

# XÂY DỰNG HỆ THỐNG NHẬN DẠNG, PHÂN TÍCH HÀNH ĐỘNG TẾ NGÃ SỬ DỤNG CẢM BIẾN GIA TỐC

Phạm Thị Hồng Ngọc, Nguyễn Đức Thường, Nguyễn Đào Nguyên Giáp, Nguyễn Hải Phong

Nhóm 3, CNTT 16-04, Khoa Công Nghệ Thông Tin

Trường Đại Học Đại Nam, Việt Nam

ThS. Lê Trung Hiếu, Gv. Nguyễn Văn Nhân

Giảng viên hướng dẫn, Khoa Công Nghệ Thông Tin

Trường Đại Học Đại Nam, Việt Nam

**Tóm tắt nội dung**—Tóm tắt: Hệ thống ghi nhận hoạt động tế ngã của con người sử dụng cảm biến giúp hỗ trợ đánh giá hành động tế ngã rồi đưa ra cảnh báo qua tin nhắn. Yêu cầu đặt ra là dữ liệu hoạt động cao, giá cả rẻ và hoạt động theo thời gian thực ngay trên các vi điều khiển có hiệu năng thấp. Từ dữ liệu thu thập, cần trích xuất các đặc trưng quan trọng để huấn luyện mô hình phân loại hành động. Một thiết bị gắn trên eo người dùng được lập trình kết nối hệ thống thông qua WiFi, cho phép nhận diện hành động tế ngã theo thời gian thực, đồng thời người dùng có thể giám sát và quản lý dữ liệu trên máy tính hoặc điện thoại thông minh.

**Từ khóa:** tế ngã, cảm biến gia tốc, học máy, thời gian thực.

## I. GIỚI THIỆU

Trong bối cảnh dân số người cao tuổi ngày càng gia tăng và yêu cầu tự động hóa trong các lĩnh vực y tế cũng như an toàn lao động, việc ứng dụng công nghệ hiện đại để giám sát và cảnh báo khi xảy ra tai nạn là rất cần thiết. Hệ thống giám sát và phân tích hành động tế ngã mà nhóm chúng tôi phát triển là một ví dụ điển hình của sự kết hợp giữa AI (Trí tuệ nhân tạo) và IoT (Internet of Things), nhằm nâng cao chất lượng chăm sóc và đảm bảo an toàn cho người sử dụng.

Về phần IoT, hệ thống sử dụng cảm biến MPU6050 được gắn trên người để thu thập dữ liệu gia tốc và con quay hồi chuyển. Dữ liệu thu thập được gửi qua module ESP32, một thiết bị có khả năng kết nối WiFi, tới máy chủ Flask. Điều này cho phép hệ thống giám sát theo thời gian thực thông qua giao diện web, giúp người quản lý theo dõi liên tục tình trạng của người sử dụng.

Phía AI, dữ liệu cảm biến được xử lý thông qua các bước tiền xử lý như chuẩn hóa, tạo cửa sổ trượt và cân bằng dữ liệu trước khi đưa vào mô hình học sâu kết hợp giữa CNN và LSTM. Kiến trúc này cho phép hệ thống nhận diện chính xác các trạng thái hành động khác nhau,

đặc biệt là các trường hợp tế ngã. Khi phát hiện hành động tế ngã, hệ thống tự động gửi cảnh báo qua các kênh như Telegram, từ đó giúp người giám hộ và nhân viên y tế có thể kịp thời can thiệp và hỗ trợ.

Hơn nữa, hệ thống có thể được ứng dụng linh hoạt trong nhiều lĩnh vực khác nhau như giám sát an toàn lao động trong môi trường công nghiệp, hỗ trợ phục hồi chức năng cho bệnh nhân sau chấn thương, cũng như hỗ trợ theo dõi và cải thiện hiệu suất cho vận động viên. Đây chính là minh chứng cho tiềm năng của ứng dụng AIoT trong việc tạo ra môi trường sống và làm việc an toàn, thông minh và hiệu quả hơn.

