

XÂY DỰNG HỆ THỐNG GIÁM, 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 giám sát và phân tích hành động tế ngã là một ứng dụng quan trọng trong chăm sóc sức khỏe, đặc biệt đối với người cao tuổi. Nghiên cứu này đề xuất một phương pháp nhận diện hành động tế ngã trong thời gian thực bằng cách sử dụng dữ liệu từ cảm biến MPU6050 gắn trên thiết bị ESP32. Dữ liệu gia tốc và con quay hồi chuyển được thu thập, truyền qua giao thức HTTP đến máy chủ để phân tích bằng mô hình học sâu CNN + LSTM triển khai trên Google Colab.

Quy trình xử lý dữ liệu bao gồm chuẩn hóa, mã hóa nhân, tạo cửa sổ trượt và cân bằng dữ liệu bằng SMOTE. Mô hình được huấn luyện để phân loại năm trạng thái: đứng, ngã về phía trước, ngã về phía sau, ngã sang trái và ngã sang phải, với độ chính xác cao trên tập kiểm tra. Kết quả được hiển thị trên giao diện web, đồng thời hệ thống có thể gửi cảnh báo qua Telegram khi phát hiện tế ngã.

Thử nghiệm cho thấy hệ thống hoạt động ổn định, đáp ứng tốt yêu cầu giám sát theo thời gian thực, hứa hẹn ứng dụng hiệu quả trong các hệ thống chăm sóc sức khỏe thông minh.

Từ khóa: tế ngã, cảm biến gia tốc, mô hình học sâu CNN+LSTM, thời gian thực.

I. GIỚI THIỆU

Trong bối cảnh chăm sóc sức khỏe ngày càng hiện đại và nhu cầu giám sát an toàn cho người cao tuổi trở nên cấp thiết, việc phát hiện và phân tích hành động tế ngã đóng vai trò then chốt trong việc hỗ trợ y tế kịp thời và giảm thiểu nguy cơ chấn thương nghiêm trọng. Hệ thống giám sát và phân tích hành động tế ngã của chúng tôi được phát triển dựa trên sự kết hợp của công nghệ IoT và các thuật toán học sâu, nhằm mang lại khả năng nhận diện các trạng thái hành động một cách chính xác và hiệu quả trong thời gian thực.

Hệ thống sử dụng cảm biến MPU6050 được gắn trên cơ thể người để thu thập dữ liệu gia tốc và con quay hồi chuyển. Dữ liệu này sau đó được truyền từ thiết bị ESP32 thông qua giao thức HTTP về máy chủ, nơi quá trình xử

lý và phân tích dữ liệu được thực hiện. Dữ liệu thu thập ban đầu được lưu dưới dạng CSV và trải qua một quy trình tiền xử lý toàn diện trước khi đưa vào huấn luyện mô hình học sâu.

Một trong những thách thức lớn trong việc triển khai hệ thống là sự đa dạng và phức tạp của dữ liệu cảm biến. Các tín hiệu thu được thường chứa nhiều nhiễu từ môi trường và có thể không phân biệt rõ ràng giữa các trạng thái hành động, chẳng hạn như sự khác biệt tinh vi giữa hành động đứng và các chuyển động nhẹ, hoặc giữa các loại tế ngã. Vì vậy, quá trình tiền xử lý dữ liệu đóng vai trò vô cùng quan trọng, giúp làm sạch và biến đổi dữ liệu thành dạng phù hợp cho mô hình học máy.

Những bước tiền xử lý dữ liệu chính của hệ thống bao gồm: -> Lọc các hoạt động hợp lệ: Dữ liệu cảm biến được phân loại ban đầu để chỉ giữ lại các hoạt động có ý nghĩa cho bài toán, loại bỏ những tín hiệu không liên quan hoặc nhiễu không mong muốn. -> Chuẩn hóa dữ liệu bằng StandardScaler: Việc chuẩn hóa giúp đưa các giá trị cảm biến về cùng một phạm vi, giảm thiểu sự chênh lệch giữa các đặc trưng và tạo điều kiện thuận lợi cho mô hình học máy trong quá trình huấn luyện. -> Tạo cửa sổ trượt (Sliding Window): Dữ liệu được chia thành các đoạn nhỏ với số mẫu nhất định (ví dụ: 50 mẫu với bước nhảy 25) để mô hình có thể phân tích chuỗi thời gian liên tục, từ đó nắm bắt được xu hướng và đặc điểm của các hành động. -> Xử lý mất cân bằng dữ liệu bằng SMOTE: Kỹ thuật SMOTE được áp dụng nhằm tăng cường dữ liệu cho các lớp có số lượng mẫu ít, góp phần cải thiện độ bền và khả năng phân biệt của mô hình học sâu.

Sau khi hoàn thành quy trình tiền xử lý toàn diện, dữ liệu được đưa vào huấn luyện mô hình CNN + LSTM. Mô hình này tận dụng ưu điểm của CNN trong việc trích xuất đặc trưng và của LSTM trong việc ghi nhớ thông tin chuỗi thời gian, giúp đạt được hiệu suất nhận diện các

hành động, đặc biệt là các trạng thái té ngã, với độ chính xác cao. Hệ thống không chỉ tập trung vào khía cạnh kỹ thuật mà còn chú trọng đến trải nghiệm người dùng. Một giao diện web trực quan được xây dựng nhằm hiển thị dữ liệu cảm biến theo thời gian thực, cung cấp biểu đồ và thông tin chi tiết về các hành động được phát hiện, đồng thời tích hợp cơ chế gửi cảnh báo qua Telegram khi phát hiện tín hiệu té ngã.

