

ĐẠI HỌC QUỐC GIA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN

LÊ HỒNG LAM

NGHIÊN CỨU PHÁT TRIỂN MỘT SỐ KỸ THUẬT
TRÍCH RÚT ĐẶC TRƯNG VÀ TỐI ƯU MÔ HÌNH
RANDOM FOREST TRONG PHÁT HIỆN SỰ KIẾN NGÃ
CỦA CON NGƯỜI BẰNG ĐIỆN THOẠI THÔNG MINH

LUẬN ÁN TIẾN SĨ
CHUYÊN NGÀNH QUẢN LÝ HỆ THỐNG THÔNG TIN

Hà Nội - 2024

ĐẠI HỌC QUỐC GIA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN

LÊ HỒNG LAM

NGHIÊN CỨU PHÁT TRIỂN MỘT SỐ KỸ THUẬT
TRÍCH RÚT ĐẶC TRƯNG VÀ TỐI ƯU MÔ HÌNH
RANDOM FOREST TRONG PHÁT HIỆN SỰ KIẾN NGÃ
CỦA CON NGƯỜI BẰNG ĐIỆN THOẠI THÔNG MINH

Chuyên ngành: Quản lý hệ thống thông tin

Mã số: 9480205.01QTD

LUẬN ÁN TIẾN SĨ

CHUYÊN NGÀNH QUẢN LÝ HỆ THỐNG THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC:

PGS. TS NGUYỄN HÀ NAM

Hà Nội - 2024

LỜI CAM ĐOAN

Tôi xin cam đoan Luận án Tiến sĩ này, với tiêu đề “*Nghiên cứu phát triển một số kỹ thuật trích rút đặc trưng và tối ưu mô hình Random Forest trong phát hiện sự kiện ngã của con người bằng điện thoại thông minh*” là công trình nghiên cứu khoa học của tôi dưới sự hướng dẫn của PGS. TS Nguyễn Hà Nam. Nội dung của luận án không có sự sao chép hay sử dụng để bảo vệ cho một học vị nào khác. Tất cả các số liệu, kết quả và tài liệu tham khảo trong luận án đều có nguồn gốc rõ ràng và đã được trích dẫn đầy đủ theo quy định.

Các kết quả nghiên cứu của tôi viết chung với các tác giả khác đã được sự nhất trí của đồng tác giả khi đưa vào luận án. Tôi chịu trách nhiệm hoàn toàn về sự trung thực và tính chính xác của những nội dung được trình bày trong luận án.

Luận án này được hoàn thành trong thời gian tôi làm Nghiên cứu sinh tại Viện Công nghệ Thông tin, Đại học Quốc gia Hà Nội.

NGƯỜI CAM ĐOAN

Nghiên cứu sinh

Lê Hồng Lam

LỜI CẢM ƠN

Quá trình học tập, nghiên cứu và hoàn thành luận án tiến sĩ này đem lại cho tôi các trải nghiệm thú vị và nhiều thách thức trong cuộc sống. Trong suốt quá trình này tôi đã nhận được sự quan tâm, giúp đỡ và động viên của rất nhiều tập thể và cá nhân. Tôi xin gửi lời cảm ơn chân thành và sâu sắc đến tất cả những người đã đồng hành cùng tôi trong suốt hành trình này.

Người đầu tiên tôi muốn bày tỏ lòng biết ơn vô hạn là thầy giáo - PGS. TS. Nguyễn Hà Nam, người đã luôn theo sát, tận tình chỉ bảo và định hướng cho tôi thực hiện công trình nghiên cứu này. Thầy là người đã truyền đạt cho tôi niềm đam mê, sự nghiêm túc, chính xác và sáng tạo trong khoa học. Thầy cũng là người đã khuyến khích và giúp đỡ tôi cả về vật chất lẫn tinh thần để vượt qua những khó khăn và thử thách trong quá trình nghiên cứu.

Tôi muốn gửi lời cảm ơn chân thành đến các thầy giáo – TS. Nguyễn Đức Nhân và các thầy, cô giáo, các anh, chị, em NCS và các em học viên cao học trong nhóm của thầy giáo Nguyễn Hà Nam. Họ đã đóng góp kiến thức, kinh nghiệm và hỗ trợ nhiều điều bổ ích để tôi ngày càng hoàn thiện.

Tiếp theo, tôi xin chân thành cảm ơn Ban lãnh đạo, Phòng Khoa học Công nghệ và Đào tạo, thầy giáo PGS. TS. Lê Hoàng Sơn và cộng sự trong LAB AI 4.0 và các phòng chức năng, chuyên môn của Viện Công nghệ Thông tin, Đại học Quốc gia Hà Nội đã tạo điều kiện thuận lợi để tôi học tập và hoàn thành luận án. Tôi cũng xin gửi lời tri ân đến các thầy cô giáo đã giảng dạy và hướng dẫn tôi trong các học phần của chương trình tiến sĩ.

Tôi xin tỏ lòng biết ơn và kính trọng đối với các hội đồng đánh giá luận án cùng các thầy cô giáo tham gia quá trình phản biện đã dành thời gian, công sức để đọc và nhận xét luận án. Các ý kiến, nhận xét, đánh giá của quý thầy cô đã giúp tôi hoàn thiện luận án và nâng cao chất lượng công trình nghiên cứu.

Ngoài ra, tôi xin cảm ơn các cơ quan, tổ chức và cá nhân đã cung cấp dữ liệu, thông tin và thiết bị cho công trình nghiên cứu này. Tôi xin ghi nhận sự giúp đỡ của các nhà khoa học, tác giả các công trình nghiên cứu tham khảo, trích dẫn trong luận án này.

Tôi sẽ không thể hoàn thành luận án này mà không nhận được sự giúp đỡ của cơ quan chủ quản. Tôi biết ơn tập thể Lãnh đạo trường Đại học Sư phạm Kỹ thuật Vinh, lãnh đạo và các đồng nghiệp tại Khoa Điện tử đã hỗ trợ, tạo điều kiện thuận lợi để tôi học tập, nghiên cứu và hoàn thành luận án này.

Đặc biệt, tôi xin bày tỏ lòng biết ơn sâu sắc đến cha, mẹ đẻ, mẹ vợ, hai con nhỏ và người vợ yêu quý của tôi là Nguyễn Thị Anh Quỳnh - người đã không quản mọi gian lao, vất vả, luôn bên cạnh, chia sẻ và hy sinh vì tôi. Em là nguồn động lực lớn nhất cho tôi vượt qua mọi khó khăn và thách thức trong suốt quá trình học tập, nghiên cứu và hoàn thành luận án. Em là người bạn đồng hành, người bạn tri kỷ của tôi.

Cuối cùng, tôi xin kết thúc dòng tâm sự bằng lời cảm ơn sâu sắc nhất đến đại gia đình, anh, chị, em nội ngoại và bạn bè của tôi. Họ đã luôn ủng hộ, khuyến khích và giúp đỡ tôi trong suốt quá trình học tập, nghiên cứu và hoàn thành luận án. Thành tựu này sẽ không thể đạt được nếu không có họ.

Hà Nội, ngày tháng năm 2024

NGHIÊN CỨU SINH

Lê Hồng Lam

MỤC LỤC

LỜI CAM ĐOAN	I
LỜI CẢM ƠN	II
MỤC LỤC	IV
DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT	VII
DANH MỤC BẢNG BIỂU	IX
DANH MỤC HÌNH VẼ, ĐỒ THỊ	XI
MỞ ĐẦU	1
Lý do chọn đề tài	1
Câu hỏi nghiên cứu	3
Mục tiêu của đề tài	3
Đối tượng và phạm vi nghiên cứu	4
Ý nghĩa khoa học và thực tiễn của đề tài	4
Đóng góp của luận án	5
Phương pháp tiếp cận	6
Phương pháp nghiên cứu	6
Bố cục của luận án	7
CHƯƠNG 1. TỔNG QUAN	9
1.1. Một số khái niệm cơ bản	9
1.2. Cấu trúc chung của hệ thống nhận dạng hoạt động	10
1.2.1. Thu dữ thập liệu cho HAR.	11
1.2.2. Một số mô hình phân loại thông dụng trong HAR	14
1.2.3. Truyền tin cảnh báo	24

1.3.	Tình hình nghiên cứu trong và ngoài nước	25
1.3.1.	Một số nghiên cứu trên thế giới	25
1.3.2.	Một số nghiên cứu ở Việt Nam	32
1.4.	Giải pháp nghiên cứu	35
1.5.	Kết luận chương	37
CHƯƠNG 2. TRÍCH CHỌN ĐẶC TRƯNG		38
2.1.	Giới thiệu	38
2.2.	Quy trình trích rút đặc trưng từ dữ liệu cảm biến gia tốc	39
2.3.	Các bộ dữ liệu sử dụng trong nghiên cứu	41
2.3.1.	Bộ dữ liệu MobiAct v2.0	42
2.3.2.	Bộ dữ liệu UP-Fall	44
2.4.	Tiền xử lý dữ liệu	45
2.4.1.	Kỹ thuật lọc nhiễu	45
2.4.2.	Phân đoạn dữ liệu	47
2.5.	Trích xuất đặc trưng	50
2.5.1.	Trong miền thời gian	50
2.5.2.	Trong miền tần số	53
2.5.3.	Tham số định hướng	55
2.5.4.	Tham số Hjorth	56
2.6.	Lựa chọn đặc trưng	57
2.7.	Thực nghiệm và đánh giá	59
2.7.1.	Mô tả thực nghiệm	59
2.7.2.	Kết quả và bàn luận	61

2.7.3.	So sánh kết quả với các nghiên cứu liên quan	73
2.8.	Kết luận chương	78
CHƯƠNG 3.	XÂY DỰNG MÔ HÌNH PHÁT HIỆN SỰ KIỆN NGÃ	80
3.1.	Giới thiệu bài toán	80
3.2.	Tối ưu mô hình mô hình phát hiện ngã	81
3.2.1.	Giới thiệu tổng quan	81
3.2.2.	Giải thuật di truyền (GA)	83
3.2.3.	Đề xuất thuật toán GA4RF	84
3.2.4.	Thực nghiệm và đánh giá	98
3.3.	Xây dựng mô hình phát hiện sự kiện ngã (FEDM)	111
3.3.1.	Đánh giá mức độ ảnh hưởng của cửa sổ dữ liệu	112
3.3.2.	Mô hình phát hiện các sự kiện ngã (FEDM).	114
3.3.3.	Thực nghiệm và đánh giá kết quả	119
3.4.	Kết luận chương	124
KẾT LUẬN VÀ KIẾN NGHỊ		126
DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN		128
PHỤ LỤC: XÂY DỰNG ỨNG DỤNG DEMO		129
TÀI LIỆU THAM KHẢO		137

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt	Diễn giải	
	Tiếng Anh	Tiếng Việt
ABC	Artificial bee colony	Giải thuật bầy ong nhân tạo
FEDM	Fall event detection model	Mô hình phát hiện sự kiện ngã
ACC	Accuracy	Độ chính xác
ADL	Activities of daily living	Các hoạt động thường ngày
FDS	Fall detection system	Hệ thống phát hiện ngã
AI	Artificial intelligence	Trí tuệ nhân tạo
AL	Active loss	Tổn thất chủ động
ANN	Artificial neural network	Mạng nơ-ron nhân tạo
ARM	Activity recognition module	Mô-đun nhận dạng hoạt động
BLE	Bluetooth low energy	Bluetooth năng lượng thấp
CART	Classification and regression tree	Cây phân loại và hồi quy
CNN	Convolutional neural network	Mạng nơ-ron tích chập
DALYs	Disability-adjusted life years	Số năm sống được điều chỉnh theo mức độ tàn tật
DL	Deep learning	Học sâu
DT	Decision tree	Cây quyết định
DOR	Data overlapping ratio	Tỉ lệ xếp chồng dữ liệu
DWS	Data window size	Kích thước cửa sổ dữ liệu
ECG	Electrocardiogram	Điện tâm đồ
FDS	Fall detection system	Hệ thống phát hiện ngã
FFT	Fast Fourier transform	Biến đổi Fourier nhanh
FN	False negative	Phân loại sai thành âm tính
FP	False positive	Phân loại sai thành dương tính
GA	Genetic algorithm	Giải thuật di truyền
GPS	Global positioning system	Hệ thống định vị toàn cầu
GRU	Gated recurrent unit	Đơn vị cổng hồi tiếp
GS	Grid search	Tìm kiếm theo lưới
GWO	Grey wolf optimizer	Thuật toán sói màu xám
HAR	Human activity recognition	Nhận dạng hoạt động của con người
HMM	Hidden markov model	Mô hình Markov ẩn
ICA	Independent Component Analysis	Phân tích thành phần độc lập

Từ viết tắt	Diễn giải	
	Tiếng Anh	Tiếng Việt
IMU	Inertial measurement unit	Đơn vị đo lường quán tính
IoT	Internet of thing	Internet vạn vật
kNN	K-nearest neighbor	k láng giềng gần nhất
LR	Logistic regression	Hồi quy logistic
LSTM	Long short-term memory	Bộ nhớ dài ngắn hạn
LWS	Long window size	Kích thước cửa sổ dài
MCC	Matthew correlation coefficient	Hệ số tương quan Matthew
MH	Meta-heuristic/Metaheuristic	Siêu kinh nghiệm
ML	Machine learning	Học máy/máy học
MPA	Marine predators algorithm	Thuật toán động vật săn mồi biển
NCS	PhD student	Nghiên cứu sinh
nOSW	Non-overlapping sliding window	Cửa sổ trượt không chồng chéo
OSW	Overlapping sliding window	Cửa sổ trượt chồng chéo
PCA	Principal components analysis	Phép phân tích thành phần chính
PIR	Passive infrared	Hồng ngoại thụ động
PSO	Particle swarm optimization	Thuật toán tối ưu bầy đàn
RF	Random forest	Rừng ngẫu nhiên
RNN	Recurrent neural network	Mạng nơ-ron hồi quy
RS	Random search	Tìm kiếm ngẫu nhiên
SP	Smartphone	Điện thoại thông minh
SVM	Support vector machine	Máy vector hỗ trợ
SW	Smartwatch	Đồng hồ thông minh
sWS	Short window size	Kích thước cửa sổ ngắn
t-SNE	t-Distributed Stochastic Neighbor Embedding	giảm số chiều phi tuyến
TN	True negative	Phân loại đúng âm tính
TP	True positive	Phân loại đúng dương tính
WD	Wearable device	Các thiết bị đeo được
WHO	World health organization	Tổ chức Y tế Thế giới
WS	Window size	Kích thước cửa sổ

DANH MỤC BẢNG BIỂU

Bảng 1.1. Ma trận nhầm lẫn trong phân lớp nhị phân-----	21
Bảng 1.2. Ma trận nhầm lẫn trong phân loại đa lớp -----	21
Bảng 1.3. Các nghiên cứu gần đây về phát hiện ngã sử dụng bộ dữ liệu MobiAct và UP-Fall. -----	26
Bảng 2.1. Các bộ dữ liệu công khai được sử dụng trong ADL và phát hiện ngã. --	41
Bảng 2.2. Thông số cảm biến gia tốc và con quay hồi chuyển trên điện thoại thông minh Samsung Galaxy S3 -----	42
Bảng 2.3. Hoạt động thường ngày và ngã trong bộ dữ liệu MobiAct [92]. -----	43
Bảng 2.4. Hoạt động thường ngày và ngã trong bộ dữ liệu UP-Fall [12]. -----	45
Bảng 2.5. Các đặc trưng cơ bản trong miền thời gian. -----	51
Bảng 2.6. Các đặc trưng cơ bản trong miền tần số. -----	53
Bảng 2.7. Các đặc trưng cơ bản trong miền tham số Hjorth. -----	57
Bảng 2.8. Tập các đặc trưng được lựa chọn để trích chọn thông tin từ dữ liệu cảm biến gia tốc cho hệ thống phát hiện sự kiện ngã (FDS) -----	58
Bảng 2.9. Các bộ đặc trưng được sử dụng trong Thực nghiệm 2.1 -----	60
Bảng 2.10. Bộ tham số mặc định của các mô hình phân lớp. -----	61
Bảng 2.11. Hiệu suất phân loại từng hoạt động của năm mô hình ML khi sử dụng bộ đặc trưng TFDHD trích rút từ các bộ dữ liệu MobiAct v2.0 -----	67
Bảng 2.12. Hiệu suất phân loại từng hoạt động của năm mô hình ML khi sử dụng bộ đặc trưng TFDHD trích rút từ các bộ dữ liệu UP-Fall. -----	68
Bảng 2.13. Ảnh hưởng của DWS đến hiệu suất nhận dạng các hoạt động trong bộ dữ liệu MobiAct v2.0 -----	70

Bảng 2.14. Ảnh hưởng của DWS đến hiệu suất nhận dạng các hoạt động trong bộ dữ liệu UP-Fall-----	71
Bảng 3.1. Giá trị các tham số cài đặt cho GA-----	99
Bảng 3.2. Tổng hợp kết quả thực nghiệm trên bộ dữ liệu MobiAct v2.0 của 50 thể hệ cuối cùng trước khi thuật toán GA4RF đạt điều kiện dừng. -----	101
Bảng 3.3. Tổng hợp kết quả thực nghiệm trên bộ dữ liệu UP-Fall của 51 thể hệ cuối cùng trước khi thuật toán GA4RF đạt điều kiện dừng. -----	104
Bảng 3.4. Tỷ lệ đồng nhất (rk) khi dự đoán hoạt động ở các sWS so với LWS; “1” mô hình đoán đúng là hoạt động bình thường, “0” mô hình đoán là sự kiện ngã. -----	118
Bảng 3.5. Các mẫu hoạt động bình thường trong bộ dữ liệu MobiAct v2.0 được sử dụng để huấn luyện mô hình. -----	119
Bảng 3.6. Kết quả huấn luyện các mô hình ở các WS khác nhau dựa trên chín nhãn ADL thường gặp trong bộ dữ liệu MobiAct 2.0-----	120
Bảng 3.7. Kết quả thử nghiệm FEDM với bộ dữ liệu hoạt động bình thường. ----	122
Bảng 3.8. Kết quả thử nghiệm FEDM với bộ dữ liệu các sự kiện ngã.-----	123
Bảng 3.9. Kết quả thực nghiệm hệ thống nhận dạng các mẫu chuyển động bình thường và ngã -----	124

DANH MỤC HÌNH VẼ, ĐỒ THỊ

Hình 1.1. Cấu trúc chung của hệ thống HAR [18–21].-----	11
Hình 1.2. Các loại cảm biến thường dùng trong HAR.-----	11
Hình 1.3. Mẫu dữ liệu ngã thu bằng cảm biến gia tốc của điện thoại thông minh. -	14
Hình 1.4. Phân phối các mô hình ML và DL được sử dụng trong 305 nghiên cứu về FDS (từ 2015 đến 2020)[7]-----	15
Hình 1.5. Hệ thống nhận dạng hoạt động áp dụng kỹ thuật học sâu [22]. -----	16
Hình 1.6. Hệ thống nhận dạng hoạt động áp dụng kỹ thuật học máy [22]. -----	17
Hình 1.7. Khung kiến trúc của mô hình phát hiện ngã của người dùng dựa trên SP và kỹ thuật ML.-----	36
Hình 2.1. Quy trình trích rút đặc trưng.-----	40
Hình 2.2. Phân phối dữ liệu trên hai bộ dữ liệu: (a) MobiAct v2.0; (b) UP-Fal [12, 92] -----	42
Hình 2.3. Phân đoạn dữ liệu bằng kỹ thuật cửa sổ không xếp chồng dữ liệu (nOSW) -----	48
Hình 2.4. Phân đoạn dữ liệu bằng kỹ thuật cửa sổ xếp chồng dữ liệu (OSW) -----	48
Hình 2.5. Các DWS phổ biến trong HAR [7]-----	49
Hình 2.6. Hệ trục tọa độ biểu diễn các trục quay của SP trong không gian -----	55
Hình 2.7. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại tất cả các hoạt động trong bộ dữ liệu MobiAct (DWS 128 mẫu và DOR 80%). -----	62
Hình 2.8. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại tất cả các hoạt động trong bộ dữ liệu UP-Fall (DWS 128 mẫu và DOR 80%).	62

Hình 2.9. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại các hoạt động bình thường trong bộ dữ liệu MobiAct (DWS 128 mẫu và DOR 80%). -----	64
Hình 2.10. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại các hoạt động bình thường trong bộ dữ liệu UP-Fall (DWS 128 mẫu và DOR 80%). -----	64
Hình 2.11. Hiệu suất các mô hình phân loại khi sử dụng 10 bộ đặc trưng để phân loại các sự kiện ngã trong bộ dữ liệu MobiAct (DWS 128 mẫu và DOR 80%). -----	65
Hình 2.12. Hiệu suất các mô hình phân loại khi sử dụng 10 bộ đặc trưng để phân loại các sự kiện ngã trong bộ dữ liệu UP-Fall (DWS 128 mẫu và DOR 80%). -----	66
Hình 2.13. Ảnh hưởng của DOR đến hiệu suất phân loại các hoạt động trong bộ dữ liệu MobiAct v2.0.-----	72
Hình 2.14. Ảnh hưởng của DOR đến hiệu suất nhận dạng của mô hình. -----	73
Hình 2.15. Hiệu suất chi tiết của Chatzaki và cộng sự [13] và mô hình đề xuất của NCS trong việc phát hiện ADL và ngã. -----	74
Hình 2.16. Ma trận nhầm lẫn của 16 hoạt động trong bộ MobiAct v2.0. (a) Chatzaki và cộng sự [13], (b) Phương pháp NCS đề xuất.. -----	75
Hình 2.17. Hiệu suất chi tiết của Lai và cộng sự [9] và bộ đặc trưng TFDHD đề xuất trong việc phát hiện ADL và các sự kiện ngã. -----	77
Hình 2.18. Ma trận nhầm lẫn của 11 hoạt động trong bộ dữ liệu UP-Fall. Hàng là các lớp dự đoán; cột là các lớp thực tế. Ma trận con 5×5 (FH, FF, FB, FS và FE) là các lớp ngã. (a) Kết quả của Lai và cộng sự [9] sử dụng mô hình Res-TCN. (b) Mô hình RF dự đoán dựa trên bộ đặc trưng TFDHD đề xuất. -----	78

Hình 3.1. Thuật toán GA4RF -----	84
Hình 3.2. Quá trình lai ghép trong GA.-----	90
Hình 3.3. Ảnh hưởng của hàm logarit trong việc tăng cường khả năng phân biệt các giá trị fitness gần nhau (với $k = 0.1$). -----	98
Hình 3.4. Giá trị hàm thích nghi theo <i>NMCC</i> trong 50 thế hệ cuối cùng trước khi thỏa mãn điều kiện dừng tại số lần lặp tối đa trên bộ dữ liệu MobiAct v2.0. -----	100
Hình 3.5. Giá trị hàm thích nghi theo <i>NMCC</i> trong 50 thế hệ cuối cùng trước khi thỏa mãn điều kiện dừng tại số lần lặp tối đa trên bộ dữ liệu UP-Fall.-----	103
Hình 3.6. So sánh hiệu suất của thuật toán GA4RF với thuật toán của nhóm tác giả Azmat [20], Chatzaki [13] và Le [CT1] thực hiện trên bộ dữ liệu MobiAct v2.0. -----	108
Hình 3.7. So sánh hiệu suất của thuật toán GA4RF với thuật toán của nhóm tác giả OP-Tanish [119], Res-TCN [120] và Le [CT1] thực hiện trên bộ dữ liệu UP-Fall. -----	109
Hình 3.8. Biểu diễn một sự kiện ngã điển hình của người đi bộ được thu thập bằng gia tốc kế ba trục của SP. -----	113
Hình 3.9. Ý tưởng thiết kế mô hình phát hiện các sự kiện ngã (FEDM). Trong hình vẽ, LWS là kích thước cửa sổ dài, sWS là kích thước cửa sổ ngắn. ----	114
Hình 3.10. Mô hình tổng quan phát hiện các sự kiện ngã-----	115
Hình 3.11. Quy trình thực hiện nhận dạng và phát hiện sự kiện ngã -----	118

MỞ ĐẦU

Lý do chọn đề tài

Già hóa dân số và các bệnh lý liên quan đến tuổi tác như tim mạch, cao huyết áp đang là vấn đề toàn cầu. Những bệnh lý này có thể dẫn đến đột quỵ hoặc ngã, đặc biệt ở người cao tuổi. Ngã là nguyên nhân hàng đầu gây chấn thương và tử vong ở nhóm này, ảnh hưởng nghiêm trọng đến sức khỏe và chất lượng cuộc sống của họ.

Theo Tổ chức Y tế Thế giới (WHO), nguy cơ ngã và chấn thương nghiêm trọng do ngã gia tăng theo độ tuổi [1]. Mỗi năm, cứ 3 người trên 65 tuổi thì có 1 người và cứ 2 người trên 80 tuổi thì có 1 người bị ngã ít nhất một lần [2]. Gần 40% trong số này bị ngã tái phát [3]. Đáng lo ngại hơn, gần 50% người bị ngã không thể tự đứng dậy [4], dẫn đến các biến chứng như hạ thân nhiệt, mất nước, loét tì đè và viêm phổi, làm tăng nguy cơ nhập viện [5].

Trên thế giới, ước tính mỗi năm có 684.000 người tử vong do ngã, chủ yếu ở các nước thu nhập thấp và trung bình, tập trung tại khu vực Tây Thái Bình Dương và Đông Nam Á. Ngã cũng là nguyên nhân dẫn đến hơn 38 triệu DALYs (số năm sống được điều chỉnh theo mức độ tàn tật) bị mất mỗi năm, cao hơn so với tổng số năm sống chung với các di chứng do tai nạn giao thông, chết đuối, bỏng và ngộ độc [6].

Tại Việt Nam, tình trạng ngã ở người cao tuổi là một vấn đề đáng báo động, cần được quan tâm và có biện pháp phòng ngừa hiệu quả để giảm thiểu những hậu quả nghiêm trọng về sức khỏe và chất lượng cuộc sống..

Hậu quả do ngã gây ra có thể được giảm đáng kể nếu được phát hiện sớm. Nhận thức được tầm quan trọng của việc này, nghiên cứu sinh (NCS) đã lựa chọn đề tài “*Nghiên cứu phát triển một số kỹ thuật trích rút đặc trưng và tối ưu mô hình Random Forest trong phát hiện sự kiện ngã của con người bằng điện thoại thông minh*” nhằm góp sức cùng cộng đồng trong công tác phòng chống và giảm thiểu hậu quả của ngã, đặc biệt là đối với nhóm người yếu thể và người cao tuổi.

Điện thoại thông minh (SP) ngày nay đóng vai trò thiết yếu trong cuộc sống hàng ngày. Chúng được sử dụng để quản lý thời gian, công việc, theo dõi sức khỏe và kết nối mọi người. Theo thống kê, SP là vật dụng đầu tiên chúng ta cầm khi thức dậy vào buổi sáng và là thứ cuối cùng được sử dụng trước khi đi ngủ [7]. Nhờ được trang bị cảm biến quán tính, điện thoại thông minh có khả năng nhận biết các hoạt động thường ngày của người dùng. Đây là cơ sở để phát triển các ứng dụng hỗ trợ chăm sóc sức khỏe, theo dõi hoạt động thể chất, phát hiện nguy cơ ngã và đưa ra cảnh báo kịp thời.

Hệ thống nhận dạng hoạt động của con người thường được xây dựng dựa trên phương pháp học sâu và học máy, mỗi phương pháp có ưu - nhược điểm riêng. Các mô hình học sâu có khả năng tự động trích xuất và học các đặc trưng phức tạp từ dữ liệu đầu vào, tự học và điều chỉnh tham số theo dữ liệu, giúp cải thiện khả năng phát hiện và phân loại các sự kiện ngã. Tuy nhiên, mô hình học sâu đòi hỏi lượng dữ liệu lớn để huấn luyện, tiêu tốn nhiều tài nguyên tính toán, công suất và thời gian huấn luyện. Điều này là một thách thức lớn khi dữ liệu ngã thực tế rất khó thu thập. Các thuật toán học sâu thường không phù hợp để phát triển mô hình phát hiện ngã trên các thiết bị có tài nguyên tính toán và nguồn điện hạn chế như điện thoại thông minh [8–11].

Ngược lại, các mô hình học máy thường có hiệu suất phân loại tốt trên các bộ dữ liệu nhỏ, dễ tích hợp và triển khai trên các thiết bị có sẵn với tài nguyên hạn chế như điện thoại thông minh. Tuy nhiên, hiệu suất phân loại và khả năng nhận biết sự kiện ngã của các mô hình học máy phụ thuộc nhiều vào thiết kế đặc trưng thủ công, đòi hỏi sự hiểu biết sâu sắc về dữ liệu [12–15].

Luận án này tập trung vào việc phát triển một mô hình phát hiện sự kiện ngã dựa trên điện thoại thông minh. Đây là một bài toán phức tạp, đòi hỏi giải quyết nhiều vấn đề như tối ưu năng lượng, dữ liệu khan hiếm, tài nguyên hạn chế và nhiều câu hỏi nghiên cứu khác cần tìm lời giải.

Câu hỏi nghiên cứu

Câu hỏi nghiên cứu thứ nhất: Làm thế nào để cải tiến và phát triển các kỹ thuật trích rút đặc trưng trung mạnh mẽ và hiệu quả từ dữ liệu thô thu thập bằng cảm biến quán tính của điện thoại thông minh để cải thiện khả năng phân loại các sự kiện ngã?

Câu hỏi nghiên cứu thứ một: Những yếu tố nào ảnh hưởng đến độ chính xác và hiệu suất của mô hình Random Forest trong việc phát hiện sự kiện ngã, và làm thế nào để tối ưu hóa các yếu tố này nhằm nâng cao hiệu quả của hệ thống?

Câu hỏi nghiên cứu thứ hai: Làm cách nào để phát triển một mô hình phát hiện sự kiện ngã có khả năng thích ứng với các điều kiện thực tế, đặc biệt khi dữ liệu về sự kiện ngã khó thu thập và gán nhãn trong thế giới thực?

Mục tiêu của đề tài

Luận án này hướng đến việc xây dựng một hệ thống phát hiện ngã thực tiễn và dễ sử dụng, tận dụng các công nghệ có sẵn trên điện thoại thông minh và giải quyết các thách thức liên quan đến dữ liệu thực tế. Để đạt được mục đích này, luận án tập trung vào ba mục tiêu cụ thể như sau:

Mục tiêu thứ nhất: Phát triển và cải tiến các kỹ thuật trích rút đặc trưng mạnh mẽ và hiệu quả từ dữ liệu cảm biến quán tính trên điện thoại thông minh để cải thiện khả năng phân loại các sự kiện ngã.

Mục tiêu thứ hai: Xác định các yếu tố ảnh hưởng đến độ chính xác và hiệu suất của mô hình Random Forest trong việc phát hiện sự kiện ngã, và tối ưu hóa các yếu tố này để nâng cao hiệu quả của hệ thống.

Mục tiêu thứ ba: Phát triển một mô hình phát hiện sự kiện ngã có khả năng thích ứng với điều kiện thực tế, đặc biệt khi dữ liệu về sự kiện ngã khó thu thập và gán nhãn trong thế giới thực.

Đối tượng và phạm vi nghiên cứu

Đối tượng chính của nghiên cứu là các sự kiện ngã của con người, được phát hiện thông qua dữ liệu thu thập từ các cảm biến trên điện thoại thông minh, như gia tốc kế và con quay hồi chuyển. Nghiên cứu tập trung vào việc đánh giá và phát triển các phương pháp trích xuất thông tin hữu ích từ dữ liệu thô thu thập từ các cảm biến này nhằm làm nổi bật các dấu hiệu nhận biết sự kiện ngã. Bên cạnh đó, nghiên cứu cũng tập trung vào phân tích và tối ưu hóa mô hình máy học Random Forest để nâng cao độ chính xác và hiệu suất của hệ thống phát hiện ngã dựa trên các đặc trưng đã được trích xuất. Ngoài ra, phạm vi nghiên cứu giới hạn trong việc sử dụng điện thoại thông minh làm thiết bị thu thập dữ liệu, tận dụng các cảm biến có sẵn và khả năng kết nối, cài đặt phần mềm trên thiết bị để thực hiện quá trình phát hiện và báo hiệu ngã.

Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài tập trung vào việc phát triển và cải tiến các kỹ thuật trích rút đặc trưng mạnh mẽ từ dữ liệu cảm biến quán tính trên điện thoại thông minh. Điều này đóng góp phần làm phong phú thêm tài liệu nghiên cứu, mở rộng kiến thức và phương pháp tiếp cận trong lĩnh vực xử lý tín hiệu và học máy, đặc biệt là trong bài toán phân loại sự kiện phức tạp như phát hiện ngã.

Nghiên cứu phân tích và đánh giá các yếu tố ảnh hưởng đến hiệu suất của mô hình Random Forest trong bối cảnh phát hiện ngã sẽ cung cấp hiểu biết sâu sắc về cách áp dụng và điều chỉnh mô hình này cho các bài toán phân loại tương tự. Điều này không chỉ có giá trị cho nghiên cứu hiện tại mà còn có thể áp dụng cho các lĩnh vực khác trong học máy.

Việc phát triển mô hình có khả năng thích ứng với dữ liệu ngã khó thu thập và gán nhãn trong thế giới thực đóng góp vào nghiên cứu về học máy trong điều kiện dữ liệu hạn chế. Điều này mở ra hướng đi mới trong việc xử lý dữ liệu không cân bằng và thiếu nhãn.

Việc sử dụng điện thoại thông minh làm thiết bị thu thập dữ liệu và phát hiện ngã giúp hệ thống trở nên dễ tiếp cận và tiết kiệm chi phí, đồng thời tận dụng các công nghệ có sẵn mà người dùng đã sở hữu, từ đó khuyến khích việc áp dụng rộng rãi trong cộng đồng.

Sản phẩm của đề tài có thể phát triển thành phần mềm miễn phí cài đặt trên điện thoại hoặc đồng hồ thông minh, giúp người dùng tự theo dõi hoạt động và phòng ngừa các rủi ro ngã. Hệ thống phát hiện ngã không chỉ hỗ trợ cá nhân mà còn có thể tích hợp vào các dịch vụ y tế cộng đồng, góp phần nâng cao chất lượng chăm sóc sức khỏe và an toàn cho xã hội.

Đóng góp của luận án

Đóng góp thứ nhất: Đề xuất một phương pháp trích xuất đặc trưng mới từ dữ liệu gia tốc 3 trục bằng cách kết hợp các đặc trưng từ miền thời gian, miền tần số, cùng với các tham số định hướng và tham số Hjorth. Việc kết hợp đa dạng các loại đặc trưng này giúp tăng cường khả năng phân tích và nhận dạng các sự kiện ngã một cách hiệu quả. Kết quả từ đóng góp này đã được công bố trong các công trình [CT1], [CT2], [CT3].

Đóng góp thứ hai: Đề xuất thuật toán GA4RF bằng cách sử dụng giải thuật di truyền để tìm bộ siêu tham số tối ưu cho Random Forest. GA4RF thể hiện khả năng tìm kiếm bộ siêu tham số tối ưu nhờ cấu trúc nhiễm sắc thể, không gian tìm kiếm và hàm thích nghi được thiết kế theo hướng mới phù hợp với đặc thù của các bộ dữ liệu ngã. Bên cạnh đó, luận án còn phát triển mô hình phát hiện sự kiện ngã (FEDM) dựa trên dữ liệu các hoạt động bình thường hàng ngày. FEDM kết hợp ưu điểm của thuật toán GA4RF đã tối ưu, cùng với kỹ thuật so khớp dữ liệu linh hoạt và kỹ thuật ngưỡng. Điều này cho phép FEDM phân biệt hiệu quả giữa các hoạt động bình thường (dữ liệu huấn luyện) và các sự kiện ngã (dữ liệu mới), mang lại khả năng phát hiện ngã chính xác ngay cả khi dữ liệu về sự kiện ngã khan hiếm. Kết quả từ các đóng góp này đã được công bố trong các công trình [CT2], [CT3], [CT4], và [CT5].

Phương pháp tiếp cận

Nghiên cứu sinh tiếp cận đề tài này theo một quy trình có hệ thống, bắt đầu từ việc tổng hợp và phân tích các nghiên cứu hiện có về nhận dạng hoạt động và phát hiện ngã. Quá trình này giúp đánh giá được ưu nhược điểm của các phương pháp hiện tại, từ đó xây dựng nền tảng cho việc đề xuất giải pháp mới.

Dựa trên những đánh giá này, NCS sẽ tập trung khám phá và cải tiến các kỹ thuật tiền xử lý dữ liệu, trích rút và lựa chọn đặc trưng, nhằm chuẩn bị dữ liệu chất lượng cao cho quá trình huấn luyện mô hình.

Việc lựa chọn mô hình phân loại phù hợp cũng đóng vai trò quan trọng. NCS sẽ thử nghiệm với nhiều mô hình khác nhau như Random Forest, J48, SVM, kNN và ANN để so sánh và chọn ra mô hình hiệu quả nhất cho bài toán phát hiện ngã. Sau khi đã xác định được mô hình phù hợp, luận án sẽ tập trung vào việc tìm kiếm giải pháp tối ưu hóa để nâng cao hiệu suất của mô hình này.

Cuối cùng, để đảm bảo tính thực tiễn của hệ thống, nghiên cứu sẽ tập trung vào việc phát triển mô hình có khả năng hoạt động tốt trên dữ liệu thực tế, bao gồm cả các trường hợp ngã khó thu thập.

Phương pháp nghiên cứu

Nghiên cứu sẽ kết hợp giữa phương pháp lý thuyết và thực nghiệm. Về mặt lý thuyết, nghiên cứu sinh sẽ tiến hành nghiên cứu chuyên sâu về các khái niệm và nguyên lý liên quan đến xử lý tín hiệu, khai phá dữ liệu, học máy và trí tuệ nhân tạo, đặc biệt là các phương pháp trích xuất và lựa chọn đặc trưng, tối ưu các mô hình phân loại và phát triển mô hình phát hiện sự kiện ngã. Quá trình này đòi hỏi việc tham khảo và đánh giá kỹ lưỡng các nghiên cứu trong và ngoài nước, nhằm tìm ra giải pháp phù hợp nhất cho bài toán phát hiện ngã bằng điện thoại thông minh.

Về mặt thực nghiệm, nghiên cứu sẽ bắt đầu từ việc sử dụng dữ liệu thô thu thập từ cảm biến gia tốc của điện thoại thông minh và các thiết bị đeo. Dữ liệu này sẽ được

tiến hành phân đoạn và trích xuất đặc trưng theo các hướng đề xuất trong phân lý thuyết. Sau đó, NCS sẽ tiến hành huấn luyện các thuật toán phân loại và các mô hình đề xuất trên bộ dữ liệu đã được xử lý. Cuối cùng của quá trình nghiên cứu sẽ tập trung vào việc đánh giá và so sánh hiệu suất phân loại hoạt động và phát hiện ngã của các giải pháp đề xuất. Việc so sánh này sẽ được thực hiện với các nghiên cứu liên quan trên cùng một bộ dữ liệu, đảm bảo tính công bằng và khách quan trong đánh giá. Kết quả từ quá trình này sẽ giúp xác định hiệu quả của các kỹ thuật trích rút đặc trưng, phương pháp tối ưu và hiệu quả của mô hình đã đề xuất.

Bố cục của luận án

Ngoài phần MỞ ĐẦU trình bày về Lý do chọn đề tài, Câu hỏi nghiên cứu, Mục tiêu của đề tài, Đối tượng và phạm vi nghiên cứu, Ý nghĩa khoa học và thực tiễn của đề tài, Đóng góp của luận án, Phương pháp tiếp cận và Phương pháp nghiên cứu. Nội dung chính của luận án được phân thành ba chương như mô tả trong Hình MĐ.1.

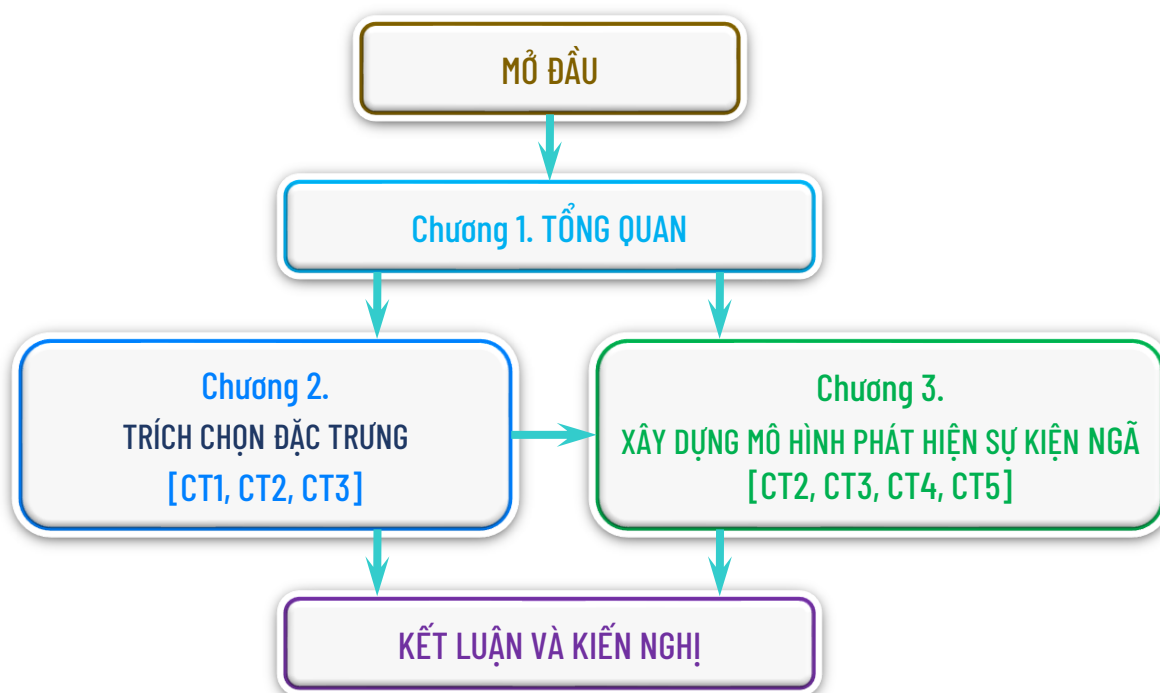
Chương 1 giới thiệu các khái niệm cơ bản về hành vi bình thường và bất thường của con người trong cuộc sống hàng ngày, xác định rõ các loại hành vi được xem là bình thường và bất thường để làm cơ sở cho các chương tiếp theo. Chương này cũng tổng hợp các nghiên cứu liên quan trong và ngoài nước để cung cấp cái nhìn tổng quan về lĩnh vực nghiên cứu. Ngoài ra, chương 1 còn trình bày tổng quan về kiến trúc chung của một hệ thống nhận dạng hoạt động. Mục tiêu chính là làm rõ lý do cho việc (i) sử dụng điện thoại thông minh (SP) để phát hiện sự kiện ngã, (ii) chỉ sử dụng cảm biến gia tốc trên SP để thu thập dữ liệu và trích xuất đặc trưng, và (iii) lựa chọn các mô hình học máy (ML) phù hợp cho hệ thống nhận dạng sự kiện ngã.

Chương 2 tập trung vào các phương pháp tiền xử lý, phân đoạn dữ liệu chuỗi thời gian và Quy trình trích rút đặc trưng. Chương này cung cấp giải pháp để trả lời Câu hỏi nghiên cứu thứ nhất với mục tiêu tìm ra: (i) tập đặc trưng phù hợp để nhận dạng các sự kiện ngã, (ii) kích thước cửa sổ dữ liệu (DWS) với sự kiện ngã, (iii) tỷ lệ xếp chồng dữ liệu (DOR) phù hợp, và (iv) thuật toán phân loại hiệu quả nhất cho phương

pháp đề xuất. Các phương pháp và kết quả trình bày trong chương này liên quan đến các công trình [CT1], [CT2], [CT3].

Chương 3 giới thiệu giải pháp tối ưu hóa mô hình Random Forest nhằm nâng cao khả năng phân loại các sự kiện ngã và phát triển mô hình phát hiện ngã phù hợp với dữ liệu khó thu thập như dữ liệu ngã. Mục tiêu của chương này là trả lời cho Câu hỏi nghiên cứu thứ hai và Câu hỏi nghiên cứu thứ ba, cung cấp các giải pháp nhằm cải thiện độ chính xác của hệ thống phát hiện ngã. Các nội dung và kết quả nghiên cứu liên quan đến các công trình [CT2], [CT3], [CT4], và [CT5].

Cuối cùng, phần Kết luận và kiến nghị tóm tắt những kết quả đạt được của luận án, đồng thời phân tích những hạn chế và các vấn đề chưa hoàn thiện. Từ đó, đề xuất các hướng nghiên cứu và giải pháp bổ sung để hoàn thiện nghiên cứu trong tương lai.