Với sự kết hợp chặt chẽ giữa phần cứng tiên tiến và các thuật toán phân tích dữ liệu hiện đại, hệ thống của chúng tôi không chỉ đáp ứng được nhu cầu giám sát, cảnh báo kịp thời mà còn mở ra nhiều hướng ứng dụng thực tiễn, góp phần cải thiện chất lượng cuộc sống và đảm bảo an toàn cho cộng đồng. Những nghiên cứu và phát triển về hệ thống này mở đường cho các ứng dụng tương lai trong việc cá nhân hóa dịch vụ chăm sóc sức khỏe và theo dõi tình trạng sức khỏe liên tục. Các cải tiến trong thuật toán AI giúp hệ thống ngày càng chính xác hơn trong việc nhận diện các trạng thái khẩn cấp, từ đó giảm thiểu tối đa rủi ro cho người sử dụng. Ngoài ra, việc tích hợp giao diện web thân thiện giúp người quản lý dễ dàng kiểm soát và theo dõi dữ liệu thu thập được theo thời gian thực. Sự linh hoạt trong thiết kế cho phép hệ thống dễ dàng mở rộng và tích hợp với các công nghệ mới, đáp ứng nhu cầu ngày càng đa dạng của thị trường. Như vậy, dự án không chỉ là minh chứng cho sự tiến bộ của công nghệ mà còn là nền tảng vững chắc để hướng tới một tương lai thông minh và an toàn hơn cho cộng đồng.

Trong nghiên cứu này, nhóm đề xuất phương pháp kết hợp huấn luyện mô hình bằng CNN + LSTM. Tại vì CNN

trích xuất các đặc trưng không gian từ dữ liệu cảm biến, còn LSTM ghi nhớ và phân tích mối quan hệ thời gian giữa các đặc trưng đó. Sự kết hợp này giúp cải thiện độ chính xác trong việc phát hiện té ngã, hỗ trợ triển khai hệ thống giám sát, từ đó cung cấp cảnh báo kịp thời và hiệu quả cho người dùng.

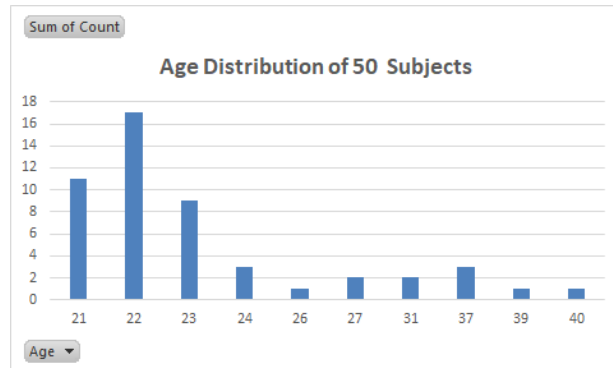
Như vậy, hệ thống sẽ được triển khai theo các bước sau: (1) thu thập dữ liệu, (2) tiền xử lý, (3) trực quan hóa dữ liệu, (4) huấn luyện mô và (5) thực nghiệm đánh giá kết quả mô hình đã huấn luyện.

## II. CÁC NGHIÊN CỨU LIÊN QUAN

Có rất nhiều phương pháp đã được đề xuất cho bài toán phát hiện té ngã. Các phương pháp này tập trung vào sử dụng cảm biến gia tốc và học máy. Chúng tôi tham khảo 2 phương pháp bên dưới này để bổ trợ cho dự án của chúng tôi.

### 2.1. Nghiên cứu Continuous Multimodal Multi-view Dataset of Human Fall

Dự án là sự hợp tác giữa Khoa Thị giác Máy tính (COMVIS) – Viện MICA và Viện Công nghệ Bưu chính Viễn thông (PTIT), nhằm xây dựng tập dữ liệu **CMDFALL** để hỗ trợ nghiên cứu phát hiện ngã. Ngã là một trong những rủi ro lớn đối với người cao tuổi sống một mình, có thể gây hậu quả nghiêm trọng nếu không được xử lý kịp thời. Do đó, nhiều nghiên cứu đã phát triển thuật toán nhận diện ngã dựa trên camera hoặc cảm biến gia tốc. Tuy nhiên, vẫn thiếu một tập dữ liệu chuẩn, quy mô lớn để đánh giá hiệu quả của các thuật toán này.



Hình 1. Biểu đồ phân bố độ tuổi của đối tượng.

CMDFALL được xây dựng để giải quyết vấn đề trên, bao gồm 20 hành động (cả hoạt động thường ngày và ngã giả lập), thu thập từ 50 người (30 nam, 20 nữ, độ tuổi 21-40). Dữ liệu được ghi nhận đồng thời từ **camera Microsoft Kinect** và **cảm biến gia tốc không dây**, giúp cung cấp nhiều góc nhìn và nguồn dữ liệu khác nhau. Mục tiêu của tập dữ liệu này là hỗ trợ cộng đồng nghiên cứu trong việc phát triển và đánh giá các thuật toán phát hiện ngã, góp phần nâng cao an toàn cho người cao tuổi.

**Phản ứng & Phân loại hành động:** Hệ thống thu thập dữ liệu CMDFALL sử dụng **7 camera Microsoft Kinect** và **2 cảm biến gia tốc không dây** để ghi lại mỗi hành động. Tập dữ liệu gồm 20 hành động, trong đó có 8 kiểu ngã khác nhau và 12 hoạt động hàng ngày hoặc tương tự ngã.