Đáng chú ý, nghiên cứu của chúng tôi đã thu hút sự quan tâm rộng rãi trong truyền thông. Các bài báo chuyên ngành, hội thảo công nghệ và các kênh truyền thông đã đánh giá cao khả năng ứng dụng thực tiễn của hệ thống trong việc nâng cao chất lượng cuộc sống cho người cao tuổi và cải thiện quy trình giám sát sức khỏe. Sự thành công của hệ thống không chỉ được thể hiện qua các kết quả thử nghiệm thực tế mà còn mở ra nhiều cơ hội hợp tác với các đơn vị y tế và công nghệ, góp phần thúc đẩy sự phát triển của các giải pháp chăm sóc sức khỏe thông minh. Những đóng góp chính của nghiên cứu bao gồm:

- 1) Quy trình tiền xử lý dữ liệu cảm biến toàn diện: Giúp lọc các hoạt động hợp lệ, chuẩn hóa dữ liệu, tạo cửa sổ trượt và xử lý mất cân bằng dữ liệu nhằm tăng cường chất lượng đầu vào cho mô hình học sâu.
- 2) Ứng dụng mô hình học sâu CNN + LSTM: Tích hợp ưu điểm của các kiến trúc CNN và LSTM để phân tích dữ liệu chuỗi thời gian từ cảm biến MPU6050, đạt được hiệu suất nhận diện cao.
- 3) Triển khai giao thức truyền dữ liệu hiệu quả: Sử dụng giao thức HTTP giúp giảm thiểu độ trễ trong quá trình truyền dữ liệu từ ESP32 đến máy chủ, đảm bảo tính thời gian thực của hệ thống.
- 4) Phát triển giao diện người dùng trực quan: Cho phép theo dõi dữ liệu cảm biến và kết quả nhận diện hành động một cách trực quan, hỗ trợ việc cảnh báo kịp thời khi có tình huống nguy hiểm.

Với những giải pháp kỹ thuật tiên tiến và quy trình xử lý dữ liệu chặt chẽ, hệ thống giám sát và phân tích hành động té ngã hứa hẹn mang lại hiệu quả cao trong việc ứng dụng vào các hệ thống chăm sóc sức khỏe thông minh và giám sát an toàn, đặc biệt là đối với người cao tuổi.

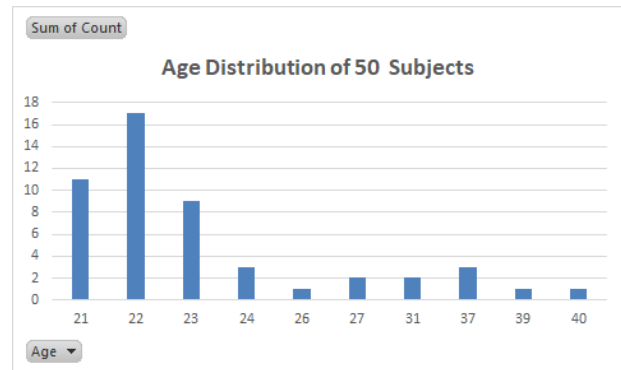
II. CÁC NGHIÊN CỨU LIÊN QUAN

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 cứng & 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.

Hệ thống này được thiết kế để phát hiện ngã bằng cách ghi lại và phân tích hình ảnh trực tiếp được truyền từ ESP32 đến máy chủ, tận dụng mô hình phát hiện đối tượng YOLOv11. Khác với các mô hình cụ thể về tư thế

truyền thống, YOLOv11 nổi bật trong việc nhận diện nhiều cá nhân cùng lúc, nâng cao độ chính xác của việc phát hiện ngã trong các môi trường đông đúc.



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 này kết hợp điện toán biên với các thuật toán phát hiện tiên tiến để cung cấp khả năng phát hiện ngã hiệu quả và đáng tin cậy. Khả năng giám sát nhiều đối tượng cùng lúc nâng cao sự an toàn trong các cơ sở y tế và nhà ở, cải thiện đáng kể thời gian phản ứng và giảm thiểu rủi ro liên quan đến ngã.

III. PHƯƠNG PHÁP NGHIÊN CỨU

Trong nghiên cứu này, chúng tôi đề xuất một phương pháp phát hiện hành động té ngã trong thời gian thực, tích hợp đầy đủ các thành phần từ thu thập dữ liệu, xử lý dữ liệu, huấn luyện mô hình đến dự đoán và hiển thị kết quả. Hệ thống sử dụng thiết bị ESP32 để thu thập dữ liệu từ cảm biến MPU6050 (bao gồm gia tốc và con quay hồi chuyển), sau đó truyền dữ liệu qua giao thức HTTP đến máy chủ. Tại máy chủ, dữ liệu được tiền xử lý qua các bước làm mịn, chuẩn hóa và phân đoạn, trước khi được đưa vào mô hình CNN + LSTM nhằm phân loại sáu hành động chính: đứng, ngồi, đi bộ, chạy bộ, té ngã và nhảy.