Hình MĐ. 1. Bố cục của luận án

Chương 1. TỔNG QUAN

Chương này giới thiệu các khái niệm cơ bản về hành vi bình thường và bất thường của con người trong cuộc sống hàng ngày, đồng thời thống nhất các khái niệm để làm cơ sở cho các chương tiếp theo của luận án. Chương cũng trình bày cấu trúc chung của hệ thống nhận dạng hoạt động của con người (HAR) và các nghiên cứu liên quan đến phát hiện ngã. Trọng tâm của chương là trình bày lý do sử dụng điện thoại thông minh và cảm biến gia tốc kế để phát triển hệ thống phát hiện ngã, cùng với việc lựa chọn các mô hình học máy (ML) phù hợp. Ngoài ra, chương này còn giới thiệu các phương pháp đánh giá hiệu năng của mô hình, thảo luận về tình hình nghiên cứu hiện tại, từ đó dẫn dắt đến vấn đề cần nghiên cứu và các đóng góp chính của luận án.

1.1. Một số khái niệm cơ bản

Đánh giá một hành động là bình thường hay bất thường không đơn giản vì nó phụ thuộc vào quan điểm của từng người đối với sự kiện hoặc hành động đó. Vì vậy, để phân biệt rõ hơn giữa hành vi bình thường và bất thường, một số khái niệm sau đây cần được thống nhất.

- (i) *Hành vi* là những phản ứng, cách cư xử, biểu hiện ra ngoài của một người trong một hoàn cảnh cụ thể [16];
- (ii) *Bình thường* là những sự việc, hành động không khác thường, không có gì đặc biệt [16];
- (iii) *Bất thường* là những sự việc, hành động không theo lệ thường, không giống bình thường [16];
- (iv) *Hành vi bình thường* là các hoạt động thường ngày của con người như ngồi, nằm, đứng, đi bộ, chạy, nhảy, đi lên cầu thang, đi xuống cầu thang, v.v.
- (v) *Hành vi bất thường* là các hoạt động không bình thường so với các hoạt động bình thường đã liệt kê, chẳng hạn như ngã.

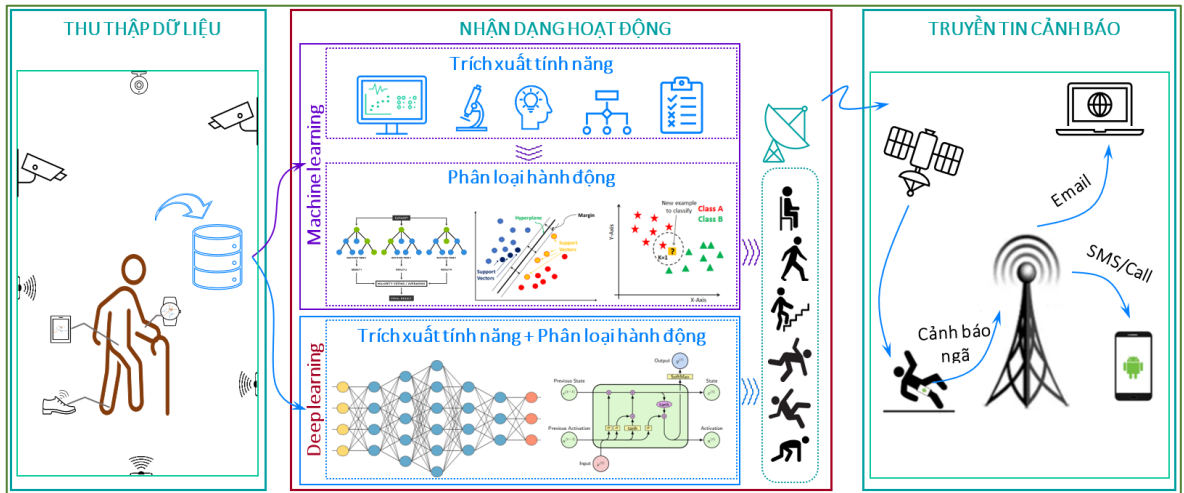
(vi) *Ngã* được định nghĩa là cơ thể của con người đột ngột di chuyển từ vị trí cao xuống vị trí thấp mà không có chủ ý hoặc do vô tình [6, 17]. Ngã là tình trạng không mong muốn và tiềm ẩn nhiều nguy cơ gây chấn thương cho người già và người bệnh. Ngã có thể gây ra những hậu quả nghiêm trọng, ảnh hưởng đến sức khỏe thể chất, thậm chí đe dọa tính mạng.

1.2. Cấu trúc chung của hệ thống nhận dạng hoạt động

Hệ thống phát hiện ngã (FDS) là một phần trong hệ thống nhận dạng hoạt động của con người (HAR). Phần này sẽ giới thiệu tổng quan về HAR và các thành phần cấu thành của nó. HAR thường bao gồm ba phần chính, được minh họa trong Hình 1.1:

- (1) Thu thập dữ liệu: Hệ thống này có nhiệm vụ ghi nhận các tín hiệu mô tả hoạt động của con người thông qua các cảm biến. Dữ liệu này sẽ hỗ trợ cho quá trình phát hiện hoạt động và các sự kiện ngã.
- (2) Mô hình nhận dạng hoạt động: Phần này thường sử dụng các thuật toán học máy (Machine Learning - ML) và học sâu (Deep Learning - DL) để phân tích dữ liệu đã thu thập. Mục đích là xác định trạng thái hoạt động của người dùng, bao gồm việc đánh giá nguy cơ ngã.
- (3) Hệ thống truyền thông tin cảnh báo: Đảm bảo rằng khi phát hiện nguy cơ, thông điệp cảnh báo sẽ được gửi ngay lập tức đến người sử dụng hoặc những người liên quan. Điều này cho phép họ có thể cung cấp sự trợ giúp kịp thời trong trường hợp xảy ra sự cố.

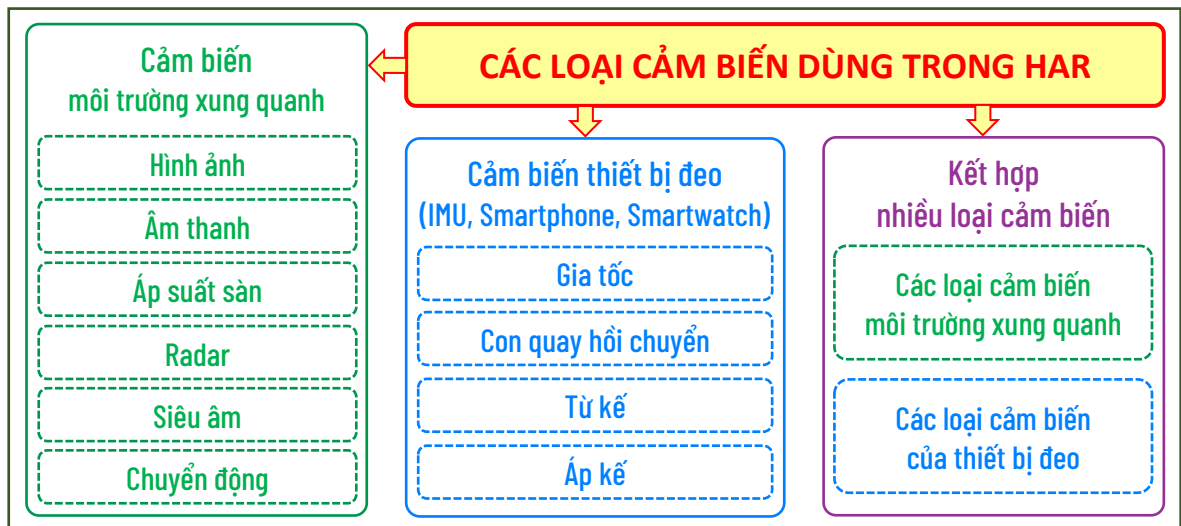
Bằng cách kết hợp ba thành phần này, hệ thống HAR có thể theo dõi, phân tích và phản ứng hiệu quả đối với các hoạt động của con người, đặc biệt là trong việc phát hiện và ngăn ngừa các sự kiện ngã.



Hình 1.1. Cấu trúc chung của hệ thống HAR [18–21].

1.2.1. Thu dữ thập liệu cho HAR.

Hiệu suất của HAR phụ thuộc nhiều vào phương thức sử dụng cảm biến để thu thập dữ liệu. Các hệ thống HAR thường được phân thành các nhóm dựa trên vị trí gắn cảm biến thu dữ liệu, điển hình là: (a) dựa trên các loại cảm biến môi trường xung quanh, (b) dựa trên các loại cảm biến trên thiết bị đeo được, và (c) công nghệ cảm biến lai [22, 23]. Các loại cảm biến thông dụng của mỗi nhóm được thống kê như Hình 1.2.



Hình 1.2. Các loại cảm biến thường dùng trong HAR.

1.2.1.1. Các loại cảm biến môi trường xung quanh

Hệ thống phát hiện ngã sử dụng nhiều loại cảm biến môi trường như cảm biến hình ảnh (camera), âm thanh, áp suất sàn, chuyển động PIR, radar, siêu âm để theo dõi hoạt động và phát hiện trường hợp ngã của con người [24]. Hệ thống này thường phù hợp để theo dõi các hoạt động trong không gian nhỏ như viện dưỡng lão, các căn phòng chăm sóc bệnh nhân và nhà riêng. Hướng tiếp cận này có một số hạn chế như chi phí lắp đặt và bảo trì cao, không gian giám sát bị giới hạn và chịu nhiều ảnh hưởng từ nhiễu môi trường [24]. Ngoài ra, hệ thống nghe, nhìn thường gây cho đối tượng được chăm sóc cảm thấy quyền riêng tư bị xâm phạm [23, 24].

Camera đòi hỏi băng thông và chi phí xử lý lớn, dễ bị ảnh hưởng bởi ánh sáng, bóng tối và các vật cản che khuất tầm nhìn, khiến việc phát hiện trường hợp ngã một cách chính xác gặp khó khăn. Cảm biến âm thanh dễ chịu tác động bởi tiếng ồn bên ngoài và có thể khó phân biệt tiếng ngã với các âm thanh khác, dễ bỏ lọt trường hợp ngã không gây ra tiếng động mạnh. Cảm biến áp suất sàn chỉ có thể phát hiện khi ngã xảy ra trên sàn và dễ bị kích hoạt bởi các hoạt động khác như đi lại. Cảm biến PIR lại nhạy cảm với chuyển động của vật nuôi hoặc đồ vật xung quanh, dễ bỏ lọt trường hợp ngã chậm. Radar tiêu tốn năng lượng lớn, giá thành cao, lắp đặt phức tạp và có nguy cơ phát xạ điện từ. Cảm biến siêu âm dễ bị ảnh hưởng bởi vật cản, góc nhìn hẹp.

1.2.1.2. Các loại cảm biến trên thiết bị đeo

Trong những năm gần đây, nhiều nghiên cứu tập trung vào thử nghiệm các giải pháp để xây dựng FDS từ tín hiệu quán tính. Điều này chủ yếu do hai yếu tố: (i) chi phí phần cứng ngày càng thấp và (ii) sự phổ biến rộng rãi của các thiết bị di động được trang bị cảm biến quán tính [7]. Với FDS dựa trên thiết bị có thể đeo được (bao gồm SP, SW và cảm biến trên Đơn vị đo lường quán tính (IMU)), dữ liệu về ADL (bao gồm sự kiện ngã) của đối tượng giám sát được thu thập thông qua cảm biến gia tốc, cảm biến con quay hồi chuyển và cảm biến định hướng. Phạm vi giám sát không giới hạn, tính cơ động cao, ít chịu ảnh hưởng của nhiễu từ môi trường và chi phí rẻ là ưu điểm lớn nhất của hệ thống này.

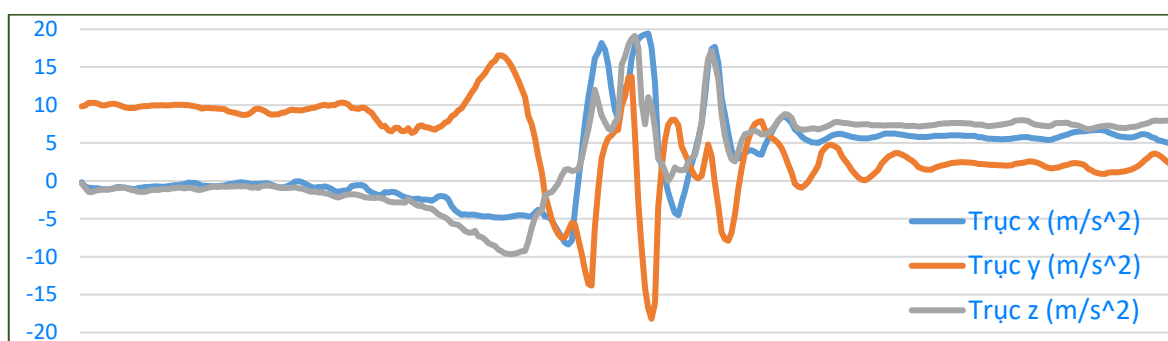
Chất lượng của dữ liệu thu thập từ thiết bị đeo (WD) chuyên dụng như IMU là rất tốt. Tuy nhiên, WD chuyên dụng thường không hoạt động độc lập, nó phải kết hợp với các hệ thống internet vạn vật (IoT), và/hoặc SP để xây dựng FDS và truyền bản tin cảnh báo. Bên cạnh đó, kiến trúc phần cứng của các hệ thống IoT có thể tích và trọng lượng lớn sẽ gây nên sự khó chịu và bất tiện cho người sử dụng [25]. Nguồn cấp cho hệ thống này cũng là một trong những thách thức mà các nhà nghiên cứu gặp khó khăn. Do các cảm biến đeo thường di động khi được sử dụng trong môi trường gia đình, nên việc sử dụng điện năng là một mối quan tâm đối với các thiết bị này. Trong các thiết bị đeo, việc trang bị pin lớn cho một thiết bị nhỏ có thể khiến thiết bị trở nên cồng kềnh hơn. Ngược lại, để giảm kích thước thiết bị sẽ cần phải sử dụng pin nhỏ hơn, lúc đó yêu cầu người dùng phải sạc lại thường xuyên [26].

1.2.1.3. Dữ liệu thu bằng cảm biến gia tốc

Gia tốc kế là một loại cảm biến dùng để ghi lại sự thay đổi vận tốc của một vật thể trong một khoảng thời gian. Loại cảm biến này được dùng phổ biến trong HAR, vì nó đo chuyển động có hướng của trạng thái chuyển động của đối tượng theo thời gian [27, 28]. Cảm biến gia tốc tích hợp trong điện thoại thông minh đo lường gia tốc của thiết bị theo ba trục (x, y, z). Những thành phần này cung cấp thông tin chi tiết sự thay đổi về tốc độ theo từng hướng, giúp theo dõi chính xác chuyển động của thiết bị. Giá trị gia tốc thường được đo bằng mét trên giây bình phương (m/s^2), thể hiện lực tác động lên thiết bị từ chuyển động hoặc tác động từ môi trường xung quanh. Ngoài ra, cảm biến gia tốc cũng cung cấp thông tin về hướng và góc nghiêng của thiết bị, bao gồm các góc xoay (φ -roll), góc nghiêng (θ -pitch) và góc đảo (ψ -yaw).

Độ phân giải và dải động của cảm biến quyết định chất lượng thu dữ liệu. Dải động là khoảng gia tốc mà cảm biến có thể phát hiện được, ảnh hưởng đến khả năng theo dõi chuyển động. Độ phân giải biểu thị độ nhạy của cảm biến đối với thay đổi nhỏ về gia tốc, đóng vai trò quan trọng trong việc nhận biết chi tiết nhỏ trong chuyển động.

Một yếu tố khác cũng ảnh hưởng đến chất lượng dữ liệu là tần số lấy mẫu. Tần số lấy mẫu cao cung cấp thông tin chi tiết hơn, nhưng cũng tiêu tốn nhiều năng lượng hơn. Tần số lấy mẫu 20Hz đủ để ghi lại các hoạt động đơn giản như đi bộ, đứng và ngồi. Tuy nhiên, để có được dữ liệu chi tiết về các chuyển động nhanh và thay đổi đột ngột, tần số lấy mẫu thường phải từ 100Hz trở lên. Một mẫu dữ liệu sử dụng gia tốc kế ba trục điển hình của điện thoại thông minh được biểu diễn như Hình 1.3.

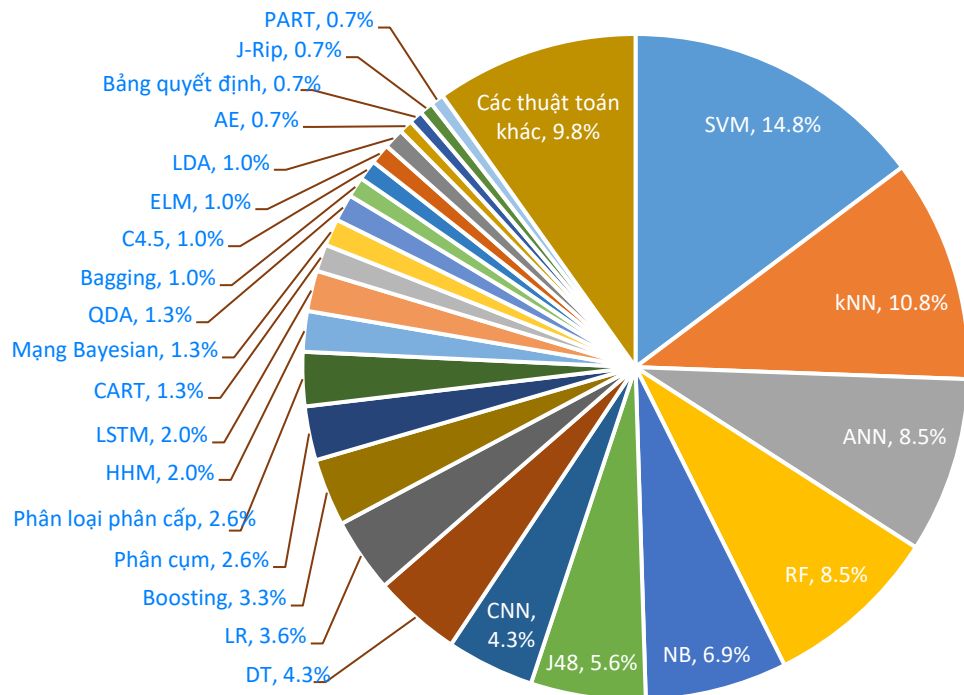


Hình 1.3. Mẫu dữ liệu ngã thu bằng cảm biến gia tốc của điện thoại thông minh.

1.2.2. Một số mô hình phân loại thông dụng trong HAR

Mỗi loại dữ liệu có một đặc điểm riêng và được mã hóa thành các nhãn. Hệ thống nhận dạng dựa vào các mẫu và nhãn của dữ liệu để xác định đó là loại dữ liệu bình thường hay ngã. Dữ liệu thu thập tốt, gán nhãn chính xác thì khả năng “học” và phát hiện hoạt động của mô hình sẽ chính xác. Nhân lực và chi phí phục vụ để thu thập, gán nhãn là rất lớn vì quá trình này thường được thực hiện thủ công bởi các chuyên gia trong cùng lĩnh vực. Thực tế, để thu thập và gán nhãn các dữ liệu ngã của người già và người bệnh là rất khó. Dựa vào điều kiện thực tế và sự phổ biến của dữ liệu, các phương pháp phát hiện ngã có thể rơi vào một trong 3 trường hợp [29]: Phân loại có giám sát, phân loại bán giám sát và phân loại không giám sát.

Mỗi bài toán phân loại sự kiện ngã có sự lựa chọn thuật toán và cách giải riêng. Ferrari và cộng sự [7] đã có những khảo sát và đánh giá mức độ phổ biến của các thuật toán ML và DL truyền thống được sử dụng để HAR được thống kê như Hình 1.4.



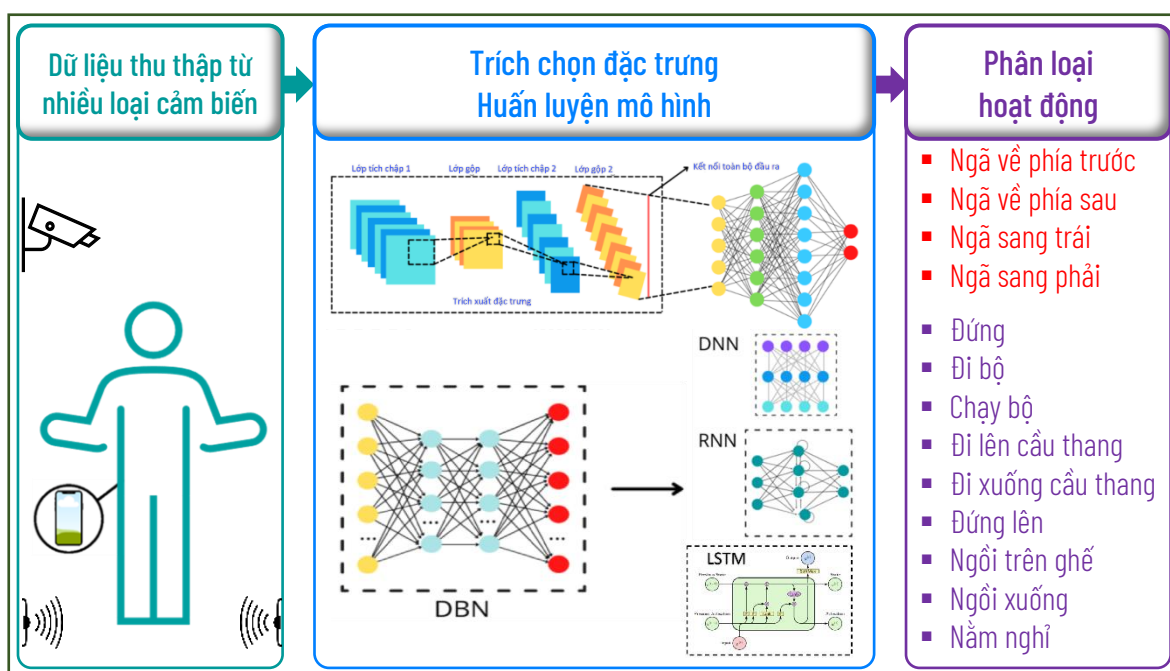
Hình 1.4. Phân phối các mô hình ML và DL được sử dụng trong 305 nghiên cứu về FDS (từ 2015 đến 2020)[7]

Gần đây, phương pháp siêu kinh nghiệm (MH) đã mở ra một hướng đi mới đầy hứa hẹn cho FDS. MH giúp cải thiện hiệu quả khả năng phát hiện ngã bằng cách tối ưu hóa việc trích xuất, lựa chọn đặc trưng từ dữ liệu và điều chỉnh tham số mô hình.

1.2.2.1. Các mô hình học sâu (DL)

Học sâu (DL) đã đạt được những tiến bộ vượt bậc trong những năm gần đây, mang lại hiệu quả cao trong nhiều lĩnh vực như nhận dạng đối tượng trực quan, xử lý ngôn ngữ tự nhiên và suy luận logic [30].

Cấu trúc cơ bản của DL bao gồm các tế bào thần kinh nhân tạo (ANN) được kết nối thành mạng lưới nhiều lớp. Mỗi tế bào thần kinh thực hiện tính toán dựa trên đầu vào và truyền kết quả đến các tế bào thần kinh khác. Các trọng số của mạng lưới được điều chỉnh đồng thời để tối ưu hóa kết quả cho dữ liệu đầu vào, tạo nên khả năng học hỏi và dự đoán của mô hình. Một hệ thống nhận dạng hoạt động và phát hiện ngã áp dụng kỹ thuật học sâu điển hình được mô tả như Hình 1.5.



Hình 1.5. Hệ thống nhận dạng hoạt động áp dụng kỹ thuật học sâu [22].

Mặc dù DL cho thấy tiềm năng to lớn trong việc nhận dạng các hoạt động phức tạp của con người, nhưng cũng tồn tại nhược điểm đáng chú ý. Đầu tiên, để đạt hiệu suất tốt, các mô hình học sâu thường yêu cầu lượng lớn dữ liệu đã được gán nhãn để huấn luyện mô hình một cách hiệu quả. Việc thu thập và gán nhãn dữ liệu ngã rất thể phức tạp, tốn kém và mất nhiều thời gian. Thứ hai, quá trình huấn luyện và triển khai các mô hình học sâu đòi hỏi phần cứng mạnh và tốn nhiều năng lượng, làm tăng chi phí vận hành. Điều này không phù hợp với các hệ thống có tài nguyên tính toán và nguồn điện hạn chế như thiết bị di động hoặc thiết bị nhúng/IoT.

Ngoài ra, các mô hình học sâu thường khó hiểu được lý do đằng sau dự đoán của chúng, khó xác định chính xác chúng đang học các đặc trưng gì để ra quyết định. Điều này có thể gây khó khăn trong việc xác định lỗi và cải thiện hiệu suất của mô hình. Bên cạnh đó, chúng dễ bị ảnh hưởng bởi nhiễu trong dữ liệu huấn luyện, dẫn đến giảm độ chính xác trong phát hiện.

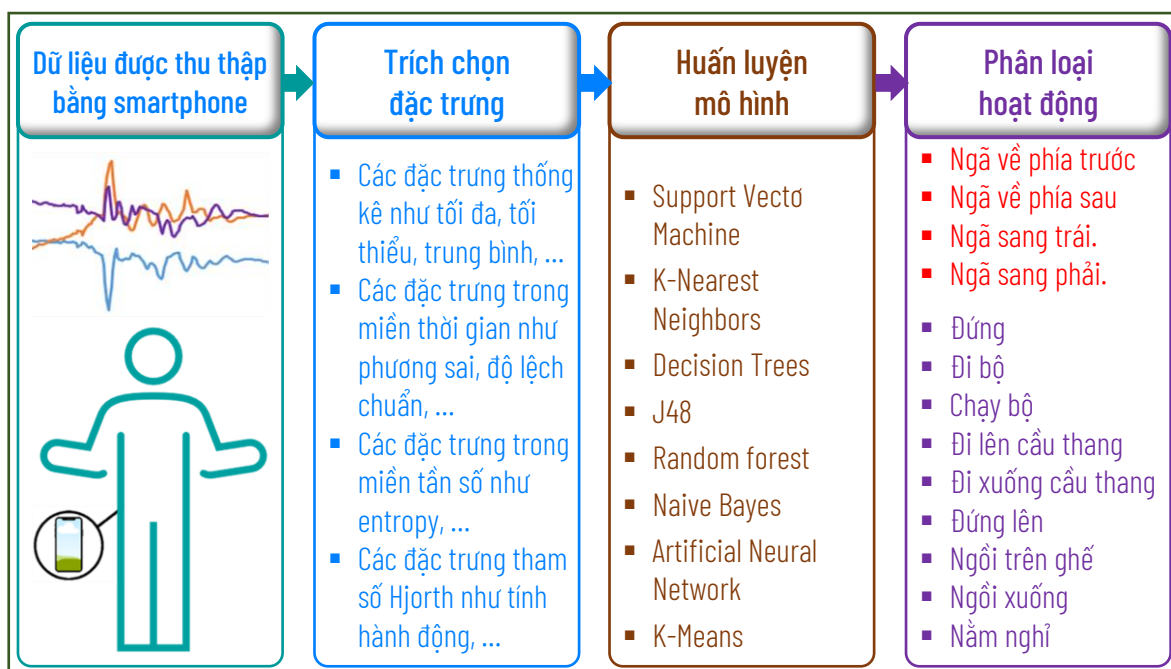
Bên cạnh đó, các mô hình học sâu giống như "hộp đen" làm cho con người khó hiểu lý do sau các dự đoán của chúng và khó xác định chính xác chúng đang học các đặc

trung gì để đưa ra quyết định. Điều này có thể tạo ra khó khăn trong việc xác định lỗi và cải thiện hiệu suất của mô hình. Ngoài ra, chúng dễ bị ảnh hưởng bởi nhiễu trong dữ liệu huấn luyện, dẫn đến giảm độ chính xác trong phát hiện ngã.

1.2.2.2. Các mô hình học máy (ML)

Dữ liệu HAR được thu thập từ nhiều loại cảm biến khác nhau, mỗi loại có đặc điểm và thông số kỹ thuật riêng. Điều này thể hiện mối liên kết phức tạp phi tuyến tương ứng với từng hoạt động khó có thể biểu diễn thông qua các quy tắc chung. Do đó, các kỹ thuật ML được ứng dụng để giải quyết vấn đề này. Hình 1.6 mô tả một FDS điển hình sử dụng kỹ thuật ML.

ML là một kỹ thuật học có giám sát, các thuật toán phân loại cơ bản được ứng dụng để học từ các dữ liệu đã được gán nhãn và sau đó thực hiện các dự đoán, các quyết định từ các thông tin đã học. Mô hình được đào tạo bởi các thuật toán học tập phù hợp hoạt động thông qua quy trình tối ưu hóa được xác định trước để giảm thiểu sai sót và nâng cao hiệu quả và độ chính xác của hệ thống tổng thể.



Hình 1.6. Hệ thống nhận dạng hoạt động áp dụng kỹ thuật học máy [22].

Năm thuật toán ML phổ biến nhất áp dụng cho FDS theo thống kê trong Hình 1.4, gồm: SVM, kNN, RF, ANN và J48 được lựa chọn để đánh giá phương pháp đề xuất.

Máy vectơ hỗ trợ (SVM) là một thuật toán ML được giám sát và nó được sử dụng nhiều nhất trong các nghiên cứu về HAR [7]. SVM là một bộ phân loại mạnh mẽ để tìm ra siêu phẳng phù hợp với việc tách các vectơ đặc trưng của các loại hành động khác nhau. Do đó, SVM xác định hàm phân loại tuyến tính tốt nhất có thể để phân biệt hiệu quả giữa các phần tử riêng biệt của hai lớp dữ liệu có thể phân tách tuyến tính trong tập huấn luyện. SVM được sử dụng chủ yếu trong phân loại nhị phân, vì nó chỉ cho phép một siêu mặt phẳng duy nhất trong mỗi bộ vấn đề. Để phân loại nhiều hoạt động, người ta sẽ thực hiện phương pháp mã hóa lỗi. Khi sử dụng mã hóa one-vs-one, mỗi lớp được phân loại tương đối với một lớp khác dẫn đến một kết quả với một phiếu tín nhiệm nhất định. Kết quả với tổng số phiếu bầu lớn nhất được coi là lớp dự đoán. Nói chung, khi muốn phân loại nhiều lớp bằng SVM cần phải sử dụng một tập các SVM.

K láng giềng gần nhất (kNN) là một trong những thuật toán học có giám sát đơn giản nhất trong ML và được sử dụng nhiều trong các nghiên cứu về HAR. Tập dữ liệu huấn luyện được xác định bởi n thuộc tính được lưu trữ trong một không gian mẫu của n chiều. Tập dữ liệu kiểm thử đã cho (không xác định) được so sánh với tập huấn luyện dựa trên sự giống nhau của nó, với k mẫu gần nhất làm cơ sở phân loại. Khi huấn luyện mô hình, kNN *không học* từ dữ liệu huấn luyện, mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. Thuật toán kNN tìm kiếm các phiên bản huấn luyện k bên trong không gian mẫu có sự tương đồng và gần nhất với một phiên bản thử nghiệm bất kỳ. Khoảng cách Euclidean là số liệu thường được sử dụng để xác định k hàng xóm gần nhất. KNN có thể hiển thị độ chính xác lên đến 98% và ít lỗi thống kê hơn trong việc phân loại ADL.

Rừng ngẫu nhiên (RF) là một phiên bản mở rộng của DT (cải tiến của phương pháp tổng hợp bootstrap (bagging)) có thể dự đoán các trường hợp trong tương lai với nhiều bộ phân loại thay vì một bộ phân loại duy nhất để đạt được độ chính xác tốt

nhất. RF là một thuật toán học có giám sát [31] nhằm xây dựng một số lượng lớn các DT không tương quan như một tập hợp tại thời điểm huấn luyện với đầu ra là phân loại các cây riêng lẻ [32]. Thuật toán này được phát triển bởi Leo Breiman, ông cũng chính là đồng tác giả của phương pháp CART [33] nổi tiếng. RF sử dụng 2 bước ngẫu nhiên, một là ngẫu nhiên theo mẫu (sample) dùng phương pháp bootstrap có hoàn lại (with replacement), hai là lấy ngẫu nhiên một lượng thuộc tính từ Tập đặc trưng ban đầu. Các tập dữ liệu con (sub-dataset) được tạo ra từ 2 lần ngẫu nhiên này có tính đa dạng cao, ít liên quan đến nhau, giúp giảm lỗi phương sai (variance). Các cây phân loại và hồi quy (CART) được xây dựng từ tập các tập dữ liệu con này tạo thành rừng. Mỗi cây riêng biệt có một dự đoán của lớp và lớp có nhiều phiếu bầu nhất sẽ trở thành dự đoán của mô hình. Không giống như các bộ phân loại DT khác, mỗi cây trong RF chỉ có thể chọn một tập hợp con ngẫu nhiên của các đặc trưng để có thể tăng sự thay đổi giữa các cây trong mô hình. Do đó, việc phân loại sẽ mang lại độ chính xác cao hơn khi xét đến sự tương quan thấp giữa các cây. Việc kết hợp các mô hình CART này để cho kết quả cuối cùng nên RF được gọi là phương pháp học tập thể. Mỗi DT trong RF không sử dụng tất cả dữ liệu huấn luyện, cũng như không sử dụng tất cả các thuộc tính của dữ liệu để xây dựng cây. Vì vậy mỗi cây có thể sẽ dự đoán không tốt. Nhưng, kết quả cuối cùng của mô hình RF lại tổng hợp từ nhiều DT bằng phương pháp bỏ phiếu (voting) cho bài toán phân loại và lấy giá trị trung bình (average) cho bài toán hồi quy. Thông tin từ các cây sẽ bổ sung cho nhau, dẫn đến mô hình có low bias và low variance, hay mô hình có kết quả dự đoán tốt.

Mạng nơ-ron nhân tạo (ANN - Artificial Neural Network) được lấy cảm hứng từ các chức năng của bộ não con người và được sử dụng cho cả phân loại và hồi quy trong ML. Chúng sao chép các chiến lược hoạt động của các tế bào thần kinh sinh học trong não, truyền tín hiệu kích hoạt dưới dạng các chức năng và mã hóa kiến thức bên trong các liên kết mạng. Kukreja và cộng sự [34] đề xuất các mô hình ANN bao gồm một số nút phi tuyến cho mục đích tính toán, song song, và được sắp xếp dưới dạng nhiều lớp (đầu vào, ẩn và đầu ra). Các nút trong ANN được kết nối thông qua các trọng số ngẫu nhiên và được cập nhật thông qua việc lan truyền ngược. ANN đủ

manh để phân biệt phi tuyến cho các vấn đề phân loại. Các đơn vị ẩn thực hiện ánh xạ phi tuyến thông qua một hàm cơ sở phi tuyến, các đầu vào và đầu ra của ANN là một sự kết hợp tuyến tính. Có thể có nhiều hơn một lớp ẩn với trọng số tương ứng của chúng và mỗi lớp có thể tính hàm phi tuyến để xử lý phức tạp hơn từ lớp đầu vào.

J48 là một thuật toán phân loại dựa trên cây quyết định (DT - decision tree) được sử dụng phổ biến trong ML. DT là một trình phân loại dựa trên mô tả cấu trúc của một cái cây, đó là gốc, thân cây, cành cây và lá. Mỗi nút bên trong của cây đại diện cho một thuộc tính và mỗi nút lá đại diện cho một nhãn lớp. Dự đoán được thực hiện thông qua các thuật toán suy luận quy nạp. DT có khả năng phân loại dễ hiểu như cách trích xuất từ các mẫu dựa trên đặc điểm [35, 36]. Ngoài ra, DT không chỉ được chứng minh là hiệu quả trong nhiều lĩnh vực, mà còn có ít tham số hơn [37]. Có hai quy tắc chính được xem xét trong quá trình xây dựng DT. Một là tiêu chí dừng để xác định thời điểm ngừng sinh trưởng của cây và phát sinh các nút lá [38]. Hai là cách gán nhãn lớp trong các nút lá [39]. Quy tắc đầu tiên có nghĩa là sự phát triển của cây phải được kết thúc nếu tất cả các mẫu thuộc cùng một lớp [40]. Quy tắc thứ hai nhấn mạnh tầm quan trọng của việc đặt ra một ngưỡng [41]. DT có một số biến thể, chẳng hạn như ID3 [38], C4.5 [42] và CART [33].

1.2.2.3. Các phương pháp đánh giá hiệu năng của mô hình

Các độ đo như ma trận nhầm lẫn, độ chính xác (Accuracy), độ chính xác dương tính (Precision), độ nhạy (Recall), độ đặc hiệu (Specificity) và điểm trung bình điều hòa F1 (F1-score) thường được sử dụng để phản ánh hiệu năng của mô hình. Đặc điểm của các thước đo này trong bài toán phân loại được khái quát như sau:

a. Ma trận nhầm lẫn (confusion matrix)

Ma trận nhầm lẫn là một bảng chứa các thông tin về kết quả phân lớp thực tế và kết quả phân lớp theo dự đoán của thuật toán phân lớp đối với các mẫu dữ liệu. Ma trận nhầm lẫn được xây dựng như Bảng 1.1. đối với phân loại nhị phân (phân loại hai lớp), và phân loại đa lớp như Bảng 1.2. .

Ma trận nhầm lẫn được xây dựng từ các thành phần như sau:

- TP (True Positive) – Mẫu dữ liệu của sự kiện ngã, mô hình dự đoán đúng (True) là sự kiện ngã (Positive) (dự đoán đúng).
- TN (True Negative) – Mẫu dữ liệu của hoạt động bình thường, mô hình dự đoán đúng (True) là hoạt động bình thường (Negative) (dự đoán đúng).
- FP (False Positive) – Mẫu dữ liệu của hoạt động bình thường, mô hình dự đoán sai (False) thành sự kiện ngã (Positive) (Lỗi loại I – Dương tính giả).
- FN (False Negative) – Mẫu dữ liệu của sự kiện ngã, mô hình dự đoán sai (False) thành hoạt động bình thường (Negative) (Lỗi loại II – Âm tính giả).

Bảng 1.1. Ma trận nhầm lẫn trong phân lớp nhị phân

		Dự đoán	
		P (+)	N (-)
Thực tế	P (+)	TP	FN (Lỗi loại II)
	N (-)	FP (Lỗi loại I)	TN

Bảng 1.2. Ma trận nhầm lẫn trong phân loại đa lớp

		Dự đoán			
		Lớp 1	Lớp 2	...	Lớp n
Thực tế	Lớp 1	TP		FN	
	Lớp 2				
	...	FP		TN	
	Lớp n				

True (T) cho biết thuật toán đã phân loại ĐÚNG. False (F) cho biết thuật toán đã phân loại SAI. Positive (P) cho biết thuật toán đã phân loại đối tượng vào lớp P. Negative

(N) cho biết thuật toán đã phân loại đối tượng vào lớp N. Sự phân loại này được biểu diễn dưới dạng ma trận 2x2 như trình bày ở Bảng 1.1 gọi là ma trận nhầm lẫn (confusion matrix).

b. Độ chính xác (Accuracy)

Độ chính xác (ACC - Accuracy) là thước đo phổ biến để đánh giá hiệu suất của mô hình phân loại. Nó được tính bằng tỉ lệ số mẫu được phân loại đúng (TP và TN) trên tổng số mẫu trong bộ dữ liệu. Thước đo này được xác định theo Công thức (1.1) [68]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1.1)$$

Đối với các bộ dữ liệu không cân bằng, như trong trường hợp các sự kiện ngã chiếm số lượng mẫu ít nhưng quan trọng, chỉ số ACC có thể cao mặc dù việc phát hiện hoạt động ngã không chính xác. Do đó, ACC chỉ phù hợp với bài toán phân loại cân bằng. Đối với bộ dữ liệu không cân bằng, các chỉ số như độ chính xác dương tính (precision), độ phủ/độ nhạy (recall), độ đặc hiệu (specificity) và điểm trung bình điều hòa (F1-score) thường được ưu tiên sử dụng để đánh giá hiệu suất mô hình phát hiện ngã một cách chính xác hơn.

c. Độ chính xác dương tính (precision)

Độ chính xác dương tính (precision) là tỉ lệ giữa số lượng các dự đoán đúng Positive (TP) trên tổng số lượng các dự đoán (bao gồm TP và FP). Trong FDS, precision đo lường khả năng của mô hình phân loại xác định chính xác các sự kiện ngã mà không gây ra các dự đoán sai (ví dụ: dự đoán sai một sự kiện ngã là hoạt động bình thường). Precision được xác định theo Công thức (1.2) [43]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1.2)$$

d. Độ nhạy (recall)

Độ phủ/độ nhạy (Sensitivity, recall, true positive rate) là tỉ lệ giữa số lượng các dự đoán đúng positive (TP) trên tổng số lượng các trường hợp positive thực tế trong bộ

dữ liệu (bao gồm TP và FN). Trong FDS, recall đo lường khả năng của mô hình phân loại tìm ra tất cả các hoạt động cụ thể cụ thể (ví dụ: tìm ra tất cả các trạng thái ngã) mà không bỏ sót. Recall được xác định theo Công thức (1.3) [43]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (1.3)$$

e. Độ đặc hiệu (specificity)

Trong bài toán phát hiện ngã hay không ngã. Độ nhạy được áp dụng cho mô hình để tự động phát hiện người bị ngã. Độ nhạy đạt 100% có nghĩa là toàn bộ những người ngã đều được phát hiện. Tuy nhiên, chỉ một mình độ nhạy không thể cho biết toàn bộ kết quả nhận dạng. Cần có thêm độ đặc hiệu, đó là tỉ lệ những trường hợp thực sự không bị ngã và có kết quả phân loại không ngã trong toàn bộ các trường hợp bị ngã. Độ đặc hiệu (Specificity, true negative rate) được tính theo Công thức (1.4) [43]:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (1.4)$$

f. Điểm trung bình điều hòa (F1-score)

F1-score là một phép đo kết hợp giữa Precision và Recall. Nó tính trung bình điều hòa giữa hai chỉ số này và là một cách để đánh giá hiệu suất tổng thể của mô hình phân loại. F1-score được xác định theo Công thức (1.5) [43]:

$$F_1\text{-score} = 2 \cdot \frac{(\text{Precision} \cdot \text{Recall})}{(\text{Precision} + \text{Recall})} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}} \quad (1.5)$$

F1-score có giá trị nằm trong khoảng từ 0 đến 1, và giá trị càng cao thì mô hình phân loại càng tốt. Nhiều nghiên cứu sử dụng thước đo F1-score để đánh giá hiệu suất của các mô hình ML, cả trong phân lớp nhị phân và đa lớp [44–47]. Tuy nhiên, độ tin cậy về hiệu suất của mô hình sử dụng thước đo F1-score giảm đáng kể trong các bộ dữ liệu không cân bằng. Thực tế, F1-score có thể cung cấp thông tin không chính xác khi một dự đoán có nhiều TP, ít TN hoặc ngược lại. Lúc này, các nhà nghiên cứu hướng đến sử dụng MCC [43] để đánh giá hiệu suất của các mô hình phân lớp dữ liệu không cân bằng. MCC đem đến một giải pháp hữu hiệu để khắc phục vấn đề không

cân bằng dữ liệu [48]. Ngay cả khi dự đoán có nhiều TP, ít TN hoặc ngược lại, các mô hình phân loại dữ liệu không cân bằng sử dụng thước đo MCC luôn cho ra kết quả dự đoán một cách tổng thể khách quan và công bằng hơn [43].

g. Hệ số tương quan Matthew (MCC)

Hệ số tương quan Matthew (MCC) [43] là một thước đo thống kê được sử dụng để đánh giá hiệu suất của mô hình phân loại, đặc biệt là trong bài toán phân loại nhị phân. MCC thường được áp dụng khi số lượng mẫu của các lớp trong dữ liệu không cân bằng. MCC được tính toán bằng cách sử dụng ma trận nhầm lẫn, được xác định theo Biểu thức (1.6) [43]

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1.6)$$

MCC có giá trị từ -1 đến 1. Giá trị 1 thể hiện một mô hình hoàn hảo, 0 có nghĩa là mô hình không khác biệt so với việc dự đoán ngẫu nhiên, và -1 chỉ ra một sự đảo ngược hoàn toàn giữa các dự đoán và thực tế. MCC là một thước đo thống kê mạnh mẽ, hữu ích, đáng tin cậy và trung thực. Nó có thể phản ánh chính xác bất kỳ thiếu sót dự đoán nào trong bất kỳ bộ dữ liệu nào [43]. MCC xem xét tỉ lệ cân bằng của bốn loại ma trận nhầm lẫn (TP, TN, FP, FN), trong khi phép đo điểm F1 bỏ qua số lượng âm tính thực sự (TN), do đó có cái nhìn sâu sắc trong việc đánh giá vấn đề. Chỉ số MCC sẽ chỉ cao nếu mô hình hoạt động tốt trên cả hai yếu tố tích cực và tiêu cực.

1.2.3. Truyền tin cảnh báo

Mô-đun truyền tin cảnh báo là một thành phần quan trọng trong hệ thống HAR. Tuy nó không thuộc đối tượng nghiên cứu trong luận án này nhưng NCS cũng giới thiệu khái quát nhằm mô tả cấu trúc chung của hệ thống HAR một cách tổng quát.