### 2.2. Nghiên cứu Real-Time Fall Detection

Một hệ thống phát hiện ngã mạnh mẽ dựa trên IoT sử dụng **YOLOv11** và **ESP32-CAM** cho việc giám sát thời gian thực trong môi trường nhiều người. Dự án này ghi lại các luồng hình ảnh từ ESP32, xử lý chúng ở phía máy chủ bằng Python, và sử dụng các thuật toán phát hiện tiên tiến để xác định chính xác các trường hợp ngã. Với khả năng phát trực tiếp và phân tích thời gian thực, hệ thống rất phù hợp cho việc giám sát liên tục trong các ứng dụng chăm sóc sức khỏe, chăm sóc người cao tuổi và an ninh, đảm bảo phát hiện và phản ứng sự cố hiệu quả.



Hình 2. Ảnh đeo thiết bị của dự án

### Thành phần chính:

- **ESP32-CAM:** Truyền dữ liệu hình ảnh qua Wi-Fi đến máy chủ.
- **YOLOv11:** Mô hình nhận diện đối tượng giúp phát hiện ngã trong khung hình.
- **Python:** Xử lý hình ảnh, phát hiện ngã và quản lý dữ liệu.
- **Flask:** Cung cấp giao diện web để giám sát trực tiếp.
- **Xử lý đa luồng:** Đảm bảo hệ thống hoạt động mượt mà, xử lý đồng thời truyền dữ liệu, phát hiện ngã và cung cấp dịch vụ web.

Hệ thống kết hợp tính toán biên và thuật toán phát hiện tiên tiến, giúp giám sát hiệu quả và giảm rủi ro liên quan đến ngã.

### III. PHƯƠNG PHÁP NGHIÊN CỨU

Từ các nghiên cứu liên quan, nhóm chúng tôi đã đưa ra một phương pháp tối ưu nhất là sử dụng ESP32 và cảm biến MPU6050 để tự thu dữ liệu cảm biến rồi huấn luyện mô hình. Tiếp theo, xây dựng một hệ thống phát hiện té ngã sử dụng Flask, TensorFlow và Telegram để gửi cảnh báo khi té ngã xảy ra. Đây là các bước thực hiện của hệ thống:

#### A. Người dùng đeo thiết bị

Người dùng đeo thiết bị có cảm biến đo gia tốc ở eo. Cảm biến thu thập dữ liệu chuyển động của người dùng (Acceleration).

#### B. Khối phần cứng (Hardware block)

Bao gồm ESP32 và cảm biến gia tốc.

Cảm biến đo dữ liệu chuyển động và gửi về vi điều khiển ESP32.

ESP32 xử lý sơ bộ và gửi dữ liệu đến máy chủ thông qua giao tiếp HTTP.

#### C. Khối giao tiếp (Communication block)

Đây là máy chủ trung gian nhận dữ liệu từ ESP32.

Máy chủ có thể là Webserver chạy Flask hoặc Node.js.

Dữ liệu được chuyển tiếp đến hệ thống xử lý.

#### D. Khối dữ liệu (Data block)

Lưu trữ dữ liệu cảm biến ở định dạng CSV.

Dữ liệu được sử dụng để huấn luyện và phân loại bằng Machine Learning.

#### E. Khối Machine Learning (Machine Learning block)

Dữ liệu từ Data block được dùng để huấn luyện mô hình AI. Mô hình sử dụng kết hợp CNN và LSTM để phân loại hoạt động của người dùng hoặc phát hiện bất thường. Kết quả dự đoán được gửi lại hệ thống.