Trọng tâm của hệ thống là nhận diện chính xác hành động té ngã, giúp cảnh báo kịp thời trong những tình huống nguy hiểm, đặc biệt hữu ích cho người cao tuổi và các đối tượng có nguy cơ té ngã cao. Kết quả dự đoán được làm mịn để giảm hiện tượng báo động giả và tăng độ ổn định, sau đó hiển thị trên giao diện người dùng đồ họa, cho phép theo dõi dữ liệu cảm biến và trạng thái

hành động theo thời gian thực. Các bước chi tiết của phương pháp này sẽ được trình bày trong các mục dưới đây.

A. Kiến trúc hệ thống

Hệ thống giám sát và nhận diện hành động là một giải pháp ứng dụng công nghệ IoT (Internet of Things) và học sâu (Deep Learning) để theo dõi chuyển động của người dùng và phát hiện các tình huống nguy hiểm như té ngã. Trong những năm gần đây, việc sử dụng thiết bị đeo thông minh để theo dõi sức khỏe và giám sát hoạt động trở nên ngày càng phổ biến, đặc biệt là đối với người cao tuổi, bệnh nhân có nguy cơ té ngã cao, hoặc các vận động viên cần phân tích chuyển động.

Hệ thống được thiết kế với cấu trúc phân tầng, bao gồm các khối chức năng chính:

1) Khối phần cứng (Hardware Block): Hệ thống sử dụng cảm biến MPU6050 để thu thập dữ liệu gia tốc và con quay hồi chuyển từ người đeo thiết bị. Các giá trị gia tốc theo 3 trục (X, Y, Z) và dữ liệu con quay hồi chuyển theo 3 trục góc quay giúp xác định chuyển động và tư thế của người dùng.

Thiết bị đeo: Người dùng đeo một thiết bị nhỏ gọn chứa cảm biến MPU6050 và vi điều khiển ESP32.

ESP32: Bo mạch ESP32 thu thập dữ liệu cảm biến với tần số lấy mẫu cao, xử lý sơ bộ và gửi dữ liệu đến máy chủ thông qua mạng Wi-Fi hoặc MQTT.

2) Khối dữ liệu (Data Block): Khối dữ liệu lưu trữ thông tin cảm biến trong tệp CSV hoặc cơ sở dữ liệu NoSQL để sử dụng cho việc huấn luyện mô hình.

3) Khối truyền thông (Communication Block): Khối truyền thông chịu trách nhiệm gửi dữ liệu cảm biến từ thiết bị ESP32 lên máy chủ để xử lý.

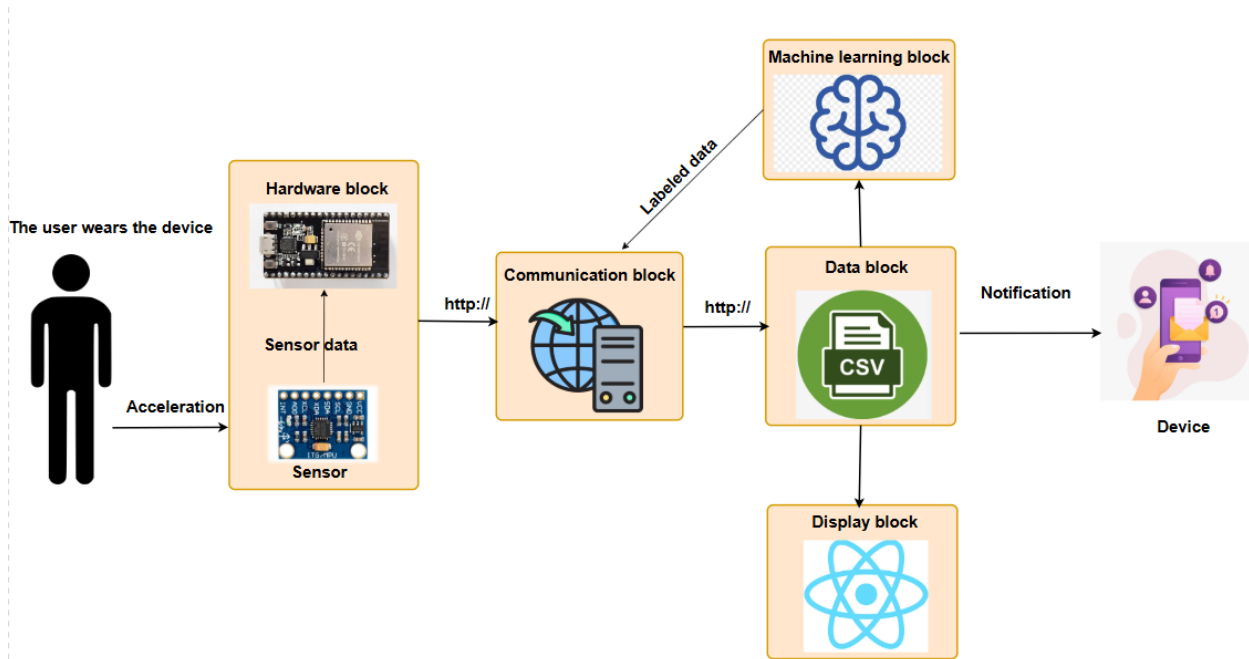
Giao thức truyền: Sử dụng HTTP hoặc MQTT để đảm bảo truyền dữ liệu thời gian thực.