Trong trường hợp phát hiện đối tượng bị ngã, các hệ thống truyền tin được kích hoạt để gửi các bản tin cảnh báo đến người chăm sóc, người giám sát thông qua các kết nối không dây. Kỹ thuật gửi thông báo là một phần quan trọng trong HAR. Nó đảm

bảo người chăm sóc hoặc nhân viên y tế có thể biết được khi có nguy cơ ngã xảy ra và có thể đưa ra phản ứng kịp thời. Dưới đây là một số phương tiện thông báo phổ biến mà HAR có thể sử dụng:

- (1) Điện thoại di động (Mobile Phones): Hệ thống có thể gửi cảnh báo thông qua cuộc gọi điện thoại hoặc tin nhắn đến điện thoại di động của người chăm sóc hoặc người dùng. Cảnh báo qua điện thoại di động có thể cung cấp thông tin chi tiết về tình trạng ngã và cần thiết để người chăm sóc hoặc nhân viên y tế có thể đưa ra phản ứng nhanh chóng.
- (2) Email: Hệ thống có thể gửi cảnh báo qua email đến người chăm sóc hoặc nhân viên y tế. Email có thể chứa thông tin chi tiết về sự kiện ngã và hướng dẫn cần thiết để xử lý tình huống.
- (3) Tin nhắn văn bản (SMS): Cảnh báo qua tin nhắn văn bản là một phương tiện hiệu quả để thông báo nguy cơ ngã. Tin nhắn văn bản có thể chứa thông tin cơ bản về sự kiện và hướng dẫn cần thiết.
- (4) Thông báo trực tiếp đến trạm kiểm soát: Trong một môi trường công nghiệp hoặc thương mại, HAR có thể được liên kết trực tiếp với một trạm kiểm soát. Khi có nguy cơ ngã, hệ thống có thể gửi thông báo trực tiếp đến trạm kiểm soát, từ đó kích hoạt quy trình phản ứng nhanh chóng.

1.3. Tình hình nghiên cứu trong và ngoài nước

1.3.1. Một số nghiên cứu trên thế giới

Để đánh giá hiệu quả của các phương pháp đề xuất, NCS đã khảo sát và tìm ra hai bộ dữ liệu công khai được nhiều nhóm nghiên cứu đã sử dụng là MobiAct v2.0 và UP-Fall detection. Hai bộ dữ liệu này được thu thập một cách khoa học, đa dạng và phù hợp với thực tế. Bộ dữ liệu MobiAct v2.0 được thu thập từ các cảm biến quán tính trên điện thoại thông minh. Bộ UP-Fall được thu thập từ thiết bị đo IMU. Hai bộ dữ liệu này phù hợp với mục tiêu và đối tượng nghiên cứu trong luận án này. NCS sử

dụng hai bộ dữ liệu với phương pháp thu thập khác nhau với mục đích đảm bảo tính khách quan của kết quả; giảm thiên vị khi đánh giá hiệu suất hệ thống. Chi tiết về hai bộ dữ liệu này được trình bày trong Mục 2.3.

So sánh hiệu suất của các hệ thống phát hiện ngã (FDS) là một vấn đề khó khăn do mỗi nghiên cứu sử dụng một bộ dữ liệu khác nhau [49, 50]. Nhiều tác giả cho biết họ không thể so sánh trực tiếp với các nghiên cứu trước đó do sự khác biệt trong phương pháp thu thập dữ liệu, nhóm đối tượng mục tiêu và môi trường nghiên cứu [50, 51]. Pannurat và cộng sự [51] chỉ ra rằng, yếu tố ảnh hưởng đến hiệu suất là số lượng mẫu được sử dụng để huấn luyện mô hình. Kết quả đạt được từ các nghiên cứu dựa trên dữ liệu gia tốc kể cũng khó so sánh nếu không sử dụng chung một bộ dữ liệu. Chính vì lẽ đó, NCS đã tập trung khảo sát các nghiên cứu liên quan có sử dụng một trong hai bộ dữ liệu này như trình bày ở Bảng 1.3.

Bảng 1.3. Các nghiên cứu gần đây về phát hiện ngã sử dụng bộ dữ liệu MobiAct và UP-Fall.

Nhóm nghiên cứu	Bộ dữ liệu	Số (ADL /Ngã)	Cảm biến	Số đặc trưng	Miền đặc trưng	DWS/ DOR	Mô hình tốt nhất	Phân loại	Hiệu suất
Nhóm nghiên cứu dựa trên bộ dữ liệu MobiAct									
Chatzaki, 2017 [13]	MobiAct V2.0	12/4	A, G	39	T, F	1s, 80%	k-NN	Đa lớp	96,8% (F1)
Shi, 2020 [52]	MobiAct	4/4	A, G	Dữ liệu thô cho CNN, 3 đặc trưng cho CAM	T	1s, 50%	CNN + CAM	Nhị phân	95,55% (ACC)
Mahfuz, 2018 [14]	MobiAct	9/4	A	58	T	10s	ANN	Nhị phân	98,75% (ACC)
Hassan, 2019 [21]	MobiAct V2.0	12/4	A	58	T	10s	Hybrid deep CNN-LSTM	Nhị phân	97% (F1)
Xu, 2019 [8]	MobiAct	6/1	A	Dữ liệu thô	-	4s	CNN-LSTM	Nhị phân	98,98% (ACC)
Wu, 2019 [53]	MobiAct	9/4	A, G	Dữ liệu thô	-	3,45s	LSTM	Đa lớp	98,83% (ACC) 90,33% (F1)

Nhóm nghiên cứu	Bộ dữ liệu	Số (ADL /Ngã)	Cảm biến	Số đặc trưng	Miền đặc trưng	DWS/ DOR	Mô hình tốt nhất	Phân loại	Hiệu suất
Nhóm nghiên cứu dựa trên bộ dữ liệu UP-Fall detection									
Lê, 2021 [54]	UP-Fall	6/5	A, L, Av, Bs	14	T, F	-	k-NN + APGWO	Nhị phân	99,3% (ACC) 84,79% (F1)
Santoyo-Ramón, 2021 [55]	UP-Fall	6/5	A	12	T, F	-	SVM + Major voting ensemble	Nhị phân	9,59% (Sen) 98,41% (Spe)
Nahian, 2021 [56]	UP-Fall	6/5	A	39	T, F	-	LR	Nhị phân	100% (Sen) 99% (Spe)
L. Martínez-Villaseñor, 2019 [12]	UP-Fall (IMU)	6/5	A, G	756	T, F	1s, 50%	ANN	Đa lớp	95,73% (ACC) 71,79% (F1)
Lai, 2021 [9]	UP-Fall	6/5	A	Dữ liệu thô	-	-	Res-TCN (Adam CNN)	Nhị phân và đa lớp	Nhị phân: 97,75% (ACC) 98,68% (F1) Đa lớp: 94,40% (ACC) 86,73% (F1)
Kraft, 2020 [10]	UP-Fall	6/5	A	Dữ liệu thô	-	-	CNN dùng ReLU	Nhị phân	99% (F1)

Ghi chú: DWS – kích thước của số dữ liệu, DOR – tỉ lệ xếp chồng dữ liệu, A – cảm biến gia tốc, G – con quay hồi chuyển, M – cảm biến từ, L – cảm biến độ sáng, ACC – Accuracy, CAM – Class activation mapping, ResNet – Residual Network, Bs – Brain Sensor, Av – Angular Velocity, APGWO – Adaptive Particle Swarm-Grey Wolf Optimization, SMV – Signal Magnitude Vector, Sen – Sensitivity, Spe – Specificity, T – miền thời gian, F – miền tần số.

Nhiều nghiên cứu về FDS dựa trên bộ dữ liệu MobiAct chỉ sử dụng thước đo Độ chính xác (Accuracy - ACC) để đánh giá hiệu suất của mô hình [8, 14, 52]. Mặc dù ACC là một chỉ số phổ biến để đánh giá hiệu suất của các mô hình phân loại. Tuy nhiên, với các bộ dữ liệu không cân bằng, thước đo ACC không còn phù hợp để đánh giá hiệu suất của mô hình [9]. Trong các bộ dữ liệu MobiAct và UP-Fall, số lượng mẫu của các hoạt động thường ngày chiếm hơn 90%. Trong khi đó, số lượng mẫu của các hành vi ngã cần được phát hiện chính xác lại ít hơn 10%. Các hoạt động thường

ngày (ADLs) thường dễ phân loại hơn so với các hành vi ngã [12, 57]. Do đó, kết quả tổng thể theo thước đo ACC thường rất cao, nhưng kết quả phát hiện các trường hợp ngã lại thấp [9, 12, 53, 57]. Có thể thấy, chỉ số ACC không phù hợp để đánh giá hiệu suất của mô hình phát hiện ngã khi sử dụng bộ dữ liệu không cân bằng. Trong trường hợp này, sử dụng các độ đo như Độ chính xác dương tính (Precision), Độ nhạy (Sensitivity), Độ đặc hiệu (Specificity) và Điểm trung bình điều hòa (F1-score) là thích hơn [58].

Le và cộng sự [54] đề xuất tập đặc trưng mới nhằm giảm số thuộc tính đầu vào dựa trên mô hình APGWO. Họ sử dụng ba loại dữ liệu có trong bộ UP-Fall để huấn luyện mô hình. Các bộ phân loại Hồi quy Logistic (LR), K láng giềng gần nhất (kNN), Máy vectơ hỗ trợ (SVM), Cây quyết định (DT), Naive Bayes (NB), Rừng ngẫu nhiên (RF), Multilayer Perceptron (MLP) và mô hình đề xuất (APGWO) được sử dụng để đánh giá hiệu quả của tập đặc trưng đề xuất. Hiệu suất tốt nhất theo các số liệu Độ chính xác (Accuracy) và Điểm F1 (F1-score) lần lượt là 99% và 96%. Mặc dù Le và cộng sự đề xuất một mô hình nhận dạng dựa trên phân loại nhị phân, nhưng chỉ số F1-score không ấn tượng. Hơn nữa, nghiên cứu này cũng sử dụng kết hợp nhiều loại dữ liệu cảm biến khác nhau nên việc áp dụng trong thực tế trở nên khó khăn.

Diponkor Bala và GM Waliullah [59] đã trích xuất các đặc trưng miền thời gian và miền tần số của bốn hoạt động là đứng (STD), đi bộ (WAL), chạy bộ (JOG) và nhảy (JUM) từ bộ dữ liệu MobiAct v2.0. Sau đó, họ tiến hành giảm kích thước dữ liệu bằng phương pháp PCA và Fisher's LDA. Hiệu suất đạt được là 99,305% (ACC) khi sử dụng mô hình phân loại kNN. Số lượng mẫu của hai hoạt động STD và WAL lớn hơn nhiều so với JOG và JUM, do đó có thể coi là không cân bằng. Vì vậy, chỉ số ACC không phản ánh chính xác hiệu quả của bộ thuộc đặc trưng đề xuất. Với dữ liệu không cân bằng, cần xem xét thêm các chỉ số khác như Precision, Recall, F1-Score để có kết quả khách quan nhất. Hơn nữa, bốn hoạt động mà Diponkor Bala và G.M Waliullah [59] lựa chọn để xây dựng mô hình nhận dạng có tính chu kỳ, ít phức tạp nên dễ phân loại hơn. Nếu áp dụng mô hình này cho các hoạt động dễ gây nhầm lẫn

như đi lên cầu thang, đi xuống cầu thang, đứng lên, ngồi xuống hoặc các hoạt động có sự thay đổi đột ngột như ngã có thể hiệu quả sẽ không cao.

Panhwar và cộng sự [60] đã đề xuất một phương pháp phát hiện ngã sử dụng cảm biến quán tính của điện thoại thông minh, bao gồm gia tốc kế, con quay hồi chuyển và cảm biến định hướng. Họ sử dụng cả SVM và mạng nơ-ron nhân tạo (ANN) để kiểm thử. Mô hình dựa trên mạng nơ-ron nhân tạo cho kết quả phân loại tổng thể tốt nhất là 96,07% trên thước đo ACC. Nhóm nghiên cứu này chỉ lựa chọn một tập con của bộ dữ liệu MobiAct bao gồm hai hoạt động ngã (FOL, BSC) và ba hoạt động sinh hoạt hàng ngày (STD, WAL và SIT) để xây dựng mô hình. Dữ liệu cho ba hoạt động thường ngày này có trọng số lớn hơn nhiều so với hai hoạt động ngã. Do đó, kết quả thu được chưa phản ánh chính xác khả năng phát hiện ngã của phương pháp đề xuất.

Shi và cộng sự [52] đã xây dựng một mạng nơ-ron tích chập (CNN) với ánh xạ kích hoạt lớp (CAM). Họ sử dụng dữ liệu ngã và các hoạt động sinh hoạt hàng ngày (ADL), bao gồm đứng (STD), đi bộ (WAL), nhảy (JUM) và chạy bộ (JOG) từ bộ dữ liệu MobiAct để huấn luyện mô hình. Mô hình của họ đạt hiệu suất là 95,55% ở thước đo ACC. Tuy nhiên, phương pháp mà họ đề xuất không mang lại kết quả tốt khi phân loại ngã [8–10, 14, 55, 56].

Martínez-Villaseñor và cộng sự [12] giới thiệu bộ dữ liệu phát hiện ngã UP-Fall được thu thập bằng nhiều loại cảm biến khác nhau như IMU, EEG, CAM, v.v. Bộ phân loại MLP đạt hiệu suất tốt nhất ở thước đo ACC là 95,0%. Tuy nhiên, trên các chỉ số Precision, Sensitivity, và F1-score đạt kết quả lần lượt là 77,7%, 69,9%, và 72,8%, thấp hơn nhiều so với độ đo ACC. Hiệu suất phân loại các sự kiện ngã theo phương pháp của họ tương đối thấp. Điều đó chứng minh bộ đặc trưng và mô hình mà họ đề xuất không phù hợp cho phân loại đa lớp dựa trên các bộ dữ liệu không cân bằng.

Nhóm tác giả J.T Perry và cộng sự [61] cho rằng hệ thống phát hiện ngã (FDS) dựa trên gia tốc kế kết hợp với con quay hồi chuyển có thể tăng khả năng phát hiện các cú ngã. Tuy nhiên, họ không trình bày lý do tại sao khi sử dụng thêm con quay hồi

chuyển thì hiệu suất của FDS lại tăng lên. Nhóm tác giả A. Chelli và cộng sự [62] chỉ ra rằng hiệu suất của các hệ thống ML tăng lên khi trích xuất đặc trưng từ dữ liệu thu thập bằng gia tốc kế và con quay hồi chuyển. Số lượng đặc trưng đầu vào mà họ đã sử dụng là rất nhiều (328 đặc trưng). Số lượng đặc trưng cần trích xuất càng lớn và càng phức tạp thì thời gian tính toán càng lâu, không phù hợp với tiêu chí tiết kiệm năng lượng, tài nguyên tính toán hạn chế và đáp ứng nhanh của FDS. Vì vậy, trong luận án này sẽ tìm các giải pháp để chỉ sử dụng dữ liệu thu từ gia tốc kế nhằm giảm chi phí tính toán, chi phí năng lượng cho điện thoại thông minh.

Có thể nhận thấy các nghiên cứu gần đây thường tập trung vào giải bài toán phân loại hoạt động ngã và không ngã bằng phương pháp phân loại nhị phân. Rất ít nghiên cứu đề cập đến bài toán phân loại đa lớp trong việc phát hiện từng loại hoạt động, bao gồm các kiểu ngã [9, 12, 53, 57]. Trong thực tế, các mô hình phân loại đa lớp thường có hiệu suất thấp hơn các mô hình phân loại nhị phân.

Việc trích xuất và lựa chọn bộ đặc trưng ảnh hưởng đáng kể đến hiệu năng của hệ thống phát hiện ngã [15, 63, 64]. Từ thống kê ở Bảng 1.3 cho thấy cần trích xuất một số lượng lớn các đặc trưng trong bài toán đa lớp mới đạt độ chính xác dự đoán trên 94% cả cho tập dữ liệu MobiAct và UP-Fall. Hiệu suất dự đoán cao với số lượng đặc trưng dưới 40 chỉ có thể đạt được trong các bài toán phân loại nhị phân. Chatzaki và cộng sự [13] sử dụng 39 đặc trưng trong miền thời gian và tần số để đạt độ chính xác dự đoán trung bình là 96,8%, tuy nhiên độ chính xác cao nhất của phát hiện ngã không vượt quá 84%. Do đó, luận án này hướng tới phát triển một giải pháp trích rút đặc trưng hiệu quả với tài nguyên tính toán thấp phù hợp cho việc phát hiện ngã trong bối cảnh phân loại đa lớp phức tạp trên thiết bị di động.

Xu hướng hiện nay trong các bài toán phân loại, các tác vụ quan trọng như lựa chọn đặc trưng, trích xuất đặc trưng, tinh chỉnh các siêu tham số được tiến hành bằng cách sử dụng MH. Kỹ thuật MH cung cấp giải pháp tốt nhất của vấn đề phức tạp bằng cách thực hiện các lần lặp lại dựa trên kinh nghiệm. Do đó, MH được xem là giải pháp tốt

để tìm kiếm bộ siêu tham số tối ưu của HAR. Cách tiếp cận của MH là đơn giản, linh hoạt, không đạo hàm và ít xảy ra trường hợp tối ưu cục bộ [65].

De Leonardis và cộng sự [66] cho thấy việc sử dụng GA để trích xuất 69 đặc trưng từ 300 đặc trưng làm cho quá trình phân loại các hoạt động nhanh hơn khi sử dụng bộ phân loại kNN. Kết quả thực nghiệm cho thấy kNN có thể phân loại chính xác khoảng 97% hoạt động khi sử dụng các vectơ đặc trưng đã tối ưu. Nhóm tác giả Putra và Vesilo [67] cho thấy chi phí tính toán giảm đáng kể và tỉ lệ phát hiện sự kiện ngã (điểm F1) thực sự khách quan khi sử dụng kỹ thuật lựa chọn dựa trên GA.

Nhóm tác giả El-Maaty và Wassal [68] đã đề xuất một mô hình lai có tên gọi là GA-PCA để rút gọn tập đặc trưng. Ban đầu họ lựa chọn một tập lớn các đặc trưng từ chuỗi thời gian, sau đó sử dụng các phép biến đổi trực giao dựa trên PCA để giảm bớt các đặc trưng yếu từ tập đã chọn. Cuối cùng họ dùng phương pháp xác nhận chéo dựa trên GA để tiếp tục loại bỏ các đặc trưng hiệu suất thấp. Kết quả thực nghiệm cho thấy độ chính xác tăng lên 5% khi sử dụng phương pháp MH. Độ chính xác, thời gian thực thi và mức tiêu thụ bộ nhớ là một trong những thách thức lớn khi phát triển thuật toán nhận dạng dựa trên điện thoại thông minh.

Nguyen và cộng sự [69] đã giải quyết bài toán đó bằng cách áp dụng Giải thuật di truyền để tối ưu hóa vectơ đặc trưng. Với phương pháp đã đề xuất, số lượng vectơ đặc trưng đã giảm 49% so với ban đầu mà kết quả phân loại của SVM vẫn không có nhiều thay đổi. Độ chính xác của mô hình Markov ẩn liên tục 2 giai đoạn tăng lên 1.8% khi sử dụng bộ vectơ đặc trưng tối ưu. Zhang và cộng sự [70] cho rằng có thể tìm ra bộ siêu tham số tối ưu một cách tự động, nhanh chóng và độ chính xác cao hơn để nhận biết hoạt động của cánh tay trong thời gian thực khi áp dụng GA.

Batool và cộng sự [71] đã thiết kế một hệ thống có tên là MFCC để trích xuất đặc trưng mới tạo thành các vectơ đặc trưng. Bộ vectơ đặc trưng được phân loại bởi SVM kết hợp với PSO thông qua việc tạo hạt. Họ đã sử dụng PSO và cách tiếp cận MH để giảm thời gian, độ phức tạp và tăng độ chính xác cho hệ thống. Kết quả thực nghiệm

cho thấy độ chính xác hơn 87% khi đánh giá bộ dữ liệu cảm biến chuyển động. Nhóm tác giả Ning và cộng sự [72] đã lựa chọn các đặc trưng và tham số tối ưu dựa trên trí thông minh bầy hạt nhai phân rời rạc để huấn luyện các bộ phân loại như SVM. Bộ phân loại SVM được tối ưu hóa theo phương pháp của họ có độ chính xác đáng kể so với SVM không được tối ưu hóa. Các hệ thống phát hiện ngã gặp nhiều thách thức với bộ dữ liệu không cân bằng được thu thập từ các thiết bị đeo được nếu áp dụng các thuật toán truyền thống.

Từ việc khảo sát các nghiên cứu liên quan, có thể thấy phần lớn các nghiên cứu sử dụng kỹ thuật MH để tìm kiếm và trích xuất các đặc trưng cho hệ thống HAR dựa trên ML và DL. NCS gặp rất nhiều khó khăn khi tìm các bài viết liên quan đến bài toán tối ưu hóa siêu tham số của mô hình RF bằng các kỹ thuật MH. Vì vậy, trong luận án này, NCS đã tìm cách giải bài toán tối ưu mô hình RF bằng cách áp dụng kỹ thuật MH – cụ thể là sử dụng GA để tìm bộ siêu tham số tối ưu cho mô hình RF.

1.3.2. Một số nghiên cứu ở Việt Nam

Ở Việt Nam, một trong những nhóm nghiên cứu mạnh về giải pháp nhận dạng hoạt động con người và phát hiện ngã phải kể đến PGS.TS Phạm Văn Cường và các đồng nghiệp tại Học viện Công nghệ Bưu chính Viễn thông. Trong các công trình nghiên cứu của mình, nhóm đã đề xuất nhiều giải pháp xây dựng hệ thống nhận dạng hoạt động và phát hiện ngã.

Cụ thể, trong nghiên cứu [73], họ đã đề xuất một phương pháp tiếp cận dựa trên cảm biến đeo để theo dõi ngã trong thời gian thực và nhận dạng chi tiết các hoạt động. Bằng cách sử dụng thiết bị cảm biến Wii Remotes đeo trên cổ tay và hông, họ thu thập dữ liệu và trích xuất 58 đặc trưng từ tín hiệu gia tốc kế. Sau đó, họ sử dụng mô hình Markov ẩn (HMM) để ánh xạ các đặc trưng này thành các trạng thái ẩn, tương ứng với 13 hoạt động khác nhau và để phát hiện ngã.

Trong nghiên cứu [74], nhóm tác giả đề xuất mô hình CNN kết hợp với mạng bộ nhớ dài ngắn hạn (LSTM) để giải quyết bài toán phát hiện các vận động bất thường từ dữ

liệu cảm biến không thuần nhất, bao gồm cảm biến gia tốc, cảm biến con quay hồi chuyển và cảm biến từ tính. Mô hình CNN học và biểu diễn đặc trưng từ dữ liệu cảm biến theo không gian, còn LSTM dùng đặc trưng theo thời gian để suy diễn hành vi.

Trong công bố [75], tác giả đề xuất một hệ thống sử dụng dữ liệu cảm biến gia tốc từ SP Samsung Galaxy Note II. Phương pháp nhận dạng hoạt động của họ bao gồm bốn bước: xử lý dữ liệu, phân đoạn, trích xuất đặc trưng và phân loại. Từ tín hiệu gia tốc kếp, hệ thống trích xuất một loạt đặc trưng (không nêu rõ số lượng) được sử dụng cho các HMM và thuật toán DT C4.5.

Gần đây nhóm tác giả Phạm Văn Cường đề xuất một mô hình DL, tăng cường dữ liệu và lấy mẫu lại dữ liệu để giải quyết vấn đề dữ liệu ngã khó thu thập trong thế giới thực và sự mất cân bằng nghiêm trọng giữa dữ liệu các hoạt động bình thường và các hành vi ngã [76]. Phương pháp học tập đa tác vụ này nhằm mục đích thu hẹp khoảng cách về hiệu suất phát hiện ngã giữa dữ liệu mô phỏng và dữ liệu trong thế giới thực. Nhóm tác giả chia tập dữ liệu thành nhiều tập huấn luyện cho từng tác vụ, và chứng minh cách tiếp cận này cho kết quả tốt hơn so với mô hình đơn tác vụ huấn luyện trên tất cả các tập dữ liệu. Sau khi trích xuất đặc trưng bằng CNN thời gian (TCN) và Mô-đun chú ý thời gian (TAM) để trích xuất đặc trưng và phát hiện các sự kiện ngã từ nhiều bộ dữ liệu khác nhau.

Trong công trình [77], nhóm tác giả đề xuất một khung phương thức đa cảm biến để nhận dạng hoạt động chi tiết của con người và phát hiện hành vi bất thường bằng cách kết hợp dữ liệu khung xương và cảm biến gia tốc ở cấp độ đặc trưng (gọi là hợp nhất ở cấp đặc trưng). Họ sử dụng các CNN thời gian (TCN) bao gồm các thành phần tích chập gây giãn nở có tính nhân quả để học đặc trưng và xử lý các đặc tính theo thời gian. Bản đồ đặc trưng được học và biểu diễn bằng các lớp tích chập trong TCN sẽ được cho vào hai lớp kết nối hoàn toàn để dự đoán.

Ngoài ra, nhóm nghiên cứu Đào Vũ Trường Sơn và cộng sự [42] ở Trường Đại học Quốc tế, Đại học Quốc gia Thành phố Hồ Chí Minh đề xuất sử dụng các thuật toán

GWO và APGWO để chọn ra những đặc trưng quan trọng cho FDS. Hệ thống dùng các phương pháp ML khác nhau như hồi quy logistic (LR), kNN, SVM, DT, NB, RF, và MLP để phân loại hành vi ngã và không ngã. Họ có 36 đặc trưng, sau đó họ chọn được 17 đặc trưng bằng GWO và 14 đặc trưng bằng APGWO để huấn luyện các mô hình ML. là hai mô hình có hiệu suất tốt nhất. Họ khuyến nghị nên sử dụng hai mô hình này cho nghiên cứu về FDS. Độ chính xác phân loại và điểm F1 tốt nhất họ đạt lần lượt là 99% và 96% trên bộ dữ liệu UP-Fall.

Tại Đại học Khoa học Tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh, nhóm tác giả T.V Ha và đồng nghiệp [78] đã thực hiện quá trình tiêu chuẩn hóa cho tập dữ liệu cảm biến bằng cách điều chỉnh giá trị trung bình của thuộc tính để đạt được giá trị 0 và độ lệch chuẩn tương ứng với 1 (một). Quá trình này giúp tạo ra các đặc trưng có ý nghĩa từ dữ liệu cảm biến. Họ đã chọn 28 thuộc tính cụ thể từ dữ liệu cảm biến và sử dụng CNN để trích xuất các đặc trưng từ dữ liệu máy ảnh. Nhóm tác giả đã áp dụng hai thuật toán là XGBoost và CatBoost để phân loại các hoạt động dựa trên dữ liệu cảm biến. Để phân loại các hoạt động từ dữ liệu cảm biến và hai camera, nhóm tác giả đã sử dụng mạng nơ-ron sâu.

Nhóm nghiên cứu Trần Ngọc Đức và Phan Đình Duy ở Đại học Công nghệ Thông tin, Đại học Quốc gia TP. Hồ Chí Minh đã giới thiệu một mô hình xác định sáu ADL bằng thuật toán SVM dựa trên phân tích dữ liệu từ các cảm biến chuyển động trong SP [79]. Mô hình của họ xác định các hoạt động như đi bộ, đứng, ngồi, nằm, đi bộ lên cầu thang, xuống cầu thang. Tuy nhiên, họ chưa xem xét các hành vi bất thường như người ngã trong ADL.

Nguyen cộng sự [69] đã giải quyết bài toán đó bằng cách áp dụng GA để tối ưu hóa vector đặc trưng. Với phương pháp đề xuất, số lượng vector đặc trưng giảm 49% so với ban đầu mà kết quả phân loại của SVM vẫn không thay đổi nhiều. Độ chính xác của HMM liên tục 2 giai đoạn tăng lên 1.8% khi sử dụng bộ vector đặc trưng tối ưu.

Nhóm nghiên cứu Trần Thị Thanh Hải tại Đại học Bách khoa Hà Nội đã nghiên cứu về phát hiện người ngã từ dữ liệu hình ảnh [80]. Họ đã sử dụng camera 3D (cảm biến Kinect) để thu thập dữ liệu và xử lý ảnh để nhận biết các hoạt động và hành vi của con người. Họ đã phát hiện được các trường hợp người ngã từ dữ liệu hình ảnh. Tuy nhiên, hệ thống này có chi phí cao và khó áp dụng trong thực tế vì có thể nhiều đối tượng hoạt động trong vùng camera giám sát.

1.4. Giải pháp nghiên cứu

Xu hướng gần đây trong nghiên cứu về hệ thống phát hiện ngã (FDS) là sử dụng điện thoại thông minh (SP) do sự phổ biến của chúng và sự đa dạng của các cảm biến nhúng bên trong [81–84]. SP không chỉ là vật dụng thiết yếu hàng ngày mà còn được người dùng lớn tuổi chấp nhận rộng rãi, do đó ít xâm phạm quyền riêng tư hơn so với các thiết bị theo dõi chuyên dụng.

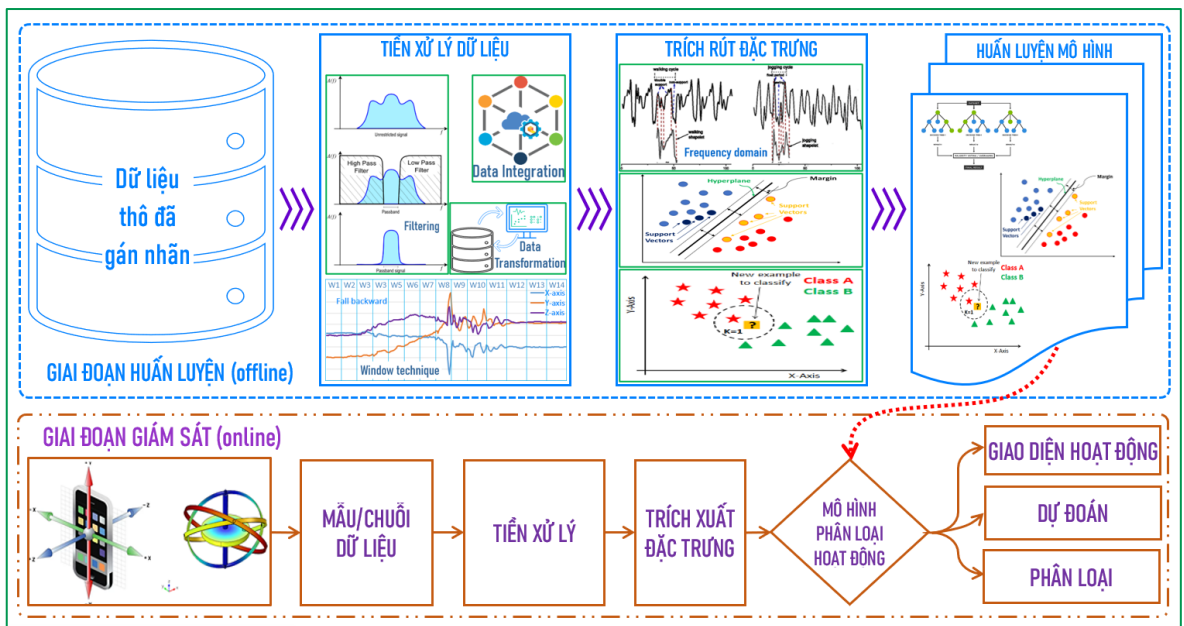
Điện thoại thông minh được trang bị nhiều loại cảm biến có độ chính xác cao và có thể kết nối với các hệ thống cảm biến độc lập khác thông qua mạng không dây năng lượng thấp (chẳng hạn như Bluetooth Low Energy - BLE) [23]. Khả năng lưu trữ lớn, bộ vi xử lý tốc độ cao, dễ dàng cài đặt ứng dụng và khả năng kết nối (wifi, 4G, 5G, SMS, BLE, NFC) để truyền tin cảnh báo là những lợi thế lớn của SP [85, 86].

Nhiều nghiên cứu về hệ thống phát hiện ngã (FDS) dựa trên SP cho rằng việc sử dụng kết hợp gia tốc kế và con quay hồi chuyển có thể giúp tăng khả năng phát hiện các sự kiện ngã [9, 55, 56, 62]. Tuy nhiên, việc sử dụng đồng thời nhiều cảm biến như gia tốc kế và con quay hồi chuyển trong FDS, mặc dù có thể tăng khả năng phát hiện ngã, lại dẫn đến việc tăng lượng dữ liệu đầu vào và số lượng đặc trưng cần trích xuất. [12, 62, 85]. Khi dữ liệu tăng lên, việc áp dụng các thuật toán học sâu trở nên cần thiết để nâng cao hiệu suất của hệ thống. Thêm vào đó, cảm biến con quay hồi chuyển tiêu hao nhiều năng lượng hơn so với gia tốc kế [87, 88]. Vì vậy, việc sử dụng nhiều cảm biến cùng lúc và xử lý dữ liệu bằng thuật toán học sâu yêu cầu phần cứng mạnh mẽ, dẫn đến việc tiêu thụ nhiều năng lượng, tăng chi phí tính toán và chi phí vận hành.

Điều này không thích hợp với các hệ thống có hạn chế về tài nguyên tính toán và nguồn điện, như thiết bị di động hoặc thiết bị nhúng/IoT.

Với những lý do đã nêu, có thể thấy các thuật toán học sâu không phù hợp với nguồn dữ liệu khan hiếm như các hành vi ngã, và các hệ thống có tài nguyên tính toán, nguồn điện hạn chế như điện thoại di động.

Do đó, luận án này đề xuất sử dụng thuật toán ML và chỉ sử dụng cảm biến gia tốc để phát triển mô hình phát hiện ngã trên điện thoại thông minh. Khung kiến trúc của mô hình này dựa trên SP và kỹ thuật ML được mô tả như Hình 1.7, bao gồm 2 giai đoạn: (1) giai đoạn huấn luyện mô hình và (2) giai đoạn giám sát.



Hình 1.7. Khung kiến trúc của mô hình phát hiện ngã của người dùng dựa trên SP và kỹ thuật ML.

Trong giai đoạn huấn luyện, dữ liệu thu thập từ cảm biến gia tốc được xử lý và phân tích để trích xuất các đặc trưng và huấn luyện mô hình ML. Trong giai đoạn giám sát, dữ liệu mới được thu thập và xử lý tương tự, sau đó được đưa vào mô hình đã huấn luyện để dự đoán xem có xảy ra sự kiện ngã hay không. Nếu phát hiện ngã, hệ thống sẽ kích hoạt cảnh báo đến các số điện thoại đã được cài đặt trước.

Phương pháp này giúp giảm thiểu chi phí tính toán và năng lượng, đồng thời tận dụng tính phổ biến và tiện lợi của điện thoại thông minh để cung cấp một giải pháp phát hiện ngã hiệu quả và thiết thực.

1.5. Kết luận chương

Chương này đã trình bày tổng quan về các khái niệm cơ bản liên quan đến nhận dạng hoạt động và hành vi bất thường của con người, đặc biệt là hành vi ngã. Cấu trúc tổng quan và các thành phần của một hệ thống nhận dạng hoạt động của con người (HAR) cũng đã được giới thiệu. Chương cũng đã thảo luận về các loại cảm biến, mô hình học máy (ML) thường được sử dụng trong HAR, cùng với các phương pháp đánh giá hiệu năng của mô hình.

Bằng cách xem xét các phương pháp, mô hình và kỹ thuật tiêu biểu, đồng thời đánh giá ưu điểm và nhược điểm của các nghiên cứu trong và ngoài nước liên quan đến bài toán nhận dạng hoạt động, luận án đã xác định được hướng tiếp cận phù hợp cho việc phát triển một hệ thống phát hiện ngã hiệu quả và đáng tin cậy dựa trên cảm biến gia tốc của điện thoại thông minh.

Chương tiếp theo sẽ trình bày chi tiết về phương pháp trích rút đặc trưng từ dữ liệu thô thu thập bằng cảm biến gia tốc của điện thoại thông minh, đặt nền tảng cho việc xây dựng hệ thống nhận dạng hoạt động và phát hiện sự kiện ngã.

Chương 2. TRÍCH CHỌN ĐẶC TRUNG

Các hệ thống phát hiện ngã dựa trên cảm biến quán tính của điện thoại thông minh có nhiều ưu điểm như tiết kiệm chi phí, dễ dàng cài đặt, hoạt động liên tục và khả năng tích hợp với các ứng dụng khác như gọi điện, gửi tin nhắn, định vị GPS. Tuy nhiên, chúng cũng đối mặt với nhiều thách thức như dữ liệu phức tạp, chứa nhiều nhiễu và không ổn định. Điều này làm cho việc trích xuất các đặc trưng có thể phân biệt tốt các sự kiện ngã và các hoạt động khác trở nên khó khăn. Ngoài ra, việc tiết kiệm năng lượng để tăng tuổi thọ pin của điện thoại thông minh cũng là gặp nhiều thử thách.

Chương này tập trung tìm các giải pháp để giải quyết các khó khăn đã nêu bằng cách đề xuất quy trình trích rút đặc trưng từ dữ liệu cảm biến gia tốc, bao gồm tiền xử lý dữ liệu, trích xuất đặc trưng, lựa chọn đặc trưng và lựa chọn mô hình học máy phù hợp. Nghiên cứu sử dụng hai bộ dữ liệu công khai là MobiAct v2.0 và UP-Fall để đánh giá hiệu quả của các giải pháp đề xuất. Các giải pháp này cũng đã được công bố trong các công trình nghiên cứu [CT1], [CT2], và [CT3].

2.1. Giới thiệu

Đặc trưng là một khái niệm quan trọng trong khai phá dữ liệu, chúng phản ánh thuộc tính của đối tượng và giúp phân biệt giữa các đối tượng tương đồng. Ví dụ, màu sắc, hình dạng, kích thước và mùi hương là các đặc trưng phản ánh đặc điểm của loài hoa. Sử dụng tập hợp các đặc trưng thay vì dữ liệu thô sẽ cải thiện độ chính xác của quá trình phân loại [89]. Một đặc trưng có thể bao gồm nhiều đặc trưng khác hoặc là sự tổng hợp của nhiều thuộc tính dữ liệu, giúp giảm chiều của dữ liệu gốc. Bằng cách sử dụng các đặc trưng quan trọng, hệ thống sẽ ưu tiên các đặc trưng có trọng số cao hơn trong quá trình dự đoán.

Trích xuất đặc trưng là phương pháp chuyển đổi không gian đặc trưng ban đầu sang một không gian mới. Nói cách khác, đây là phương pháp tạo ra đặc trưng mới từ các

đặc trưng ban đầu. Trong các FDS, dữ liệu cảm biến được chuyển thành dữ liệu trong không gian đặc trưng mới thông qua một bộ đặc trưng đã chọn. Giả sử một phân đoạn dữ liệu có dạng như Biểu thức (2.1) [7].

$$s = \{s_1, s_2, \dots, s_n\} \in \mathbb{R}^n \quad (2.1)$$

một đặc trưng được trích xuất trên phân đoạn đó có dạng:

$$y = f(s) (y_1, y_2, \dots, y_m) \text{ với } i = 1 \div m \quad (2.2)$$

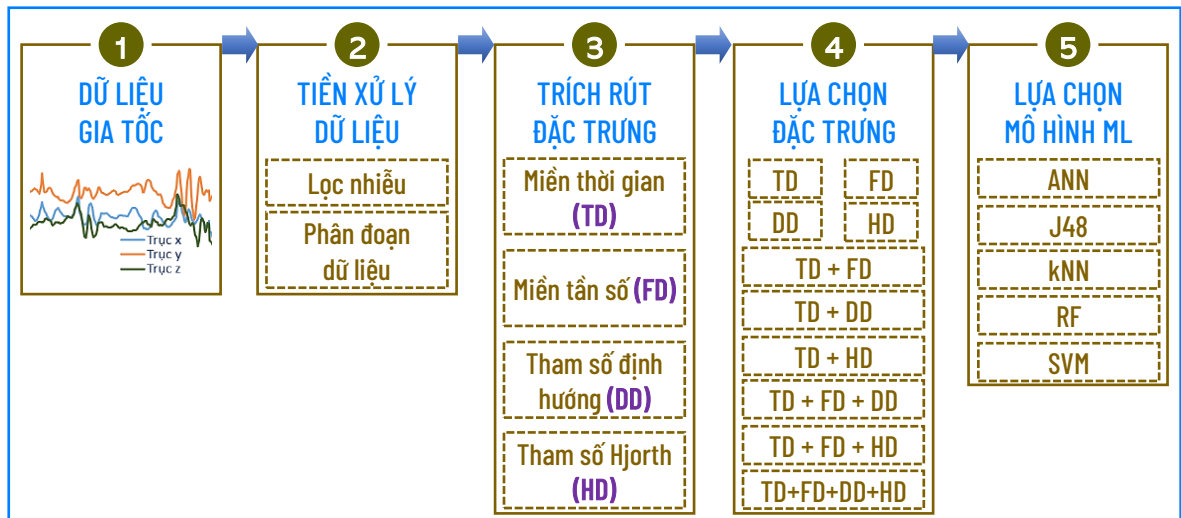
trong đó f là một hàm ánh xạ (hay còn gọi là hàm đặc trưng) biến đổi dữ liệu s từ không gian n chiều sang không gian m chiều (m thường nhỏ hơn n). Kích thước không gian đặc trưng $m \leq n$, nghĩa là việc trích xuất các đặc trưng sẽ làm giảm kích thước không gian dữ liệu thô.

Trích xuất đặc trưng đóng vai trò quan trọng trong các FDS dựa trên ML vì nó giúp xác định bộ đặc trưng tinh gọn từ dữ liệu cảm biến, giảm lỗi phân loại và độ phức tạp tính toán. Để nhận dạng chính xác, một vector g_i được chọn sao cho dữ liệu gốc ánh xạ tới các vùng khác nhau của không gian đặc trưng.

Các hàm đặc trưng cần được lựa chọn để không gian mới thể hiện thông tin hoạt động và thuộc tính đặc biệt của từng đối tượng. Hệ thống sử dụng thông tin từ các đặc trưng này để phân tích, nhận dạng và xác định các hoạt động. Lựa chọn đặc trưng phù hợp sẽ đảm bảo tính toàn vẹn và sẵn có của dữ liệu là ưu tiên hàng đầu. Trong các kỹ thuật xử lý tín hiệu cảm biến, các đặc tính dữ liệu được chọn theo tính chất thời gian, tần số và rời rạc và đã được sử dụng trong nhiều nghiên cứu [90, 91].

2.2. Quy trình trích rút đặc trưng từ dữ liệu cảm biến gia tốc

Quá trình trích rút đặc trưng từ dữ liệu cảm biến gia tốc cho hệ thống phát hiện sự kiện ngã của con người được thực hiện theo quy trình như minh họa ở Hình 2.1.



Hình 2.1. Quy trình trích rút đặc trưng.

- (1) **Nguồn dữ liệu gia tốc:** Sử dụng cảm biến gia tốc ba trục từ các thiết bị như điện thoại thông minh, đồng hồ thông minh, hoặc Đơn vị đo lường quán tính (IMU) để thu thập dữ liệu về chuyển động của người dùng.
- (2) **Tiền xử lý dữ liệu:** Bước này bao gồm việc lọc nhiễu để loại bỏ các tín hiệu không mong muốn và chia dữ liệu chuỗi thời gian thành các đoạn nhỏ để dễ dàng xử lý.
- (3) **Trích rút đặc trưng:** Mỗi phân đoạn dữ liệu ở bước tiền xử lý được sử dụng để trích rút đặc trưng theo miền thời gian, miền tần số, tham số định hướng và tham số Hjorth.
- (4) **Lựa chọn đặc trưng:** Sau khi trích rút, dựa trên các thực nghiệm để tiến hành lựa chọn những đặc trưng phù hợp với phân loại ngã. Loại bỏ các đặc trưng không mang nhiều thông tin hoặc có độ tương quan cao.
- (5) **Lựa chọn mô hình phù hợp với tập đặc trưng:** Áp dụng năm mô hình học máy phổ biến (SVM, kNN, RF, ANN và J48) để đánh giá mức độ ảnh hưởng của các tập đặc trưng đề xuất.