#### F. Gửi thông báo đến thiết bị

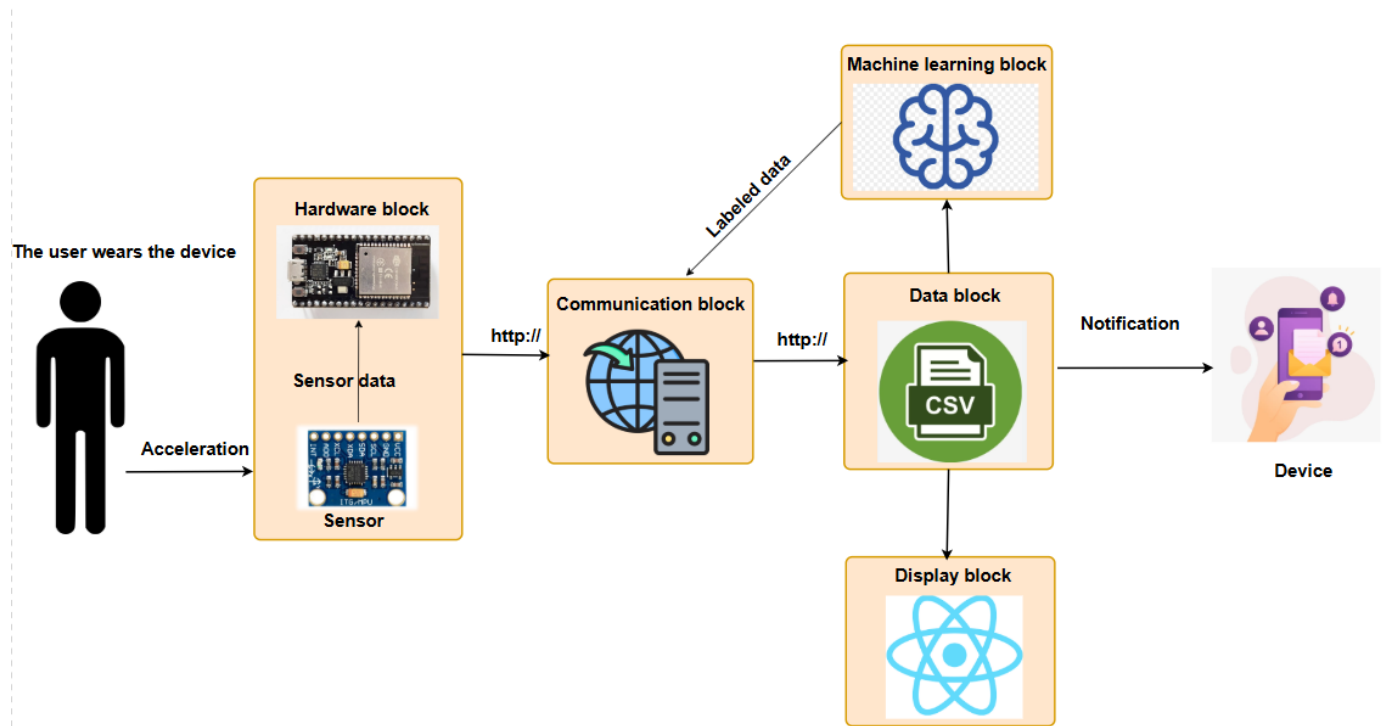
Nếu mô hình phát hiện bất thường hoặc có sự kiện quan trọng, hệ thống gửi thông báo đến người dùng.

Thông báo có thể được gửi qua ứng dụng di động hoặc giao diện web.

#### G. Khối hiển thị (Display block)

Sử dụng React để hiển thị dữ liệu theo thời gian thực.

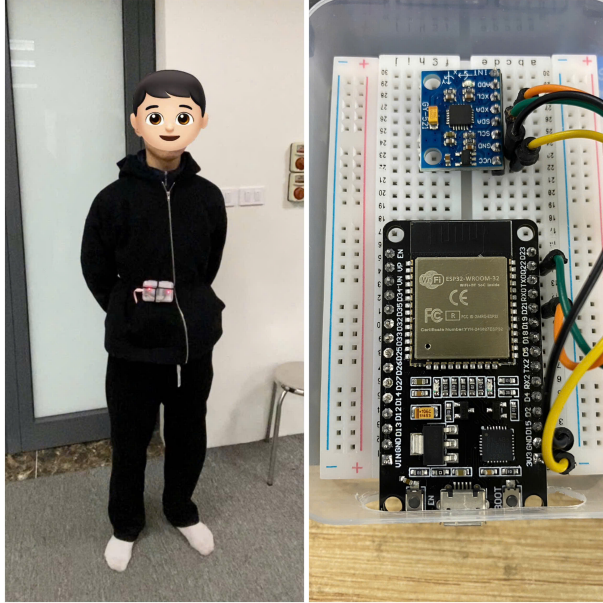
Có thể bao gồm biểu đồ, thông tin cảnh báo hoặc lịch sử hoạt động.



Hình 3. Sơ đồ hệ thống

### 3.1. Hệ thống thu dữ liệu

Thiết bị giám sát được gắn trên eo người dùng và thu thập dữ liệu gia tốc theo ba trục Ax, Ay, Az. Dữ liệu được phân đoạn bằng phương pháp cửa sổ trượt và xử lý mất cân bằng bằng SMOTE. Mỗi vectơ đặc trưng được huấn luyện bằng thuật toán cây quyết định để nâng cao hiệu suất phân loại.