Kết nối mạng: ESP32 có thể kết nối Wi-Fi hoặc 4G (nếu tích hợp thêm module mạng). Khối dữ liệu (Data Block): Lưu trữ và xử lý dữ liệu, phục vụ huấn luyện mô hình.

4) Khối học máy (Machine Learning Block): Ứng dụng mô hình Transformer để nhận diện hành động.

5) Khối hiển thị và cảnh báo (Display và Notification Block): Hiển thị kết quả phân tích trên ứng dụng và gửi thông báo khi phát hiện té ngã.

Trọng tâm của hệ thống là nhận diện chính xác hành động té ngã, giúp cảnh báo kịp thời trong những tình huống nguy hiểm, đặc biệt hữu ích cho người cao tuổi và các đối tượng có nguy cơ té ngã cao. Kết quả dự đoán được làm mịn để giảm hiện tượng báo động giả và tăng độ ổn định, sau đó hiển thị trên giao diện người dùng đồ họa, cho phép theo dõi dữ liệu cảm biến và trạng thái hành động theo thời gian thực. Các bước chi tiết của phương pháp này sẽ được trình bày trong các mục dưới đây. Sơ đồ hệ thống tổng quát được minh họa trong Hình 1 dưới đây:



Hình 3: Sơ đồ của hệ thống giám sát và phân tích hành động té ngã

B. Quy trình thu dữ liệu

Để đảm bảo chất lượng dữ liệu đầu vào cho mô hình, quy trình thu thập dữ liệu được thiết kế cẩn thận với các bước cụ thể như sau:

1) Lựa chọn đối tượng: Dữ liệu được thu thập từ 1 đối tượng, là nam, trong độ tuổi từ 20 đến 40, nhằm đảm bảo tính đa dạng về thể chất và phong cách chuyển động. Mỗi đối tượng thực hiện sáu hành động (ngã về phía trước, ngã về phía sau, ngã sang bên trái, ngã sang bên phải, đứng im) trong các điều kiện khác nhau, bao gồm cả trong nhà và ngoài trời. Tổng cộng, tập dữ liệu bao gồm 12, mẫu. Số lượng mẫu cho mỗi hành động từ đối tượng được trình bày trong Bảng I.

Bảng I: Phân bố số lượng mẫu theo từng hành động

Hành động	Số mẫu trung bình mỗi người
Ngã về phía trước	6,855
Ngã về phía sau	2,542
Ngã sang bên trái	1,228
Ngã sang bên phải	915
Đứng im	660

2) Thu thập dữ liệu:

Cài Đặt Thiết Bị: Thiết bị ESP32 được gắn trên eo đối tượng để ghi nhận dữ liệu từ cảm biến gia tốc và con quay hồi chuyển, vị trí này được chọn vì ghi nhận tốt các chuyển động khi té ngã.

Ghi Nhận Dữ Liệu: Mỗi hành động được ghi lại trong thời gian phù hợp (xem Bảng II), dữ liệu được gắn nhãn thủ công qua ứng dụng ghi chú thời gian thực, đảm bảo gắn nhãn chính xác để hỗ trợ huấn luyện mô hình.

Lưu Trữ Dữ Liệu: Dữ liệu thô được lưu dưới dạng tệp JSON trong thư mục `data_action`, chứa thông tin thời gian, giá trị cảm biến, và nhãn hành động, sau đó được xử lý sơ bộ (làm mịn, chuẩn hóa) trước khi huấn luyện.

C. Trực quan hóa dữ liệu:

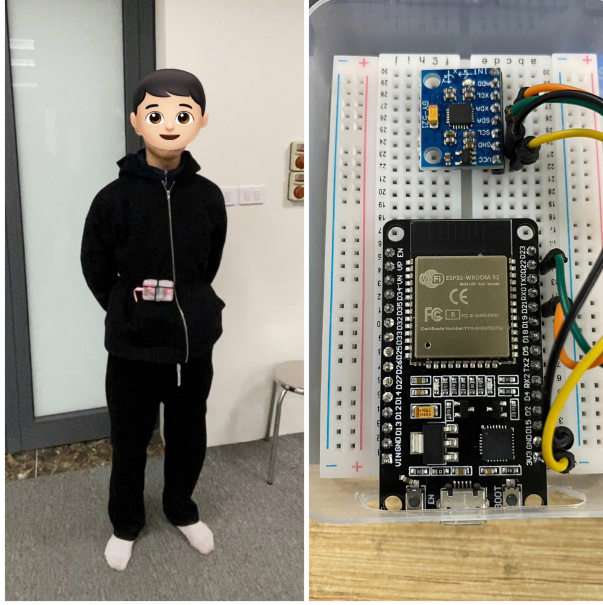
Giúp kiểm tra và phân tích đặc trưng tín hiệu cảm biến cho từng hành động té ngã.

D. Tiền xử lý dữ liệu

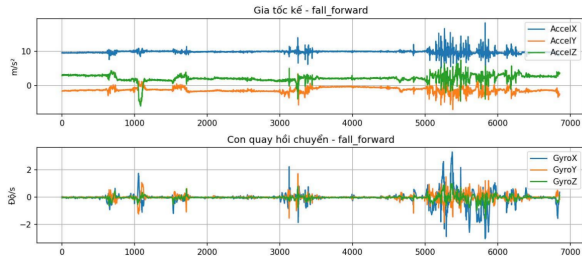
Dữ liệu cảm biến thô từ ESP32 bao gồm sáu kênh: ba trục gia tốc (AccelX, AccelY, AccelZ) đo gia tốc tuyến tính (m/s²).