2.3. Các bộ dữ liệu sử dụng trong nghiên cứu

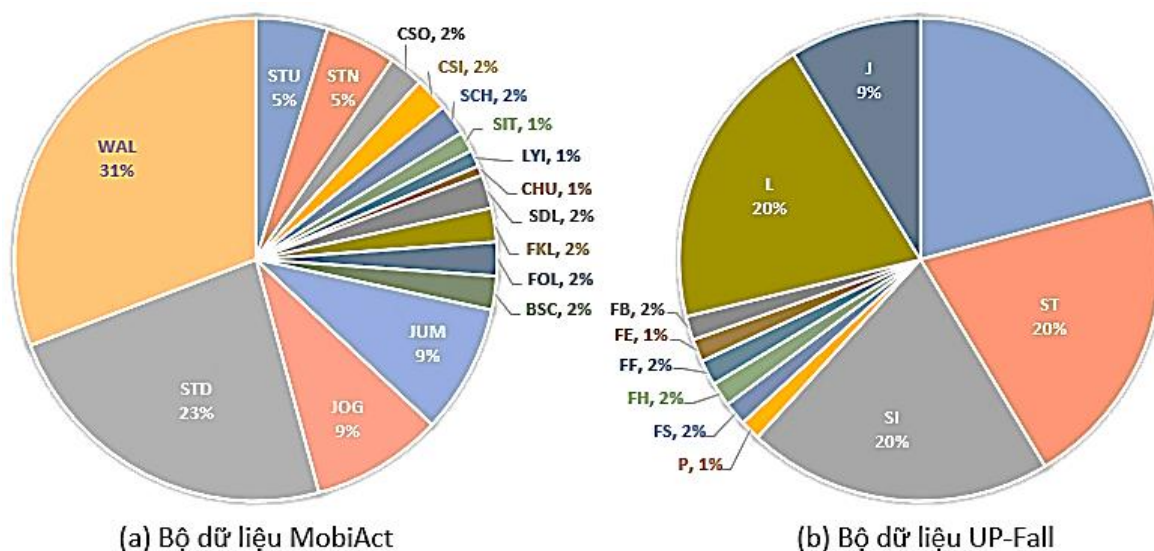
Trong luận án này và trong các nghiên cứu về chủ đề phát hiện các sự kiện ngã, NCS đã sử dụng hai bộ dữ liệu công khai là MobiAct v2.0 [92] và UP-Fall detection [12] để thực nghiệm - đánh giá các giải pháp đề xuất. Cả hai bộ dữ liệu này đã được rộng rãi sử dụng trong nghiên cứu liên quan đến nhận dạng hoạt động và phát hiện ngã, nhờ vào kích thước lớn và mức độ phổ biến của chúng. Điểm nổi bật của hai bộ dữ liệu là sự đa dạng về số lượng hoạt động, số lượng tình nguyện viên tham gia và phương pháp thu thập dữ liệu. Bộ dữ liệu MobiAct v2.0 được thu thập từ các cảm biến của điện thoại thông minh, trong khi bộ dữ liệu UP-Fall được thu thập từ các cảm biến của thiết bị Đơn vị đo lường quán tính (IMU). Chủ sở hữu và tác giả của các bộ dữ liệu này đã cho phép NCS sử dụng chúng trong các nghiên cứu. Các bộ dữ liệu này đã được xem xét và phê duyệt bởi các chuyên gia về đạo đức nghiên cứu của con người. Thông tin về hai bộ dữ liệu được trình bày trong Bảng 2.1.

Bảng 2.1. Các bộ dữ liệu công khai được sử dụng trong ADL và phát hiện ngã.

Bộ dữ liệu	Năm công bố	ADL	Ngã	Số người thu thập	Giới tính		Tuổi	Chiều cao (cm)	Cân nặng (kg)
					Nam	Nữ			
MobiAct	2016	12	4	66	48	19	20 – 47	158 – 193	50 – 120
UP-Fall	2019	6	5	17	9	8	18 – 24	157 – 175	53 – 99

Tỉ lệ phân bố mẫu dữ liệu của các hoạt động trong hai bộ dữ liệu MobiAct và UP-Fall được thể hiện ở Hình 2.2.

Đối với bộ dữ liệu MobiAct, số lượng mẫu của bốn hành vi ngã chỉ chiếm 8% tổng số mẫu. Trong khi đó, bốn hoạt động cơ bản hàng ngày như Đi bộ (WAL), Đứng (STD), Chạy bộ (JOG), Nhảy (JUM) chiếm 72% mẫu. Đối với bộ dữ liệu UP-Fall, số lượng mẫu của năm sự kiện ngã chỉ chiếm 9% tổng số mẫu. Sáu hoạt động cơ bản hàng ngày như Đi bộ (W), Đứng (ST), Ngồi (SI), Nhảy (J), Nằm (L), chiếm 90% mẫu. Có thể nhận thấy, MobiAct và Up-Fall đều là những bộ dữ liệu có sự không cân bằng lớn giữa các mẫu hoạt động bình thường và các sự kiện ngã.



Hình 2.2. Phân phối dữ liệu trên hai bộ dữ liệu: (a) MobiAct v2.0; (b) UP-Fal [12, 92]

2.3.1. Bộ dữ liệu MobiAct v2.0

MobiAct v2.0 [92] là một bộ dữ liệu công khai được thu thập từ điện thoại thông minh khi người tham gia thực hiện các loại hoạt động khác nhau và các tình huống ngã. Bộ dữ liệu này dựa trên phiên bản MobiAct đã được công bố trước đó (Vavoulas và cộng sự, 2016). Bộ dữ liệu MobiAct v2.0 được thu thập từ cảm biến gia tốc, con quay hồi chuyển và cảm biến định hướng của điện thoại Samsung Galaxy S3. Thông số kỹ thuật của cảm biến gia tốc và cảm biến con quay hồi chuyển trên điện thoại thông minh Samsung Galaxy S3 được tóm tắt như trong Bảng 2.2.

Bảng 2.2. Thông số cảm biến gia tốc và con quay hồi chuyển trên điện thoại thông minh Samsung Galaxy S3

Thông số	Loại cảm biến	
	Gia tốc	Con quay hồi chuyển
Nhà sản xuất	STMicroelectronics	InvenSense
Dải đo	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$	$\pm 250^\circ/s, \pm 500^\circ/s, \pm 1000^\circ/s, \pm 2000^\circ/s$
Độ nhạy	1024 LSB/g	16,4 LSB/ $^\circ/s$
Độ phân giải	12 bit	16 bit
Tốc độ lấy mẫu	1Hz đến 1 kHz	lên đến 8 kHz
Công suất tiêu thụ	0,75 mA (chế độ đo liên tục) 0,5 μ A (chế độ chờ)	3 mA (chế độ đo liên tục) 0,5 μ A (chế độ chờ)

Phiên bản MobiAct v2.0 bao gồm 4 loại ngã khác nhau, 12 hoạt động sinh hoạt hàng ngày (ADL) khác nhau và một kịch bản cuộc sống hàng ngày từ tổng cộng 66 đối tượng với hơn 3200 lần thử nghiệm (3200 mẫu), tất cả đều được ghi lại bằng điện thoại thông minh.

Các nhãn hoạt động được thu thập trong bộ dữ liệu này được tóm tắt như Bảng 2.3. Các hoạt động sinh hoạt hàng ngày được chọn dựa trên các tiêu chí:

- (1) Ưu tiên các hoạt động có đặc điểm tương tự như ngã. Những hoạt động này bao gồm các chuỗi động tác mà tình nguyện viên thường giữ yên động tác, ở các tư thế khác nhau, chẳng hạn như ngồi xuống ghế hoặc bước vào xe hơi.
- (2) Các hoạt động thay đổi trạng thái đột ngột hoặc nhanh chóng và tương tự như ngã, chẳng hạn như nhảy và chạy bộ.
- (3) Các hoạt động phổ biến nhất trong cuộc sống hàng ngày như đi bộ, đứng, lên và xuống cầu thang.

Bảng 2.3. Hoạt động thường ngày và ngã trong bộ dữ liệu MobiAct [92].

Nhãn	Thời gian	Số mẫu [*]	Mô tả	Loại hoạt động
BSC	10 giây	6308	Ngã ngửa trong khi đang ngồi trên ghế	Các sự kiện ngã
FKL	10 giây	6342	Ngã về phía trước khi đang đứng, đầu gối tiếp xúc đầu tiên.	
FOL	10 giây	6324	Ngã về phía trước khi đang đứng, sử dụng tay để chống.	
SDL	10 giây	6315	Ngã sang một bên khi đang đứng, gập chân	
CHU	6 giây	1800	Chuyển tiếp từ ngồi sang đứng (đứng lên từ ghế)	Hoạt động bình thường (hàng ngày)
CSI	6 giây	6382	Bước vào trong xe	
CSO	6 giây	6444	Bước ra khỏi xe hơi	
JOG	30 giây	24652	Chạy bộ	
JUM	30 giây	24163	Nhảy liên tục	
LYI	10 giây	3106	Hoạt động được thực hiện ở khoảng thời gian nằm sau ngã	
SCH	6 giây	5799	Chuyển tiếp từ đứng sang ngồi (ngồi trên ghế)	

Nhãn	Thời gian	Số mẫu [*]	Mô tả	Loại hoạt động
SIT	1 phút	3787	Ngồi trên ghế với những cử động nhẹ nhàng	
STD	5 phút	65562	Đứng với ma cử động nhẹ nhàng	
STN	10 giây	13311	Đi xuống cầu thang (10 bậc)	
STU	10 giây	13462	Đi lên cầu thang (10 bậc)	
WAL	5 phút	86178	Đi bộ bình thường	

[*]: Đây là số mẫu dữ liệu sử dụng trong luận án, nó được thu thập bằng cảm biến gia tốc của điện thoại thông minh. NCS không thống kê số mẫu của dữ liệu thu bằng cảm biến con quay hồi chuyển và cảm biến định hướng.

Bộ dữ liệu MobiAct v2.0 [92] được thu thập từ gia tốc kế, con quay hồi chuyển và cảm biến định hướng của SP Samsung Galaxy S3 với tần số lấy mẫu khoảng 85Hz cho mọi hoạt động. Việc lấy mẫu các sự kiện ngã được thực hiện ba lần cho mỗi kiểu hoạt động, mỗi mẫu kéo dài 10 giây. Các hoạt động đứng và đi bộ chỉ được lấy mẫu một lần trong 5 phút. Các ADL khác có thời gian lấy mẫu từ sáu giây đến 30 giây, tùy thuộc vào từng loại hoạt động. Tỷ lệ mẫu của các sự kiện ngã chỉ chiếm khoảng 8% tổng số dữ liệu được thu thập.

Luận án này chỉ sử dụng dữ liệu thô thu từ cảm biến gia tốc để trích rút đặc trưng. Tần số lấy mẫu cho tất cả các hoạt động được giữ nguyên 85Hz.

2.3.2. Bộ dữ liệu UP-Fall

Bộ dữ liệu UP-Fall [12] bao gồm 11 hoạt động - sáu hoạt động thường ngày cũng như năm sự kiện ngã khác nhau của con người được thu thập bởi 17 thanh niên khỏe mạnh. Họ áp dụng phương pháp thu thập đa phương thức bằng cách sử dụng cảm biến đeo được, cảm biến môi trường và cảm biến thị giác. Thông tin về các nhãn hoạt động trong bộ dữ liệu này được tóm tắt ở Bảng 2.4.

Để thu thập dữ liệu cho bộ UP-Fall, Martínez-Villaseñor và cộng sự [12] đã sử dụng năm cảm biến đeo trên người Mbientlab MetaSensor để thu thập dữ liệu thô từ gia tốc kế và con quay hồi chuyển. Họ cũng sử dụng tai nghe điện não đồ (EEG)

NeuroSky MindWave để đo tín hiệu sóng não ở trán. Ngoài ra, họ lắp đặt sáu cảm biến hồng ngoại dưới dạng lưới 400mm trên sàn phòng thí nghiệm để đo lường những thay đổi trong hoạt động và đặt hai camera Microsoft LifeCam Cinema được đặt ở độ cao 1820 mm so với mặt sàn để thu thập hình ảnh của đối tượng theo góc nhìn ngang và góc nhìn chính diện. Tất cả các loại cảm biến hoạt động đồng thời để thu thập dữ liệu cho từng loại hoạt động. Tần số lấy mẫu tiêu chuẩn cho tất cả các mẫu là 100Hz.

Bảng 2.4. Hoạt động thường ngày và ngã trong bộ dữ liệu UP-Fall [12].

Nhãn		Thời gian	Số mẫu [*]	Mô tả	Loại hoạt động
A01	FH	10	152	Ngã về phía trước dùng tay để chống	Các sự kiện ngã
A02	FF	10	159	Ngã về phía trước sử dụng đầu gối	
A03	FB	10	149	Ngã về phía sau	
A04	FS	10	146	Ngã sang một bên	
A05	FE	10	144	Ngã ngồi (ngồi vào ghế trống)	
A06	W	60	1954	Đi bộ	Hoạt động bình thường (hàng ngày)
A07	ST	60	1922	Đứng	
A08	SI	60	1915	Ngồi	
A09	P	10	130	Nhặt đồ vật	
A10	J	30	826	Nhảy	
A11	L	60	1872	Nằm	

[*]: Đây là số mẫu dữ liệu sử dụng trong luận án, nó được thu thập bằng cảm biến gia tốc của thiết bị đeo để bên trong túi quần bên phải. NCS không thống kê số mẫu của dữ liệu thu bằng các loại cảm biến khác.

Luận án này chỉ sử dụng dữ liệu từ gia tốc kế 3 trục của thiết bị IMU đặt trong túi quần bên phải của các tình nguyện viên với tần số lấy mẫu giữ nguyên 100Hz.

2.4. Tiền xử lý dữ liệu

2.4.1. Kỹ thuật lọc nhiễu

Tín hiệu cảm biến của điện thoại thông minh thường gặp nhiều yếu tố gây nhiễu, đòi hỏi sử dụng các kỹ thuật lọc phù hợp để làm sạch và làm mịn. Tín hiệu từ cảm biến gia tốc mô tả hoạt động con người bao gồm chuyển động, trọng lực và nhiễu [93]. Để

xây dựng mô hình nhận dạng hoạt động và phát hiện ngã hiệu quả thì phải tách được các thành phần này [94]. Một kỹ thuật thường được sử dụng là lọc dữ liệu để loại bỏ nhiễu. Tuy nhiên, nếu áp dụng kỹ thuật lọc không phù hợp có thể làm mất thông tin hữu ích của dữ liệu.

Các bộ lọc được đặt tên theo phạm vi tần số của các tín hiệu mà chúng cho phép đi qua, đồng thời chặn hoặc làm suy giảm các tín hiệu còn lại. Ba loại bộ lọc phổ biến là (i) bộ lọc thông thấp (low-pass filter), (ii) bộ lọc thông cao (high-pass filter) và (iii) bộ lọc thông dải (band-pass filter).

- (i) *Bộ lọc thông thấp*: Chỉ cho phép tín hiệu có tần số nhỏ hơn tần số cắt (f_c) đi qua và chặn mọi tín hiệu có tần số cao hơn f_c . Hầu hết năng lượng của tín hiệu gia tốc do chuyển động của con người tạo ra nằm trong dải tần số dưới 15Hz [80, 81]. Vì vậy, các nghiên cứu về nhận dạng hoạt động của con người sử dụng bộ lọc thông thấp thường chọn tần số cắt từ 15Hz đến 20Hz [96–98].
- (ii) *Bộ lọc thông cao*: Chỉ cho phép tín hiệu có tần số lớn hơn tần số cắt (f_c) đi qua, đồng thời chặn mọi tín hiệu có tần số thấp hơn f_c . Bộ lọc thông cao thường được dùng để tách tín hiệu gia tốc bị lẫn với thành phần lực trọng trường và nhiễu tần số thấp do chuyển động của cơ hoặc nhịp tim có tần số cắt nằm trong khoảng từ 0.1 đến 0.5Hz [94, 98–100].
- (iii) *Bộ lọc thông dải*: Cho phép các tín hiệu nằm trong một băng tần nhất định nằm giữa hai điểm tần số cắt đi qua, đồng thời chặn các tín hiệu có tần số nằm ngoài dải tần đó. Phần lớn các nghiên cứu về nhận dạng hoạt động và phát hiện sự kiện ngã sử dụng bộ lọc thông thấp để làm sạch dữ liệu thu từ cảm biến gia tốc. Trong luận án này, NCS sử dụng bộ lọc thông dải để lọc bỏ nhiễu tần số cao và tần số thấp như trình bày ở trên. Bộ lọc thông dải là sự kết hợp của bộ lọc thông thấp và bộ lọc thông cao. Băng thông của bộ lọc thông dải được giới hạn bởi tần số cắt $f_H = 0.25\text{Hz}$ và $f_L = 15\text{Hz}$. Trong đó, f_H là tần số cắt của bộ lọc thông cao và f_L là tần số cắt của bộ lọc thông thấp.

Các bộ lọc số được sử dụng có cấu trúc mô tả bởi hàm truyền tổng quát như sau:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} \dots + b_nz^{-n}}{1 + a_1z^{-1} + a_2z^{-2} \dots + a_nz^{-n}} \quad (2.3)$$

trong đó, n là bậc bộ lọc, a_i và b_i là các hệ số xác định đặc tính của bộ lọc, z^{-1} đặc trưng cho độ trễ đơn vị của mẫu dữ liệu [101] tương ứng với chu kỳ lấy mẫu dữ liệu.

Hàm truyền của bộ lọc thông thấp bậc một phổ biến được biểu diễn bằng:

$$H(z) = \frac{Y(z)}{X(z)} = 1 + \alpha z^{-1} \quad (2.4)$$

trong đó, hệ số α dùng để xác định tần số cắt của bộ lọc. Bộ lọc bậc cao hơn thường được sử dụng để nâng cao khả năng lọc nhiễu và lực trọng trường. Tuy nhiên, độ phức tạp tính toán sẽ tăng lên theo bậc của bộ lọc. Để giảm bớt độ phức tạp, NCS sử dụng bộ lọc bậc hai để làm sạch dữ liệu, nó có hàm truyền [101] như Biểu thức (2.5).

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \quad (2.5)$$

Quá trình lọc được biểu diễn theo biểu thức sau:

$$y[i] = b_0x[i] + b_1x[i-1] + b_2x[i-2] - a_1y[i-1] - a_2y[i-2] \quad (2.6)$$

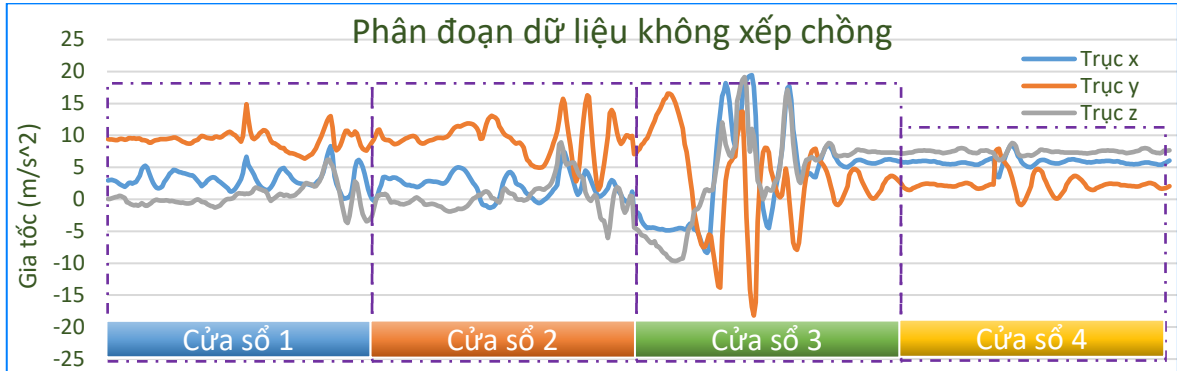
trong đó: $x[i]$ là giá trị mẫu trước khi lọc tại thời điểm i ; $y[i]$ là giá trị mẫu sau khi lọc tại thời điểm i và các giá trị mẫu này có thể được thu thập từ các cảm biến khác nhau như gia tốc kế, từ kế hay con quay hồi chuyển. Luận án này sử dụng bộ lọc thông dải dựa trên cấu trúc của bộ lọc bậc hai để chọn một dải tần như trên, điều này giúp cho việc nâng cao hiệu quả của các tham số Hjorth và các đặc trưng mang tính định hướng.

2.4.2. Phân đoạn dữ liệu

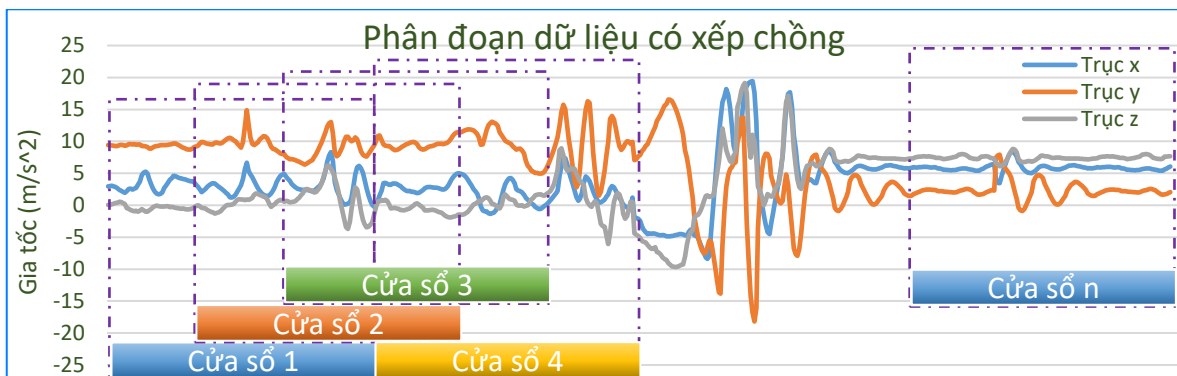
2.4.2.1. Cửa sổ dữ liệu

Có hai phương pháp chính để thiết lập cửa sổ dữ liệu: (i) phân chia theo kích thước cố định và (ii) phân chia theo kích thước động. Phương pháp phân chia theo kích thước động bao gồm cửa sổ xác định hoạt động và cửa sổ xác định sự kiện. Kích

thước cửa sổ cố định chia thành hai loại: cửa sổ trượt không chồng chéo (nOSW) và cửa sổ trượt có chồng chéo (OSW) [102, 103] như mô tả ở Hình 2.3 và Hình 2.4.



Hình 2.3. Phân đoạn dữ liệu bằng kỹ thuật cửa sổ không xếp chồng dữ liệu (nOSW)



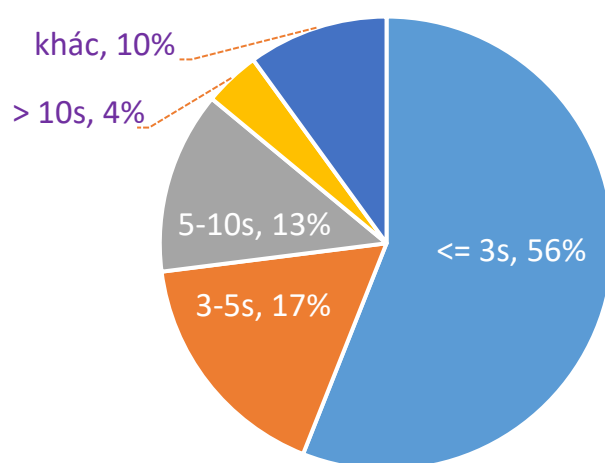
Hình 2.4. Phân đoạn dữ liệu bằng kỹ thuật cửa sổ xếp chồng dữ liệu (OSW)

Cửa sổ xác định hoạt động bắt đầu và kết thúc dựa trên mẫu thay đổi hoạt động, trong khi cửa sổ xác định sự kiện tạo ra quanh một sự kiện cụ thể. Dữ liệu được phân thành các đoạn có kích thước nhất định gọi là cửa sổ dữ liệu hoặc cửa sổ trượt, các cửa sổ có thể chồng lên nhau như Hình 2.4. Trong HAR, cửa sổ trượt là phương pháp phân đoạn dữ liệu phổ biến, đặc biệt là cho các hoạt động có chu kỳ và tính [102].

2.4.2.2. Kích thước cửa sổ dữ liệu (DWS)

Dữ liệu hoạt động của người dùng được ghi lại bằng cảm biến gia tốc theo các trục x, y, z và liên tục theo thời gian. Để nhận dạng hoạt động chính xác, chúng ta cần phân tích đặc điểm của dữ liệu trong khoảng thời gian diễn ra hoạt động. Dữ liệu từ cảm biến được chia thành nhiều đoạn nhỏ bằng kỹ thuật cửa sổ, mỗi kích thước cửa sổ

(DWS) có độ dài cố định. Đặc trưng của mỗi hoạt động được trích xuất từ từng cửa sổ dữ liệu. DWS ảnh hưởng đến độ phức tạp tính toán và hiệu suất nhận dạng. DWS quá ngắn sẽ thiếu thông tin về các hoạt động, ngược lại DWS quá lớn sẽ chứa nhiều hoạt động trong cùng một cửa sổ [22]. Nghiên cứu của Saha và cộng sự [22] cho thấy DWS khoảng 1-2 giây thường mang lại hiệu suất phân loại tốt nhất. Ferrari và cộng sự đã khảo sát mức độ phổ biến của DWS trong nghiên cứu về HAR và kết quả cho thấy hơn 50% công trình nghiên cứu sử dụng $DWS \leq 3$ giây (Hình 2.5).



Hình 2.5. Các DWS phổ biến trong HAR [7]

2.4.2.3. Cửa sổ trượt và xếp chồng dữ liệu

Tùy thuộc vào cách thu thập và loại dữ liệu mà các nghiên cứu lựa chọn giữa cửa sổ cố định hay không cố định để thiết kế hệ thống. Luận án này chọn phương pháp phân đoạn dữ liệu thành các cửa sổ có kích thước cố định, nó thường chia thành hai loại: không chồng chéo dữ liệu (nOSW) và có chồng chéo dữ liệu (OSW) như mô tả trong Hình 2.3 và Hình 2.4 [102–104].

Hiệu suất của FDS dựa trên phương pháp nOSW thường tốt hơn khi thời gian diễn ra các hoạt động có sự tương đồng. Ngược lại, nếu độ dài và tính chất của các hoạt động không đồng đều và không rõ ràng, thì phương pháp OSW sẽ hiệu quả hơn nOSW [90]. Đối với sự kiện như ngã, chúng xảy ra nhanh, bất ngờ và khó dự đoán. Thời gian diễn ra của mỗi sự kiện ngã có thể khác nhau tùy thuộc vào mỗi người. Do đó, phương pháp OSW để phân đoạn dữ liệu sẽ hiệu quả hơn phương pháp nOSW.

2.5. Trích xuất đặc trưng

Trích xuất đặc trưng là quá trình biến đổi không gian đặc trưng ban đầu sang một không gian khác, nói cách khác, đây là quá trình xây dựng một tập đặc trưng mới từ tập đặc trưng ban đầu. Mục đích của việc trích xuất đặc trưng là xác định các bộ đặc trưng tinh gọn hơn từ dữ liệu cảm biến đầu vào, nhằm giảm thiểu lỗi phân loại, giảm độ phức tạp tính toán và cải thiện hiệu suất của hệ thống nhận dạng.

Giả sử một phân đoạn dữ liệu cảm biến được biểu diễn dưới dạng $s = \{s_1, s_2, \dots, s_n\}$ [7]. Một đặc trưng được trích xuất từ phân đoạn này có thể được biểu diễn dưới dạng $y = f(s) \{y_1, y_2, \dots, y_m\}$. Trong đó, f là một hàm ánh xạ (hàm đặc trưng) biến đổi dữ liệu s từ không gian n chiều sang không gian m chiều (m thường nhỏ hơn n). Điều này mô tả việc trích xuất đặc trưng giúp giảm kích thước của không gian dữ liệu thô.

Để quá trình nhận dạng hoạt động đạt hiệu quả, hàm đặc trưng $f(s)$ cần được lựa chọn sao cho dữ liệu gốc được ánh xạ vào các vùng riêng biệt trong không gian đặc trưng mới. Nói cách khác, các đặc trưng được trích xuất phải mang tính đại diện cao, thể hiện rõ ràng các đặc điểm, tính chất đặc trưng của từng loại hoạt động.

Việc lựa chọn một tập đặc trưng phù hợp là rất quan trọng, đảm bảo dữ liệu mới vừa toàn vẹn vừa có khả năng phân biệt tốt giữa các hoạt động khác nhau. Trong các phương pháp và kỹ thuật xử lý tín hiệu cảm biến hiện nay, các đặc trưng thường được lựa chọn dựa trên các tính chất thời gian, tần số và rời rạc hóa của dữ liệu [90, 91].

2.5.1. Trong miền thời gian

Trong miền thời gian, có nhiều đặc trưng được sử dụng để phân loại hoạt động người dùng từ dữ liệu cảm biến gia tốc. Bảng 2.5 tóm tắt và thống kê những đặc trưng được các chuyên gia đánh giá là quan trọng và có ảnh hưởng lớn đến khả năng phân loại hoạt động [18].

Bảng 2.5. Các đặc trưng cơ bản trong miền thời gian.

Tên đặc trưng	Mô tả	Công thức tính	Mã
Tối đa	Giá trị tối đa của một phân đoạn dữ liệu trên mỗi trục	$\max_{j=1, \dots, n}(s_j)$	(2.7)
Tối thiểu	Giá trị tối thiểu của một phân đoạn dữ liệu trên mỗi trục	$\min_{j=1, \dots, n}(s_j)$	(2.8)
Phạm vi	Phạm vi của một phân đoạn dữ liệu trên mỗi trục	$R = \max - \min$	(2.9)
Độ lệch mẫu	giá trị độ lệch giữa các mẫu cực đại và cực tiểu trong một phân đoạn dữ liệu trên mỗi trục	$\Delta s = \max(s) - \min(s)$	(2.10)
Trung bình	Giá trị trung bình trên một cửa sổ mẫu dữ liệu trên mỗi trục	$\bar{s} = \frac{1}{n-1} \sum_{i=1}^n s_i$	(2.11)
Độ lệch tuyệt đối trung bình	Giá trị trung bình trên một cửa sổ mẫu dữ liệu trên mỗi trục	$\Delta_{MDA} = \sqrt{\frac{1}{1-n} \sum_{i=1}^n s_i - \bar{s} }$	(2.12)
Phương sai	trung bình độ lệch khỏi giá trị trung bình pha phương	$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (s_i - \bar{s})^2$	(2.13)
Độ lệch chuẩn	khả năng thay đổi của một bộ dữ liệu và phân bố xác suất của nó	$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (s_i - \bar{s})^2}$	(2.14)
Độ hiệu dụng	Giá trị trung bình pha phương gốc của một phân đoạn dữ liệu trên mỗi trục	$S_{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n s_i^2}$	(2.15)
Tỉ lệ đỉnh trên trung bình	Giá trị của biên độ tín hiệu lớn nhất so với giá trị trung bình của một phân đoạn dữ liệu trên mỗi trục	$PAR = \frac{\max(s)}{\bar{s}}$	(2.16)
Hệ số tương quan mẫu	tỉ lệ của hiệp phương sai giữa các tín hiệu dọc theo 2 trục (ví dụ trục x và y)	$\rho_{x,y} = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$	(2.17)
Tương quan chéo	phép đo mức độ tương tự giữa hai tín hiệu	$R_{x,y} = \max_{d=1}^{n-1} \left(\sum_{i=1}^n x_i \cdot y_{i-d} \right)$	(2.18)
Độ tương quan	tính theo các cặp trục tọa độ (x,y), (x,z) hay (y,z) hoặc giữa hai thành phần gia tốc dọc và ngang	$R = \frac{\sum_{i=1}^n (s_x^i - \bar{s}_x)(s_y^i - \bar{s}_y)}{\sqrt{\sum_{i=1}^n (s_x^i - \bar{s}_x)^2 \sum_{i=1}^n (s_y^i - \bar{s}_y)^2}}$	(2.19)
Điểm cắt không	điểm mà phân nửa một tín hiệu đi qua	$D = \sum_{i=1}^N \left \frac{\text{sgn}(s_p [i]) - \text{sgn}(s_p [i-1])}{2} \right $	(2.20)

Tên đặc trưng	Mô tả	Công thức tính	Mã
Diện tích độ lớn tín hiệu	tổng diện tích được bao bởi độ lớn của mỗi tín hiệu gia tốc ở 3 trục	$SMA = \frac{1}{t} \left[\sum_{k=x}^z \int_0^t s_k(t) dt \right]$	(2.21)
Năng lượng hoạt động	năng lượng của trạng thái hoạt động theo hai thành phần gia tốc dọc và ngang	$SMA_k = \frac{1}{2} \sum_{i=2}^N (s_k [i-1] + s_k [i]) * (t [i] - t [i-1])$	(2.22)
Tổng độ lớn vectơ tín hiệu	Tổng của các thành phần dọc theo các trục x, y và z	$SVM = \frac{1}{n} \sum_{i=1}^n \sqrt{x_i^2 + y_i^2 + z_i^2}$	(2.23)
Tổng độ lớn vectơ tín hiệu vi sai	Sự khác biệt về tổng độ lớn của các thành phần dọc theo các trục x, y và z	$DSVM = \frac{1}{t} \left(\int_0^t (\sum SVM') dt \right)$	(2.24)
Giá trị tích lũy	Độ tích lũy của các giá trị trong thành phần góc quay thay đổi theo thời gian	$I_g = \sum_{i=1}^{n-1} I[i]$	(2.25)

Ghi chú: n – số mẫu dữ liệu trong phân đoạn cửa sổ; M – tốc độ lấy mẫu

Trong số những đặc trưng được tổng hợp ở Bảng 2.5, giá trị trung bình (\bar{s}) được sử dụng như một đặc trưng độc lập hoặc làm cơ sở để trích rút các đặc trưng khác. Đặc trưng này được tính toán nhanh và yêu cầu ít bộ nhớ. Phương sai (σ^2) và độ lệch chuẩn (σ) là các đặc trưng thể hiện sự phân tán của tín hiệu quanh giá trị trung bình. Những đặc trưng này có khả năng phát hiện các hoạt động có biên độ lớn như đi bộ, chạy bộ hay nhảy, đồng thời cũng cung cấp thông tin về độ ổn định của tín hiệu từ cảm biến gia tốc, giúp phát hiện các sự kiện ngã trong chuyển động của người dùng.

Độ hiệu dụng (S_{RMS}) là đặc trưng thể hiện tính chất của một chuỗi giá trị, giúp phân biệt giữa các hoạt động bình thường với sự kiện ngã. Tỷ lệ đỉnh trên trung bình (PAR) là đặc trưng thể hiện sự chênh lệch giữa giá trị cao nhất và giá trị trung bình của tín hiệu, giúp nhận dạng các hoạt động ngã có biên độ lớn. Độ tương quan (R) là đặc trưng thể hiện mối quan hệ giữa hai tín hiệu khác nhau và có thể sử dụng để phát hiện các hoạt động có thay đổi theo hướng nhất định, như đi bộ, chạy, hoặc có thay đổi theo nhiều hướng như lên, xuống cầu thang.

Diện tích độ lớn tín hiệu (SMA) là đặc trưng thể hiện chi phí năng lượng trong các hoạt động thường ngày. Đặc trưng này có khả năng phân biệt giữa trạng thái nghỉ và hoạt động của người dùng, đồng thời phản ánh năng lượng của trạng thái hoạt động theo hai thành phần gia tốc dọc và ngang. Hai thành phần năng lượng này có thể sử dụng để phân biệt các hoạt động có sự khác biệt giữa chúng, như hoạt động nhảy so với các hoạt động khác.

2.5.2. Trong miền tần số

Với tín hiệu gia tốc, việc trích rút đặc trưng trong miền tần số thường được thực hiện thông qua phép biến đổi Fourier. Phép biến đổi này cho phép xác định cách tín hiệu thay đổi theo thời gian dưới dạng một chuỗi các thành phần tần số.

Kỹ thuật trích xuất đặc trưng trong miền tần số được sử dụng để xác định bản chất lặp lại theo chu kỳ của tín hiệu, chẳng hạn như khi đi bộ, chạy, nhảy. Các đặc trưng quan trọng khi trích xuất trong miền tần số được tóm tắt và thống kê trong Bảng 2.6.

Bảng 2.6. Các đặc trưng cơ bản trong miền tần số.

Tên đặc trưng	Mô tả	Công thức tính	Mã
Entropy thông tin	Entropy thông tin chuẩn hóa của các thành phần FFT rời rạc	$H = - \sum_{i=1}^n p(s_i) \log_2 p(s_i)$	(2.26)
Khai triển Fourier nhanh	Khai triển Fourier rời rạc trên tập N mẫu dữ liệu	$\text{FFT}[k] = \sum_{m=0}^{n-1} x[m] \cdot w[m] \cdot \exp(-j(2\pi/n) \cdot k \cdot m)$	(2.27)
Năng lượng phổ	Tổng bình phương các hệ số phổ được chuẩn hóa theo chiều dài của số mẫu.	$E_s = \sum_{k=1}^M X[k] ^2$	(2.28)
Năng lượng phổ trung bình	Được tính bằng cách lấy trung bình cộng của tất cả các giá trị năng lượng trong phổ tín hiệu.	$\bar{E} = \frac{2 \sum_{k=2}^{n/2} X[k] ^2}{n}$	(2.29)
Trung bình phổ	Giá trị trung bình của phân phối FFT.	$\mu_f = \frac{1}{n} \sum_{k=1}^M \text{FFT}_j$	(2.30)

Phép biến đổi Fourier là công cụ chính được sử dụng trong việc trích xuất đặc trưng từ miền tần số. Phép biến đổi này không chỉ giúp biểu diễn các đặc tính cốt lõi của phổ tín hiệu (như thành phần trung bình hay các thành phần tần số quan trọng) mà còn cho phép phân tích cụ thể các chu kỳ hoặc các phần lặp lại trong tín hiệu. Điều này được thực hiện bằng cách sử dụng kỹ thuật khai triển Fourier rời rạc thông qua các giải thuật như khai triển Fourier nhanh (FFT) và khai triển tần số thời gian nhanh (STFT), áp dụng cho các phân đoạn tín hiệu trong một khoảng thời gian nhất định.

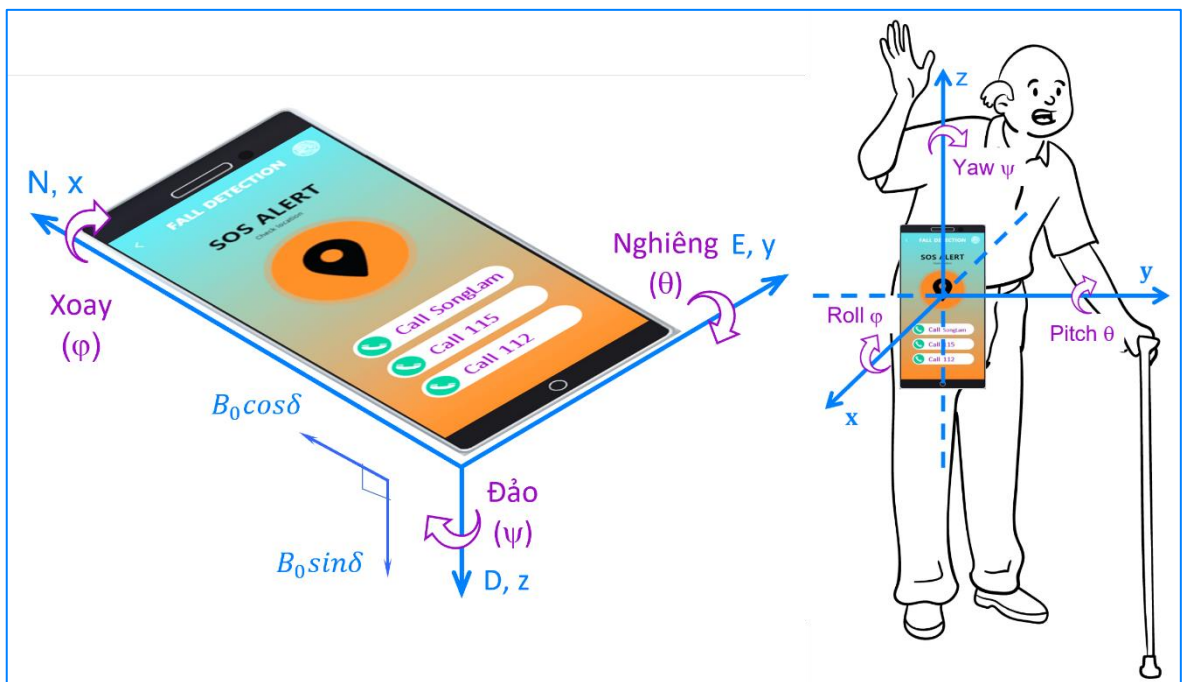
Các thành phần tần số quan trọng là các giá trị khác 0 trong miền tần số, biểu thị các chu kỳ cơ bản hoặc các khoảng lặp lại của tín hiệu. Các giá trị này cho biết tần số và biên độ của các chu kỳ lặp lại, tức là số lần và mức độ lặp lại trong một đơn vị thời gian. Ví dụ, nếu tín hiệu gia tốc kế có một thành phần tần số quan trọng ở mức 2Hz và biên độ 0.5 m/s^2 , có nghĩa là tín hiệu lặp lại 2 lần trong một giây và dao động 0.5 m/s^2 quanh giá trị trung bình, có thể do người dùng đang đi bộ.

Ngoài ra, có những đại lượng khác được sử dụng để mô tả phổ của tín hiệu, bao gồm năng lượng phổ, Entropy thông tin, và các đại lượng liên quan đến độ lớn và pha của các hệ số phổ. Năng lượng phổ đo lường năng lượng của tín hiệu trong miền tần số, trong khi Entropy thông tin giúp phân biệt các tín hiệu có cùng năng lượng nhưng thuộc các hoạt động khác nhau. Độ lớn và pha của các hệ số Fourier cũng có thể được sử dụng để mô tả các đặc điểm của tín hiệu như độ lớn vector tín hiệu (SVM) và độ lớn vector sai số tín hiệu (DSVM).

Đại lượng Entropy được tính bằng cách sử dụng độ lớn của các hệ số Fourier rời rạc sau khi loại bỏ thành phần DC và sử dụng định nghĩa về Entropy thông tin chuẩn hóa. Entropy giúp phân biệt các tín hiệu có cùng mức năng lượng nhưng thể hiện các mẫu hoạt động khác nhau, dù chúng có cùng giá trị năng lượng. Đại lượng này hỗ trợ phân biệt các tín hiệu theo bản chất hoạt động mà chúng thể hiện.

2.5.3. Tham số định hướng

Gia tốc kế trên điện thoại thông minh (SP) phát hiện gia tốc của người dùng theo 3 trục x , y , z . Các hoạt động của người dùng như lắc lư, đứng lên, ngồi xuống, đi bộ, chạy bộ, ngồi xuống đột ngột hoặc ngã sẽ làm thay đổi các giá trị gia tốc theo các trục này. Khi người dùng lắc lư sang trái hoặc sang phải các giá trị gia tốc theo trục x sẽ thay đổi. Các động tác ngồi xuống hoặc đứng lên của người dùng làm ảnh hưởng đến các giá trị đo được trên trục y . Các hoạt động như đi bộ, chạy bộ, sau đó đột ngột ngồi xuống hoặc ngã tạo biến đổi cho các đại lượng trên trục z . Các thay đổi về định hướng được mô tả thông qua các góc xoay (roll), góc nghiêng (pitch), và góc đảo (yaw), tương ứng với các trục x , y và z của điện thoại như thể hiện ở Hình 2.6.



Hình 2.6. Hệ trục tọa độ biểu diễn các trục quay của SP trong không gian

Sự quay này thường được đo bởi cảm biến con quay hồi chuyển. Tuy nhiên, việc kích hoạt thêm cảm biến con quay hồi chuyển để thu thập dữ liệu làm tăng nhu cầu tiêu thụ năng lượng của thiết bị. Góc định hướng của SP có thể được ước tính gần đúng dựa trên dữ liệu gia tốc kế 3 trục. Do đó, NCS sử dụng phương pháp này để tính các góc định hướng của điện thoại và trích rút các đặc trưng cần thiết. Cụ thể, từ các kết

hợp của ma trận xoay và các giá trị cảm biến gia tốc đo được trên ba trục x, y và z. Góc xoay ($x-\varphi$) và góc nghiêng ($y-\theta$) có thể tính theo hàm arctan như biểu diễn ở Công thức (2.31) hoặc (2.32).

$$\varphi [i] = \tan^{-1} \left(\frac{s_y^i}{\sqrt{(s_x^i)^2 + (s_z^i)^2}} \right) \text{ và } \theta [i] = \tan^{-1} \left(\frac{-s_x^i}{s_z^i} \right) \quad (2.31)$$

hoặc

$$\varphi [i] = \tan^{-1} \left(\frac{s_y^i}{s_z^i} \right) \text{ và } \theta [i] = \tan^{-1} \left(\frac{-s_x^i}{\sqrt{(s_y^i)^2 + (s_z^i)^2}} \right) \quad (2.32)$$

Trong đó, giá trị đo được của cảm biến gia tốc trên trục z vẫn có lực trọng lực g. Tuy nhiên, trong thực tế người dùng có thể đặt điện thoại ở bất kỳ hướng nào. Do đó, có thể tính toán thêm góc đảo bằng công thức được mô tả trong Phương trình (2.33).

$$\varphi_z [i] = \varphi' [i] - \bar{\varphi}' = \tan^{-1} \left(\frac{s_z^i}{\sqrt{(s_x^i)^2 + (s_y^i)^2}} \right) \quad (2.33)$$

Sau khi đã tính toán các thành phần góc xoay, góc nghiêng và góc đảo, các đặc trưng sẽ được xác định tương tự như việc xử lý dữ liệu từ cảm biến gia tốc như Phương sai (s^2), giá trị trung bình (\bar{a}), tích phân (I g) và các tham số Hjorth [CT1].

Đây là cơ sở để NCS chọn lựa giải pháp chỉ sử dụng cảm biến gia tốc thay vì sử dụng đồng thời cảm biến gia tốc và con quay hồi chuyển như các nghiên cứu khác.

2.5.4. Tham số Hjorth

Tham số Hjorth [106] là một phương pháp mô tả đặc điểm thống kê của tín hiệu trong miền thời gian, thường được áp dụng trong việc phân tích các tín hiệu y sinh. Phương pháp này bao gồm ba thành phần chính là tính hoạt động (a), tính di động (M), và độ phức hợp (C) như tóm tắt và thống kê trong Bảng 2.7.

