt đạt 83% và 85%, và các điểm số **F1** lần lượt đạt 0.90 và 0.85.

**1) Hiệu suất tổng thể (Overall Performance):**

* Qua huấn luyện so sánh, mạng thời gian tích chập (TCN) đã chứng minh được khả năng mạnh mẽ hơn trong việc nắm bắt các đặc điểm của chuỗi thời gian dài hạn hiệu quả.
* **Recall** của mô hình **SeqTCN** trên bộ dữ liệu tự kiểm tra và bộ dữ liệu **UMAFall** lần lượt đạt **83%** và **85%**.
* Các điểm số **F1** của mô hình đạt **0.90** trên bộ dữ liệu tự kiểm tra và **0.85** trên bộ dữ liệu **UMAFall**.

**2) Rủi ro thấp (Low Risk):**

* Vì đây là một mô hình phát hiện té ngã, **Recall** nên được tăng lên càng nhiều càng tốt trong khi vẫn đảm bảo độ chính xác (**Precision**) ở mức hợp lý.
* Sau khi so sánh, các điểm số **F1** của mô hình **TCN** cao hơn, và **Recall** của cả hai bộ dữ liệu đều đạt trên **80%**. Đây là lý do tại sao mô hình **TCN** cuối cùng được chọn.

****

**Phân tích kết quả:**

1. **SeqTCN (mô hình của chúng tôi):**
   * **Accuracy:** Mô hình của chúng tôi đạt được độ chính xác cao nhất trên cả hai bộ dữ liệu (**0.92** cho **UMAFall** và **0.98** cho dữ liệu của chúng tôi).
   * **Precision:** Độ chính xác của mô hình **SeqTCN** đạt **0.84** trên bộ dữ liệu **UMAFall** và **1** trên dữ liệu của chúng tôi, cho thấy mô hình có khả năng dự đoán các trường hợp té ngã đúng cao.
   * **Recall:** Độ nhạy của mô hình đạt **0.85** trên bộ dữ liệu **UMAFall** và **0.83** trên dữ liệu của chúng tôi, cho thấy mô hình có khả năng phát hiện đúng các trường hợp té ngã thực tế.
   * **F1 Score:** Điểm số F1, một thước đo kết hợp giữa Precision và Recall, cho thấy mô hình của chúng tôi có hiệu suất cao nhất (**0.85** cho **UMAFall** và **0.90** cho dữ liệu của chúng tôi).
2. **So sánh với các mô hình khác:**
   * **SVM** có độ chính xác thấp hơn (**0.76** cho **UMAFall** và **0.83** cho dữ liệu của chúng tôi) và rất thấp về Recall và F1 Score trên bộ dữ liệu **UMAFall**.
   * **Decision Tree** có độ chính xác cao (**0.91** trên cả hai bộ dữ liệu), nhưng Recall trên dữ liệu của chúng tôi chỉ đạt **0.50**, làm giảm điểm F1 (**0.16**).
   * **LSTM** có độ chính xác tương đương với **SVM** nhưng Recall trên **UMAFall** rất thấp (**0.01**), dẫn đến điểm số F1 cực thấp (**0.02**).

### ****6. Kết luận (Conclusion)****

Bài báo này đã trình bày một nghiên cứu toàn diện về việc thu thập chuyển động dựa trên các cảm biến nhúng, bao gồm con quay hồi chuyển, gia tốc kế, và cảm biến áp suất. Chúng tôi đã xây dựng một hệ thống phát hiện té ngã hoàn chỉnh có tên là **TSFallDetect**.

Chúng tôi đã thực hiện các nghiên cứu thực nghiệm trên các bộ dữ liệu hiện có và các bộ dữ liệu được thu thập một cách hệ thống, và kết quả cho thấy mô hình của chúng tôi có những lợi thế so với các phương pháp truyền thống, điều này xác nhận tính khả thi và hiệu quả của hệ thống của chúng tôi.

Mạng tích chập thời gian (**Temporal Convolutional Network - TCN**) thể hiện khả năng mạnh mẽ trong việc nắm bắt các đặc trưng của chuỗi thời gian dài hạn hiệu quả, qua đó xác nhận tiềm năng của mạng này cho việc thu thập chuyển động dựa trên các cảm biến nhúng.