1) Loại các hoạt động hợp: Giữ lại các hàng có nhãn hoạt động thuộc danh sách các hoạt động hợp lệ `fall_forward`, `fall_backward`, `fall_left`, `fall_right`, `standing` giúp tập trung vào những hành động quan trọng và loại bỏ các nhiễu không cần thiết. **Số lượng của số sau khi cân bằng: 1365.**

2) Chuẩn hóa dữ liệu: Loại bỏ cột `Time`(không cần thiết cho bài toán) và chuẩn hóa các đặc trưng cảm biến: `AccelX`, `AccelY`, `AccelZ`, `GyroX`, `GyroY`,



Hình 4: Thiết bị thu dữ liệu



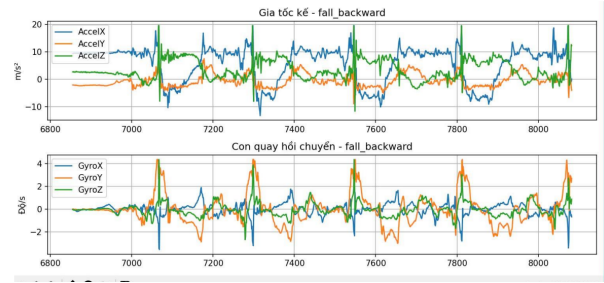
Hình 5: Biểu đồ trực quan hóa dữ liệu ngã về phía trước

GyroZ bằng StandardScaler. Mục đích là đưa các giá trị về cùng một thang đo để cải thiện hiệu quả của các mô hình học máy.

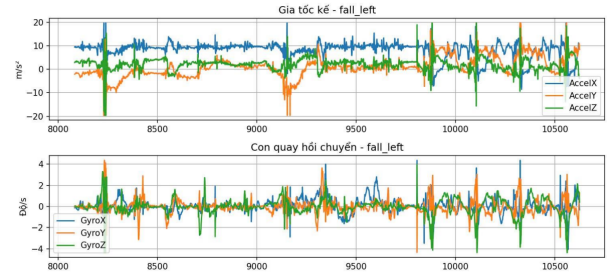
3) Mã hóa nhãn động: Sử dụng LabelEncoder để chuyển đổi các nhãn dạng chuỗi (text) thành số nguyên, giúp cho quá trình xử lý và huấn luyện mô hình dễ dàng hơn.

4) Tạo cửa sổ trượt (Sliding Window): Với kích thước cửa sổ (window size) là 50 mẫu và bước nhảy (step size) là 25 mẫu, dữ liệu được chia thành các đoạn (window) liên tiếp. Mỗi cửa sổ chứa dữ liệu của 50 mẫu cảm biến và nhãn của cửa sổ được gán bằng nhãn của mẫu cuối cùng trong cửa sổ. Quá trình này tạo ra các mẫu dữ liệu dạng 3 chiều: (số cửa sổ, 50, 6).

5) Xử lý mất cân bằng dữ liệu với SMOTE: Do số lượng mẫu của các lớp có thể không đồng đều, sử dụng SMOTE để tăng cường các lớp có số lượng mẫu ít bằng cách tạo ra các mẫu mới dựa trên các mẫu hiện có. Vì SMOTE chỉ hoạt động trên dữ liệu 2 chiều, nên các cửa sổ được reshape từ (num_windows, 50, 6) về dạng (num_windows, 300) trước khi áp dụng SMOTE,



Hình 6: Biểu đồ trực quan hóa dữ liệu ngã về phía sau



Hình 7: Biểu đồ trực quan hóa dữ liệu ngã sang bên trái

sau đó chuyển lại về dạng ban đầu.

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

6: Lưu dữ liệu đã cân bằng: Dữ liệu đã qua xử lý được lưu dưới dạng file numpy (.npz) để sử dụng cho quá trình huấn luyện mô hình sau này. In ra số lượng cửa sổ và kích thước của dữ liệu sau cân bằng

E. Huấn luyện mô hình CNN + LSTM

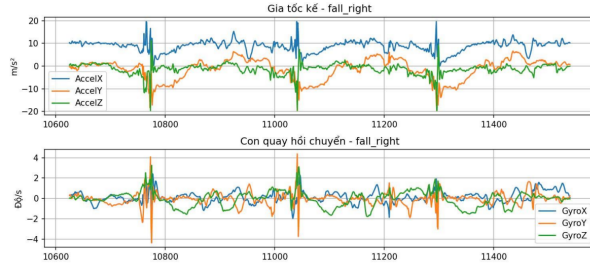
CNN + LSTM là mô hình kết hợp khả năng trích xuất đặc trưng không gian của CNN và khả năng ghi nhớ chuỗi thời gian của LSTM, giúp nhận diện hành động té ngã hiệu quả hơn.