Tính hoạt động đo lường phương sai của tín hiệu theo thời gian, thường được sử dụng để đánh giá bề mặt phổ công suất trong miền thời gian. Nó biểu thị mức độ của các

thành phần tần số cao trong tín hiệu, tức là chu kỳ lặp lại nhanh của nó. Ví dụ, nếu tính hoạt động cao, có thể đo tín hiệu gia tốc kế có nhiều thành phần tần số cao, thường xuyên xuất hiện khi người dùng chạy hoặc nhảy.

Tính di động liên quan đến độ lệch chuẩn của phổ công suất, chỉ ra mức độ biến động của tần số. Tính di động biểu thị sự tuần hoàn hoặc không của tín hiệu theo thời gian. Ví dụ, nếu tín hiệu gia tốc kế có tính di động cao, có nghĩa là tín hiệu có nhiều sự thay đổi, có thể do người dùng đang thay đổi hoạt động. Tính di động càng cao, tín hiệu càng có nhiều biến thiên đột ngột.

Độ phức hợp cho biết tín hiệu thu được có giống với dạng sóng hình sin hay không. Ví dụ, nếu độ phức tạp gần bằng 1, tín hiệu có dạng gần giống sóng hình sin, có thể xuất hiện khi người dùng đi bộ. Mặc dù có thể tính toán trong miền thời gian, tham số Hjorth có thể phản ánh được đặc trưng của tín hiệu trên miền tần số.

Bảng 2.7. Các đặc trưng cơ bản trong miền tham số Hjorth.

Tên đặc trưng	Mô tả	Công thức tính	Mã
Tham số hoạt động	Phương sai của hàm thời gian, chỉ bề mặt phổ công suất trong miền thời gian.	$A = \frac{\sum_{i=1}^{n-1} d_{0i}^2}{n-1}$	(2.34)
Tham số di động	Căn bậc hai tỉ lệ của phương sai đạo hàm bậc một tín hiệu và phương sai của tín hiệu.	$M = \sqrt{\frac{\sum_{i=1}^{n-2} d_{1i}^2}{A \cdot (n-2)}}$	(2.35)
Tham số phức hợp	Mô tả sự tương đồng về hình dạng của một tín hiệu tương tự với sóng hình sin.	$C = \sqrt{\frac{\sum_{i=1}^{n-3} d_{2i}^2 \cdot (n-2)}{\sum_{i=1}^{n-2} d_{1i}^2 \cdot (n-3)}}$	(2.36)

2.6. Lựa chọn đặc trưng

Kích thước lớn của tập đặc trưng là một thách thức lớn đối với các nhà nghiên cứu, vì nó không chỉ làm tăng thời gian huấn luyện mô hình mà còn dễ dẫn đến hiện tượng quá khớp. Trong bối cảnh đó, giảm số chiều trở thành bước tiền xử lý quan trọng, đặc biệt trong các bài toán phức tạp như phân loại hoạt động con người, giúp giảm kích thước dữ liệu. Hai phương pháp chính được áp dụng để giảm số chiều là trích xuất và

lựa chọn đặc trưng, cả hai đều góp phần nâng cao hiệu suất của hệ thống phân loại hoạt động con người (HAR) bằng cách loại bỏ các đặc trưng không liên quan hoặc thừa, từ đó tăng độ chính xác của mô hình máy học.

Trong nỗ lực nâng cao khả năng phát hiện các sự kiện ngã, NCS đã thực hiện việc trích rút một lượng lớn đặc trưng bao gồm 43 đặc trưng trong miền thời gian, 15 đặc trưng trong miền tần số, 10 đặc trưng từ tham số định hướng và 18 đặc trưng từ tham số Hjorth. Qua quá trình thử nghiệm và đánh giá liên tục, NCS đã chọn ra 28 đặc trưng trong miền thời gian, 7 đặc trưng trong miền tần số, 6 đặc trưng từ tham số định hướng và 3 đặc trưng từ tham số Hjorth. Các đặc trưng này được chứng minh là có hiệu quả trong phân loại hoạt động và sự kiện ngã, phù hợp với các mô hình học máy cơ bản. Chi tiết về các đặc trưng được lựa chọn đã được tổng hợp trong Bảng 2.8.

Bảng 2.8. Tập các đặc trưng được lựa chọn để trích chọn thông tin từ dữ liệu cảm biến gia tốc cho hệ thống phát hiện sự kiện ngã (FDS)

Miền	Ký hiệu	Tên/định nghĩa	Công thức	Kiểu đặc trưng	Số đặc trưng
Thời gian (28 đặc trưng)	Δs	Độ lệch mẫu	(2.10)	s_x, s_y, s_z	3
	\bar{s}	Giá trị trung bình	(2.11)	s_x, s_y, s_z, s_{rms}	4
	σ^2	Phương sai	(2.13)	s_x, s_y, s_z	3
	σ	Độ lệch chuẩn	(2.14)	s_x, s_y, s_z	3
	PAR	Tỉ lệ đỉnh trên trung bình	(2.16)	s_x, s_y, s_z	3
	R	Độ tương quan	(2.19)	$(s_x, s_y), (s_x, s_z), (s_z, s_y)$	3
	D	Điểm cắt không	(2.20)	s_x, s_y, s_z	3
	SMA	Diện tích độ lớn tín hiệu	(2.21)	s_x, s_y, s_z, s_{rms}	4
	SVM	Độ lớn vectơ tín hiệu	(2.23)	s_{rms}	1
	DSVM	Độ lớn vectơ tín hiệu vi sai	(2.24)	s_{rms}	1
Tần số (7)	H	Entropy thông tin	(2.26)	s_x, s_y, s_z	3
	E_s	Năng lượng phổ	(2.28)	s_x, s_y, s_z, s_{rms}	4

Miền	Ký hiệu	Tên/định nghĩa	Công thức	Kiểu đặc trưng	Số đặc trưng
Định hướng (6)	\bar{s}_D	Tham số định hướng theo giá trị trung bình	(2.31) (2.32) (2.11)	φ, θ	2
	σ_D^2	Tham số định hướng theo phương sai	(2.31) (2.32) (2.13)	φ, θ	2
	I	Tích phân tích lũy	(2.31) (2.32) (2.25)	φ, θ	2
Tham số Hjorth (3)	A	Hoạt động	(2.34)	s_{rms}	1
	M	Di động	(2.35)	s_{rms}	1
	C	Phức hợp	(2.36)	s_{rms}	1

Ghi chú:

- s_x, s_y, s_z – đặc trưng được trích xuất trên trục x, y, z tương ứng.
- s_{rms} - đặc trưng được trích xuất theo giá trị hiệu dụng.
- φ, θ - đặc trưng được trích xuất theo góc xoay và góc nghiêng

2.7. Thực nghiệm và đánh giá

2.7.1. Mô tả thực nghiệm

Thực nghiệm 2.1: Đánh giá hiệu suất của các tập đặc trưng

Các đặc trưng trên miền thời gian ảnh hưởng nhiều đến phân loại các hoạt động, hoạt động của con người. Sau khi đánh giá kết quả phân lớp của từng tập đặc trưng riêng lẻ trên từng miền, NCS kết hợp các đặc trưng trong miền thời gian với các miền/tham số khác để đánh giá sự tác động của chúng.

Bảng 2.9 mô tả số lượng đặc trưng và sự kết hợp đặc trưng giữa các miền để tạo ra bộ đặc trưng mới. Dữ liệu thô của bộ MobiAct v2.0 và bộ UP-Fall được phân đoạn thành các cửa sổ có kích thước 256 mẫu và tỉ lệ chồng chéo 80%. Từ dữ liệu thô trích xuất thành 10 bộ đặc trưng để kiểm tra xem bộ đặc trưng nào mang lại hiệu suất phân loại tốt nhất. Bộ đặc trưng có hiệu năng tốt nhất sẽ được sử dụng để đánh giá mức độ ảnh hưởng của DWS và mức độ ảnh hưởng của DOR.

Bảng 2.9. Các bộ đặc trưng được sử dụng trong Thực nghiệm 2.1

Thực nghiệm	Bộ đặc trưng	Số đặc trưng	Miền trích xuất
(1)	TD	28	Thời gian
(2)	FD	7	Tần số
(3)	DD	6	Định hướng
(4)	HD	3	Hjorth
(5)	TFD	35	Thời gian + Tần số
(6)	TDD	34	Thời gian + Định hướng
(7)	THD	31	Thời gian + Hjorth
(8)	TFDD	41	Thời gian + Tần số + Định hướng
(9)	TFHD	38	Thời gian + Tần số + Hjorth
(10)	TFDHD	44	Thời gian + Tần số + Định hướng + Hjorth

Thực nghiệm 2.2: Đánh giá ảnh hưởng của kích thước cửa sổ dữ liệu (DWS)

Bộ đặc trưng tốt nhất thu được từ Thực nghiệm 2.1 được sử dụng cho thực nghiệm này. Tỷ lệ xếp chồng dữ liệu được thiết lập ở mức 80%. Sáu DWS (từ 32 mẫu (2^5) đến 1024 mẫu (2^{10})) được sử dụng để kiểm tra xem DWS nào là tốt nhất trên hai bộ dữ liệu MobiAct v2.0 và UP-Fall.

Thực nghiệm 2.3: Đánh giá mức độ ảnh hưởng của tỉ lệ xếp chồng dữ liệu (DOR)

Tương tự như Thực nghiệm 2.2, thực nghiệm này cũng sử dụng bộ đặc trưng tốt nhất thu được từ Thực nghiệm 2.1. Cửa sổ trượt được thiết lập ở kích thước 128 mẫu (tương đương 1,5 giây cho tập dữ liệu MobiAct và 1,28 giây cho tập dữ liệu UP-Fall). Chín tỉ lệ chồng chéo (từ 10% đến 90%) sẽ được thiết lập để kiểm tra xem DOR bằng bao nhiêu là tốt nhất trên hai bộ dữ liệu MobiAct v2.0 và UP-Fall.

Năm thuật toán phân loại được sử dụng phổ biến trong HAR là SVM, kNN, RF, ANN và J48, như như giới thiệu trong Mục 1.2.2 được sử dụng để thực hiện các thử nghiệm.

Trong mỗi thử nghiệm, các tham số mặc định của thuật toán phân loại được áp dụng dựa trên cấu hình mặc định của bộ công cụ Weka, và giá trị của chúng được trình bày trong Bảng 2.10. Quy trình xác thực chéo 10 lần cùng với thước đo F1-score được sử dụng để đánh giá hiệu suất.

Bảng 2.10. Bộ tham số mặc định của các mô hình phân lớp.

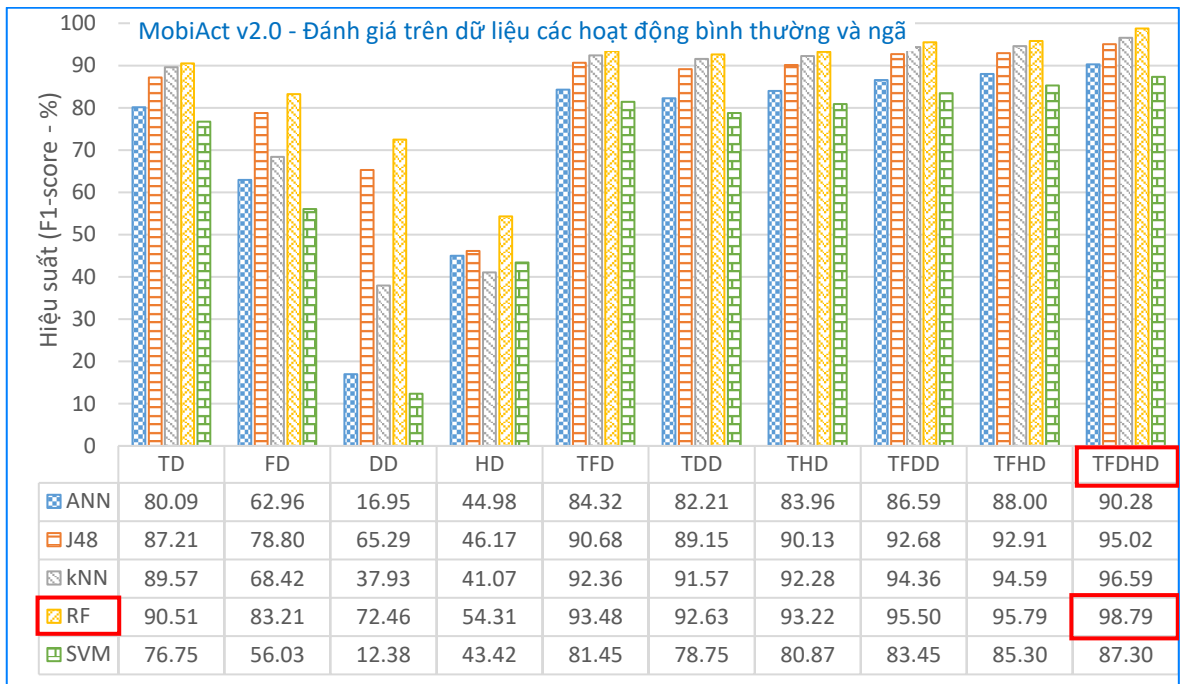
TT	Thuật toán	Tham số
1	RF	Số lượng cây trong rừng là 100; xác nhận chéo 10 lần; độ sâu tối đa của cây không giới hạn; số lượng đặc trưng tối đa được đặt bằng 0 (đồng nghĩa với việc các nút được mở rộng cho đến khi tất cả các lá đạt sự thuần khiết hoặc các lá có số mẫu nhỏ hơn <code>min_samples_split</code>); số lượng mẫu tối thiểu mà một nút cần giữ lại sau khi tách là 1; số lượng mẫu tối thiểu cần thiết để tách thành các nút con là $1e-3$.
2	J48	Ngưỡng tin cậy để cắt tỉa cây $C = 0.25$; số lượng bản sao tối thiểu $M = 2$
3	kNN	Số lượng láng giềng gần nhất $k = 1$; $W = 0$; (Mặc định: <code>weka.core.neighboursearch.LinearNNSearch</code>)
4	ANN	Learningrate $L = 0.3$; Momentum $M = 0.2$; Số lớp ẩn: 1; Số node ẩn: 40
5	SVM	Độ phức tạp ($C = 1.0$); Dung sai ($L = 0,001$); Epsilon cho lỗi làm tròn ($P = 1.0E-12$). Kernel: <code>weka.classifiers.functions.supportVector.PolyKernel - E1.0 C 250007</code> ;

2.7.2. Kết quả và bàn luận

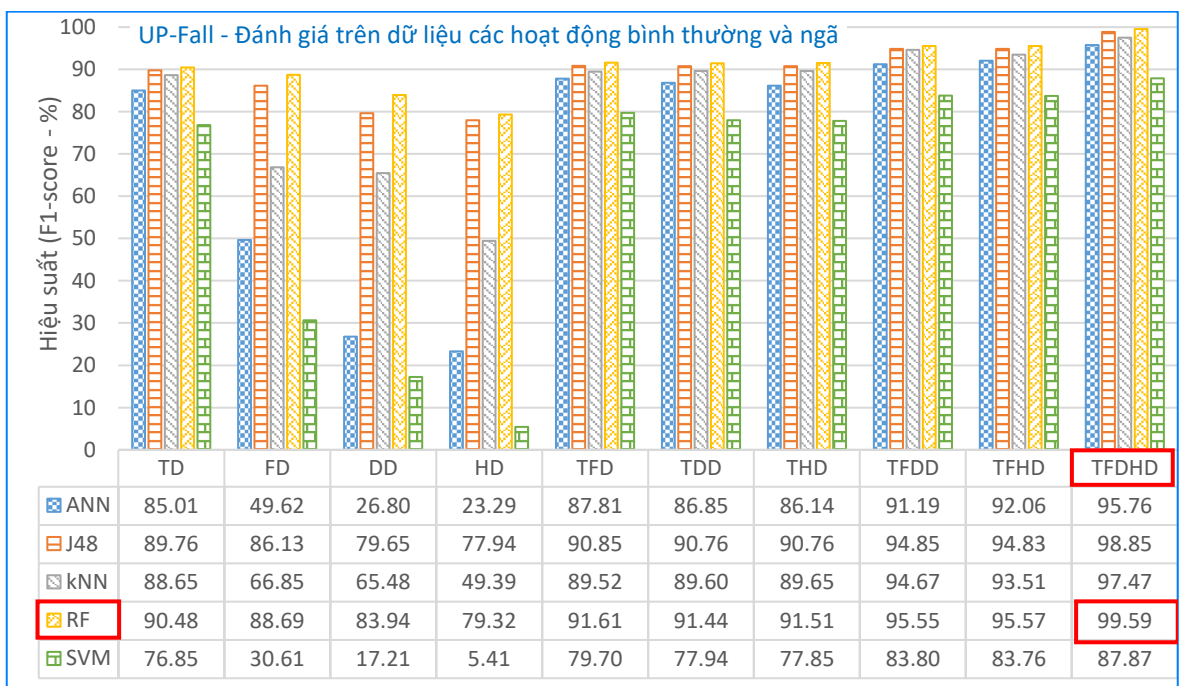
2.7.2.1. Ảnh hưởng của tập đặc trưng

a. Đánh giá trên các hoạt động (bao gồm các sự kiện ngã và hoạt động bình thường)

Dữ liệu thô trong hai bộ MobiAct v2.0 và UP-Fall được trích xuất thành 10 bộ đặc trưng như đã trình bày trong Bảng 2.9. Dữ liệu thô được phân đoạn ở DWS 128 mẫu và DOR 80%. Các kết quả thử nghiệm được trình bày trong Hình 2.7 và Hình 2.8.



Hình 2.7. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại tất cả các hoạt động trong bộ dữ liệu MobiAct (DWS 128 mẫu và DOR 80%).



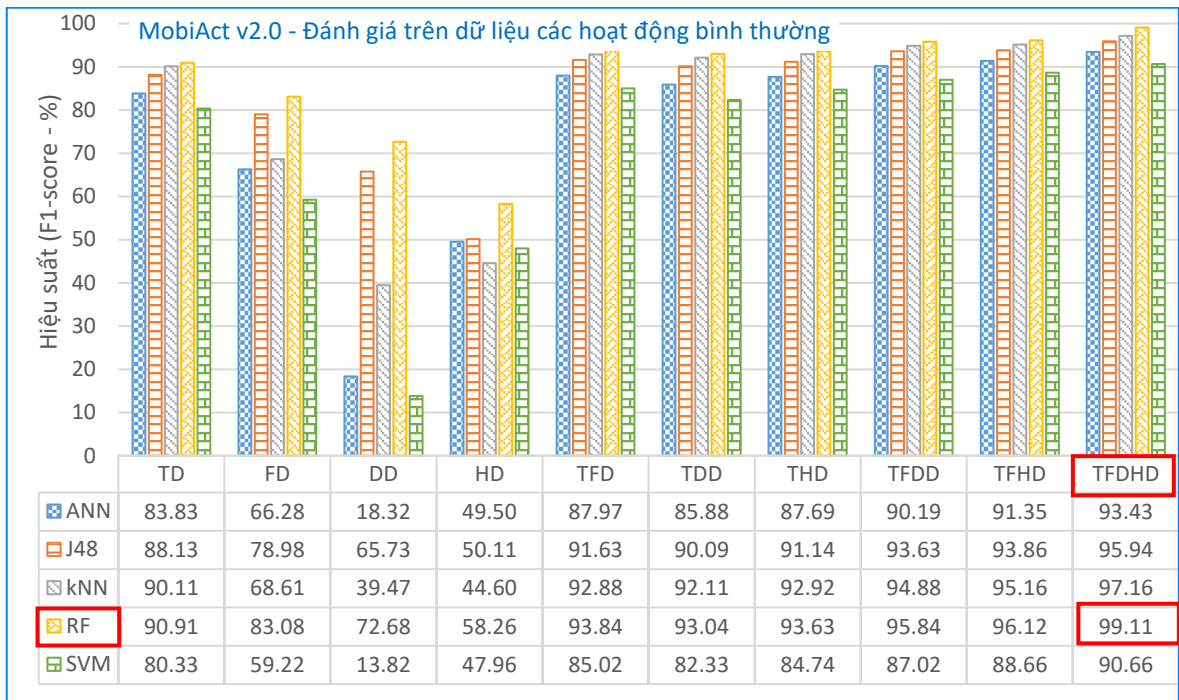
Hình 2.8. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại tất cả các hoạt động trong bộ dữ liệu UP-Fall (DWS 128 mẫu và DOR 80%).

Kết quả thực nghiệm trên hai bộ dữ liệu MobiAct v2.0 và UP-Fall cho thấy sự ảnh hưởng tương đồng của 10 bộ đặc trưng đến hiệu suất của năm bộ phân loại được áp dụng. Trong số đó, bộ đặc trưng trích xuất từ miền thời gian (TD) đem lại hiệu suất cao hơn so với đặc trưng từ miền tần số (FD) và miền tham số Hjorth (HD), trong khi đặc trưng dựa trên tham số định hướng (DD) cho kết quả kém nhất. Điều này là có thể giải thích bởi vì số lượng đặc trưng ở miền này chiếm đa số. Khi kết hợp đặc trưng TD với FD (TFD), TD với DD (TDD), và TD với HD (THD), hiệu suất phân loại của các thuật toán được cải thiện đáng kể. Trong số các kết hợp này, TFD có hiệu năng tốt hơn một ít so với THD và TDD nhưng không đáng kể.

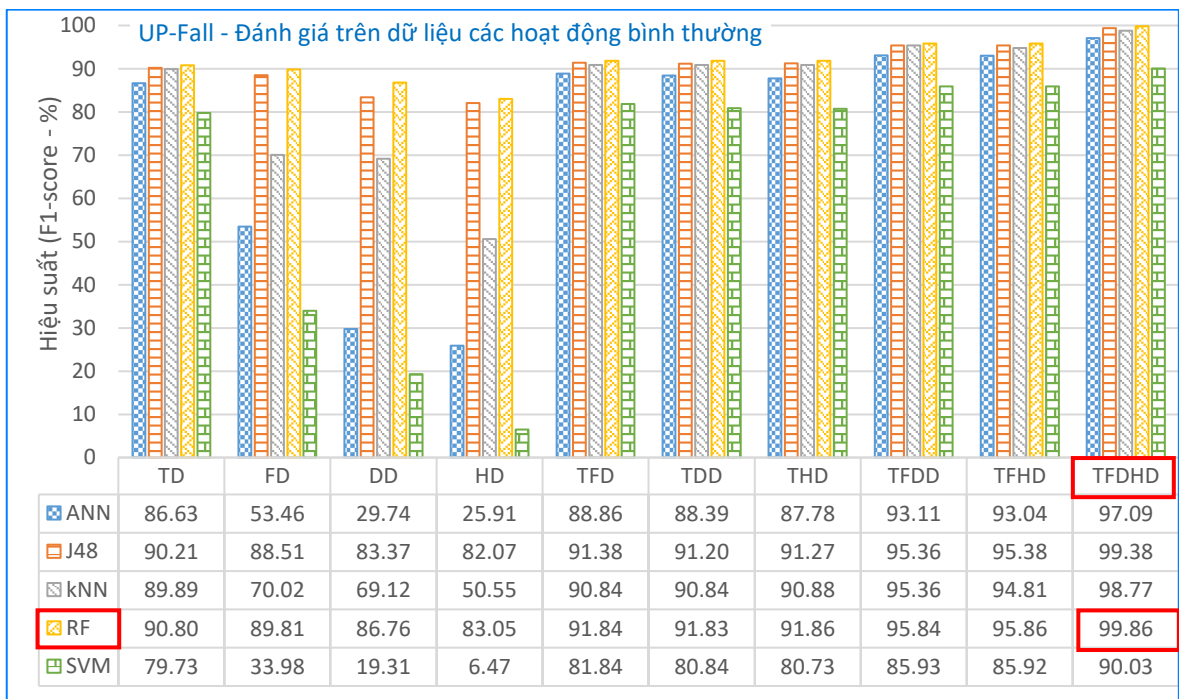
Sự kết hợp của tất cả các đặc trưng từ bốn miền, tức là TFDHD, đạt hiệu suất tốt nhất trong tất cả các thuật toán phân loại được thử nghiệm. Rõ ràng, việc bổ sung thêm các đặc trưng từ tham số định hướng và tham số Hjorth đã đóng góp vào việc nâng cao đáng kể khả năng phân loại của các thuật toán.

b. Đánh giá trên các hoạt động bình thường

Kết quả thực nghiệm cho thấy hiệu suất của các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại hoạt động bình thường, được trình bày trong Hình 2.9 và Hình 2.10 có sự tương đồng với hiệu suất khi đánh giá trên tất cả các hoạt động, bao gồm cả sự kiện ngã và hoạt động bình thường (Hình 2.7 và Hình 2.8). Điều này phản ánh đúng đặc điểm không cân bằng của các bộ dữ liệu được sử dụng, trong đó dữ liệu về hoạt động bình thường chiếm ưu thế so với dữ liệu về sự kiện ngã. Hiệu suất của các mô hình phân loại thường nghiêng về phía các lớp có số lượng mẫu lớn hơn.



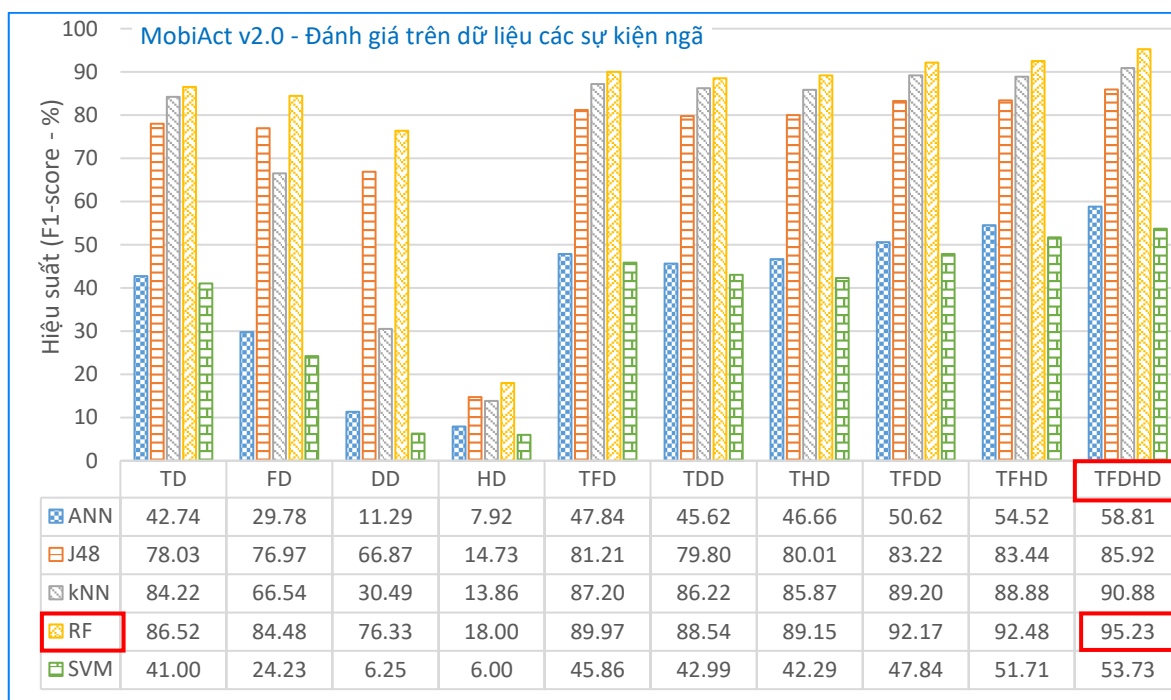
Hình 2.9. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại các hoạt động bình thường trong bộ dữ liệu MobiAct (DWS 128 mẫu và DOR 80%).



Hình 2.10. Hiệu suất các mô hình ML khi sử dụng 10 bộ đặc trưng để phân loại các hoạt động bình thường trong bộ dữ liệu UP-Fall (DWS 128 mẫu và DOR 80%).

c. *Đánh giá trên các sự kiện ngã*

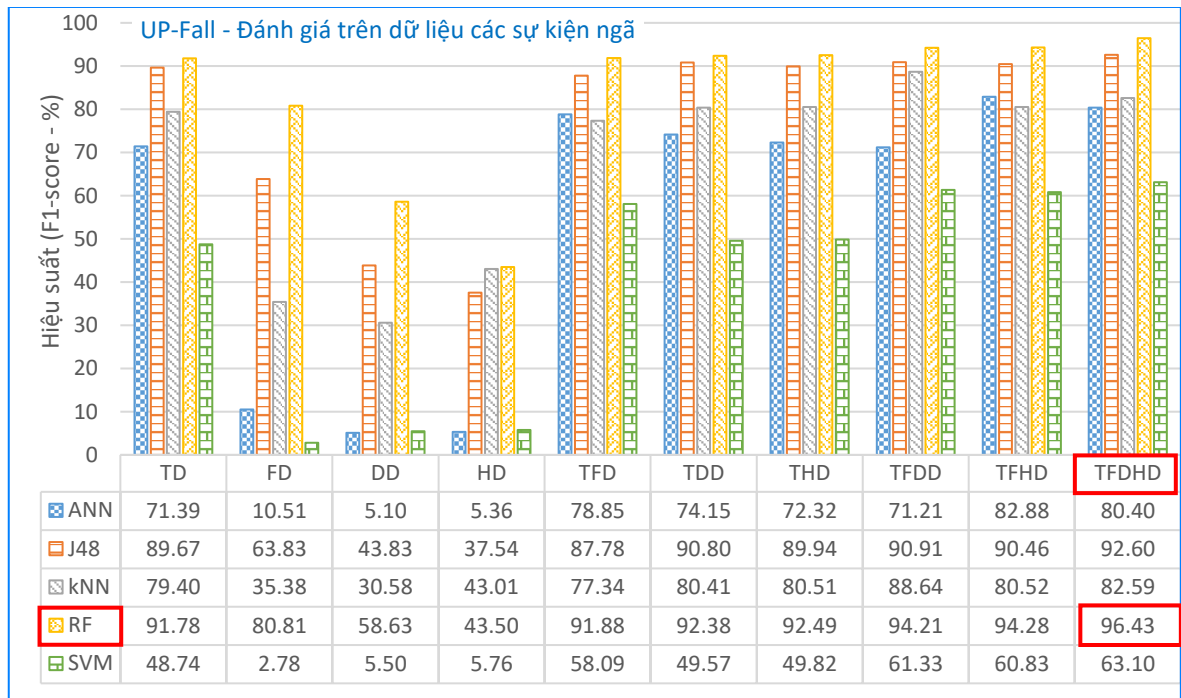
Kết quả thực nghiệm trình bày trong Hình 2.11 và Hình 2.12 cho thấy sự khác biệt đáng kể trong hiệu suất của các mô hình học máy (ML) khi sử dụng 10 bộ đặc trưng để phân loại các sự kiện ngã so với khi đánh giá trên tất cả các hoạt động, bao gồm cả sự kiện ngã và hoạt động bình thường (Hình 2.7 và Hình 2.8).



Hình 2.11. Hiệu suất các mô hình phân loại khi sử dụng 10 bộ đặc trưng để phân loại các sự kiện ngã trong bộ dữ liệu MobiAct (DWS 128 mẫu và DOR 80%).

Có thể nhận thấy hiệu suất của các mô hình sử dụng các bộ đặc trưng trích xuất từ bộ dữ liệu MobiAct được phân thành hai nhóm: nhóm có hiệu suất cao hơn bao gồm kNN, J48, và RF; và nhóm có hiệu suất thấp hơn gồm ANN và SVM. Đáng chú ý, sự chênh lệch hiệu suất giữa RF và SVM là rất lớn, với RF vượt trội hơn 70% ở bộ đặc trưng DD và hơn 78% ở bộ đặc trưng FD khi thử nghiệm trên MobiAct v2.0 và UP-Fall tương ứng. Khi sử dụng bộ đặc trưng TFDHD, hiệu suất của SVM được cải thiện đáng kể so với RF. Tuy nhiên, mức độ chênh lệch hiệu năng của hai thuật toán này vẫn rất lớn, với trên 33% ở cả hai bộ dữ liệu MobiAct v2.0 và UP-Fall.

Từ các kết quả trên, có thể nhận thấy bộ đặc trưng TFDHD luôn cho hiệu suất tốt nhất trong tất cả các bộ được thử nghiệm. Ngoài ra, thuật toán phân loại RF cũng thể hiện sự phù hợp với các bộ dữ liệu được sử dụng. Trong mọi trường hợp, RF luôn cho hiệu năng tốt nhất so với bốn thuật toán phân loại còn lại.



Hình 2.12. Hiệu suất các mô hình phân loại khi sử dụng 10 bộ đặc trưng để phân loại các sự kiện ngã trong bộ dữ liệu UP-Fall (DWS 128 mẫu và DOR 80%).

Bảng 2.11 cung cấp thông tin về hiệu suất phân loại của năm mô hình học máy khác nhau khi sử dụng bộ đặc trưng TFDHD trích xuất từ bộ dữ liệu MobiAct v2.0. Kết quả được đánh giá dựa trên F1-Score cho từng loại hoạt động riêng lẻ và tổng hợp hiệu suất trung bình cho các nhóm hoạt động khác nhau: các sự kiện ngã, hoạt động bình thường (ADL), và tổng hợp các hoạt động hỗn hợp. Dữ liệu gia tốc thô vẫn được phân đoạn ở DWS = 128 mẫu, DOR = 80%.

Nhìn chung, các hoạt động mang tính tuần hoàn như đi bộ, nhảy và các hoạt động tĩnh như đứng, ngồi được nhận dạng chính xác hơn so với các hoạt động có sự thay đổi trạng thái đột ngột như ngã, đứng lên từ ghế hoặc ngồi xuống ghế.

Bảng 2.11. Hiệu suất phân loại từng hoạt động của năm mô hình ML khi sử dụng bộ đặc trưng TFDHD trích rút từ các bộ dữ liệu MobiAct v2.0

Nhãn	Mô tả	Nhóm	Hiệu suất nhận dạng (F1-Score %)					
			ANN	J48	kNN	RF	SVM	
BSC	Ngã ngựa trong khi đang ngồi trên ghế	Các sự kiện ngã	58.50	86.23	91.90	94.92	52.60	
FKL	Ngã về phía trước khi đang đứng, đầu gối tiếp xúc đầu tiên.		54.77	85.64	89.55	95.58	49.66	
FOL	Ngã về phía trước khi đang đứng, sử dụng tay để chống.		53.80	85.00	89.25	95.28	45.80	
SDL	Ngã sang một bên khi đang đứng, gập chân		68.20	86.83	92.83	95.14	66.87	
Hiệu suất phân loại ngã tính theo trung bình trọng số			58.81	85.92	90.88	95.23	53.73	
CHU	Chuyển tiếp từ ngồi sang đứng (đứng lên từ ghế)	Hoạt động bình thường (ADL)	65.31	87.39	90.51	95.18	52.67	
CSI	Bước vào trong xe		63.29	78.35	87.50	94.75	56.97	
CSO	Bước ra khỏi xe hơi		85.96	94.44	96.01	98.33	84.26	
JOG	Chạy bộ		96.70	96.50	98.20	99.49	95.39	
JUM	Nhảy liên tục		98.50	98.20	99.38	99.91	97.51	
LYI	Nằm		81.06	96.71	98.32	98.84	64.76	
SCH	Chuyển tiếp từ đứng sang ngồi (ngồi trên ghế)		69.14	89.76	92.08	96.61	65.50	
SIT	Ngồi trên ghế với những cử động nhẹ nhàng		68.20	86.83	92.83	99.84	66.87	
STD	Đứng với cử động nhẹ nhàng		87.19	97.56	98.92	99.96	74.80	
STN	Đi xuống cầu thang (10 bậc)		99.12	99.59	99.51	95.02	98.73	
STU	Đi lên cầu thang (10 bậc)		76.94	82.57	86.74	95.50	70.36	
WAL	Đi bộ bình thường		75.95	83.10	87.42	99.62	63.46	
Hiệu suất phân loại các hoạt động bình thường tính theo trung bình trọng số			93.43	95.94	97.16	99.11	90.66	
Hiệu suất phân loại các hoạt động hỗn hợp (ngã và không ngã) tính theo trung bình trọng số			90.28	95.02	96.59	98.79	87.30	

Các sự kiện ngã như ngã ngựa trong khi đang ngồi trên ghế (BSC) và ngã về phía trước khi đang đứng, đầu gối tiếp xúc đầu tiên (FKL) có hiệu suất nhận dạng thấp hơn, đặc biệt là với thuật toán ANN và SVM. SVM là thuật toán có hiệu suất thấp

kém hơn với các thuật toán khác, với hiệu suất phân loại các hoạt động hỗn hợp (ngã và không ngã) tính theo trung bình trọng số là 87.30%. Mô hình RF luôn cho kết quả phân loại tốt nhất, phân loại các hoạt động hỗn hợp (ngã và không ngã) tính theo trung bình trọng số lên đến 97.78%, cao nhất trong tất cả các bộ phân loại.

Bảng 2.12 trình bày hiệu suất phân loại của năm thuật toán học máy khác nhau khi sử dụng bộ đặc trưng TFDHD trích xuất từ dữ liệu UP-Fall. Dữ liệu gia tốc thô vẫn được phân đoạn ở DWS =128 mẫu, DOR = 80%. Kết quả được biểu thị qua thước đo F1-Score cho từng loại hoạt động và được tổng hợp theo các nhóm sự kiện ngã, các hoạt động bình thường (ADL), và hoạt động tổng hợp gồm cả ngã và không ngã.

Bảng 2.12. Hiệu suất phân loại từng hoạt động của năm mô hình ML khi sử dụng bộ đặc trưng TFDHD trích rút từ các bộ dữ liệu UP-Fall.

Nhãn	Mô tả	Nhóm	Hiệu suất nhận dạng (F1-Score %)				
			ANN	J48	kNN	RF	SVM
FH	Ngã về phía trước dùng tay chống	Các sự kiện ngã	78.76	100	88.24	99.67	67.43
FF	Ngã về phía trước sử dụng đầu gối		80.47	98.73	84.21	99.68	40.00
FB	Ngã về phía sau		82.39	89.47	84.00	96.60	79.49
FS	Ngã sang một bên		77.18	86.51	76.66	91.15	48.06
FE	Ngã ngòì (ngòì vào ghế trống)		83.27	88.11	79.42	94.58	82.31
Hiệu suất phân loại ngã tính theo trung bình trọng số			80.40	92.73	82.59	96.43	63.10
W	Đi bộ	Hoạt động bình thường (ADL)	97.57	99.69	98.47	99.95	88.22
ST	Đứng		96.95	99.38	99.17	99.95	89.37
SI	Ngòì		97.05	99.32	99.32	99.82	95.08
P	Nhặt đồ vật		68.18	90.84	88.10	97.66	22.93
J	Nhảy		95.42	99.70	96.56	99.88	76.57
L	Nằm		99.52	99.57	99.81	99.87	98.04
Hiệu suất phân loại các hoạt động bình thường tính theo trung bình trọng số			97.09	99.38	98.77	99.86	90.03
Hiệu suất phân loại các hoạt động hỗn hợp (ngã và không ngã) tính theo trung bình trọng số			95.76	98.85	97.47	99.59	87.87

Trong phân loại các sự kiện ngã, mô hình RF đạt hiệu suất cao nhất với F1-Score trung bình là 96,43%. Ngược lại, SVM có hiệu suất thấp nhất với điểm F1-Score là 63,10%. Thuật toán J48 đạt điểm hoàn hảo (100%) trong việc nhận dạng sự kiện ngã về phía trước dùng tay chống (FH), trong khi SVM chỉ đạt 67,43% cho sự kiện này.

Hoạt động nhật đồ vật (P) có đặc điểm tương đồng với sự kiện ngã, dẫn đến việc các mô hình phân loại thường nhầm đây là sự kiện ngã. Hiệu suất phân loại hoạt động này là thấp nhất trong tất cả các hoạt động, bao gồm cả sự kiện ngã.

Đối với phân loại các hoạt động bình thường (ADL), RF cũng đạt hiệu suất cao, lên đến 99,86%. Trong khi đó, SVM có hiệu suất thấp nhất trong việc nhận dạng ADL với điểm F1-Score là 90,03%. Có thể nhận thấy RF là thuật toán mạnh mẽ nhất với hiệu suất phân loại tổng hợp (cả ngã và không ngã) cao nhất là 99,59%, cho thấy khả năng phân loại tốt trong cả hai trường hợp. Ngược lại, ANN và SVM có hiệu suất thấp hơn nhiều so với các thuật toán khác. Điều này cung cấp thông tin quan trọng cho việc lựa chọn mô hình phân loại phù hợp trong các nghiên cứu tiếp theo.

Trong các thực nghiệm tiếp theo, bộ đặc trưng TFDHD sẽ được sử dụng để đánh giá mức độ ảnh hưởng của DWS và DOR đến hiệu suất nhận dạng hoạt động của năm thuật toán phân loại đã nêu.

2.7.2.2. Ảnh hưởng của kích thước cửa sổ dữ liệu (DWS)

Trong phần này, bộ đặc trưng TFDHD được sử dụng trong các thực nghiệm. Để đánh giá ảnh hưởng của kích thước cửa sổ dữ liệu (DWS) đến hiệu suất phân loại hoạt động của các mô hình RF, tỉ lệ xếp chồng dữ liệu (DOR) được cố định ở mức 80%. Sáu kích thước cửa sổ khác nhau, từ 32 mẫu (2^5) đến 1024 mẫu (2^{10}), đã được thử nghiệm để xác định kích thước cửa sổ nào mang lại hiệu suất phân loại tốt nhất trên hai bộ dữ liệu MobiAct v2.0 và UP-Fall. Kết quả từ các thực nghiệm này được tổng hợp và trình bày trong ở Bảng 2.13 và Bảng 2.14.

Bảng 2.13. Ảnh hưởng của DWS đến hiệu suất nhận dạng các hoạt động trong bộ dữ liệu MobiAct v2.0

Nhãn	Hiệu suất của RF (F1-scores %) với DOR = 80% và các kích thước cửa sổ khác nhau (số mẫu)					
	32 mẫu	64 mẫu	128 mẫu	256 mẫu	512 mẫu	1024 mẫu
BSC	89.30	94.04	94.92	93.60	85.08	84.13
FKL	89.08	93.31	95.58	93.93	79.79	82.56
FOL	88.47	93.71	95.28	93.23	77.04	75.00
SDL	89.19	92.87	95.14	93.94	84.23	78.42
Ngã	89.01	93.48	95.23	93.67	81.53	80.04
CHU	86.47	96.87	95.18	93.66	95.45	95.50
CSI	83.22	88.21	94.75	93.36	96.42	98.18
CSO	96.30	98.79	98.33	98.26	98.43	98.21
JOG	97.37	96.51	99.49	99.39	99.82	99.92
JUM	99.01	98.11	99.91	99.90	99.98	100
LYI	97.91	99.15	98.84	98.69	91.27	97.89
SCH	91.43	94.06	96.61	95.49	97.12	97.96
SIT	89.19	92.87	99.84	93.94	84.23	78.42
STD	98.84	99.72	99.96	99.83	98.92	99.52
STN	99.21	99.03	95.02	99.94	99.99	100
STU	84.34	88.86	95.50	94.03	95.85	94.04
WAL	84.30	88.90	99.62	94.38	96.55	93.60
ADL	96.31	96.93	99.11	98.89	99.46	99.58
Tổng hợp	95.61	96.60	98.79	98.45	98.31	98.88

Kết quả trong Bảng 2.13 cho thấy DWS ảnh hưởng đáng kể hiệu suất của mô hình phân loại. Cửa sổ cần đủ lớn để bao gồm ít nhất một chu kỳ hoạt động và phân biệt được các hoạt động khác nhau. Nếu DWS quá nhỏ, số lượng đặc trưng được trích xuất không đủ để phản ánh bản chất của hoạt động. Tuy nhiên, nếu DWS quá lớn có thể tồn tại nhiều hoạt động trong cùng một cửa sổ làm cho hiệu suất phân loại của các thuật toán sẽ giảm mạnh, đặc biệt là đối với các sự kiện ngã.

Mô hình RF đạt hiệu suất tốt nhất ở DWS = 128 mẫu cho các sự kiện ngã, hiệu suất giảm đáng kể khi DWS tăng lên 1024 mẫu. Kết quả này phản ánh chính xác bản chất của các hoạt động như ngã, thường xảy ra trong khoảng thời gian ngắn. Sử dụng kích thước cửa sổ dài có thể dẫn đến việc lẫn các hoạt động khác. Các hoạt động có sự thay đổi trạng thái nhanh như chuyển từ ngồi sang đứng (CHU) hoặc bước ra khỏi xe hơi (CSO) thích hợp với các kích thước cửa sổ ngắn như 64 mẫu hoặc 128 mẫu. Các

hoạt động thường ngày thường có tính chu kỳ và ít thay đổi về trạng thái như chạy (JOG), nhảy (JUM), đi xuống cầu thang (STN), đi lên cầu thang (STU), đi bộ (WAL) thường phù hợp với các kích thước cửa sổ dài như 512 mẫu hoặc 1024 mẫu.

Đối với bộ dữ liệu MobiAct v2.0, có thể thấy kích thước cửa sổ 128 mẫu là lý tưởng cho mô hình RF trong việc nhận dạng các sự kiện ngã. Hiệu suất phân loại ngã giảm dần khi kích thước cửa sổ tăng lên.