Hình 4. Thiết bị thu dữ liệu

### Quá trình thu thập dữ liệu

Xây dựng chương trình trên ESP32

- Chúng ta sẽ xây dựng chương trình trong Arduino để thu dữ liệu. Chương trình này chạy trên **ESP32**, sử dụng cảm biến **MPU6050** để thu thập dữ liệu gia tốc và con quay hồi chuyển. Các bạn hãy kết nối phần cứng như ảnh sau...
- Sau khi **ESP32** kết nối với **WiFi**, mở **Serial Monitor** để xem địa chỉ **IP**. Dùng trình duyệt truy cập vào địa chỉ **IP** của **ESP32**.
- Trang **Web Server** cho phép người dùng bắt đầu/dừng ghi dữ liệu với nhân hành động cụ thể (ngã về các hướng, đứng yên) thông qua giao diện web. Dữ liệu cảm biến cũng được hiển thị theo thời gian thực trên web.
- Khi bộ đệm đầy, dữ liệu được ghi vào file để tối ưu hiệu suất. Chúng ta sẽ tải file **CSV** qua nút "Download" trên web.

### 3.2. Dữ liệu

Thiết bị được đeo như hình trên. Dữ liệu sau khi thu thập được tiền xử lý để loại bỏ tín hiệu nhiễu, gán nhãn theo 5 lớp: ngã trước, ngã sau, ngã trái, ngã phải, đứng yên. Tần số lấy mẫu: 50Hz.

### 3.3. Tiền xử lý dữ liệu

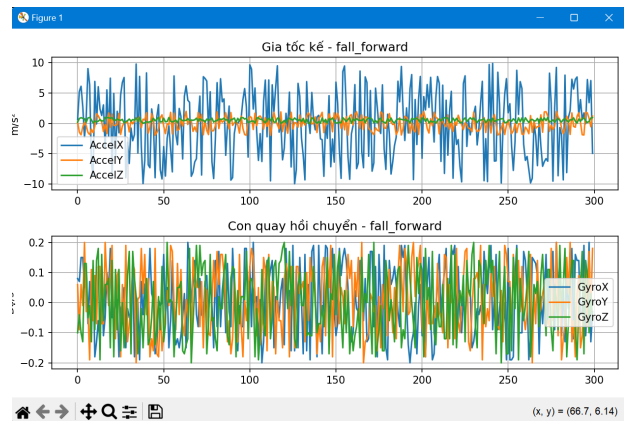
Dữ liệu được chuẩn hóa bằng StandardScaler, nhãn mã hóa bằng LabelEncoder. Cửa sổ trượt kích thước 50 mẫu, bước nhảy 25 mẫu. SMOTE được sử dụng để cân bằng dữ liệu, lưu dưới dạng NumPy (.npy) phục vụ huấn luyện.

**Số lượng cửa sổ sau khi cân bằng: 1365.**

- Shape của X: (1365, 50, 6)
- Shape của Y: (1365, )

### 3.4. Trực quan hóa dữ liệu

Giúp kiểm tra và phân tích đặc trưng tín hiệu cảm biến cho từng hành động té ngã.



Hình 5. Biểu đồ trực quan hóa dữ liệu

### 3.5. Huấn luyện mô hình CNN + LSTM

**Tập dữ liệu được sử dụng để huấn luyện**

Bảng 1  
DATASET SUMMARY

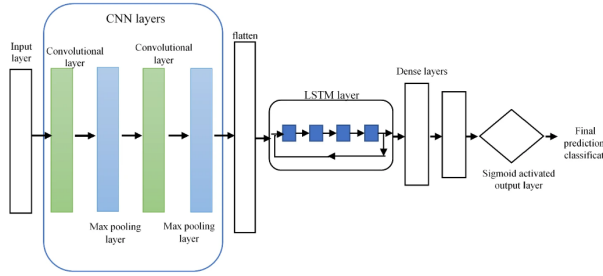
Attribute	Quantity
Classes	5
Frequency	50
Subject	1
Training set	1092
Testing set	273

**CNN + LSTM** là mô hình kết hợp khả năng trích xuất đặc trưng không gian của CNN và khả năng ghi nhớ chuỗi thời gian của LSTM, giúp nhận diện hành động té ngã hiệu quả hơn.

**CNN** phân tích đặc trưng tư thế từ ảnh hoặc tín hiệu cảm biến.

**LSTM** theo dõi sự thay đổi theo thời gian, giúp phân biệt giữa các hành động như đi bộ, ngồi xuống, đứng lên, nằm xuống chậm và té ngã.

Sự kết hợp này giúp mô hình không chỉ nhận diện tư thế tức thời mà còn hiểu được diễn biến của hành động, cải thiện độ chính xác trong phát hiện té ngã.