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

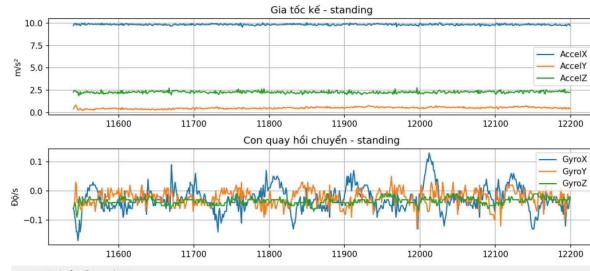
LSTM theo dõi sự thay đổi theo thời gian, giúp phân biệt giữa các hành động như đi bộ, ngồi xuống, đứng lên, nằm xuống chậm và té ngã.

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.

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.



Hình 8: Biểu đồ trực quan hóa dữ liệu ngã sang bên phải



Hình 9: Biểu đồ trực quan hóa dữ liệu đứng im

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:

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

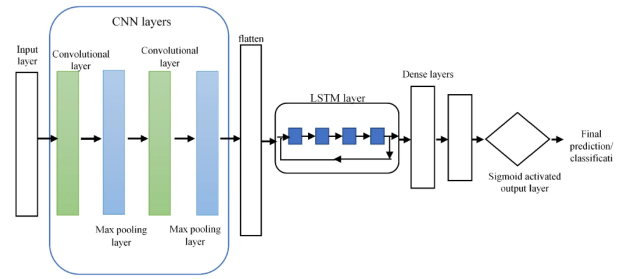
F. Dự đoán thời gian thực

1) Nhận dữ liệu: Nhận dữ liệu từ ESP32 (/sensor) Bộ nhớ đệm (Deque) Lưu trữ dữ liệu cảm biến theo dạng hàng đợi (deque) với kích thước 128.

2) Dự đoán mỗi 2 giây: Khi thu thập đủ 50 mẫu (tần số 50 Hz), tiến hành dự đoán.

3) Làm mịn kết quả: Lấy trung vị của 7 dự đoán gần nhất để làm mịn kết quả.

4) Hiển thị độ tin cậy: Tính trung bình của 7 giá trị “confidence score” để xác định độ tin cậy và hiển thị cho người dùng, hiển thị thông tin theo thời gian thực, bao gồm tên hành động, độ tin cậy ổn định (Cao/Trung



Hình 10: Sơ đồ mô hình huấn luyện CNN + LSTM

bình/Thấp), thời gian suy luận (ms), và hai biểu đồ: gia tốc (3 trục) và con quay hồi chuyển (3 trục).

5) Cảnh báo qua Telegram: Nếu phát hiện dấu hiệu té ngã, tự động gửi tin nhắn kèm theo độ tin cậy qua Telegram.

IV. THỬ NGHIỆM

A. Tập dữ liệu

Tập dữ liệu được thu thập từ một đối tượng thực hiện năm hành động: ngã về phía trước, ngã về phía sau, ngã sang bên trái, ngã sang bên phải và đứng im. Mỗi hành động được gán nhãn từ 0 đến 4 theo ánh xạ trong tệp `data_action.csv`.

Dữ liệu thô ban đầu được lưu trong thư mục `Xu_ly_du_lieu`, sau đó được xử lý (làm mịn, chuẩn hóa) và lưu vào hai tệp `X_balanced1.npy` và `y_balanced1.npy`.

Sau khi áp dụng các kỹ thuật tăng cường dữ liệu như thêm nhiễu, biến đổi thời gian và thay đổi biên độ, tập huấn luyện chứa hơn 12.000 chuỗi, trong đó 80% được dùng để huấn luyện và 20% để kiểm tra. Quá trình phân chia được thực hiện bằng hàm `train_test_split` với tham số `stratify` để đảm bảo phân bố nhân đồng đều giữa các tập.

Quy trình thu thập dữ liệu diễn ra với tần số 50 Hz, áp dụng các quy tắc cụ thể cho từng hành động như mô tả trong Bảng II. Bảng này nêu chi tiết thời gian thu thập và mục đích của từng hành động, đảm bảo dữ liệu phản ánh chính xác đặc điểm thực tế của từng hành động.

Bảng II: Quy Tắc Thu Thập Dữ Liệu

Hành động	Thời gian thực hiện	Mục đích thời gian
Ngã về phía trước	3–5 giây	Ghi lại khoảnh khắc đột ngột
Ngã về phía sau	3–5 giây	Ghi lại khoảnh khắc đột ngột
Ngã sang bên trái	3–5 giây	Ghi lại khoảnh khắc đột ngột
Ngã sang bên phải	3–5 giây	Ghi lại khoảnh khắc đột ngột
Đứng im	1 phút	Thu đủ chu kỳ, ổn định

B. Huấn luyện

Mô hình được huấn luyện trên Google Colab và xây dựng mô hình huấn luyện như sau:

1) Cài đặt các thư viện Python cần thiết để hỗ trợ huấn luyện mô hình và phân tích dữ liệu.

2) Upload file: up 2 file đã lưu ở bước tiền xử lý `X_balanced1.npy` và `y_balanced1.npy` lên Google Colab để chia tập train/test.

3) Xây dựng mô hình CNN + LSTM: 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.

Định dạng đầu vào: (50, 6)

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

Tập dữ liệu được chia để huấn luyện

Bảng III: Tóm Tắt Tập Dữ Liệu

Thuộc tính	Số lượng
Số lớp hành động	5
Tần số lấy mẫu (Hz)	50
Số đối tượng	1
Số mẫu huấn luyện	1092
Số mẫu kiểm tra	273

3) Tiến hành huấn luyện, đánh giá mô hình Mô hình cuối cùng sẽ được lưu dưới dạng `fall_detection_model.h5` trong thư mục `test`.