Bảng 2.14 trình bày kết quả phân loại các hoạt động bằng mô hình RF (F1-scores) với DOR = 80% và các kích thước cửa sổ khác nhau. Trong bảng này, số liệu được tô màu đỏ đậm biểu thị hiệu suất cao nhất cho từng nhãn hoạt động ở các kích thước cửa sổ tương ứng. Từ bảng, ta có thể nhận thấy rằng hiệu suất nhận dạng phụ thuộc vào kích thước cửa sổ. Mô hình RF đạt hiệu suất phân loại tốt nhất khi kích thước cửa sổ DWS = 64 mẫu đối với hầu hết các hoạt động trong bộ dữ liệu UP-Fall. Hiệu suất phân loại có xu hướng giảm khi kích thước cửa sổ dữ liệu tăng.

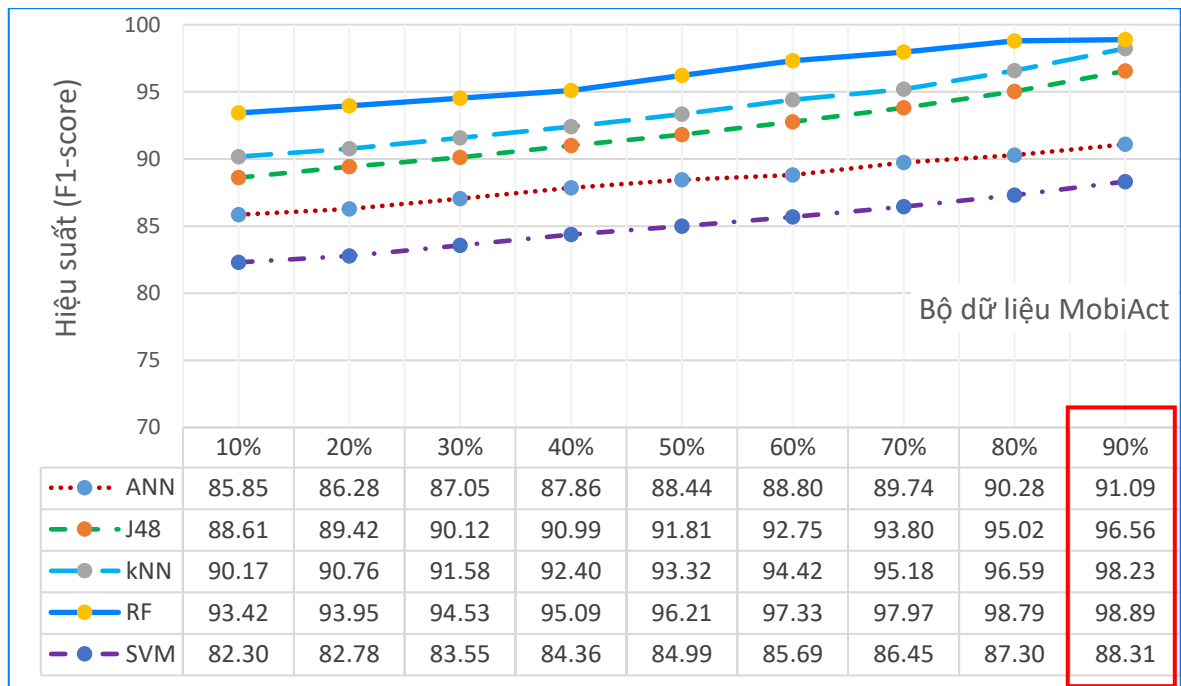
Bảng 2.14. Ảnh hưởng của DWS đến hiệu suất nhận dạng các hoạt động trong bộ dữ liệu UP-Fall

Nhãn	Hiệu suất của RF (F1-scores %) với DOR = 80% và các kích thước cửa sổ khác nhau (số mẫu)					
	32 mẫu	64 mẫu	128 mẫu	256 mẫu	512 mẫu	1024 mẫu
FH	99.71	100	99.78	96.15	100	96.97
FF	99.51	99.80	99.79	96.00	100	98.08
FB	98.88	99.70	98.94	90.91	91.84	89.80
FS	99.22	99.32	96.23	81.90	80.39	80.00
FE	98.81	99.29	97.52	89.80	88.46	86.54
Ngã	99.23	99.62	98.46	90.95	92.14	90.28
W	99.95	99.96	99.97	100	100	100
ST	99.98	100	99.99	99.94	100.00	97.65
SI	99.95	99.99	99.97	99.82	99.71	97.62
P	99.60	99.68	99.21	99.03	98.04	96.08
J	99.94	100	99.94	99.69	98.67	100.00
L	99.96	99.95	100	99.82	99.70	98.72
ADL	99.95	99.97	99.97	99.87	99.73	98.40
Tổng hợp	99.85	99.93	99.88	99.30	98.61	95.41

2.7.2.3. Ảnh hưởng của tỉ lệ xếp chồng dữ liệu (DOR)

Trong các thí nghiệm ở phần này, dữ liệu thô từ bộ MobiAct v2.0 được phân đoạn thành các cửa sổ kích thước DWS = 128 mẫu, còn bộ UP-Fall được phân thành các cửa sổ kích thước DWS = 64 mẫu. Các DWS này được lựa chọn dựa trên hiệu suất tốt nhất trong việc phân loại sự kiện ngã từ các thực nghiệm trước đó.

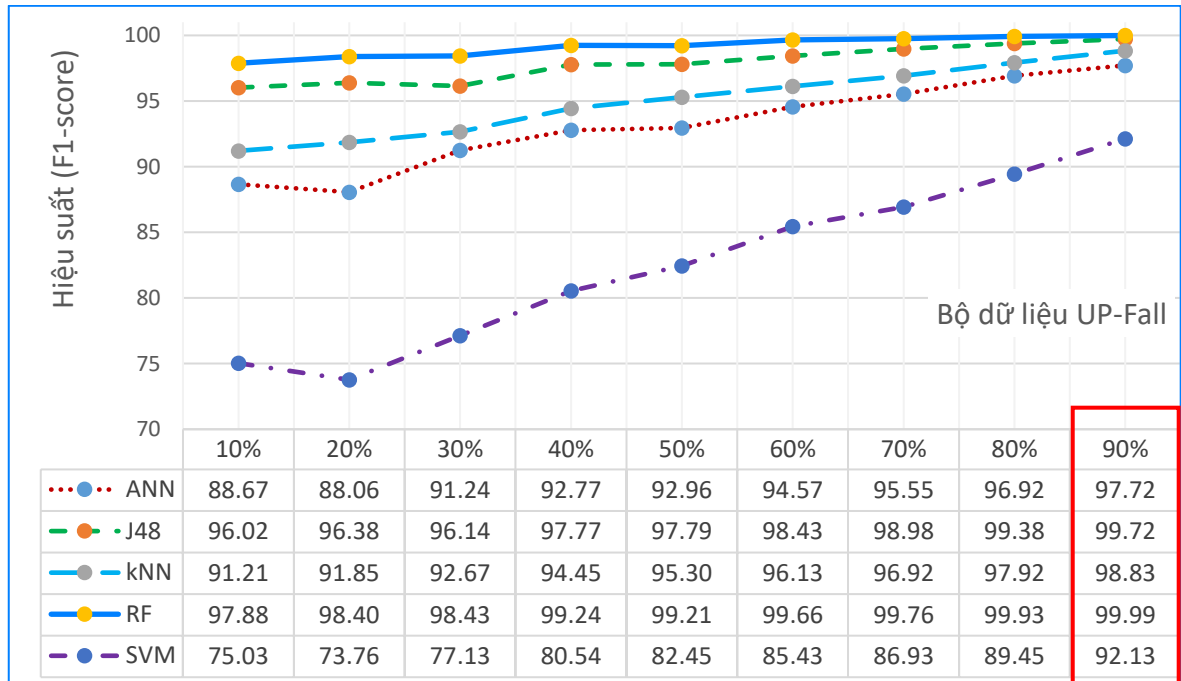
Chín tỉ lệ xếp chồng dữ liệu (từ 10% đến 90%) được sử dụng để tiến hành các thực nghiệm đánh giá sự ảnh hưởng của tỉ lệ xếp chồng dữ liệu (DOR). Các kết quả thực nghiệm được tổng hợp và trình bày ở Hình 2.13 và Hình 2.14.



Hình 2.13. Ảnh hưởng của DOR đến hiệu suất phân loại các hoạt động trong bộ dữ liệu MobiAct v2.0.

Có thể thấy rằng hiệu suất phân loại được cải thiện đáng kể khi tỉ lệ chồng chéo tăng. Tỉ lệ chồng chéo cao hơn dẫn đến việc đọc dữ liệu mới trong cửa sổ tiếp theo ngắn hơn, giúp phát hiện hiệu quả các hoạt động nhanh như ngã. Nhìn chung, hiệu suất của mô hình tỉ lệ thuận với DOR.

Trong bộ dữ liệu UP-Fall, sự khác biệt về hiệu suất giữa các thuật toán rất rõ ràng khi DOR nhỏ. Điều này là do số mẫu của mỗi hoạt động trong bộ UP-Fall ít hơn nhiều so với MobiAct, khiến các thuật toán không thể học được nhiều từ dữ liệu khi tỉ lệ chồng chéo thấp. Khi tăng DOR, độ chênh lệch giảm dần. Với DOR 90%, chênh lệch về hiệu năng giữa các thuật toán phân loại không còn quá lớn.



Hình 2.14. Ảnh hưởng của DOR đến hiệu suất nhận dạng của mô hình.

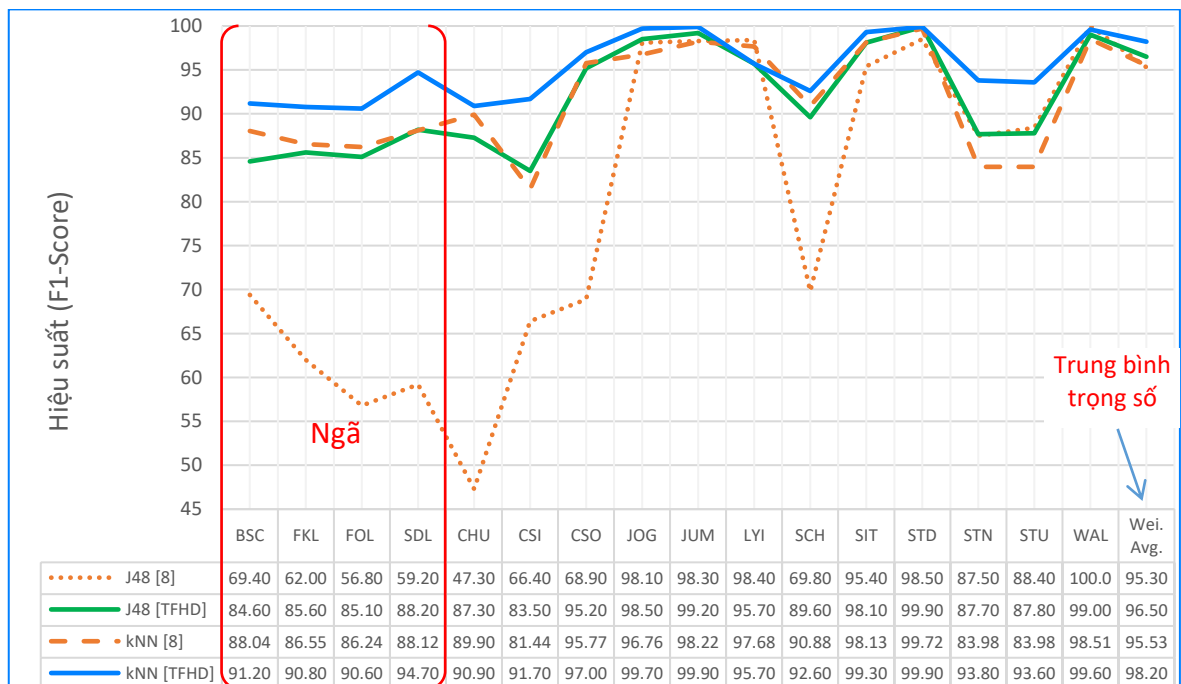
Quan sát kết quả ở Hình 2.14, có thể thấy sự ưu thế vượt trội của mô hình RF so với ANN và SVM trong mọi tỉ lệ DOR, đặc biệt khi bộ dữ liệu huấn luyện nhỏ. RF luôn cho thấy hiệu suất tốt nhất trong mọi DOR. Ngược lại, thuật toán SVM dường như không phù hợp với các điều kiện trong luận án này và nó có hiệu suất kém nhất trong mọi giả định được đặt ra.

2.7.3. So sánh kết quả với các nghiên cứu liên quan

2.7.3.1. Nghiên cứu dựa trên bộ dữ liệu MobiAct

Trong phần này, NCS sẽ so sánh phương pháp đề xuất với các nghiên cứu gần đây trên bộ dữ liệu MobiAct v2.0 [107]. Thông qua khảo sát các nghiên cứu sử dụng bộ dữ liệu MobiAct v2.0 từ scholar.google.com, NCS nhận thấy nhiều công trình sử

dụng phương pháp học máy để xây dựng mô hình phân loại nhị phân, đó là ngã hay không ngã. Vì vậy, NCS chọn nghiên cứu của Chatzaki [13] để so sánh vì cả hai có sự tương đồng trong bài toán phân loại đa lớp. Trong so sánh này, NCS đã thiết lập các thông số mô hình như kích thước cửa sổ trượt là 128 mẫu (tương đương với 1,5 giây) với tỉ lệ chồng chéo 80%, tương tự như trong nghiên cứu của Chatzaki và cộng sự [13]. Kết quả thu được từ bộ đặc trưng TFDHD đề xuất và phương pháp của nhóm tác giả Chatzaki [13] được thể hiện ở Hình 2.15.



Hình 2.15. Hiệu suất chi tiết của Chatzaki và cộng sự [13] và mô hình đề xuất của NCS trong việc phát hiện ADL và ngã.

Nhìn chung, hiệu suất nhận dạng các hoạt động mang tính ổn định hoặc tuần hoàn như chạy bộ (JOG), nhảy liên tục (JUM), nằm (LYI), ngồi (SIT), đứng (STD) hay đi bộ (WAL) có sự tương đồng giữa các nghiên cứu. Hiệu suất tổng thể theo trung bình trọng số trong phương pháp của NCS có tốt hơn của Chatzaki nhưng chênh lệch không quá lớn. Tuy nhiên, đối với các hoạt động thay đổi trạng thái nhanh và đột ngột như các sự kiện ngã, bộ đặc trưng TFDHD thể hiện sự vượt trội so với phương pháp của Chatzaki và cộng sự [13].

Đối với bộ phân loại J48, bộ đặc trưng TFDHD có thể đạt độ chính xác lên đến 88.20% trong việc phát hiện sự kiện ngã, trong khi đề xuất của Chatzaki có độ chính xác cao nhất là 69.40%. Điều tương tự cũng xảy ra khi sử dụng bộ đặc trưng TFDHD và bộ đặc trưng của Chatzaki để huấn luyện mô hình kNN, hiệu suất trong việc phát hiện sự kiện ngã trong đề xuất của NCS có thể đạt đến 94.70%, trong khi của Chatzaki và cộng sự [13] chỉ đạt tối đa 88.12%.

Không chỉ giới hạn trong việc phát hiện các sự kiện ngã, bộ đặc trưng TFDHD còn thể hiện khả năng nhận dạng các hoạt động khó phân biệt, thay đổi trạng thái nhanh như ngồi chuyển sang đứng (CHU), đứng chuyển sang ngồi (SCH) tốt hơn so với công trình của Chatzaki và cộng sự [13]. Kết quả trên Hình 2.15 cho thấy bộ đặc trưng TFDHD đề xuất trong luận án này phù hợp với các hoạt động diễn ra nhanh và thay đổi đột ngột như các sự kiện ngã. Đó là mục tiêu chính mà NCS muốn hướng đến khi nghiên cứu và đề xuất bộ đặc trưng TFDHD.

Ngoài ra, kết quả nhận dạng của 16 hoạt động trong bộ dữ liệu MobiAct v2.0 cũng được trình bày dưới dạng ma trận nhầm lẫn ở trong Hình 2.16.

Actual:	Classified as:															
	STD	BSC	LYI	SIT	CHU	CSI	FKL	FOL	JOG	JUM	WAL	STU	SCH	SDL	STN	
STD	98.43	0.04	0.01	0.01	0.06	0.17	0.30	0.05	0.10	0.03	0.02	0.26	0.22	0.05	0.05	0.21
BSC	1.86	83.16	4.33	0.00	0.00	0.19	0.04	1.90	1.94	0.00	0.00	0.00	0.91	0.00	5.21	0.46
LYI	0.02	0.36	98.70	0.01	0.00	0.01	0.00	0.30	0.26	0.00	0.00	0.00	0.01	0.00	0.31	0.02
SIT	0.05	0.00	0.01	97.71	0.15	0.84	0.48	0.00	0.00	0.00	0.00	0.00	0.77	0.00	0.00	0.00
CHU	5.81	0.00	0.00	4.05	59.15	1.85	3.35	0.00	0.00	0.09	0.00	2.64	0.18	22.80	0.00	0.09
CSI	2.16	0.01	0.00	3.13	0.56	80.19	9.45	0.00	0.01	0.07	0.01	1.68	0.19	2.47	0.00	0.07
CSO	3.18	0.01	0.00	1.62	0.73	7.30	83.48	0.03	0.00	0.03	0.00	1.36	0.17	2.02	0.01	0.07
FKL	2.62	2.02	3.17	0.00	0.00	0.09	0.05	75.57	8.95	0.00	0.00	0.00	1.65	0.00	2.80	3.08
FOL	5.56	3.88	4.08	0.00	0.00	0.05	0.00	10.66	69.10	0.00	0.00	0.00	1.43	0.00	3.62	1.63
JOG	0.12	0.00	0.00	0.00	0.01	0.05	0.03	0.00	0.00	98.50	0.28	0.78	0.10	0.01	0.00	0.11
JUM	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69	98.97	0.16	0.05	0.00	0.00	0.05
WAL	0.13	0.00	0.00	0.00	0.01	0.05	0.05	0.00	0.00	0.04	0.00	99.37	0.17	0.03	0.00	0.14
STU	1.15	0.04	0.01	0.00	0.01	0.04	0.02	0.02	0.06	0.02	0.00	2.53	92.95	0.02	0.04	3.10
SCH	1.75	0.00	0.00	5.12	6.56	2.67	2.98	0.00	0.00	0.03	0.03	1.77	0.26	78.81	0.00	0.03
SDL	2.83	7.92	4.18	0.00	0.00	0.22	0.09	3.87	3.05	0.00	0.00	0.00	2.79	0.00	72.14	2.92
STN	1.28	0.02	0.03	0.00	0.01	0.02	0.01	0.13	0.06	0.03	0.01	2.98	4.03	0.00	0.13	91.25

(a) Chatzaki et al.

Actual:	Classified as:															
	STD	BSC	LYI	SIT	CHU	CSI	FKL	FOL	JOG	JUM	WAL	STU	SCH	SDL	STN	
STD	99.92	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00
BSC	0.04	91.16	0.52	0.07	0.00	3.28	0.07	0.55	1.11	0.00	0.00	0.00	0.00	1.11	1.62	0.00
LYI	0.00	3.65	95.74	0.75	0.00	0.00	0.00	0.66	0.09	0.00	0.00	0.00	0.00	0.56	0.00	0.00
SIT	0.05	0.00	0.16	99.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.05	0.00
CHU	3.43	0.00	0.00	0.00	90.94	0.00	7.02	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.33
CSI	0.09	2.79	0.04	0.04	0.00	91.71	0.56	0.39	0.39	0.00	0.00	0.00	0.47	1.89	1.25	0.00
CSO	0.00	0.39	0.00	0.00	1.54	0.60	97.04	0.04	0.00	0.00	0.00	0.00	0.39	0.00	0.09	0.04
FKL	0.07	0.74	0.15	0.00	0.00	0.74	0.04	90.82	5.79	0.00	0.00	0.11	0.04	1.19	0.26	0.04
FOL	0.07	1.26	0.04	0.04	0.00	0.45	0.00	6.24	90.62	0.00	0.00	0.00	0.00	0.22	1.19	0.00
JOG	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.66	0.03	0.24	0.06	0.00	0.00	0.19
JUM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	99.93	0.00	0.00	0.00	0.00	0.04
WAL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.61	0.06	0.00	0.00	0.03
STU	0.14	0.00	0.00	0.00	0.07	0.02	0.00	0.00	0.00	0.07	0.00	2.37	93.57	0.00	0.07	4.73
SCH	0.00	1.96	0.20	0.25	0.00	1.96	0.00	1.70	1.15	0.00	0.00	0.00	0.00	92.60	0.80	0.00
SDL	0.00	1.71	0.11	0.04	0.00	1.01	0.26	0.48	1.01	0.00	0.00	0.00	0.07	0.56	94.73	0.15
STN	0.09	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.09	0.00	2.21	4.36	0.00	0.05	93.78

(b) Our method

Hình 2.16. Ma trận nhầm lẫn của 16 hoạt động trong bộ MobiAct v2.0. (a) Chatzaki và cộng sự [13], (b) Phương pháp NCS đề xuất..

Khi xem xét kết quả ở Hình 2.16.a và Hình 2.16.b, có thể thấy hoạt động chuyển từ tư thế ngồi sang đứng (CHU) là hoạt động khó phát hiện nhất. Trong Hình 2.16.a, kết

quả nghiên cứu của [13] đã phân loại nhầm hoạt động này sang hoạt động chuyển đổi từ đứng sang ngồi (SCH), với tỉ lệ lỗi lên đến 22,8%. Trong khi đó, khả năng phát hiện hoạt động CHU của bộ đặc trưng TFDHD tỏ ra vượt trội hơn, tỉ lệ dự đoán nhầm sang hoạt động khác (CSO - bước ra khỏi xe hơi) chỉ là 7,02%.

Chatzaki và cộng sự [13] thừa nhận rằng việc phát hiện chính xác các sự kiện ngã rất khó khăn. Độ chính xác tổng thể có trọng số của họ đối với tất cả các hoạt động (bao gồm cả ngã) có thể đạt tới 96,8%. Tuy nhiên, hiệu suất phát hiện ngã của họ tương đối thấp, chỉ đạt 69.10% đối với FOL và tối đa là 83,16% đối với BSC.

Trong khi đó, kết quả thu được từ phương pháp mà NCS đề xuất cho thấy sự cải thiện đáng kể về độ chính xác trong việc phát hiện ngã và có sự cân bằng hiệu năng trong việc phát hiện tất cả các hoạt động (bao gồm cả ngã và không ngã). Hiệu suất phân loại của mọi hoạt động đều đạt trên 90%. Điều này chứng tỏ tầm quan trọng của việc làm sạch, chuẩn hóa, trích xuất và lựa chọn đặc trưng khi sử dụng dữ liệu thô.

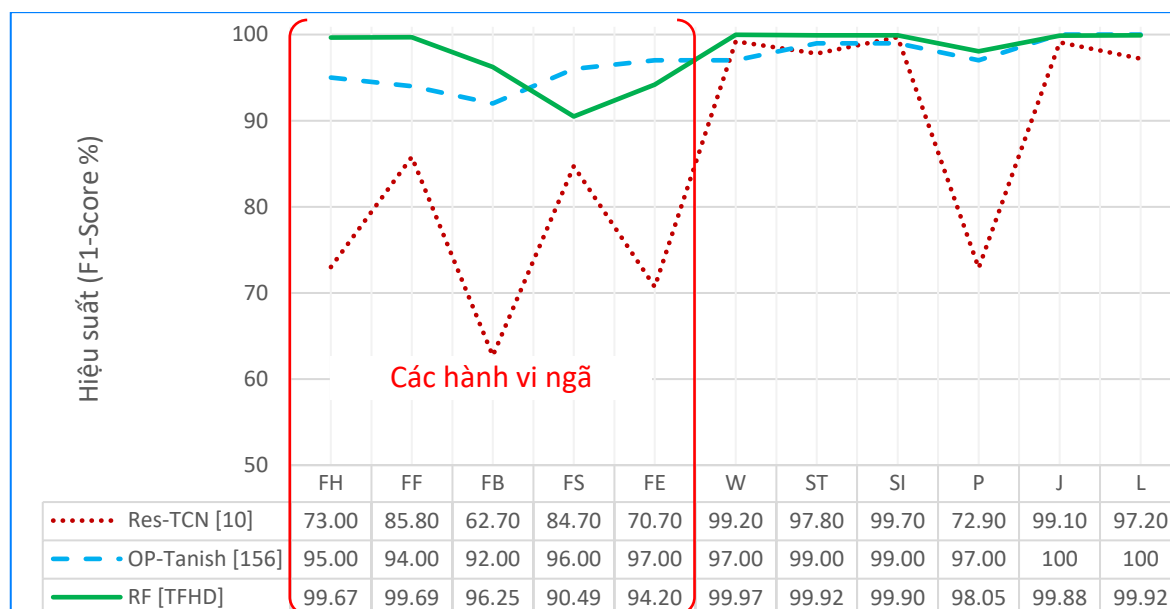
Kết quả được sử dụng để so sánh với nghiên cứu của Chatzaki và cộng sự không phải là tốt nhất. Trong luận án này, phương pháp mà NCS đề xuất có kết quả tốt nhất lên đến 98,79% ở thước đo F1-score khi sử dụng bộ dữ liệu MobiAct v2.0. Kết quả này đạt được khi sử dụng bộ đặc trưng TFDHD với kích thước cửa sổ dữ liệu 128 mẫu, tỉ lệ xếp chồng dữ liệu 80%, và được huấn luyện bằng mô hình phân loại RF.

2.7.3.2. Nghiên cứu dựa trên bộ dữ liệu UP-Fall

Tương tự với phương pháp so sánh các Nghiên cứu dựa trên bộ dữ liệu MobiAct, phần này NCS so sánh phương pháp của mình với các công trình gần đây trên bộ dữ liệu UP-Fall. Các điều kiện thử nghiệm đều được thiết lập tương đồng giữa các nghiên cứu. Hình 2.17 thể hiện kết quả nghiên cứu đã công bố của Lai và cộng sự [9] và kết quả thực nghiệm dựa trên bộ đặc trưng TFDHD đề xuất.

Kết quả thực nghiệm bộ đặc trưng TFDHD có độ chính xác cao, đảm bảo mục tiêu phù hợp với FDS. Các chỉ số Độ chính xác dương tính (precision), Độ nhạy (recall),

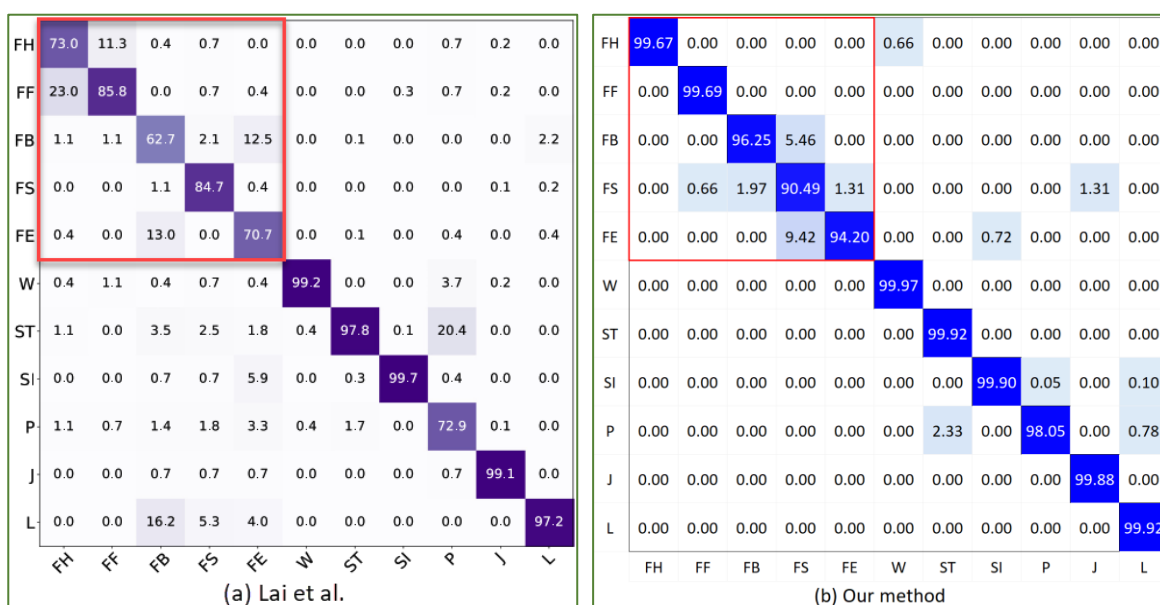
Độ đặc hiệu (specificity), Điểm trung bình điều hòa (F1-score) đều cho kết quả gần tương đồng, thấp nhất là 93,63% và cao nhất là 99,52%. Điều đó chứng tỏ tỉ lệ “âm tính giả” và “dương tính giả” của bộ đặc trưng TFDHD là tương đối thấp.



Hình 2.17. Hiệu suất chi tiết của Lai và cộng sự [9] và bộ đặc trưng TFDHD đề xuất trong việc phát hiện ADL và các sự kiện ngã.

Trong nghiên cứu của Lai và cộng sự [9], hiệu suất nhận dạng sự kiện ngã và không ngã có sự khác biệt lớn. Khả năng phát hiện ngã và các hoạt động dễ gây nhầm lẫn chưa thực sự tốt, chỉ đạt từ 62,2% đến 85,8%. Ngược lại, bộ đặc trưng TFDHD đạt hiệu suất tối thiểu là 90,49%, bao gồm cả hoạt động dễ gây nhầm lẫn như cúi xuống nhặt đồ vật (P). Hiệu suất bộ đặc trưng TFDHD này tốt hơn đáng kể so với nghiên cứu của Villaseñor [12] và cả nghiên cứu trước đó của Lai [9], đặc biệt là trong việc nhận dạng các sự kiện ngã.

Kết quả nhận dạng các hoạt động được trình bày dưới dạng ma trận nhầm lẫn trong bộ dữ liệu UP-Fall được trình bày ở Hình 2.18. Các phần tử trên đường chéo của ma trận biểu thị tỉ lệ giữa hiệu suất dự đoán và thực tế, trong khi các phần tử không nằm trên đường chéo đại diện cho các lỗi phân loại. Tỉ lệ phần trăm các phần tử nằm trên đường chéo càng cao mô hình dự đoán càng chính xác, điều này là một chỉ báo tốt về hiệu suất nhận dạng.



Hình 2.18. Ma trận nhầm lẫn của 11 hoạt động trong bộ dữ liệu UP-Fall. Hàng là các lớp dự đoán; cột là các lớp thực tế. Ma trận con 5×5 (FH, FF, FB, FS và FE) là các lớp ngã. (a) Kết quả của Lai và cộng sự [9] sử dụng mô hình Res-TCN. (b) Mô hình RF dự đoán dựa trên bộ đặc trưng TFDHD đề xuất.

Trong công bố của Lai và cộng sự [9], dữ liệu ma trận nhầm lẫn cho thấy dường như có sự phân chia thành hai cụm: các sự kiện ngã và không ngã. Mô hình của họ đã thể hiện khả năng phân loại các hoạt động bình thường gần như hoàn hảo. Tuy nhiên, bộ đặc trưng TFDHD vẫn cho hiệu suất phân loại nhỉnh hơn trong nhóm hoạt động này. Với bộ đặc trưng TFDHD, khả năng phân loại mọi hoạt động (bao gồm các sự kiện ngã) đều đạt trên 90%. Đặc biệt, khả năng phát hiện các hoạt động như đi, đứng, ngồi, nhảy, nằm đạt hiệu năng xuất sắc, từ 99,8% trở lên.

2.8. Kết luận chương

Nội dung Chương 2 đã trình bày một phương pháp trích rút đặc trưng mới bằng cách kết hợp thông tin từ miền thời gian, miền tần số, tham số định hướng và tham số Hjorth của dữ liệu gia tốc. Bộ 44 đặc trưng TFDHD được đề xuất cho thấy hiệu suất tốt hơn trong việc phát hiện ngã so với các nghiên cứu khác, đạt độ chính xác lần lượt là 95,23%, 99,11% và 98,79% trên bộ dữ liệu MobiAct v2.0 và 96,16%, 99,90% và 99,60% trên bộ dữ liệu Up-Fall.

Kết quả thực nghiệm đạt được khi sử dụng kích thước cửa sổ 128 mẫu dữ liệu (tương đương 1.5 giây với MobiAct v2.0 và 1.28 giây với UP-Fall), tỉ lệ xếp chồng dữ liệu 80%, và mô hình Random Forest (RF) để phân loại. Mô hình RF luôn cho hiệu suất phân loại tốt nhất trong mọi điều kiện thử nghiệm.

Chương tiếp theo sẽ trình bày chi tiết về phương pháp tối ưu mô hình Random Forest để cải thiện khả năng phân loại các sự kiện ngã, cũng như phát triển mô hình phát hiện sự kiện ngã phù hợp với dữ liệu khó thu thập như dữ liệu về ngã.

Chương 3. XÂY DỰNG MÔ HÌNH PHÁT HIỆN SỰ KIỆN NGÃ

Hiệu suất của mô hình Random Forest (RF) phụ thuộc nhiều vào việc lựa chọn và điều chỉnh các siêu tham số [108–112]. Với không gian tìm kiếm rộng do phạm vi giá trị của các siêu tham số rất lớn [108, 109, 113], việc tìm kiếm giá trị tối ưu trở nên phức tạp và có thể đòi hỏi hàng trăm triệu lần thử nghiệm để đạt được kết quả tốt nhất. Nếu các siêu tham số không được tối ưu, quá trình huấn luyện RF có thể kéo dài và khó hội tụ. Do đó, tối ưu hiệu suất của RF là một bài toán phức tạp cần được nghiên cứu nhiều hơn.

Ngoài ra, khi nghiên cứu hệ thống phát hiện ngã (FDS), một thách thức lớn là thiếu dữ liệu thực tế. Dữ liệu về các sự kiện ngã thực sự rất khan hiếm, khó thu thập và gán nhãn, dẫn đến việc nhiều nghiên cứu phải sử dụng các bộ dữ liệu mô phỏng sự kiện ngã thu thập trong phòng thí nghiệm. Điều này có thể làm giảm hiệu suất của mô hình khi áp dụng vào thực tế. Ngược lại, dữ liệu về các hoạt động bình thường hàng ngày lại dễ thu thập và phong phú, nên việc khai thác hiệu quả nguồn dữ liệu này là rất quan trọng.

Để giải quyết các thách thức trên, trong chương này, nghiên cứu sinh đề xuất:

- + Thuật toán tìm kiếm các siêu tham số tối ưu cho mô hình RF trong việc phân loại các sự kiện ngã (GA4RF). Kết quả nghiên cứu này đã được công bố trong các công trình [CT2], [CT3], và [CT4].
- + Mô hình phát hiện sự kiện ngã (FEDM) dựa trên dữ liệu các hoạt động thường ngày. Kết quả nghiên cứu này đã được công bố trong công trình [CT5].

3.1. Giới thiệu bài toán

Bài toán tối ưu hóa mô hình phân loại RF và xây dựng mô hình phát hiện sự kiện ngã (FEDM) được thực hiện theo hai giai đoạn:

Giai đoạn thứ nhất tập trung vào việc tối ưu hóa mô hình RF thông qua việc đề xuất thuật toán GA4RF. Mục tiêu của giai đoạn này là tìm kiếm bộ siêu tham số tối ưu cho RF nhằm nâng cao khả năng phát hiện các sự kiện ngã, đặc biệt trong trường hợp dữ liệu không cân bằng. Việc tìm kiếm giá trị tối ưu của các siêu tham số RF được thực hiện thông qua giải thuật di truyền (GA) với cấu trúc nhiễm sắc thể, không gian tìm kiếm và hàm thích nghi được thiết kế theo hướng mới phù hợp với đặc thù của các bộ dữ liệu ngã không cân bằng.

Giai đoạn thứ hai tiến hành xây dựng mô hình phát hiện sự kiện ngã (FEDM) dựa trên dữ liệu các hoạt động bình thường hàng ngày của con người. Mô hình FEDM kết hợp thuật toán GA4RF với kỹ thuật so khớp dữ liệu linh hoạt và kỹ thuật ngưỡng để nhận dạng các sự kiện ngã. Quá trình nhận dạng được thực hiện thông qua việc so khớp các mẫu dữ liệu chưa biết với các mẫu dữ liệu đã biết từ bộ dữ liệu huấn luyện, giải quyết khó khăn trong việc thu thập dữ liệu ngã thực tế.

Cả hai giai đoạn này đều nhằm cải thiện hiệu suất và độ tin cậy của hệ thống phát hiện ngã, đáp ứng tốt hơn các thách thức trong việc phát hiện và xử lý các sự kiện ngã từ dữ liệu thực tế.

3.2. Tối ưu mô hình mô hình phát hiện ngã

3.2.1. Giới thiệu tổng quan

Không gian tìm kiếm của các siêu tham số trong mô hình Random Forest (RF) là rất rộng, với khoảng giá trị rất lớn [108, 109, 113], dẫn đến việc cần thực hiện hàng trăm triệu lần thử nghiệm để tìm được bộ siêu tham số tối ưu. Điều này là không khả thi nếu sử dụng các phương pháp thủ công. Để giải quyết vấn đề này, việc áp dụng các thuật toán tối ưu hóa siêu tham số giúp tự động hóa và nâng cao hiệu suất, cũng như khả năng tổng quát hóa của mô hình.

Hai phương pháp phổ biến cho việc tối ưu hóa siêu tham số là tìm kiếm theo lưới (Grid Search - GS) và tìm kiếm ngẫu nhiên (Random Search - RS). GS thực hiện khảo

sát toàn bộ các kết hợp giá trị siêu tham số đã xác định trước, tính toán và đánh giá hiệu suất mô hình với mỗi kết hợp. Trong khi đó, RS chọn ngẫu nhiên các giá trị siêu tham số từ phạm vi cho phép để đánh giá mô hình, tiết kiệm thời gian hơn so với GS.

Mặc dù GS là một phương pháp đơn giản và dễ hiểu, nó không phù hợp cho các bài toán dữ liệu lớn, có nhiều chiều và yêu cầu tối ưu hóa nhiều biến cùng lúc [114]. James Bergstra và Yoshua Bengio [115] đã chứng minh rằng RS có thể tìm kiếm hiệu quả hơn GS trong không gian rộng, nhờ vào việc khảo sát được nhiều điểm trong không gian tìm kiếm hơn trong cùng thời gian. Tuy nhiên, kết quả của RS chưa chắc đã tốt hơn GS [116]. Cả GS và RS chỉ thích hợp khi không gian tìm kiếm hẹp, và gặp hạn chế khi xử lý không gian tìm kiếm rộng hoặc nhiều biến đầu vào, dẫn đến chi phí tính toán cao và mô hình phức tạp.

Để giải quyết các thách thức trong việc tìm kiếm siêu tham số tối ưu cho RF khi xử lý các bộ dữ liệu lớn, luận án này sử dụng phương pháp tối ưu hóa siêu kinh nghiệm (Metaheuristics - MH). Các phương pháp MH hiện đang được sử dụng nhiều trong các tác vụ quan trọng như trích xuất đặc trưng, lựa chọn đặc trưng, và tinh chỉnh siêu tham số trong các mô hình học máy. MH là các phương pháp tối ưu hóa tiên tiến, giúp khắc phục hạn chế của GS và RS bằng cách mô phỏng quá trình tiến hóa tự nhiên và tìm kiếm thông qua các phép biến đổi như lai ghép, đột biến và chọn lọc để tránh tối ưu cục bộ. MH được đánh giá cao nhờ tính đơn giản, linh hoạt, không yêu cầu tính toán đạo hàm và ít bị ảnh hưởng bởi tối ưu cục bộ [65].

Một số thuật toán MH phổ biến để tối ưu hóa siêu tham số cho RF bao gồm Thuật toán động vật săn mồi biển (Marine Predators Algorithm - MPA), Thuật toán bầy ong nhân tạo (Artificial Bee Colony - ABC), Thuật toán sói xám (Grey Wolf Optimizer - GWO), Thuật toán tối ưu bầy đàn (Particle Swarm Optimization - PSO), và Giải thuật di truyền (Genetic Algorithm - GA) [118]. Mỗi thuật toán có những ưu, nhược điểm riêng. Trong luận án này, với các đặc điểm của dữ liệu và các phương pháp trích xuất đặc trưng đã sử dụng, nghiên cứu sinh đã chọn giải thuật di truyền (GA) để đề xuất giải pháp tối ưu cho mô hình RF.

3.2.2. Giải thuật di truyền (GA)

Giải thuật di truyền (GA) là một phương pháp tối ưu hóa mạnh mẽ dựa trên siêu kinh nghiệm, lấy cảm hứng từ thuyết tiến hóa sinh học của Darwin [22, 117]. GA tận dụng dữ liệu lịch sử để hướng đến những vùng tìm kiếm tiềm năng nhất trong không gian giải pháp. Thuật toán này có khả năng hoạt động hiệu quả trong cả không gian tìm kiếm liên tục và rời rạc, đặc biệt thích hợp cho các bài toán có không gian tìm kiếm rộng và đa chiều. Một ưu điểm lớn của GA là không phụ thuộc vào thông tin đạo hàm của hàm mục tiêu, điều này cho phép GA tránh rơi vào các điểm tối ưu cục bộ và tìm ra các giải pháp toàn cục tốt hơn nhờ vào tính ngẫu nhiên và quá trình tiến hóa [65].

Quá trình tìm kiếm giải pháp tối ưu của GA tuân theo nguyên lý “khả năng sống sót của những cá thể thích nghi nhất” [118].

GA bắt đầu bằng việc tạo ra một quần thể các giải pháp tiềm năng trong không gian tìm kiếm. Mỗi cá thể được biểu diễn bằng một nhiễm sắc thể, chứa các gen tương ứng với các siêu tham số cần tối ưu hóa. Quá trình tiến hóa của quần thể diễn ra qua các thế hệ, trong đó các cá thể trải qua các quá trình chọn lọc, lai ghép và đột biến.

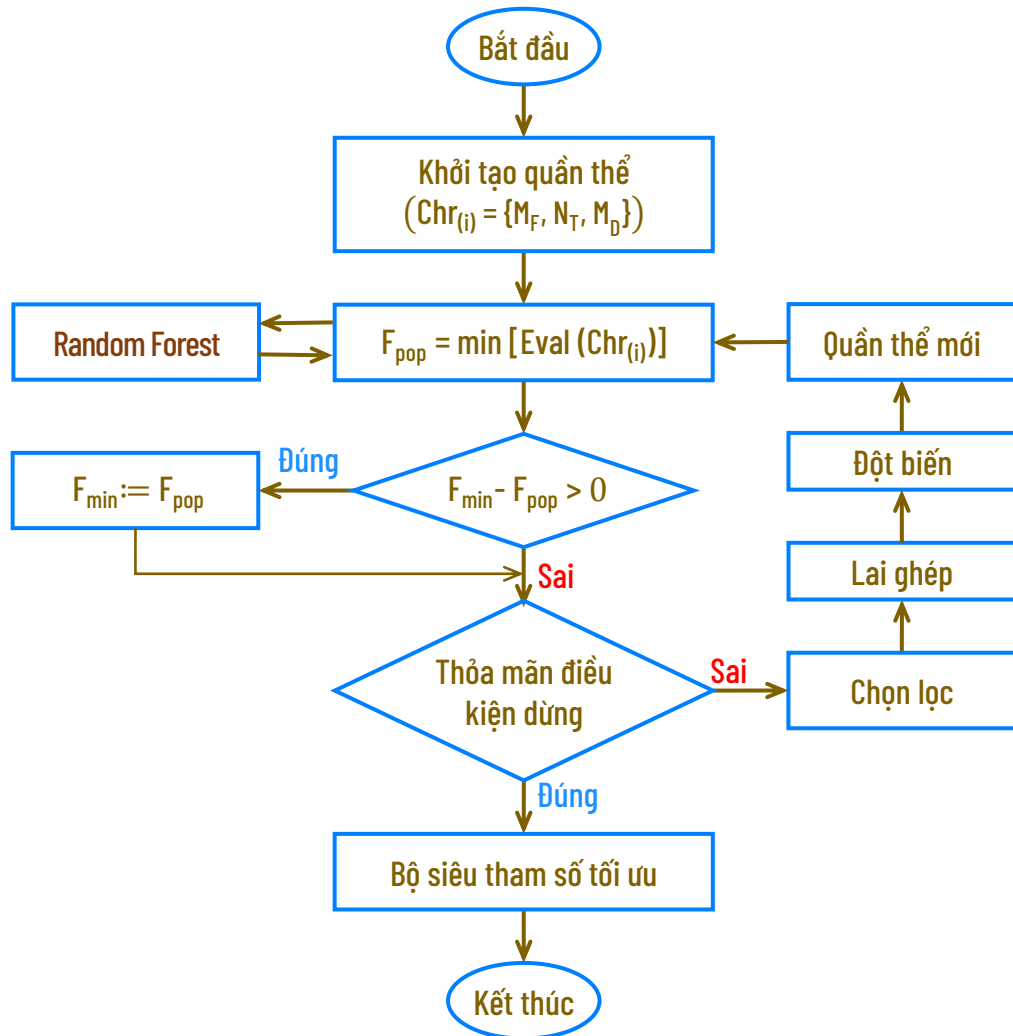
- **Chọn lọc:** Các cá thể có độ thích nghi tốt hơn (được đánh giá thông qua hàm thích nghi) có nhiều cơ hội được chọn để sinh sản cho thế hệ tiếp theo hơn.
- **Lai ghép:** Các cá thể được chọn sẽ lai ghép với nhau để tạo ra các cá thể con mới, kết hợp các đặc điểm của bố mẹ. Điều này tạo ra sự đa dạng trong quần thể và cho phép khám phá các vùng mới trong không gian siêu tham số.
- **Đột biến:** Một số gen của cá thể con có thể bị đột biến ngẫu nhiên, tạo ra sự đa dạng trong quần thể và ngăn chặn sự hội tụ sớm ở các điểm tối ưu cục bộ, giúp khám phá các vùng mới trong không gian tìm kiếm.