Hình 6. Sơ đồ mô hình huấn luyện CNN + LSTM

#### Xây dựng mô hình huấn luyện:

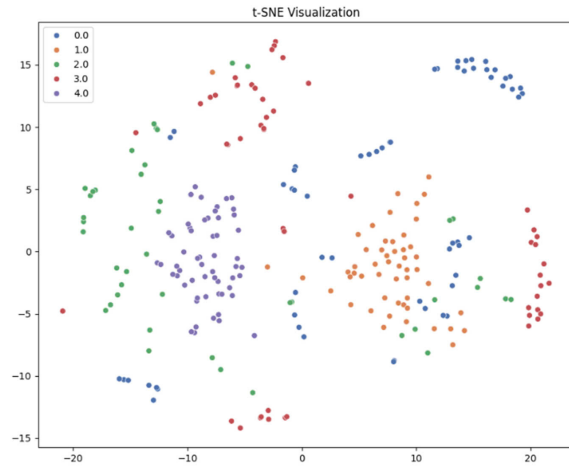
- **2 lớp Conv1D + MaxPooling1D:** Trích xuất đặc trưng từ dữ liệu chuỗi thời gian.
- **2 lớp LSTM:** Học thông tin theo thời gian.
- **Dense 128 + Dropout 0.3:** Lớp ẩn để tăng khả năng học của mô hình.
- **Dense 5 (softmax):** Lớp đầu ra với 5 nhãn, tương ứng với 5 trạng thái té ngã.

#### Định dạng đầu vào: (50, 6)

- 50: Số lượng bước thời gian.
- 6: Số lượng đặc trưng của mỗi bước.

#### 3.6. Phân loại hành động

Sử dụng **t-SNE** để giảm số chiều dữ liệu, hỗ trợ trực quan hóa và phân biệt hành động tốt hơn.



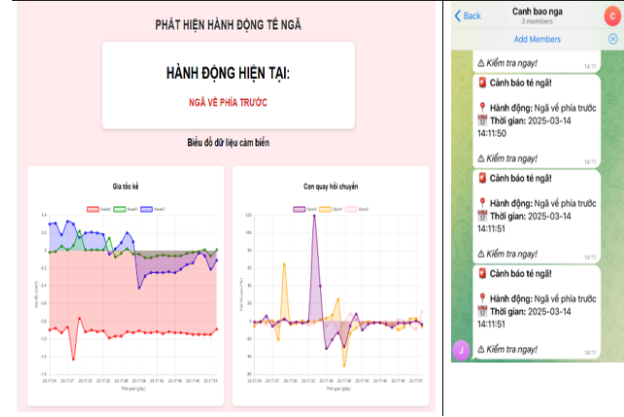
Hình 7. Sơ đồ trực quan t-SNE

#### 3.7. WebServer và ứng dụng phân loại hành động

Ứng dụng web bằng **Flask** nhận dữ liệu cảm biến từ ESP32, dự đoán hành động bằng TensorFlow và gửi cảnh

báo qua Telegram. Cung cấp giao diện giám sát dữ liệu thời gian thực.

Giao diện của ứng dụng trên điện thoại thông minh được xây dựng giúp người sử dụng dễ dàng quan sát tham số thu thập từ cảm biến theo đơn vị gia tốc g hoặc m/s<sup>2</sup> và thời gian cập nhật dữ liệu từ cảm biến.



Hình 8. Ảnh hệ thống nhận diện hành động té ngã

## IV. KẾT QUẢ VÀ ĐÁNH GIÁ

### 4.1. Phương pháp đánh giá

Sau khi huấn luyện mô hình, tiến hành dự đoán trên tập kiểm tra ( $X_{test}$ ) bằng cách lấy lớp có xác suất cao nhất làm nhãn dự đoán cuối cùng. Để đánh giá hiệu suất của mô hình, sử dụng báo cáo phân loại (Classification Report), trong đó cung cấp các chỉ số quan trọng như Precision (độ chính xác), Recall (độ nhạy) và F1-score cho từng lớp.

Precision đo lường tỷ lệ số mẫu được dự đoán đúng trên tổng số mẫu mà mô hình dự đoán là thuộc lớp đó:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

Trong đó, TP (True Positive) là số lượng dự đoán đúng dương tính, FP (False Positive) là số lượng dự đoán sai dương tính.

Recall phản ánh khả năng mô hình phát hiện đúng các mẫu thực sự thuộc một lớp nhất định:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

Với FN (False Negative) là số lượng dự đoán sai âm tính.