C. Kết quả

Bảng phía dưới 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.

Hiệu suất mô hình đánh giá trên tập kiểm tra 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.

Cụ thể, trung bình cộng của Precision trên tất cả các lớp là 98.05 %, Macro average Recall đạt 98,11 %, cho

Bảng IV: Đánh Giá Hiệu Suất Phân Loại

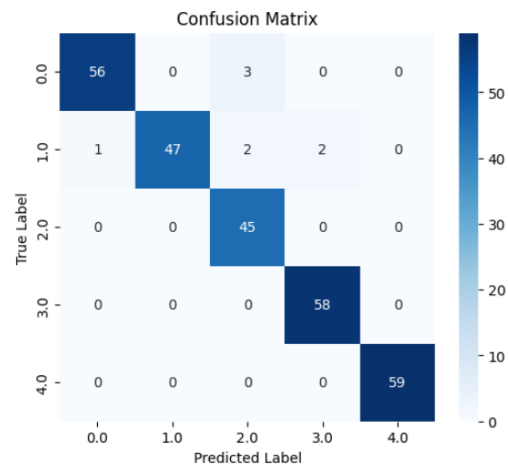
Lớp hành động	Độ chính xác	Độ nhạy	F1-score	Số mẫu
0.0 (Ngã về phía trước)	96.6%	96.6%	96.6%	59 mẫu
1.0 (Ngã về phía sau)	98.0%	94.2%	96.1%	52 mẫu
2.0 (Ngã sang bên trái)	95.7%	97.8%	96.7%	45 mẫu
3.0 (Ngã sang bên phải)	98.3%	100%	99.1%	58 mẫu
4.0 (Đứng yên)	100%	100%	100%	59 mẫu

Bảng V: Đánh Giá Chỉ Số Hiệu Suất

Chỉ số	Giá trị	Đơn vị
Độ chính xác tổng thể	98.17	%
Thời gian suy luận trung bình	2.04	ms
Độ chính xác trung bình Marco	98.05	%
Độ nhạy trung bình Marco	98.11	%
F1-score trung bình Marco	98.07	%
Độ chính xác trung bình Weighted	99.18	%
Độ nhạy trung bình Weighted	98.	%
F1-score trung bình Weighted	98.17	%

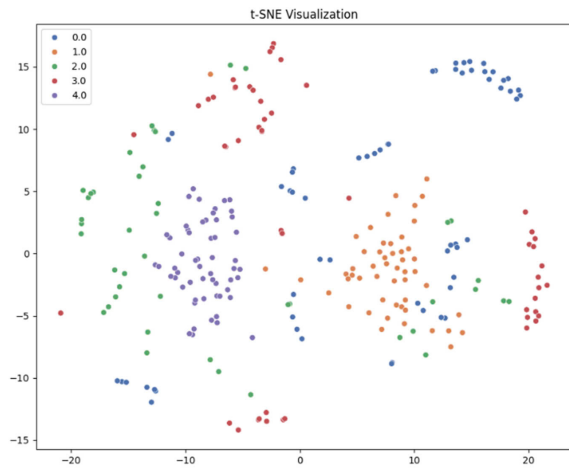
thấy mô hình có khả năng nhận diện tốt các mẫu thực sự dương tính. Macro average F1-score là 98.07%, đảm bảo sự cân bằng giữa Precision và Recall.

Ma trận nhầm lẫn với 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.



Hình 11: Sơ đồ ma trận nhầm lẫn

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 12: Sơ đồ trực quan t-SNE

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/s² và thời gian cập nhật dữ liệu từ cảm biến.



Hình 13: Ảnh hệ thống nhận diện hành động té ngã

V. KẾT LUẬN

Dựa trên các kết quả thu được, có thể kết luận rằng đề tài xây dựng hệ thống nhận diện hành động té ngã dựa trên dữ liệu cảm biến đạt được hiệu quả cao về độ chính xác trong việc phân loại các hành động. Cụ thể,

mô hình với các chỉ số Precision, Recall và F1-score đều nằm trong khoảng từ 96.1

Hơn nữa, với độ chính xác tổng thể của mô hình đạt 98.17

Về ứng dụng thực tế, hệ thống tích hợp WebServer sử dụng Flask và giao diện trên điện thoại thông minh đã cho phép thu thập và hiển thị dữ liệu cảm biến theo thời gian thực, từ đó đưa ra cảnh báo kịp thời qua Telegram khi phát hiện trường hợp té ngã. Đây là một bước tiến quan trọng trong việc ứng dụng trí tuệ nhân tạo cho việc giám sát sức khỏe và an toàn của người cao tuổi.

Hướng phát triển đề tài trong tương lai:

Cải thiện khả năng nhận diện: Nghiên cứu thêm các phương pháp tối ưu hóa mô hình, đặc biệt là cải thiện Recall cho các lớp có tỷ lệ nhận diện thấp như “Lean back”, nhằm giảm thiểu các trường hợp bỏ sót trong dự đoán.

Mở rộng dữ liệu huấn luyện: Thu thập thêm dữ liệu từ nhiều nguồn và điều kiện khác nhau để làm phong phú bộ dữ liệu, giúp mô hình học được nhiều đặc trưng phức tạp hơn trong các tình huống thực tế.