Quá trình này lặp đi lặp lại cho đến khi đạt được một giải pháp tối ưu hoặc thỏa mãn các tiêu chí dừng. Nhờ tính ngẫu nhiên và quá trình tiến hóa, GA có khả năng tránh rơi vào tối ưu cục bộ, giúp tìm ra các giải pháp toàn cục tốt hơn.

3.2.3. Đề xuất thuật toán GA4RF

3.2.3.1. Tổng quan về thuật toán GA4RF

Thuật toán GA4RF được đề xuất dựa trên thuật toán di truyền để tối ưu hóa các siêu tham số của mô hình Random Forest trong hệ thống phát hiện ngã (FDS). Các bước chính trong thuật toán GA4RF được minh họa trong Hình 3.1.



Hình 3.1. Thuật toán GA4RF

Thuật toán bắt đầu với việc định nghĩa không gian siêu tham số cho mô hình Random Forest. Không gian này bao gồm các siêu tham số chính cần được tối ưu hóa [CT4]. Sau khi xác định không gian được, một quần thể các bộ siêu tham số được khởi tạo và sẽ tiến hóa qua các thế hệ.

Bước tiếp theo là định nghĩa hàm thích nghi để đánh giá hiệu suất của mô hình với một bộ siêu tham số cho trước. Trong thuật toán này, hàm thích nghi được phát triển dựa trên hệ số tương quan Matthews (MCC). Thuật toán sẽ sử dụng hàm này để chọn các bộ siêu tham số có hiệu suất tốt nhất cho quá trình sinh sản. Trong quá trình chọn lọc, các bộ siêu tham số được chọn dựa trên điểm thích nghi của chúng. Những bộ có độ thích nghi cao hơn có nhiều khả năng được chọn để thực hiện quá trình sinh sản, lai ghép và đột biến cho các thế hệ kế tiếp.

Thuật toán lặp lại quy trình trên qua nhiều thế hệ, với hy vọng sau mỗi thế hệ sẽ tìm ra các bộ siêu tham số có độ thích nghi tốt hơn. Thuật toán dừng lại khi đạt được tiêu chí định trước, có thể là số thế hệ tối đa hoặc mức độ thích nghi thỏa mãn.

Một số ký hiệu sử dụng trong thuật toán:

- $Chr_{(i)}$: Cá thể thứ i trong quần thể.
- Eval: Đánh giá.
- F_i : Giá trị hàm thích nghi của cá thể thứ i trong quần thể hiện tại.
- F_{pop} : Giá trị hàm thích nghi nhỏ nhất trong quần thể hiện tại.
- F_{min} : Giá trị hàm thích nghi nhỏ nhất trong toàn cục.
- M : Số cá thể (số bộ siêu tham số) trong quần thể (POP).
- P_C : Xác suất để các cặp lai ghép cá thể trong quần thể lai ghép với nhau.
- P_p : Xác suất lựa chọn điểm lai ghép.
- P_M : Xác suất đột biến.
- G : Số thế hệ tiến hóa.

3.2.3.2. Quy trình thực hiện thuật toán GA4RF

a. Khởi tạo quần thể

Mỗi cá thể trong quần thể đại diện cho một giải pháp tiềm năng trong bài toán tối ưu hóa. Thuật toán chỉ có thể hoạt động hiệu quả khi cấu trúc nhiễm sắc thể được thiết kế phù hợp, điều này ảnh hưởng đáng kể đến không gian tìm kiếm. Do đó, ảnh hưởng đến hiệu quả và kết quả của quá trình tối ưu.

Việc hiểu rõ và tìm được giá trị tối ưu hóa của các siêu tham số không chỉ giúp cải thiện độ chính xác của mô hình Random Forest mà còn tăng khả năng tổng quát hóa của mô hình trên dữ liệu mới. Do đó, quá trình điều chỉnh siêu tham số là một bước quan trọng và không thể bỏ qua trong việc xây dựng và triển khai các mô hình học máy trong thực tế.

Trong đề tài này, NCS đã thực hiện đánh giá mức độ ảnh hưởng của năm siêu tham số đối với hiệu suất mô hình RF dựa trên hai bộ dữ liệu là MobiAct v2.0 và UP-Fall. Các siêu tham số được sử dụng để đánh giá là $max_features$ (M_F), max_depth (M_D), $min_samples_leaf$ (m_L), $min_samples_split$ (m_S), num_tree (N_T) (Phụ lục 1).

Quá trình thực nghiệm cho thấy việc điều chỉnh cùng lúc hai siêu tham số $min_samples_leaf$ và $min_samples_split$ làm cho hiệu suất của RF giảm mạnh so với giá trị mặc định. Ngoài ra, khi kết hợp điều chỉnh $min_samples_leaf$ hoặc $min_samples_split$ với max_depth (M_D), hoặc/và num_tree (N_T) hoặc/và $max_features$ (M_F) thì hiệu suất của RF cũng không tốt. Ngược lại, hiệu suất của RF tốt nhất khi kết hợp điều chỉnh đồng thời num_tree và $max_features$ hoặc kết hợp max_depth với num_tree và $max_features$ [CT4].

Việc xác định tổ hợp tối ưu của các siêu tham số với không gian tìm kiếm của tất cả các siêu tham số này là một thách thức lớn. Dựa vào kết quả đã khảo sát [CT4], NCS đã thu hẹp không gian tìm kiếm với cấu trúc nhiệm sắc thể gồm ba siêu tham số là số lượng cây N_T , độ sâu tối đa M_D và số lượng đặc trưng M_F dưới dạng:

$$Chr_{(i)} = \{M_F, N_T, M_D\} \quad (3.1)$$

Trong đó, N_T có thể cải thiện độ chính xác bằng cách nắm bắt các mẫu phức tạp hơn trong dữ liệu hoạt động, M_D ảnh hưởng đến mối quan hệ phức tạp giữa các đặc trưng và hoạt động cụ thể, và M_F giới hạn các đặc trưng được xem xét trong quá trình chia cây để cân bằng giữa độ chính xác và quá khớp, đặc biệt khi xử lý dữ liệu cảm biến như gia tốc kế và con quay hồi chuyển.

Trước khi thực hiện các bước chọn lọc, lai ghép và đột biến, giá trị của các siêu tham số ở dạng thập phân sẽ được chuyển sang mã nhị phân để phục vụ cho quá trình tiến hóa. Mỗi siêu tham số được mã hóa thành số bit dựa trên phạm vi giá trị cho phép của nó. Tổng chiều dài của nhiễm sắc thể được mã hóa là 22 bit như minh họa trong Hình 3.2. Sau khi hoàn thành quá trình tiến hóa, các giá trị mã nhị phân được chuyển ngược lại thành dạng thập phân và được sử dụng để cấu hình cho mô hình RF.

b. Huấn luyện mô hình Random Forest

Bước này sử dụng các “cá thể” (tức là các bộ siêu tham số) đã tạo ra ở bước trước để tiến hành phân loại các hoạt động. Mỗi bộ siêu tham số sẽ tương ứng với một lần huấn luyện mô hình. Kết quả sẽ thu được M giá trị MCC tương ứng với M bộ siêu tham số đã sử dụng, sau đó tiến hành chuẩn hóa các giá trị MCC này như Biểu thức 3.2 để đảm bảo chúng nằm trong cùng một thang đo (từ 0 đến 1), giúp thuận tiện cho các bước tính toán tiếp theo.

$$N_{MCC(i)} = \frac{MCC(i) + 1}{2} \quad (3.2)$$

c. Đánh giá độ thích nghi của mỗi cá thể

Độ thích nghi của từng bộ siêu tham số (cá thể) trong quần thể được đánh giá theo Biểu thức 3.3. Giá trị F_i càng nhỏ, cá thể có độ thích nghi càng tốt. Chi tiết về phương pháp xây dựng hàm thích nghi được trình bày tại Mục 3.2.3.3.

$$F_i = 1 + k \cdot \log_2(1 - N_{MCC(i)}) \quad (3.3)$$

Trong đó,

- F_i : Giá trị thích nghi của cá thể thứ i trong một quần thể.
- k : Hệ số hiệu chỉnh.
- $N_{MCC(i)}$: Hệ số tương quan chuẩn hóa của cá thể thứ i trong một quần thể.

d. Kiểm tra điều kiện dừng

(1) So sánh giá trị F_{pop} và F_{min} :

- + Nếu $F_{pop} < F_{min}$, thì cập nhật $F_{min} = F_{pop}$ sau đó kiểm tra các điều kiện dừng.
- + Ngược lại, $F_{pop} \geq F_{min}$ thì kiểm tra các điều kiện dừng.

(2) Các điều kiện dừng:

- + *Điều kiện dừng 1: $F_{min} = 0$*
 - Nếu $F_{min} = 0$, đã tìm thấy bộ siêu tham số tối ưu, thuật toán kết thúc.
 - Ngược lại, $F_{min} \neq 0$, kiểm tra điều kiện dừng 2
- + *Điều kiện dừng 2: $F_{min} - F_{pop} < \delta$*
 - Nếu $F_{min} - F_{pop} < \delta$ (với δ là một ngưỡng nhỏ xác định trước), bộ siêu tham số tương ứng với F_{min} là tối ưu, thuật toán kết thúc.
 - Ngược lại, $F_{min} - F_{pop} \geq \delta$, kiểm tra điều kiện dừng 3.
- + *Điều kiện dừng 3: $G =$ số thế hệ đã tiến hóa*
 - Nếu đã tiến hóa đủ số thế hệ theo yêu cầu, bộ siêu tham số tương ứng với F_{min} là tối ưu, thuật toán kết thúc.
 - Ngược lại, G nhỏ hơn số thế hệ đã tiến hóa, thực hiện các bước chọn lọc, lai ghép và đột biến để tạo thế hệ tiếp theo sau đó quay lại vòng lặp tiến hóa.

e. Chọn lọc cá thể

- + Trong bước này có nhiều phương pháp được sử dụng để lựa chọn các cá thể (bộ điều tham số) từ quần thể hiện tại để tạo ra thế hệ tiếp theo. Mục tiêu của việc chọn lọc là giữ lại những cá thể có chất lượng tốt (độ thích nghi cao) và loại bỏ những cá thể kém chất lượng. Luận án này sử dụng phương pháp “bánh xe roulette” để lựa chọn các bộ siêu tham số, quy trình lựa chọn được thực hiện như sau:

- (1) Tính nghịch đảo của F_i (Giá trị thích nghi của từng bộ siêu tham số ở Bước c. Mục đích là để bộ siêu tham số có giá trị fitness càng nhỏ thì xác suất được lựa chọn càng cao).

$$F'_i = \frac{1}{F_i} \quad (3.4)$$

Ở đây, khi $F_i = 0$ thì thuật toán GA4RF thỏa mãn điều kiện dừng nên không xảy ra trường hợp biểu thức có phép chia cho 0.

- (2) Tính tổng giá trị F'_i các bộ siêu tham số của quần thể:

$$S' = \sum_{i=1}^M F'_i \quad (3.5)$$

Trong đó, - M là số cá thể (số bộ siêu tham số) trong quần thể (POP)

- F'_i là giá trị nghịch đảo F_i của từng bộ siêu tham số.

- (3) Tính xác suất lựa chọn ($P(i)$) cho mỗi cá thể (cá thể có giá trị fitness nhỏ thì fitness nghịch đảo sẽ lớn, do đó có xác suất được chọn cao hơn):

$$P(i) = \frac{F'_i}{S'} \quad (3.6)$$

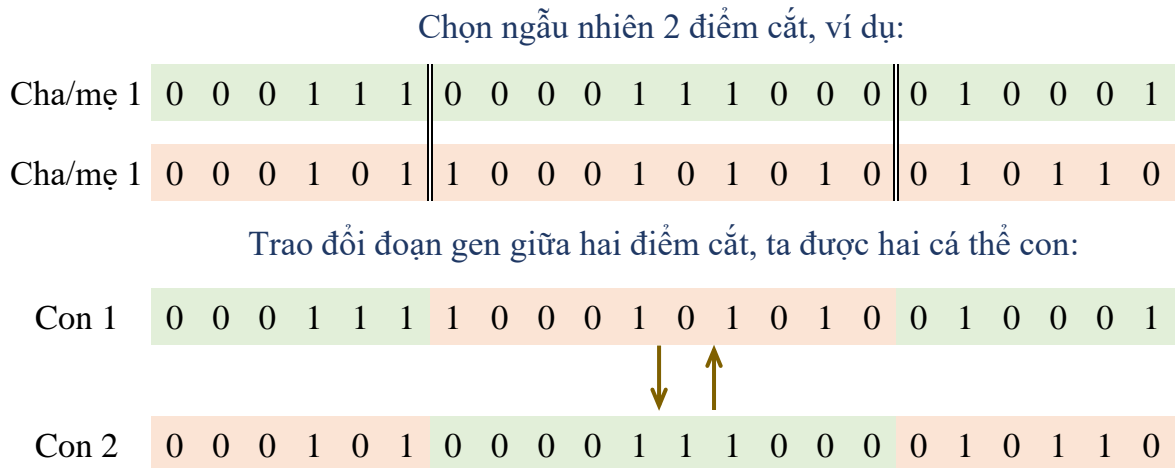
- (4) Phân chia và quay bánh xe roulette để chọn cá thể theo nguyên tắc:

- Mỗi đoạn bánh xe đại diện cho một bộ siêu tham số. Độ lớn của mỗi đoạn tương ứng với xác suất chọn ($P(i)$) của cá thể đó. Những bộ siêu tham số có fitness nhỏ hơn sẽ chiếm đoạn lớn hơn và do đó có khả năng được chọn cao hơn.
- Mỗi bộ siêu tham số có thể được chọn nhiều hơn một lần. Bộ siêu tham số tốt có nhiều cơ hội tạo ra các bản sao ở thế hệ sau hơn.
- Một con trỏ ngẫu nhiên sẽ được quay trên bánh xe. Vị trí mà con trỏ dừng lại sẽ quyết định bộ siêu tham số nào được chọn.

f. Lai ghép

- (1) Lựa chọn xác suất lai ghép P_C (tỉ lệ phân trăm các cá thể có cơ hội thực hiện lai ghép với nhau, thường chọn 60% đến 90%)
- (2) Chọn ngẫu nhiên 2 cá thể trong quần thể đã chọn ở Bước e.
- (3) Chọn ngẫu nhiên hai vị trí trong chuỗi mã hóa của các cá thể cha mẹ, được gọi là điểm cắt.
- (4) Đoạn gen nằm giữa hai điểm cắt của cá thể cha/mẹ thứ nhất được trao đổi với đoạn gen tương ứng của cá thể cha/mẹ thứ hai để tạo ra hai cá thể con mới.

Ví dụ: Có hai cá thể cha mẹ với chuỗi gen như sau:



Hình 3.2. Quá trình lai ghép trong GA.

g. Đột biến

- (1) Chọn ngẫu nhiên khoảng 1% đến 5% số lượng cá thể trong quần thể đã thực hiện ở Bước f để tạo đột biến.
- (2) Chọn hai vị trí ngẫu nhiên trong chuỗi mã hóa của cá thể. Các vị trí này không nhất thiết phải liên tiếp và có thể nằm bất kỳ đâu trong chuỗi.
- (3) Thực hiện phép lật bit tại các vị trí được chọn, cụ thể bit 0 được đổi thành 1 và bit 1 được đổi thành 0.

	Các vị trí đột biến (giả định)																							
Trước đột biến	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1		
Sau đột biến	0	0	0	1	0	1	1	0	0	0	1	0	1	0	1	0	1	0	1	1	0	1	1	0

h. Cập nhật quần thể mới, đánh giá và kết thúc vòng lặp

Các cá thể con được tạo ra sau quá trình lai ghép và các cá thể đã trải qua đột biến sẽ hình thành nên quần thể mới. Quy trình sẽ quay lại Bước b để đánh giá quần thể mới này, và tiếp tục lặp lại cho đến khi một trong ba điều kiện dừng (như đã trình bày ở Bước d) được thỏa mãn, khi đó thuật toán sẽ kết thúc. Kết quả cuối cùng là tìm được bộ siêu tham số có giá trị hàm thích nghi nhỏ nhất.

3.2.3.3. Hàm thích nghi

Một thách thức lớn trong việc tìm siêu tham số tối ưu là phát triển hàm thích nghi phù hợp với thuật toán tìm kiếm. Một lựa chọn phổ biến cho hàm thích nghi là dựa trên các chỉ số hiệu suất như F1-Score, độ chính xác (Accuracy) hoặc độ chính xác dương tính (Precision). Tuy nhiên, các chỉ số này có thể không hiệu quả để định nghĩa hàm thích nghi cho bài toán tối ưu hóa này do gặp vấn đề với các bộ dữ liệu mất cân bằng.

Trong bài toán phát hiện ngã, dữ liệu về các sự kiện ngã (dương tính) thường khan hiếm so với dữ liệu của các hoạt động bình thường (âm tính), dẫn đến sự mất cân bằng dữ liệu nghiêm trọng. Để mô hình dự đoán không thiên vị các lớp đa số và tăng khả năng phân loại các sự kiện ngã, luận án đã đặt ra hai mục tiêu chính để xây dựng hàm thích nghi.

Thứ nhất, cân bằng hiệu suất dự đoán giữa các lớp dương tính và âm tính: Hàm thích nghi khuyến khích mô hình dự đoán chính xác cả các sự kiện ngã và các hoạt động bình thường, bằng cách sử dụng tất cả các yếu tố trong ma trận nhầm lẫn (TP, TN, FP, FN). Cách tiếp cận này giúp mô hình tránh thiên vị đối với lớp có lượng dữ liệu lớn hơn (thường là các hoạt động bình thường), đồng thời đánh giá hiệu suất của mô hình một cách chính xác và toàn diện hơn.

Thứ hai, cải thiện khả năng phân biệt giữa các giá trị dự đoán gần nhau: Khi mô hình đạt hiệu suất cao, khả năng phân biệt giữa các giá trị dự đoán gần nhau trở nên quan trọng hơn. Để cải thiện khả năng này, hàm thích nghi được thiết kế để phân biệt tốt hơn giữa các giá trị dự đoán gần nhau, đặc biệt là khi các giá trị hiệu suất đạt mức cao. Cụ thể, hàm thích nghi được xây dựng dựa trên hai yếu tố chính:

- (1) *Sử dụng MCC để xử lý dữ liệu mất cân bằng:* Trong các bài toán phân loại với dữ liệu không cân bằng, các thước đo truyền thống như Accuracy hay F1-score không phản ánh đúng hiệu suất của mô hình. Accuracy không phân biệt được giữa các loại lỗi như False Positives (FP) và False Negatives (FN), do đó không cung cấp cái nhìn toàn diện về hiệu suất mô hình trên các lớp khác nhau. F1-score, mặc dù tập trung vào lớp dương tính qua việc cân bằng Precision và Recall, nhưng lại không xem xét đến True Negatives (TN), nên cũng không đánh giá được toàn bộ hiệu suất của mô hình. Hệ số tương quan Matthews (MCC) được sử dụng làm thước đo chính trong hàm thích nghi vì nó cung cấp một đánh giá toàn diện, cân nhắc tất cả các yếu tố trong ma trận nhầm lẫn và không bị ảnh hưởng bởi sự mất cân bằng dữ liệu. MCC giúp đánh giá hiệu suất của mô hình một cách khách quan và chính xác hơn, đặc biệt trong các bài toán phân loại dữ liệu mất cân bằng.
- (2) *Sử dụng logarit để tăng khả năng phân biệt giữa các giá trị fitness gần nhau:* Mặc dù MCC là một thước đo mạnh mẽ, nhưng khi đạt giá trị cao, MCC có thể chưa đủ nhạy để phân biệt giữa các giá trị gần nhau. Để khắc phục điểm yếu này, hàm thích nghi được bổ sung thành phần logarit, nhằm tăng khả năng phân biệt giữa các giá trị fitness gần nhau, giúp mô hình phản ứng nhạy hơn với các thay đổi nhỏ trong hiệu suất, đặc biệt khi tối ưu hóa trong giai đoạn cuối của quá trình tiến hóa.

a. Chuẩn hóa hệ số tương quan Matthew (MCC)

Trong nghiên cứu này, MCC được sử dụng làm thước đo chính cho hàm thích nghi trong giải thuật di truyền (GA). Việc lựa chọn MCC dựa trên các nghiên cứu của

Chicco D. và Jurman G., được công bố trong công trình [43]. Để làm rõ lý do lựa chọn MCC, phần này NCS tóm tắt những nội dung liên quan từ nghiên cứu [43] mà luận án đã áp dụng.

Ma trận nhầm lẫn là công cụ quan trọng trong việc đánh giá chất lượng của các mô hình phân loại, vì nó cung cấp cái nhìn chi tiết về các dự đoán đúng và sai của mô hình cho từng lớp. Do đó, việc sử dụng một thước đo thống nhất, có khả năng phản ánh chính xác chất lượng dự đoán là điều cần thiết.

Hai thước đo phổ biến thường được sử dụng trong đánh giá mô hình phân loại là Độ chính xác (Accuracy) và Điểm trung bình điều hòa (F1-score). Tuy nhiên, cả hai thước đo này có thể dẫn đến kết quả đánh giá sai lệch khi áp dụng trên các tập dữ liệu không cân bằng, do chúng không xem xét đầy đủ tỷ lệ giữa các mẫu thuộc lớp dương tính và âm tính. Accuracy và F1-score chỉ đáng tin cậy khi áp dụng cho các tập dữ liệu cân bằng, trong đó các lớp có số lượng mẫu tương đương.

Ngược lại, MCC không bị ảnh hưởng bởi sự mất cân bằng giữa các lớp trong dữ liệu, chẳng hạn như khi số lượng mẫu ngã (dương tính) ít hơn rất nhiều so với các mẫu không ngã (âm tính). MCC cung cấp một thước đo toàn diện và công bằng, yêu cầu mô hình phân loại phải dự đoán chính xác cả hai nhóm - mẫu âm tính và mẫu dương tính - để đạt được điểm số cao, bất kể tỷ lệ mẫu giữa các nhóm trong toàn bộ tập dữ liệu. Đây là ưu điểm quan trọng khiến MCC được lựa chọn khi làm việc với các tập dữ liệu không cân bằng như MobiAct v2.0 và UP-Fall.

Trong bài toán phân loại nhị phân, tập dữ liệu mô tả bài toán gồm s^+ mẫu dữ liệu thuộc lớp 1, gọi là dương tính, và s^- mẫu dữ liệu thuộc lớp 2, gọi là âm tính. Ví dụ, trong bài toán phát hiện ngã, các hoạt động thường ngày bình thường được gán nhãn âm tính, trong khi nhãn dương tính sẽ được gán cho các sự kiện ngã.

Mô hình phân loại dự đoán lớp của mỗi mẫu dữ liệu, gán nhãn dự đoán (dương tính hoặc âm tính) cho từng mẫu. Khi kết thúc quá trình phân lớp, mỗi mẫu sẽ thuộc một

trong bốn trường hợp TP, FP, TN, FN như đã trình bày ở Mục 1.2.2.3.a. Sự phân chia này có thể được biểu diễn dưới dạng bảng 2x2 gọi là ma trận nhầm lẫn (Bảng 1.1)

$$M = \begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}, \text{ trong đó } s^+ = TP + FN, s^- = FP + TN \text{ [43]}$$

Mục tiêu phân loại nhị phân bằng mô hình học máy là phân biệt giữa các mẫu dương tính thật (TP) và âm tính thật (TN) càng chính xác càng tốt, đồng thời giảm thiểu các trường hợp âm tính giả (FN) và dương tính giả (FP). Mô hình phân loại được coi là hoàn hảo khi $M = \begin{pmatrix} s^+ & 0 \\ 0 & s^- \end{pmatrix}$.

Giá trị của MCC nằm trong khoảng $[-1, +1]$, với:

- + $MCC = +1$: Phân loại hoàn hảo (tất cả các mẫu được dự đoán chính xác).
- + $MCC = -1$: Phân loại sai hoàn toàn (tất cả các mẫu được dự đoán sai).
- + $MCC = 0$: Thể hiện hiệu suất dự đoán của bộ phân loại mang tính ngẫu nhiên.

Trong trường hợp ma trận M có một phần tử khác không, nghĩa là tất cả các mẫu trong tập dữ liệu thuộc về một lớp, và chúng được phân loại đúng (đối với $TP \neq 0$ hoặc $TN \neq 0$, lúc này $MCC = 1$) hoặc phân loại sai (đối với $FP \neq 0$ hoặc $FN \neq 0$, lúc này $MCC = -1$).

MCC không xác định $\left(\frac{0}{0}\right)$ khi các phần tử trên cùng một hàng hoặc một cột của ma trận M có giá trị bằng 0, (M thuộc một trong bốn dạng $M = \begin{pmatrix} a & 0 \\ b & 0 \end{pmatrix}$, $M = \begin{pmatrix} 0 & a \\ 0 & b \end{pmatrix}$, $M = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix}$, $M = \begin{pmatrix} 0 & 0 \\ a & b \end{pmatrix}$ với $a, b \geq 1:n$ trong cả bốn trường hợp).

Nếu thay giá trị của các phần tử bằng 0 trong các ma trận trên bằng giá trị ϵ (epsilon) rất nhỏ, chúng ta thu được MCC như trình bày ở Biểu thức (3.7) [43]:

$$\begin{aligned} MCC &= \frac{a \cdot \epsilon - b \cdot \epsilon}{\sqrt{(a+b)(a+\epsilon)(b+\epsilon)(\epsilon+\epsilon)}} = \frac{\epsilon}{\sqrt{\epsilon}} \frac{a - b}{\sqrt{2(a+b)(a+\epsilon)(b+\epsilon)}} \\ &\approx \sqrt{\epsilon} \frac{a - b}{\sqrt{2ab(a - b)}} \rightarrow 0 \text{ khi } \epsilon \rightarrow 0 \end{aligned} \quad (3.7)$$

Với ϵ rất nhỏ, giá trị MCC thu được cho bốn trường hợp đặc biệt này xấp xỉ bằng 0. Điều này phù hợp với trực quan, vì cả bốn trường hợp đều đại diện cho các mô hình phân loại kém, dự đoán tất cả các mẫu thuộc một lớp hoặc không dự đoán được bất kỳ mẫu nào.

Trong bài toán phân loại đa lớp, MCC cũng giữ vai trò quan trọng nhờ khả năng cung cấp một đánh giá toàn diện về hiệu suất của mô hình, tương tự như trong phân loại nhị phân. Tuy nhiên, để áp dụng MCC cho phân loại đa lớp, chúng ta cần mở rộng cách tính MCC để phù hợp với nhiều lớp. Với bài toán phân loại nhị phân, ma trận nhầm lẫn có 4 hạng mục (TP, TN, FP, FN). MCC sử dụng tất cả các hạng mục này để tính toán điểm số, đảm bảo rằng mô hình được đánh giá dựa trên khả năng dự đoán chính xác cả lớp dương tính và lớp âm tính. Trong bài toán phân loại đa lớp, ma trận nhầm lẫn sẽ có kích thước lớn hơn ($C \times C$, với C là số lớp). MCC đa lớp cũng sẽ xem xét tất cả các hạng mục trong ma trận này, đảm bảo đánh giá hiệu suất trên tất cả các lớp, không chỉ tập trung vào một lớp cụ thể.

Để đảm bảo cùng một thang đo từ 0 đến 1 như F1-score hay Accuracy, khoảng giá trị $[-1, +1]$ của MCC được ánh xạ tuyến tính vào khoảng giá trị $[0, 1]$, như Biểu thức:

$$N_{\text{MCC}} = \frac{\text{MCC} + 1}{2} \quad (3.8)$$

- $N_{\text{MCC}} = 1$, mô hình phân loại hoàn hảo (các mẫu được dự đoán chính xác).
- $N_{\text{MCC}} = 0$, mô hình phân loại sai hoàn toàn (tất cả các mẫu được dự đoán sai).
- $N_{\text{MCC}} = \frac{1}{2}$, thể hiện hiệu suất dự đoán của bộ phân loại mang tính ngẫu nhiên.

b. Xây dựng hàm thích nghi

Hàm fitness (hàm đánh giá độ thích nghi) đóng vai trò quan trọng trong giải thuật di truyền (GA), giúp đánh giá hiệu suất của từng cá thể (tương ứng với một bộ siêu tham số) và hướng dẫn quá trình tìm kiếm giải pháp tối ưu. Có hai cách tiếp cận chính để xây dựng hàm thích nghi: tìm giá trị cực tiểu (giá trị fitness càng thấp, hiệu suất mô

hình càng cao) hoặc tìm giá trị cực đại (giá trị fitness càng cao, hiệu suất mô hình càng cao). Trong luận án này, NCS chọn phương pháp tìm giá trị cực tiểu để xây dựng hàm fitness. Có nhiều cách để xây dựng hàm thích nghi cho thuật toán di truyền, một cách đơn giản chúng ta có thể tính như sau:

$$F_i = 1 - N_{MCC(i)} \quad (3.9)$$

Trong đó, $N_{MCC(i)}$ là kết quả được tính theo hệ số tương quan MCC chuẩn hoá của bộ siêu tham số thứ i trong một quần thể. Hàm này có mối quan hệ tuyến tính trực tiếp với giá trị $N_{MCC(i)}$. Giá trị thích nghi F_i giảm đều từ 1 về 0 mà không có sự biến động lớn, giúp tránh được những dao động không mong muốn trong quá trình tối ưu hóa. Tuy nhiên, hàm này có một số hạn chế như không có điểm tối ưu rõ ràng và không thể hiện rõ sự phân biệt giữa các giá trị $N_{MCC(i)}$ khác nhau.

Để khắc phục những hạn chế này, luận án phát triển một hàm thích nghi mới dựa trên MCC với mục tiêu tối thiểu hóa giá trị của nó và có độ phân giải cao tại các giá trị N_{MCC} lớn (gần bằng 1). Điều này có nghĩa là hàm F_i sẽ: (1) trả về giá trị thấp hơn cho các giải pháp tốt hơn và (2) tăng cường khả năng phân biệt giữa các giá trị N_{MCC} với biến thiên nhỏ nhất. Do đó, NCS đề xuất hàm thích nghi cho thuật toán GA4RF dựa trên MCC như sau:

$$F_i = 1 + k \cdot \log_2(1 - N_{MCC(i)}) \quad (3.10)$$

Trong đó, k là hệ số hiệu chỉnh và $N_{MCC(i)}$ là hệ số tương quan chuẩn hoá cho một bộ siêu tham số thứ i trong quần thể tương ứng. Với mỗi quần thể sẽ tìm ra một giá trị thích nghi nhỏ nhất là F_{pop} :

$$F_{pop} = \min[F_i] \quad (3.11)$$

Ví dụ: Để F_i tiến về 0 khi $N_{MCC} = 0.9999$ (tránh trường hợp $N_{MCC} = 1 \leftrightarrow \log_2(0)$):

$$F_i = 1 + k \cdot \log_2(1 - N_{MCC(i)}) = 0$$

Thay $N_{MCC(i)} = 0.9999$ vào phương trình:

$$1 + k \cdot \log_2(1 - 0.9999) = 0$$

Giải phương trình trên, ta được: $k = -\frac{1}{\log_2(0.0001)} \approx 0.07526$

Trong Biểu thức 3.10, đại lượng $\log_2(1 - N_{MCC(i)})$ đóng vai trò quan trọng trong việc định hình hàm thích nghi, giúp khuếch đại sự khác biệt giữa các giá trị $N_{MCC(i)}$. Việc này làm cho mô hình nhạy cảm hơn với những thay đổi nhỏ trong giá trị MCC, đặc biệt khi MCC đạt giá trị cao, từ đó tối ưu hóa hiệu suất mô hình phát hiện ngã một cách toàn diện và chính xác hơn.

Hình 3.3 minh họa mối quan hệ giữa hàm thích nghi và giá trị MCC chuẩn hóa (N_{MCC}) trước và sau khi áp dụng hàm logarit. Biểu đồ cho thấy cách hàm logarit được sử dụng để khuếch đại sự khác biệt giữa các giá trị fitness, đặc biệt khi giá trị N_{MCC} tiến gần đến 1 (mô hình đạt hiệu suất phân loại cao).

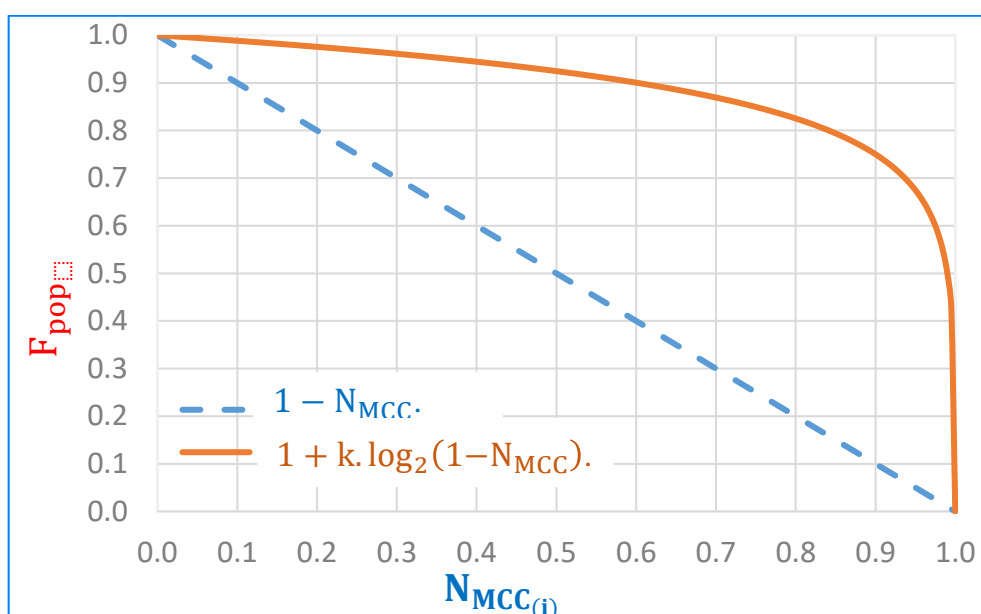
Khi N_{MCC} càng tiến gần đến 1, giá trị F_i sẽ dần tiến về 0. Điều này có nghĩa là hàm fitness sẽ giảm dần khi N_{MCC} tăng, nhưng sự thay đổi này không đơn giản là tuyến tính mà được khuếch đại hoặc giảm bớt tùy theo giá trị N_{MCC} . Sự thay đổi phi tuyến này giúp nhấn mạnh sự khác biệt giữa các giá trị thích nghi F_i khi N_{MCC} cao, làm cho quá trình tối ưu hóa nhạy cảm hơn với những cải tiến nhỏ về hiệu suất.

Việc áp dụng hàm logarit vào hàm thích nghi giúp làm rõ sự khác biệt giữa các giá trị F_i gần nhau. Khi N_{MCC} rất gần với 1, mô hình có thể đạt hiệu suất rất cao, nhưng sự khác biệt giữa các mô hình có thể trở nên nhỏ và khó phân biệt nếu chỉ dựa trên giá trị N_{MCC} gốc. Hàm logarit giúp khếch đại sự khác biệt này, làm cho quá trình tối ưu hóa trong GA trở nên nhạy cảm hơn với các cải thiện nhỏ, giúp thuật toán phân biệt rõ hơn giữa các mô hình có hiệu suất tương đối gần nhau.

Khi N_{MCC} đã đạt mức cao, các mô hình có thể trông không khác biệt nhiều nếu chỉ dựa trên các giá trị N_{MCC} trực tiếp, dẫn đến hiện tượng bão hòa, nơi thuật toán có thể dừng tối ưu. Việc sử dụng hàm logarit giúp tránh hiện tượng này bằng cách duy trì

khả năng phân biệt giữa các giá trị fitness và thúc đẩy quá trình tối ưu hóa tiếp tục, ngay cả khi đã đạt được các giá trị gần tối ưu.

Việc kết hợp N_{MCC} và hàm logarit trong hàm thích nghi là một cải tiến quan trọng, giúp thuật toán GA4RF tìm kiếm hiệu quả hơn các siêu tham số tối ưu cho mô hình RF, đặc biệt trong bối cảnh dữ liệu không cân bằng và yêu cầu phân biệt rõ ràng giữa các mô hình có hiệu suất cao.



Hình 3.3. Ảnh hưởng của hàm logarit trong việc tăng cường khả năng phân biệt các giá trị fitness gần nhau (với $k = 0.1$).

3.2.4. Thực nghiệm và đánh giá

3.2.4.1. Mô tả thực nghiệm

Quy trình tổ chức các thực nghiệm đánh giá tính hiệu quả của các thuật toán GA4RF được thực hiện như sau:

- (1) Để thiết lập không gian tìm kiếm cho GA4RF trước tiên cần xác định các siêu tham số và giới hạn của chúng. Sau khi đánh giá ảnh hưởng của từng siêu tham số đến hiệu suất của mô hình RF trên hai bộ dữ liệu MobiAct v2.0 và UP-Fall, NCS nhận thấy các siêu tham số và khoảng giá trị của chúng phù hợp cho từng

siêu tham số lần lượt là $M_F \in [0, 50]$, $N_T \in [50, 800]$ và $M_D \in [0, 50]$. Những khoảng giá trị này sẽ được sử dụng để thiết lập không gian tìm kiếm cho GA4RF, giúp thuật toán tìm kiếm các giải pháp tối ưu trong phạm vi đã xác định.

- (2) Từ không gian đã thiết lập, khởi tạo ngẫu nhiên một quần thể gồm 100 cá thể. Mỗi cá thể tương ứng với một bộ gồm ba siêu tham số $Chr = \{M_F, N_T, M_D\}$.
- (3) Từng bộ siêu tham số do GA tìm được sẽ cập nhật để huấn luyện mô hình RF. Độ thích nghi của từng cá thể được kiểm tra bằng hàm F_i ở Biểu thức (3.10). Thuật toán sẽ tính toán để tìm ra giá trị thích nghi tốt nhất của quần thể (F_{pop}) theo Biểu thức (3.11)
- (4) Sau khi đánh giá được 100 cá thể của quần thể ban đầu, phương pháp bánh xe roulette sẽ được áp dụng để chọn ra 60% bộ siêu tham số (Chr) tốt nhất (xét theo thước đo MCC) cho quá trình sinh sản tạo quần thể mới.
- (5) Quá trình sinh sản gồm có giai đoạn lai ghép và giai đoạn đột biến với tỉ lệ lai ghép là 0.8 và tỉ lệ đột biến là 0.1.
- (6) Quá trình tìm kiếm bộ siêu tham số tối ưu sẽ dừng lại khi đáp ứng một trong các tiêu chí: (1) $F_{pop} = 0$ hoặc (2) F_{pop} đạt đến một ngưỡng mục tiêu hoặc (3) trải qua 10.000 thế hệ tiến hóa.

Các tham số chính của GA sử dụng trong thực nghiệm được tóm tắt ở Bảng 3.1.

Bảng 3.1. Giá trị các tham số cài đặt cho GA

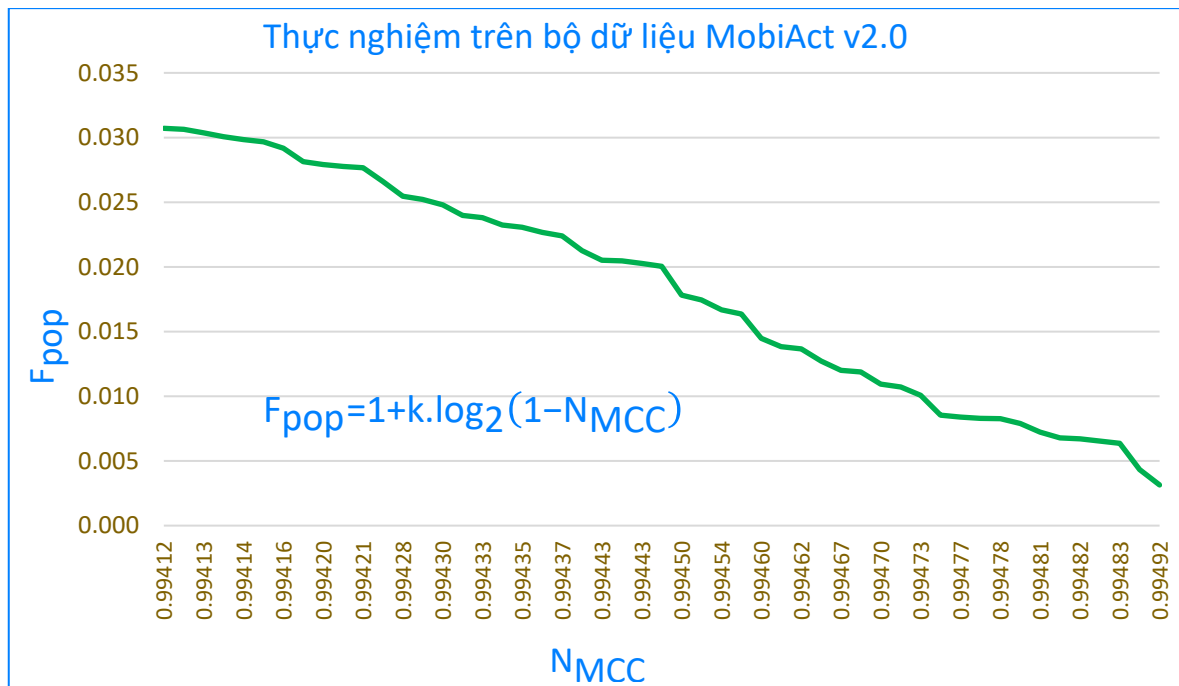
Tên tham số	Giá trị
Nhiễm sắc thể (Chr)	$\{M_F, N_T, M_D\}$
Kích thước quần thể	100
Số lần lặp lại tối đa	10000
Tỉ lệ lai ghép	0.8
Tỉ lệ đột biến	0.1

3.2.4.2. Kết quả và bàn luận

a. Đánh giá khả năng hội tụ của thuật toán GA4RF

Khả năng hội tụ của thuật toán GA4RF đóng vai trò quan trọng, phản ánh hiệu quả trong việc tối ưu hóa siêu tham số của mô hình Random Forest. Chúng tôi đã tiến hành các thí nghiệm để đánh giá hiệu suất của thuật toán GA4RF trên hai bộ dữ liệu là MobiAct v2.0 và UP-Fall. Kết quả thực nghiệm trình bày trong Hình 3.4 và Hình 3.5 cho thấy GA4RF có khả năng hội tụ tốt trên cả hai bộ dữ liệu này.

Hình 3.4 biểu diễn sự thay đổi của giá trị hàm thích nghi theo N_{MCC} trong 50 thế hệ cuối cùng trước khi thỏa mãn điều kiện dừng tại số lần lặp tối đa (10000 thế hệ) khi thực nghiệm trên bộ dữ liệu MobiAct v2.0. Trong các thế hệ này, bộ phân loại RF đạt được độ chính xác tương đối cao, thể hiện qua giá trị N_{MCC} lớn, tiến gần đến 1. Biểu đồ này cho thấy F_{pop} giảm dần và tiến gần về 0, chứng minh khả năng tối ưu hóa hiệu quả của GA4RF. Đồng thời, N_{MCC} tăng dần và tiến gần đến 1, phản ánh hiệu suất mô hình ngày càng tốt hơn.



Hình 3.4. Giá trị hàm thích nghi theo N_{MCC} trong 50 thế hệ cuối cùng trước khi thỏa mãn điều kiện dừng tại số lần lặp tối đa trên bộ dữ liệu MobiAct v2.0.

Trong bối cảnh bộ dữ liệu có sự mất cân bằng lớn, việc phân loại chính xác các hoạt động ngã riêng biệt trong số tất cả các hoạt động không chỉ đòi hỏi tối đa hóa giá trị N_{MCC} mà còn đảm bảo bộ phân loại hoạt động tốt khi sự khác biệt giữa các giá trị N_{MCC} ở hai thế hệ liên tiếp là tối thiểu. Do đó, việc đánh giá hiệu suất của bộ phân loại trong vùng mà N_{MCC} tiến đến giá trị tối ưu là rất quan trọng.

Bảng 3.2 trình bày kết quả thực nghiệm trên bộ dữ liệu MobiAct v2.0 của 50 thế hệ cuối cùng trước khi thuật toán GA4RF đạt điều kiện dừng thứ 3 (số thế hệ đã tiến hóa). Các chỉ số như MCC, F1-score, Precision và Recall đều thể hiện sự hội tụ ổn định của thuật toán.

Bảng 3.2. Tổng hợp kết quả thực nghiệm trên bộ dữ liệu MobiAct v2.0 của 50 thế hệ cuối cùng trước khi thuật toán GA4RF đạt điều kiện dừng.