F1-score là trung bình điều hòa giữa Precision và Recall:

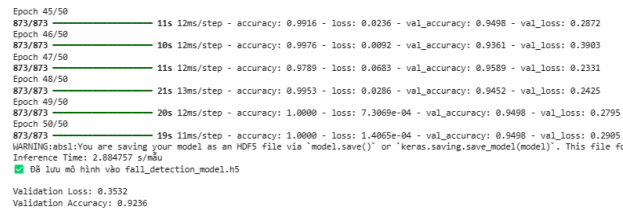
$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$



#### 4.2. Đánh giá mô hình phân loại

Mô hình CNN+LSTM phân loại 5 hành động, gồm Conv1D, MaxPooling1D, 2 lớp LSTM, Dense và Dropout. Huấn luyện với 50 epochs, batch size 1, Adam optimizer và sparse categorical crossentropy.

Bảng phía trên trình bày kết quả đánh giá mô hình phân loại hành động té ngã dựa trên dữ liệu cảm biến, với các chỉ số Precision, Recall và F1-score cho từng loại hành động. Mô hình đạt hiệu suất cao với F1-score từ 96.1% đến 100%, trong đó hành động "Stand normally" có độ chính xác tuyệt đối (100%). Các hành động té ngã cũng được phân loại tốt, đặc biệt "Lean to the right" đạt 99.1% F1-score. Tuy nhiên, hành động "Lean back" có Recall thấp nhất (94.2%), cho thấy mô hình có thể bỏ sót một số trường hợp. Nhìn chung, mô hình hoạt động hiệu quả và có tiềm năng ứng dụng trong giám sát té ngã cho người cao tuổi.



Hình 9. Bảng đánh giá mô hình CNN + LSTM

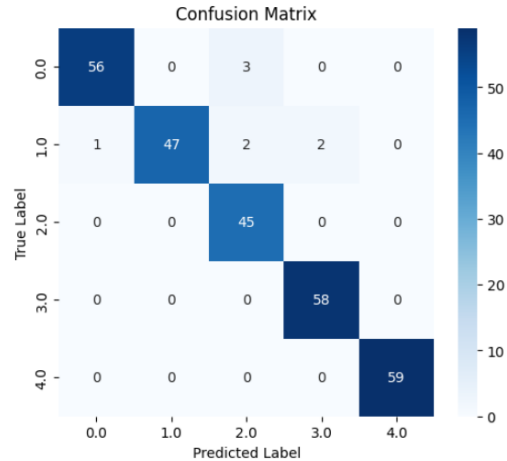
Bảng II  
CLASSIFICATION PERFORMANCE METRICS

Action class	Precision	Recall	F1-score	Support
0.0 (Lean forward)	96.6%	96.6%	96.6%	59 mẫu
1.0 (Lean back)	98.0%	94.2%	96.1%	52 mẫu
2.0 (Lean to the left)	95.7%	97.8%	96.7%	45 mẫu
3.0 (Lean to the right)	98.3%	100%	99.1%	58 mẫu
4.0 (Stand normally)	100%	100%	100%	59 mẫu

#### 4.3. Kết quả thực nghiệm

Hai lớp 3 và 4 có dự đoán hoàn toàn chính xác. Lớp 0, 1 và 2 có một vài trường hợp dự đoán nhầm, nhưng số lượng rất ít. Nhìn chung, đường chéo chính có giá trị rất cao, thể hiện mô hình có độ chính xác cao trong việc phân loại. Màu sắc đậm ở đường chéo (theo thang màu bên phải) cho thấy số mẫu đúng chiếm ưu thế vượt trội so với các mẫu sai.

Kết quả đánh giá hiệu suất mô hình được thể hiện trong bảng dưới có độ chính xác (Accuracy) của mô hình đạt 98.17% cho thấy phần lớn các dự đoán đều chính xác. Các chỉ số Precision, Recall và F1-score cũng đạt giá trị cao, cho thấy mô hình có độ cân bằng tốt giữa việc phát hiện đúng các mẫu thuộc từng lớp và hạn chế các dự đoán sai.



Hình 10. Sơ đồ ma trận nhầm lẫn

Bảng III  
BẢNG KẾT QUẢ CÁC HIỆU SUẤT HUẤN LUYỆN MÔ HÌNH

Method	Value
Accuracy	98.17%
Macro avg Precision	98.05%
Macro avg Recall	98.11%
Macro avg F1-score	98.07%
Weighted avg Precision	98.18%
Weighted avg Recall	98.17%
Weighted avg F1-score	98.17%

Cụ thể, Macro average Precision đạt 0.9805, nghĩa là trung bình cộng của Precision trên tất cả các lớp là 98.05%. Macro average Recall đạt 0.9811, cho thấy mô hình có khả năng nhận diện tốt các mẫu thực sự dương tính. Macro average F1-score là 0.9807, đảm bảo sự cân bằng giữa Precision và Recall.