Tích hợp đa cảm biến: Kết hợp dữ liệu từ các loại cảm biến khác (ví dụ: cảm biến áp lực, cảm biến từ trường) nhằm cung cấp nhiều góc nhìn hơn cho hệ thống, từ đó cải thiện hiệu năng nhận dạng.

Triển khai hệ thống thực tế: Phát triển ứng dụng trong môi trường thử nghiệm thực tế để đánh giá tính khả thi của hệ thống trong điều kiện sống của người cao tuổi, qua đó điều chỉnh và tối ưu hóa hệ thống theo phản hồi từ người sử dụng.

Ứng dụng học sâu nâng cao: Khám phá và áp dụng các kiến trúc mạng nơ-ron tiên tiến hơn, như mạng tích chập (CNN) kết hợp với mạng hồi tiếp (RNN) hay Transformer, nhằm khai thác tối đa đặc trưng động của dữ liệu cảm biến theo thời gian.

Những hướng phát triển trên hứa hẹn sẽ mở ra nhiều khả năng ứng dụng rộng rãi và hiệu quả hơn cho hệ thống giám sát té ngã, góp phần nâng cao chất lượng cuộc sống và sự an toàn cho người cao tuổi.

LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn chân thành đến tất cả mọi người đã đồng hành, hỗ trợ và góp phần vào thành công của đề tài nghiên cứu này.

Trước hết, tôi xin bày tỏ lòng biết ơn sâu sắc đến Giảng viên Lê Trung Hiếu và Giảng viên Nguyễn Văn Nhân, những người đã tận tâm hướng dẫn và chia sẻ những kiến thức quý báu trong suốt quá trình thực hiện đề tài.

Tôi cũng xin gửi lời cảm ơn đến các thành viên trong nhóm đã cùng nhau nỗ lực, trao đổi và đóng góp ý kiến để hoàn thiện đề tài một cách tốt nhất.

Một lần nữa, xin chân thành cảm ơn tất cả!

TÀI LIỆU THAM KHẢO

- [1] A. Bulling, U. Blanke, và B. Schiele, “Hướng dẫn về phát hiện hành động con người sử dụng cảm biến quán tính đeo trên người,” *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, 2014.
- [2] A. Vaswani và cộng sự, “Chú ý là tất cả những gì bạn cần,” trong *Proc. NeurIPS*, 2017, pp. 5998–6008.
- [3] L. Bao và S. S. Intille, “Phát hiện hành động từ dữ liệu gia tốc được chú thích bởi người dùng,” trong *Proc. Pervasive*, 2004, pp. 1–17.
- [4] F. J. Ordóñez và D. Roggen, “Mạng nơ-ron tích chập sâu và hồi tiếp LSTM cho phát hiện hành động đa phương thức từ thiết bị đeo,” *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [5] J. Wang, Y. Chen, S. Hao, X. Peng, và L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *J. Sens.*, vol. 2019, pp. 1–11, 2019.
- [6] O. D. Lara và M. A. Labrador, “A survey on human activity recognition in healthcare, sports, and entertainment,” *IEEE Commun. Surv. Tutor.*, vol. 15, no. 1, pp. 292–329, 2013.
- [7] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, và U. R. Alo, “Human activity recognition using inertial sensors: A comprehensive review,” *Sensors*, vol. 18, no. 10, p. 3542, 2018.
- [8] Z. Chen, Q. Zhu, Y. C. Soh, và L. Zhang, “Robust human activity recognition using smartphone and wearable devices,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1629–1640, 2017.
- [9] T. Zebin, P. J. Scully, và K. B. Ozanyan, “Human activity recognition with wearable inertial sensors using convolutional neural networks,” trong *Proc. IEEE Sens.*, 2016, pp. 1–3.
- [10] Y. LeCun, Y. Bengio, và G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [11] S. Hochreiter và J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [12] K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” trong *Proc. EMNLP*, 2014, pp. 1724–1734.
- [13] W. Jiang và Z. Yin, “Human activity recognition using a bidirectional convolutional neural network,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1871–1883, 2015.
- [14] H. Wu, Z. Zhou, và P. Li, “Time-series classification with an improved Transformer model,” *IEEE Access*, vol. 8, pp. 197614–197622, 2020.
- [15] M. Zeng et al., “Convolutional neural networks for human activity recognition on mobile devices,” trong *Proc. UbiComp*, 2014, pp. 630–639.
- [16] G. Muhammad, M. S. Hossain, và A. Alamri, “Sensor-based IoT systems for healthcare monitoring,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1928–1936, 2018.
- [17] A. Ignatov, “Real-time human activity recognition on mobile devices with deep neural networks,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2593–2604, 2018.
- [18] R. Gravina, P. Alinia, H. Ghasemzadeh, và G. Fortino, “Multi-sensor activity recognition: Synthesis and trends,” *Inf. Fusion*, vol. 52, pp. 189–205, 2019.
- [19] S. S. Saha, S. Sandha, và M. Srivastava, “Deep learning for sensor-based IoT applications,” trong *Proc. IEEE IoT*, 2020, pp. 1–8.
- [20] D. Ravi, C. Wong, F. Deligianni, et al., “Deep learning for human activity recognition: A survey from an embedded computing perspective,” *IEEE Signal Process. Mag.*, vol. 34, no. 5, pp. 50–66, 2017.