#### V. KẾT LUẬN

Đề tài "Xây dựng hệ thống giám sát và phân tích hành động té ngã" đã được triển khai thành công với mục tiêu chính là phát hiện và cảnh báo kịp thời khi người đeo thiết bị gặp sự cố té ngã. Hệ thống sử dụng cảm biến MPU6050 kết hợp với ESP32 để thu thập dữ liệu, truyền về máy chủ Flask xử lý bằng mô hình AI, từ đó hiển thị kết quả trên giao diện web và gửi cảnh báo qua Telegram.

Qua quá trình thực hiện, nhóm đã hoàn thành các bước quan trọng gồm: thu thập và xử lý dữ liệu cảm biến, huấn luyện mô hình nhận diện hành động bằng CNN + LSTM, triển khai hệ thống giám sát trên web và tích hợp chức năng cảnh báo theo thời gian thực. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, có độ chính xác cao trong việc phát hiện các hành động té ngã.

Tuy nhiên, vẫn còn một số hạn chế cần khắc phục như: tối ưu tốc độ xử lý mô hình, cải thiện độ chính xác

trong các điều kiện môi trường khác nhau và thử nghiệm trên nhiều đối tượng để tăng tính tổng quát. Trong tương lai, nhóm sẽ tiếp tục phát triển hệ thống với các tính năng nâng cao như sử dụng cảm biến bổ sung (camera, radar), ứng dụng trên thiết bị di động và cải thiện thuật toán nhận diện để nâng cao hiệu suất.

Đề tài không chỉ mang lại giá trị ứng dụng thực tiễn cao trong việc hỗ trợ giám sát người cao tuổi, bệnh nhân cần theo dõi sức khỏe mà còn mở ra hướng nghiên cứu và phát triển các giải pháp AI-IoT trong lĩnh vực y tế và chăm sóc sức khỏe.

Trong tương lai, tôi sẽ tiếp tục phát triển và cải thiện hệ thống bằng cách thử nghiệm trên các hoạt động phức tạp hơn nhằm đánh giá độ chính xác của mô hình. Bên cạnh đó, việc mở rộng tập đặc trưng sang miền tần số hoặc kết hợp cả hai miền thời gian – tần số sẽ được xem xét để nâng cao hiệu suất phân loại. Ngoài ra, trong tương lai chúng tôi sẽ nghiên cứu thêm để xây dựng mô hình hiện đại phù hợp với thời đại công nghệ số.

### LỜI CẢM ƠN

Nhóm chúng tôi xin trân thành cảm ơn Th.S Lê Trung Hiếu và Giảng viên Nguyễn Văn Nhân đã hỗ trợ rất nhiệt tình và tâm huyết trong đề tài này. Xin trân trọng cảm ơn.

### TÀI LIỆU THAM KHẢO

- [1] Pham, Minh, et al. "Cloud-based smart home environment (CoSHE) for home healthcare." *IEEE International Conference on Automation Science and Engineering*, pp. 483-488, 2016.
- [2] Peetoom, Kirsten KB, et al. "Literature review on monitoring technologies..." *Disability and Rehabilitation: Assistive Technology*, vol. 10, no. 4, pp. 271-294, 2015.
- [3] Tan, Tran Duc, and Nguyen Van Tinh. "Reliable fall detection system..." *APSIPA*, pp. 1-6, 2014.
- [4] Van Thanh, Pham, et al. "Development of a real-time, simple and high-accuracy fall detection system..." *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3329-3342, 2019.
- [5] Stiefmeier, Thomas, et al. "Wearable activity tracking in car manufacturing." *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 42-50, 2018.
- [6] Taha, Ahmed, et al. "Human activity recognition for surveillance applications." *Proceedings of the 7th International Conference on Information Technology*, pp. 577-586, 2015.
- [7] Putra, I. P. E. S., et al. "An event-triggered machine learning approach for accelerometer-based fall detection." *Sensors*, vol. 18, no. 1, 2018.
- [8] Ravi, Nishkam, et al. "Activity recognition from accelerometer data." *AAAI*, vol. 5, pp. 1541-1546, 2005.
- [9] Bao, Ling, and Stephen S. Intille. "Activity recognition from user-annotated acceleration data." *International Conference on Pervasive Computing*, pp. 1-17, 2004.
- [10] Min, Ya, et al. "Comparing the performance of machine learning algorithms..." *International Journal of Computer (IJC)*, vol. 38, no. 1, pp. 61-72, 2020.
- [11] Vavoulas, George, et al. "The MobiAct dataset..." *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, vol. 2, pp. 143-151, 2016.