TT	Thế hệ	MCC	F1-score	Precision	Recall	N_{MCC}	F_{pop}	F_{min}	$F_{min} - F_{pop}$	Ghi chú (M_F, M_D, N_T)
1	9951	0.98824	0.98896	0.98896	0.98898	0.99412	0.03065	0.03071	0.000066	16, 23, 81
2	9952	0.98825	0.98898	0.98898	0.98899	0.99413	0.03037	0.03065	0.000278	17, 24, 132
3	9953	0.98827	0.98902	0.98904	0.98905	0.99414	0.03008	0.03037	0.000293	4, 31, 75
4	9954	0.98829	0.98901	0.98902	0.98903	0.99414	0.02985	0.03008	0.000230	19, 38, 427
5	9955	0.98830	0.98903	0.98906	0.98907	0.99415	0.02966	0.02985	0.000184	4, 31, 83
6	9956	0.98833	0.98903	0.98903	0.98905	0.99416	0.02919	0.02966	0.000475	20, 25, 387
7	9957	0.98839	0.98912	0.98913	0.98914	0.99420	0.02814	0.02919	0.001046	22, 27, 287
8	9958	0.98841	0.98912	0.98912	0.98914	0.99420	0.02792	0.02814	0.000219	19, 38, 388
9	9959	0.98841	0.98918	0.98919	0.98921	0.99421	0.02777	0.02792	0.000153	4, 23, 91
10	9960	0.98842	0.98914	0.98914	0.98915	0.99421	0.02767	0.02777	0.000104	22, 22, 340
11	9961	0.98849	0.98921	0.98924	0.98924	0.99424	0.02661	0.02767	0.001057	4, 31, 90
12	9962	0.98856	0.98926	0.98927	0.98928	0.99428	0.02545	0.02661	0.001157	18, 24, 391
13	9963	0.98857	0.98931	0.98932	0.98933	0.99429	0.02522	0.02545	0.000233	5, 17, 425
14	9964	0.98860	0.98931	0.98932	0.98933	0.99430	0.02479	0.02522	0.000433	17, 25, 430
15	9965	0.98864	0.98935	0.98936	0.98937	0.99432	0.02399	0.02479	0.000797	18, 27, 445
16	9966	0.98866	0.98937	0.98937	0.98939	0.99433	0.02382	0.02399	0.000173	16, 37, 387
17	9967	0.98869	0.98939	0.98942	0.98942	0.99435	0.02323	0.02382	0.000590	4, 28, 91
18	9968	0.98870	0.98941	0.98944	0.98944	0.99435	0.02305	0.02323	0.000174	4, 30, 91
19	9969	0.98872	0.98941	0.98943	0.98944	0.99436	0.02268	0.02305	0.000373	0, 29, 67
20	9970	0.98874	0.98944	0.98945	0.98946	0.99437	0.02239	0.02268	0.000291	18, 37, 500
21	9971	0.98881	0.98952	0.98954	0.98955	0.99440	0.02125	0.02239	0.001143	4, 31, 95
22	9972	0.98885	0.98954	0.98956	0.98956	0.99443	0.02052	0.02125	0.000726	6, 31, 70
23	9973	0.98885	0.98955	0.98955	0.98956	0.99443	0.02046	0.02052	0.000059	17, 31, 557
24	9974	0.98887	0.98960	0.98960	0.98962	0.99443	0.02027	0.02046	0.000196	12, 30, 83

TT	Thế hệ	MCC	F1-score	Precision	Recall	N_{MCC}	F_{pop}	F_{min}	$F_{min} - F_{pop}$	Ghi chú (M_F, M_D, N_T)
25	9975	0.98888	0.98962	0.98962	0.98964	0.99444	0.02005	0.02027	0.000215	12, 30, 82
26	9976	0.98901	0.98970	0.98971	0.98972	0.99450	0.01782	0.02005	0.002226	7, 31, 115
27	9977	0.98903	0.98971	0.98971	0.98972	0.99452	0.01746	0.01782	0.000368	15, 23, 294
28	9978	0.98908	0.98976	0.98977	0.98978	0.99454	0.01668	0.01746	0.000773	12, 24, 384
29	9979	0.98909	0.98977	0.98979	0.98980	0.99455	0.01637	0.01668	0.000314	0, 31, 91
30	9980	0.98920	0.98988	0.98989	0.98990	0.99460	0.01447	0.01637	0.001900	0, 29, 123
31	9981	0.98924	0.98990	0.98991	0.98992	0.99462	0.01384	0.01447	0.000633	7, 35, 189
32	9982	0.98925	0.98994	0.98994	0.98996	0.99462	0.01367	0.01384	0.000165	12, 24, 352
33	9983	0.98930	0.98997	0.98998	0.98999	0.99465	0.01273	0.01367	0.000945	7, 38, 350
34	9984	0.98934	0.99001	0.99001	0.99003	0.99467	0.01200	0.01273	0.000721	7, 38, 390
35	9985	0.98935	0.99001	0.99002	0.99003	0.99468	0.01188	0.01200	0.000129	7, 28, 297
36	9986	0.98940	0.99004	0.99006	0.99006	0.99470	0.01094	0.01188	0.000931	7, 27, 188
37	9987	0.98942	0.99008	0.99009	0.99010	0.99471	0.01071	0.01094	0.000236	6, 35, 148
38	9988	0.98945	0.99010	0.99012	0.99012	0.99473	0.01008	0.01071	0.000630	13, 34, 537
39	9989	0.98954	0.99019	0.99019	0.99021	0.99477	0.00855	0.01008	0.001529	11, 25, 320
40	9990	0.98955	0.99019	0.99020	0.99021	0.99477	0.00840	0.00855	0.000154	11, 39, 269
41	9991	0.98955	0.99021	0.99022	0.99022	0.99478	0.00830	0.00840	0.000099	12, 37, 543
42	9992	0.98955	0.99020	0.99022	0.99022	0.99478	0.00828	0.00830	0.000018	5, 34, 384
43	9993	0.98957	0.99023	0.99024	0.99024	0.99479	0.00790	0.00828	0.000382	12, 33, 510
44	9994	0.98961	0.99026	0.99027	0.99028	0.99481	0.00722	0.00790	0.000677	8, 26, 190
45	9995	0.98963	0.99028	0.99029	0.99029	0.99482	0.00679	0.00722	0.000434	8, 26, 186
46	9996	0.98964	0.99029	0.99030	0.99031	0.99482	0.00671	0.00679	0.000081	5, 29, 529
47	9997	0.98965	0.99031	0.99032	0.99033	0.99482	0.00654	0.00671	0.000167	5, 27, 428
48	9998	0.98966	0.99033	0.99034	0.99035	0.99483	0.00637	0.00654	0.000168	5, 25, 430
49	9999	0.98977	0.99042	0.99042	0.99044	0.99488	0.00433	0.00637	0.002043	0, 26, 186
50	10000	0.98983	0.99049	0.99050	0.99051	0.99492	0.00314	0.00433	0.001185	5, 26, 420

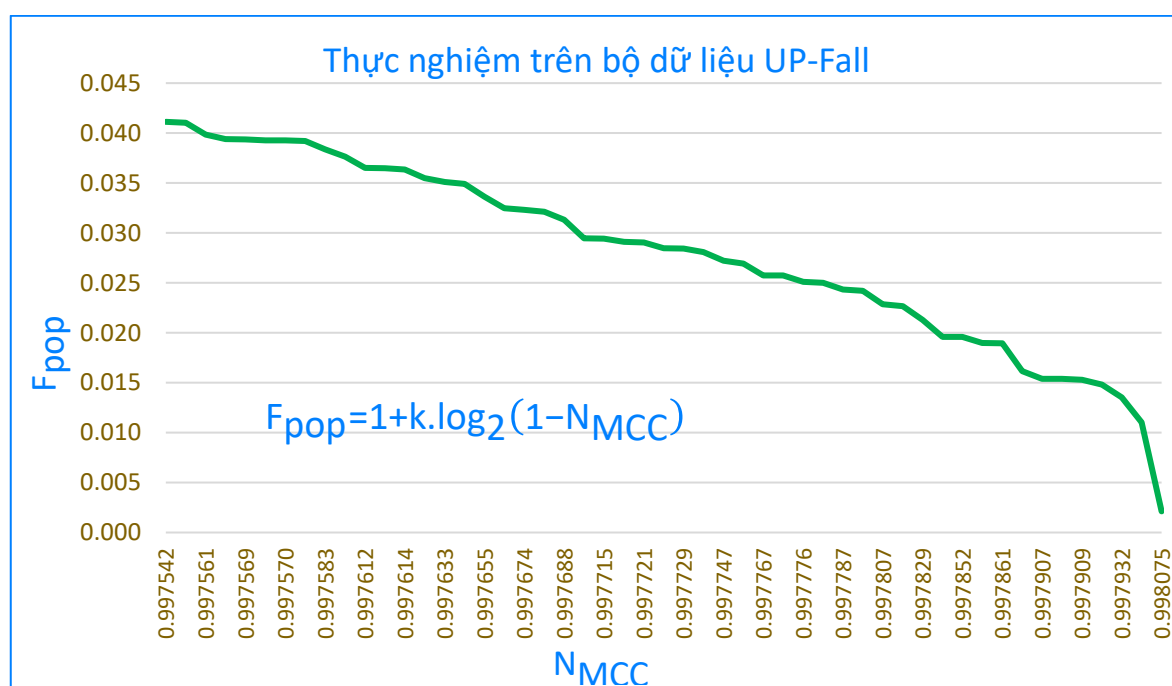
Kết quả trình bày trong Bảng 3.2 cho thấy giá trị N_{MCC} tăng lên qua mỗi thế hệ, từ 0.99412 đến 0.99492, trong khi giá trị hàm thích nghi (F_{pop}) dần giảm về 0. Điều này chứng tỏ thuật toán GA4RF đang liên tục cải thiện bộ siêu tham số qua mỗi thế hệ và tránh được việc mắc kẹt tại cực trị địa phương.

Việc lựa chọn ngưỡng δ có tác động lớn đến quá trình tìm kiếm siêu tham số tối ưu. Khi đặt ngưỡng δ lớn, thuật toán GA4RF sẽ kết thúc nhanh hơn, tiết kiệm thời gian và tài nguyên tính toán. Tuy nhiên, điều này có thể dẫn đến việc dừng quá sớm, bỏ qua các bộ siêu tham số tiềm năng tốt hơn. Ví dụ, nếu đặt $\delta = 0.005$, thuật toán có thể dừng sau chỉ 1000 thế hệ, trong khi bộ siêu tham số tối ưu có thể xuất hiện ở thế hệ

thứ 9000. Kết quả thí nghiệm cho thấy nếu đặt ngưỡng $\delta = 0.000018$ thuật toán sẽ dừng tại thế hệ thứ 9994, mà chưa đạt đến số lần lặp tối đa. Ngược lại, khi đặt ngưỡng δ nhỏ, thuật toán GA4RF có cơ hội khám phá không gian tìm kiếm kỹ lưỡng hơn, tìm ra bộ siêu tham số tối ưu hơn. Tuy nhiên, điều này cũng đồng nghĩa với việc thời gian tìm kiếm kéo dài, có thể dẫn đến tốn kém về mặt tài nguyên tính toán.

Tầm quan trọng của việc lựa chọn ngưỡng δ thể hiện ở nhiều khía cạnh. Đầu tiên, nó giúp cân bằng giữa chất lượng kết quả và thời gian thực hiện thuật toán. Thứ hai, một ngưỡng δ phù hợp có thể ngăn chặn việc tối ưu hóa quá mức, tránh hiện tượng overfitting trên tập huấn luyện. Cuối cùng, ngưỡng δ đóng vai trò quan trọng trong việc xác định điểm dừng phù hợp, đảm bảo thuật toán hội tụ về một giải pháp tốt trong thời gian hợp lý.

Sự biến đổi của giá trị hàm thích nghi dựa trên N_{MCC} trong 50 thế hệ cuối cùng trước khi đạt đến điều kiện dừng tại số lần lặp tối đa khi thực nghiệm với bộ dữ liệu UP-Fall được trình bày trong Hình 3.5.



Hình 3.5. Giá trị hàm thích nghi theo N_{MCC} trong 50 thế hệ cuối cùng trước khi thỏa mãn điều kiện dừng tại số lần lặp tối đa trên bộ dữ liệu UP-Fall.

Kết quả trên bộ dữ liệu UP-Fall có sự nhất quán với những gì quan sát được trên bộ dữ liệu MobiAct v2.0. Hình 3.5 minh họa sự giảm dần của F_{pop} và sự tăng dần của N_{MCC} . Mặc dù F_{pop} giảm dần và ổn định, nhưng có những biến động nhỏ ở các thể hệ cuối, chứng tỏ thuật toán không bị mắc kẹt tại các điểm tối ưu cục bộ.

Tương tự, Bảng 3.3 cung cấp kết quả thực nghiệm trên bộ dữ liệu UP-Fall, cũng tập trung vào 50 thể hệ cuối cùng. Các chỉ số hiệu suất MCC tiếp tục được theo dõi để đánh giá sự hội tụ của thuật toán. Tương tự kết quả thực nghiệm trên bộ MobiAct v2.0, các chỉ số MCC, F1-score, Precision và Recall đều thể hiện sự hội tụ ổn định của thuật toán. Giá trị MCC duy trì ở mức cao, từ 0.9951 đến 0.9961, và N_{MCC} tăng từ 0.997543 đến 0.998075, tiến sát đến 1, phản ánh khả năng phân loại chính xác cao của mô hình. F_{pop} giảm từ 0.041044 xuống 0.002120, cho thấy quá trình tối ưu hóa hiệu quả. F_{min} cuối cùng đạt 0.002120, nhỏ hơn so với MobiAct v2.0.

Tương tự, việc lựa chọn ngưỡng δ phù hợp là then chốt để đảm bảo thuật toán GA4RF không dừng lại quá sớm. Nếu chọn ngưỡng $\delta = 0.005$ thuật toán có thể dừng sau chỉ 1000 thể hệ, hoặc nếu $\delta = 0.000005$, thuật toán sẽ dừng sớm tại thể hệ thứ 9.957, mà không đạt đến số lần lặp tối đa. Điều này có thể dẫn đến việc bỏ lỡ các giải pháp tiềm năng và bộ siêu tham số không tối ưu.

Bảng 3.3. Tổng hợp kết quả thực nghiệm trên bộ dữ liệu UP-Fall của 51 thể hệ cuối cùng trước khi thuật toán GA4RF đạt điều kiện dừng.

TT	Thể hệ	MCC	F1-score	Precision	Recall	N_{MCC}	F_{pop}	F_{min}	$F_{min} - F_{pop}$	Ghi chú (M_F, M_D, N_T)
1	9951	0.9951	0.9955	0.9955	0.9955	0.997543	0.041044	0.041124	0.000080	10, 48, 147
2	9952	0.9951	0.9955	0.9955	0.9955	0.997561	0.039848	0.041044	0.001196	8, 11, 450
3	9953	0.9951	0.9955	0.9955	0.9955	0.997568	0.039386	0.039848	0.000461	8, 22, 303
4	9954	0.9951	0.9955	0.9955	0.9955	0.997569	0.039350	0.039386	0.000036	8, 12, 108
5	9955	0.9951	0.9955	0.9955	0.9955	0.997570	0.039257	0.039350	0.000092	9, 11, 365
6	9956	0.9951	0.9955	0.9955	0.9662	0.997570	0.039252	0.039257	0.000005	9, 20, 243
7	9957	0.9951	0.9955	0.9955	0.9955	0.997571	0.039204	0.039252	0.000048	8, 20, 602
8	9958	0.9952	0.9955	0.9955	0.9955	0.997583	0.038379	0.039204	0.000825	8, 30, 270
9	9959	0.9952	0.9956	0.9956	0.9956	0.997595	0.037631	0.038379	0.000749	14, 27, 615
10	9960	0.9952	0.9956	0.9956	0.9956	0.997612	0.036501	0.037631	0.001129	9, 12, 747

TT	Thế hệ	MCC	F1-score	Precision	Recall	N_{MCC}	F_{pop}	F_{min}	$F_{min} - F_{pop}$	Ghi chú (M_F, M_D, N_T)
11	9961	0.9952	0.9956	0.9956	0.9956	0.997612	0.036483	0.036501	0.000018	10, 30, 85
12	9962	0.9952	0.9956	0.9956	0.9956	0.997614	0.036344	0.036483	0.000139	8, 11, 320
13	9963	0.9953	0.9956	0.9956	0.9956	0.997627	0.035497	0.036344	0.000847	8, 13, 516
14	9964	0.9953	0.9956	0.9956	0.9956	0.997633	0.035093	0.035497	0.000404	8, 30, 104
15	9965	0.9953	0.9956	0.9957	0.9956	0.997635	0.034917	0.035093	0.000176	9, 14, 221
16	9966	0.9953	0.9957	0.9957	0.9957	0.997655	0.033613	0.034917	0.001304	11, 50, 409
17	9967	0.9953	0.9957	0.9958	0.9957	0.997671	0.032477	0.033613	0.001136	10, 34, 410
18	9968	0.9953	0.9957	0.9957	0.9957	0.997674	0.032306	0.032477	0.000170	10, 14, 90
19	9969	0.9954	0.9957	0.9958	0.9957	0.997676	0.032116	0.032306	0.000190	10, 30, 110
20	9970	0.9954	0.9957	0.9957	0.9957	0.997688	0.031331	0.032116	0.000785	9, 15, 151
21	9971	0.9954	0.9958	0.9959	0.9958	0.997715	0.029468	0.031331	0.001863	11, 34, 256
22	9972	0.9954	0.9958	0.9959	0.9958	0.997715	0.029419	0.029468	0.000049	10, 34, 189
23	9973	0.9954	0.9958	0.9959	0.9958	0.997720	0.029115	0.029419	0.000304	10, 10, 221
24	9974	0.9954	0.9958	0.9958	0.9958	0.997721	0.029036	0.029115	0.000079	12, 11, 478
25	9975	0.9955	0.9958	0.9959	0.9958	0.997729	0.028451	0.029036	0.000585	10, 26, 591
26	9976	0.9955	0.9958	0.9958	0.9958	0.997729	0.028445	0.028451	0.000006	8, 30, 108
27	9977	0.9955	0.9958	0.9958	0.9958	0.997734	0.028080	0.028445	0.000365	9, 0, 287
28	9978	0.9955	0.9958	0.9958	0.9675	0.997747	0.027220	0.028080	0.000860	9, 26, 215
29	9979	0.9955	0.9958	0.9958	0.9958	0.997751	0.026941	0.027220	0.000279	8, 13, 76
30	9980	0.9955	0.9959	0.9959	0.9959	0.997767	0.025743	0.026941	0.001199	11, 34, 318
31	9981	0.9955	0.9959	0.9960	0.9959	0.997768	0.025732	0.025743	0.000011	11, 43, 404
32	9982	0.9956	0.9959	0.9960	0.9959	0.997776	0.025114	0.025732	0.000618	10, 40, 394
33	9983	0.9956	0.9959	0.9960	0.9959	0.997778	0.024992	0.025114	0.000122	14, 18, 414
34	9984	0.9956	0.9959	0.9959	0.9959	0.997787	0.024340	0.024992	0.000653	8, 30, 107
35	9985	0.9956	0.9959	0.9960	0.9959	0.997789	0.024194	0.024340	0.000145	11, 14, 108
36	9986	0.9956	0.9959	0.9960	0.9959	0.997807	0.022873	0.024194	0.001321	8, 30, 124
37	9987	0.9956	0.9959	0.9959	0.9959	0.997810	0.022677	0.022873	0.000197	8, 14, 77
38	9988	0.9957	0.9960	0.9961	0.9961	0.997829	0.021276	0.022677	0.001401	11, 35, 386
39	9989	0.9957	0.9961	0.9961	0.9961	0.997852	0.019591	0.021276	0.001686	11, 41, 138
40	9990	0.9957	0.9960	0.9960	0.9961	0.997852	0.019589	0.019591	0.000001	8, 14, 108
41	9991	0.9957	0.9960	0.9961	0.9961	0.997860	0.018982	0.019589	0.000608	8, 15, 120
42	9992	0.9957	0.9961	0.9961	0.9961	0.997861	0.018935	0.018982	0.000046	11, 30, 90
43	9993	0.9958	0.9961	0.9961	0.9962	0.997897	0.016168	0.018935	0.002767	8, 13, 108
44	9994	0.9958	0.9962	0.9962	0.9962	0.997907	0.015406	0.016168	0.000762	11, 0, 158
45	9995	0.9958	0.9962	0.9962	0.9962	0.997908	0.015393	0.015406	0.000013	11, 32, 150
46	9996	0.9958	0.9961	0.9962	0.9962	0.997909	0.015304	0.015393	0.000089	9, 32, 395
47	9997	0.9958	0.9962	0.9962	0.9962	0.997915	0.014823	0.015304	0.000481	8, 35, 133
48	9998	0.9959	0.9962	0.9962	0.9962	0.997932	0.013539	0.014823	0.001284	0, 30, 108
49	9999	0.9959	0.9962	0.9962	0.9963	0.997964	0.011045	0.013539	0.002494	8, 14, 110
50	10000	0.9961	0.9965	0.9965	0.9965	0.998075	0.002120	0.011045	0.008925	11, 30, 84

Tóm lại, việc lựa chọn ngưỡng δ trong GA4RF là một yếu tố quan trọng, ảnh hưởng trực tiếp đến hiệu quả của quá trình tối ưu hóa siêu tham số cho mô hình Random Forest. Một ngưỡng ε quá nhỏ có thể kéo dài thời gian chạy nhưng tìm được giải pháp tối ưu hơn. Ngược lại, ngưỡng δ quá lớn có thể dẫn đến việc dừng sớm và bỏ lỡ các giải pháp tiềm năng. Bằng cách chú trọng đến việc lựa chọn ε , người sử dụng có thể tối ưu hóa hiệu quả của GA4RF, đạt được bộ siêu tham số tốt nhất trong khoảng thời gian hợp lý, từ đó nâng cao hiệu suất tổng thể của mô hình.

Từ kết quả có thể nhận thấy thuật toán GA4RF thích ứng tốt với các bộ dữ liệu khác nhau, thể hiện tính linh hoạt của thuật toán và khả năng tối ưu hóa hiệu quả siêu tham số cho mô hình Random Forest trong bài toán phát hiện ngã trên dữ liệu không cân bằng. Thuật toán không chỉ hội tụ tốt mà còn thể hiện khả năng tìm kiếm toàn cục mạnh mẽ, tránh mắc kẹt tại các tối ưu cục bộ. Hiệu suất cao của mô hình được thể hiện qua chỉ số MCC cao trên cả hai bộ dữ liệu, cho thấy tiềm năng ứng dụng rộng rãi của GA4RF trong các bài toán phân loại phức tạp.

3.2.4.3. So sánh kết quả với các nghiên cứu khác

Trong phần này, chúng tôi tiến hành so sánh hiệu suất của thuật toán GA4RF với một số nghiên cứu khác sử dụng hai bộ dữ liệu phổ biến trong lĩnh vực phát hiện ngã là MobiAct v2.0 và UP-Fall.

a. So sánh với các nghiên cứu sử dụng bộ dữ liệu MobiAct v2.0

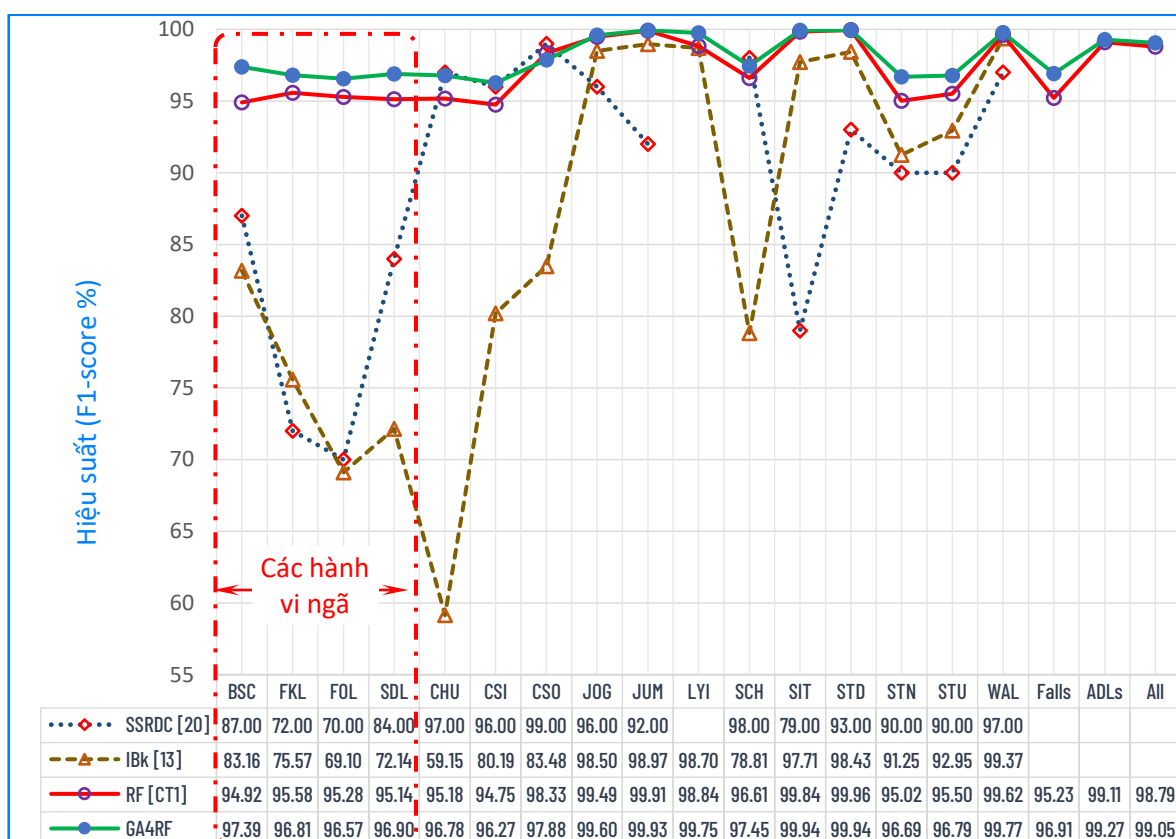
Hình 3.6 so sánh hiệu suất của thuật toán GA4RF với các thuật toán từ các nhóm tác giả Azmat [20], Chatzaki [13] và Le [CT1] khi thực hiện trên bộ dữ liệu MobiAct v2.0. Hình ảnh thể hiện hiệu suất phân loại (F1-score) của các thuật toán đối với các hoạt động khác nhau, bao gồm cả các sự kiện ngã và các hoạt động thường ngày (ADLs). Các thuật toán được so sánh bao gồm SSRDC [20], IBk [13], RF với siêu tham số mặc định [CT1] và GA4RF. Các thuật toán được đánh giá dựa trên thước đo F1-score cho các hoạt động khác nhau, bao gồm cả các sự kiện ngã và các hoạt động thường ngày (ADLs). Các nhãn Falls, ADLs và All lần lượt thể hiện hiệu suất trung

bình tính theo trọng số của các sự kiện ngã, hoạt động bình thường và tổng hợp (bao gồm cả hai loại hoạt động) theo thước đo F1-score.

Hai thuật toán SSRDC [20], IBk [13] có hiệu suất thấp hơn đáng kể so với GA4RF và RF trên hầu hết các hoạt động. Đặc biệt, chúng gặp khó khăn trong việc phân loại các hoạt động phức tạp và các hành vi ngã. Nhìn tổng quan, GA4RF thể hiện hiệu suất tốt hơn so với các thuật toán còn lại trên hầu hết các hoạt động. F1-score của GA4RF luôn nằm ở vị trí cao nhất hoặc gần cao nhất trong phần lớn các loại hoạt động, cho thấy sự ổn định trong việc phân loại các hoạt động, ngay cả đối với các hoạt động phức tạp hoặc dễ gây nhầm lẫn.

Mặc dù hiệu suất phân loại tổng thể của thuật toán GA4RF không quá chênh lệch so với RF (tăng từ 98.79% lên 99.05%), nhưng khả năng phát hiện các sự kiện ngã của thuật toán GA4RF lại tốt hơn đáng kể so với RF (tăng từ 95.23% lên 96.791%). Từ Hình 3.6, có thể nhận thấy hiệu suất phân loại các hoạt động của thuật toán đề xuất GA4RF cải thiện đáng kể so với mô hình RF sử dụng tham số mặc định, đặc biệt là trong việc phát hiện các sự kiện ngã. Điều này chứng tỏ hiệu quả của việc sử dụng giải thuật di truyền để tối ưu hóa siêu tham số, giúp mô hình RF đạt hiệu suất phân loại tốt hơn.

Kết quả từ Hình 3.6 cho thấy GA4RF không chỉ nâng cao hiệu suất phân loại của RF mà còn cải thiện đáng kể khả năng phát hiện các sự kiện ngã, một trong những thách thức chính của bài toán. Bằng cách thiết kế hàm thích nghi phù hợp với các hoạt động khó và ít dữ liệu, GA4RF tập trung vào tối ưu hóa các yếu tố quan trọng nhất. Mặc dù có thể có một số đánh đổi nhỏ trong việc phát hiện các hoạt động bình thường (ADLs), sự suy giảm này không đáng kể và được bù đắp bằng sự cải thiện vượt trội trong phát hiện ngã. Kết quả này chứng minh tính hiệu quả của việc kết hợp giải thuật di truyền (GA) và Random Forest (RF), cũng như vai trò quan trọng của việc tối ưu hóa siêu tham số trong việc nâng cao hiệu suất phân loại trong các ứng dụng phát hiện ngã. GA4RF cho thấy tiềm năng lớn trong việc phát triển các hệ thống phát hiện ngã đáng tin cậy.



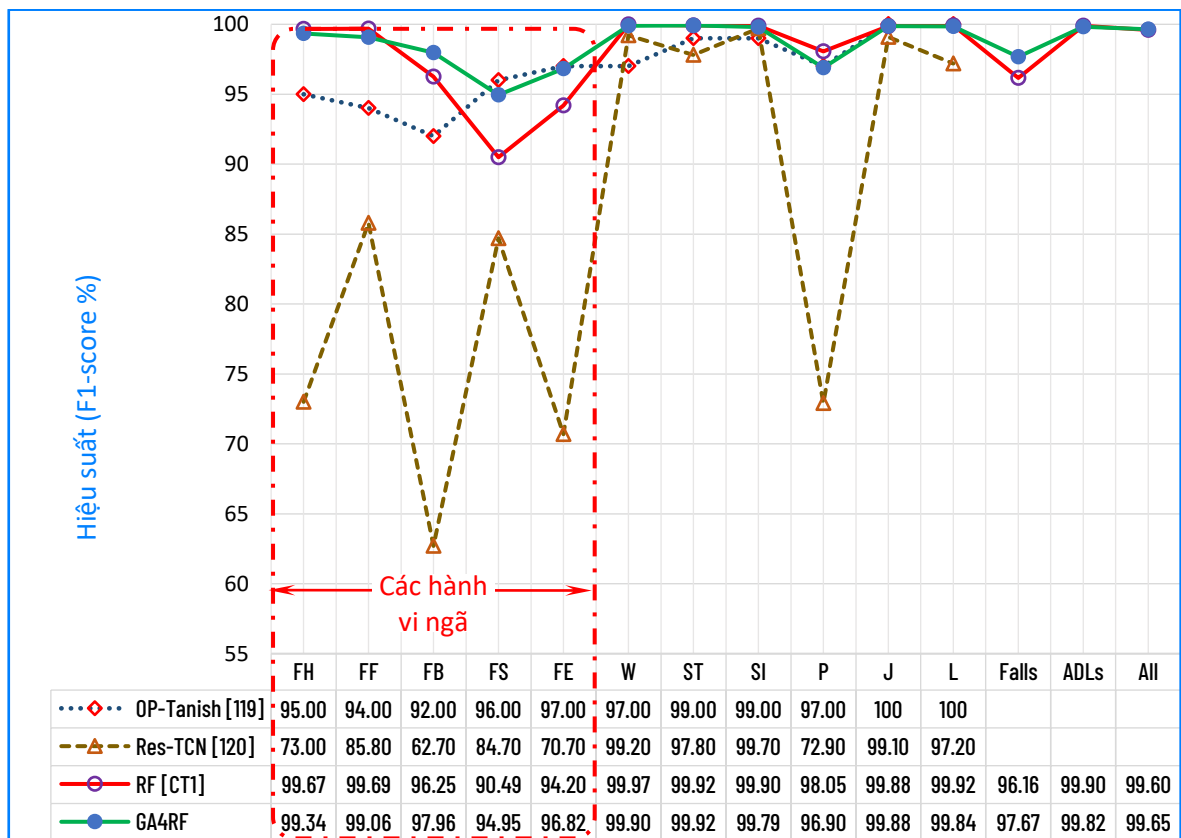
Hình 3.6. So sánh hiệu suất của thuật toán GA4RF với thuật toán của nhóm tác giả Azmat [20], Chatzaki [13] và Le [CT1] thực hiện trên bộ dữ liệu MobiAct v2.0.

b. So sánh với các nghiên cứu sử dụng bộ dữ liệu UP-Fall

Hình 3.7 so sánh hiệu suất của thuật toán GA4RF với các thuật toán OP-Tanish [119], Res-TCN [120], và Random Forest (RF) sử dụng siêu tham số mặc định [CT1] khi thực hiện trên bộ dữ liệu UP-Fall. Biểu đồ trong hình này thể hiện giá trị F1-score của các thuật toán đối với các hoạt động khác nhau, bao gồm cả các sự kiện ngã và các hoạt động thường ngày (ADLs).

Dựa vào biểu đồ trong Hình 3.7 có thể nhận thấy thuật toán Res-TCN không thực sự ổn định trong việc phân loại các hoạt động, đặc biệt khi phân loại các sự kiện ngã. Thuật toán này cho thấy sự khó khăn khi gặp phải các cú ngã về phía sau (FB), hiệu suất chỉ đạt được 62.7%. Điều này cho thấy Res-TCN có thể không phù hợp cho các ứng dụng phát hiện ngã đòi hỏi độ tin cậy cao.

Các mô hình RF (tham số mặc định) [CT1], OP-Tanish [119] và GA4RF có sự tương đồng trong việc phát hiện ADL như đứng (ST), ngồi (SI), nhặt đồ vật (P), nhảy (J), nằm (L). Các thuật toán này đều có hiệu suất cao hơn 97%. Thuật toán OP-Tanish cho thấy hiệu suất vượt trội khi phát hiện các hoạt động nhảy (J) và nằm (L) với F1-score độ chính xác 100%. Tuy nhiên, hiệu suất của OP-Tanish khi phân loại các sự kiện ngã lại kém hơn đáng kể so với GA4RF. Mặc dù RF với các tham số mặc định cũng cho thấy hiệu suất tốt trên hầu hết các hoạt động nhưng vẫn kém hơn GA4RF, đặc biệt là trong việc phát hiện ngã. Cụ thể, GA4RF nâng cao F1-score từ 96,16% lên 97,67% cho các sự kiện ngã. Kết quả này cho thấy việc tối ưu hóa bằng giải thuật di truyền giúp mô hình không chỉ duy trì mà còn nâng cao hiệu suất phân loại trong các bài toán khó khăn.



Hình 3.7. So sánh hiệu suất của thuật toán GA4RF với thuật toán của nhóm tác giả OP-Tanish [119], Res-TCN [120] và Le [CT1] thực hiện trên bộ dữ liệu UP-Fall.

Xét về tổng thể, GA4RF đạt hiệu suất phân loại các hoạt động ổn định và tốt hơn các thuật toán khác, đặc biệt trong việc phát hiện các sự kiện ngã. Trong nhóm các sự kiện ngã, GA4RF đạt F1-score từ 94,95% đến 99,34%, thể hiện sự ổn định và hiệu quả cao. So với Res-TCN, có F1-score chỉ từ 62,70% đến 85,80%, GA4RF cho thấy sự vượt trội rõ rệt. Kết quả này đặc biệt quan trọng vì các sự kiện ngã thường khó phân loại và là trọng tâm của hệ thống phát hiện ngã.

Mặc dù GA4RF không cải thiện nhiều trong các hoạt động bình thường (ADLs), sự gia tăng hiệu suất trong các sự kiện ngã là rất quan trọng và có ý nghĩa lớn trong bối cảnh phát hiện ngã. So với các thuật toán khác, sự cải thiện này khẳng định GA4RF là một phương pháp hiệu quả và có thể ứng dụng rộng rãi trong các hệ thống phát hiện ngã thực tế.

Kết quả từ Hình 3.7 minh chứng cho tính hiệu quả của sự kết hợp giữa giải thuật di truyền và Random Forest. GA4RF không chỉ cải thiện hiệu suất tổng thể mà còn đặc biệt hiệu quả trong việc phát hiện các sự kiện ngã, đáp ứng yêu cầu và thách thức đặc thù của các hệ thống giám sát an toàn. Bằng cách tối ưu hóa các siêu tham số thông qua giải thuật di truyền và thiết kế hàm thích nghi chuyên biệt, GA4RF tập trung vào các hoạt động ít dữ liệu và khó phân loại, khắc phục được những hạn chế của mô hình RF mặc định và các thuật toán khác.

GA4RF đã chứng tỏ ưu thế trong việc phân loại hoạt động, đặc biệt là khả năng phát hiện ngã chính xác và ổn định, ngay cả khi xử lý các hoạt động phức tạp và dữ liệu không cân bằng. Việc kết hợp thành công giữa GA và RF, cùng với thiết kế hàm thích nghi phù hợp đã mang lại những cải tiến đáng kể so với các phương pháp truyền thống và các nghiên cứu trước đó. Mặc dù có thể có sự đánh đổi nhỏ về hiệu suất trên một số hoạt động thường ngày, nhưng sự tập trung vào việc phát hiện ngã - một yếu tố quan trọng trong các ứng dụng thực tế - đã khẳng định GA4RF là một giải pháp tiềm năng cho các hệ thống giám sát và hỗ trợ người dùng trong tương lai.

3.3. Xây dựng mô hình phát hiện sự kiện ngã (FEDM)

Một hệ thống phát hiện ngã (FDS) thực sự tốt khi được huấn luyện bằng dữ liệu thu thập trong thế giới thực. Tuy nhiên, việc thu thập dữ liệu ngã trong thực tế rất khó khăn, gần như không thể thu được do tính bất ngờ của các sự kiện ngã [121]. Để khắc phục điều này, các nghiên cứu thường xây dựng các kịch bản mô phỏng lại sự kiện ngã và sử dụng các tình nguyện viên trẻ tuổi thực hiện trong các phòng thí nghiệm.

Trong quá trình thu dữ liệu bằng mô phỏng, các tình nguyện viên biết trước sự kiện ngã và sẽ sẵn sàng các tư thế để giảm thiểu tổn thương cho cơ thể. Hơn nữa, để bảo vệ tình nguyện viên, các kịch bản thu dữ liệu thường sử dụng những tấm nệm mềm và dày để tình nguyện viên ngã vào. Điều này hoàn toàn khác biệt so với các tình huống xảy ra trong thế giới thực. Vì vậy, luôn có một khoảng cách giữa dữ liệu thực và mô phỏng, điều này đòi hỏi các FDS cần phải xem xét và khắc phục [76]. Một kết quả huấn luyện mô hình tốt trên dữ liệu mô phỏng là chưa đủ. FDS đáng tin cậy phải được đánh giá từ dữ liệu ngã xảy ra trong thực tế.

Ngoài ra, các loại hoạt động và hành vi của con người thường mang các đặc trưng riêng, để phân loại chúng một cách hiệu quả đòi hỏi sự linh hoạt trong việc thay đổi kích thước cửa sổ dữ liệu (DWS). Ví dụ, những hành động xảy ra trong thời gian ngắn như sự kiện ngã cần có kích thước cửa sổ ngắn (sWS) để trích xuất đặc trưng mà không bị ảnh hưởng bởi nhiễu và hoạt động khác. Tuy nhiên, nếu chỉ tập trung vào sWS, các hoạt động có tính tuần hoàn như đi bộ, chạy, nhảy lại không phù hợp. Thông thường, các nghiên cứu sử dụng chung một DWS cho cả hai loại hoạt động, dẫn đến sự suy giảm hiệu suất của mô hình.

Vì vậy, phần này luận án đề xuất một mô hình phát hiện sự kiện ngã (FEDM) nhằm khắc phục các vấn đề đã nêu. Mô hình đề xuất có sự kết hợp giữa kỹ thuật ML, so khớp linh hoạt và ngưỡng tối ưu để xác định các hoạt động bình thường và hành vi ngã dựa trên chuỗi hoạt động cơ bản. FEDM thực hiện so sánh mẫu hành vi chưa được gán nhãn với các mẫu hành vi bình thường đã được gán nhãn. Cụ thể, dữ liệu

về các hoạt động bình thường hàng ngày (ADL) đã gán nhãn được sử dụng để huấn luyện các mô-đun nhận dạng hoạt động (ARM). Các ARM đã được huấn luyện có nhiệm vụ kiểm tra dữ liệu mới có sự khác biệt so với dữ liệu đã huấn luyện hay không.

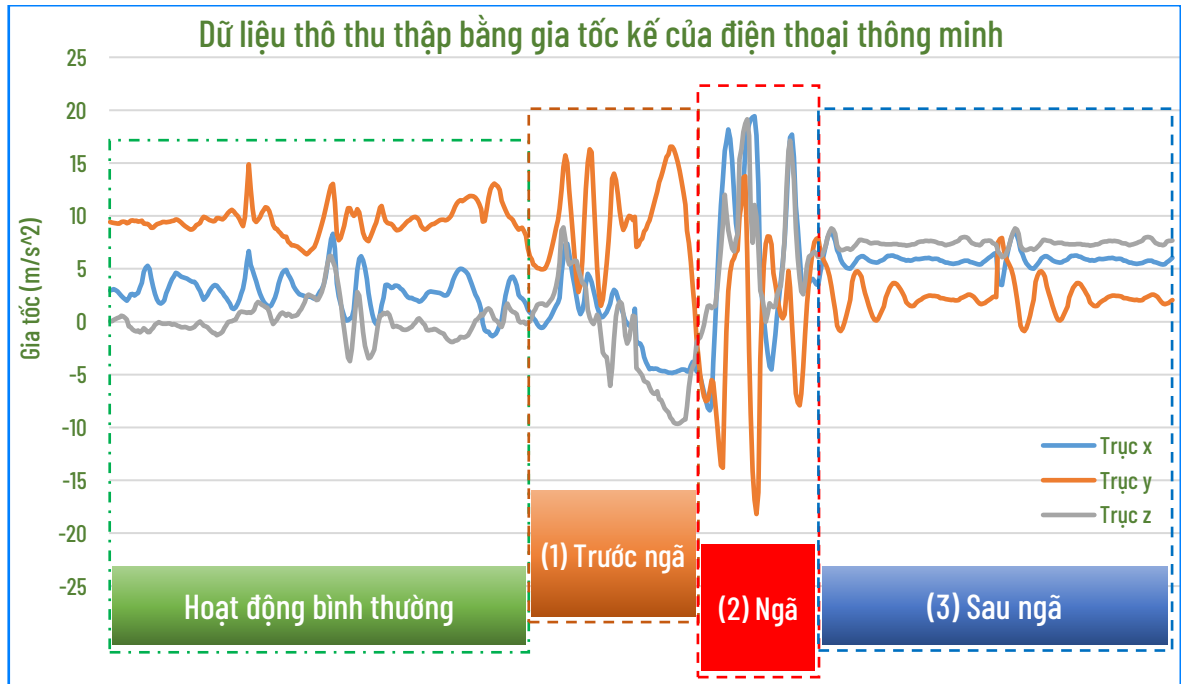
Dữ liệu thu thập được phân đoạn thành các kích thước cửa sổ dài (LWS) theo thời gian để phân tích. FEDM sẽ kiểm tra mẫu dữ liệu mỗi cửa sổ LWS để đưa ra dự đoán là bình thường hay ngã. Nếu hành vi được phát hiện là bình thường, phân đoạn dữ liệu có kích thước LWS sẽ được chia tiếp thành các sWS. Kết quả dự đoán trong từng sWS sẽ được tổng hợp để so sánh với kết quả ở LWS, từ đó đưa ra kết luận hành vi đoán được là bình thường hay ngã.

3.3.1. Đánh giá mức độ ảnh hưởng của cửa sổ dữ liệu

Cùng một loại hoạt động, mỗi người có cách thực hiện riêng, hoặc sử dụng các kiểu điện thoại khác nhau, hoặc thời điểm thu dữ liệu khác nhau sẽ cho ra các chuỗi dữ liệu có đặc điểm khác nhau. Để có bộ dữ liệu nhất quán trước khi huấn luyện mô hình, dữ liệu thô sau thu thập cần phải tiền xử lý, chẳng hạn như sử dụng kỹ thuật khác nhau như phân đoạn cửa sổ dữ liệu. Mỗi cửa sổ tương ứng với một phân đoạn dữ liệu trong một khoảng thời gian hoặc một số mẫu nhất định nào đó, có kích thước được ký hiệu là DWS. Các đặc trưng của mỗi hoạt động được trích xuất trên từng cửa sổ dữ liệu và được sử dụng để huấn luyện mô hình học máy. Việc lựa chọn DWS rất quan trọng, vì kích thước này ảnh hưởng đến hiệu suất nhận dạng của hệ thống. DWS quá ngắn sẽ không cung cấp đủ thông tin cho các hoạt động riêng lẻ, trong khi DWS quá dài có thể chứa nhiều hoạt động khác nhau trong cùng một cửa sổ.

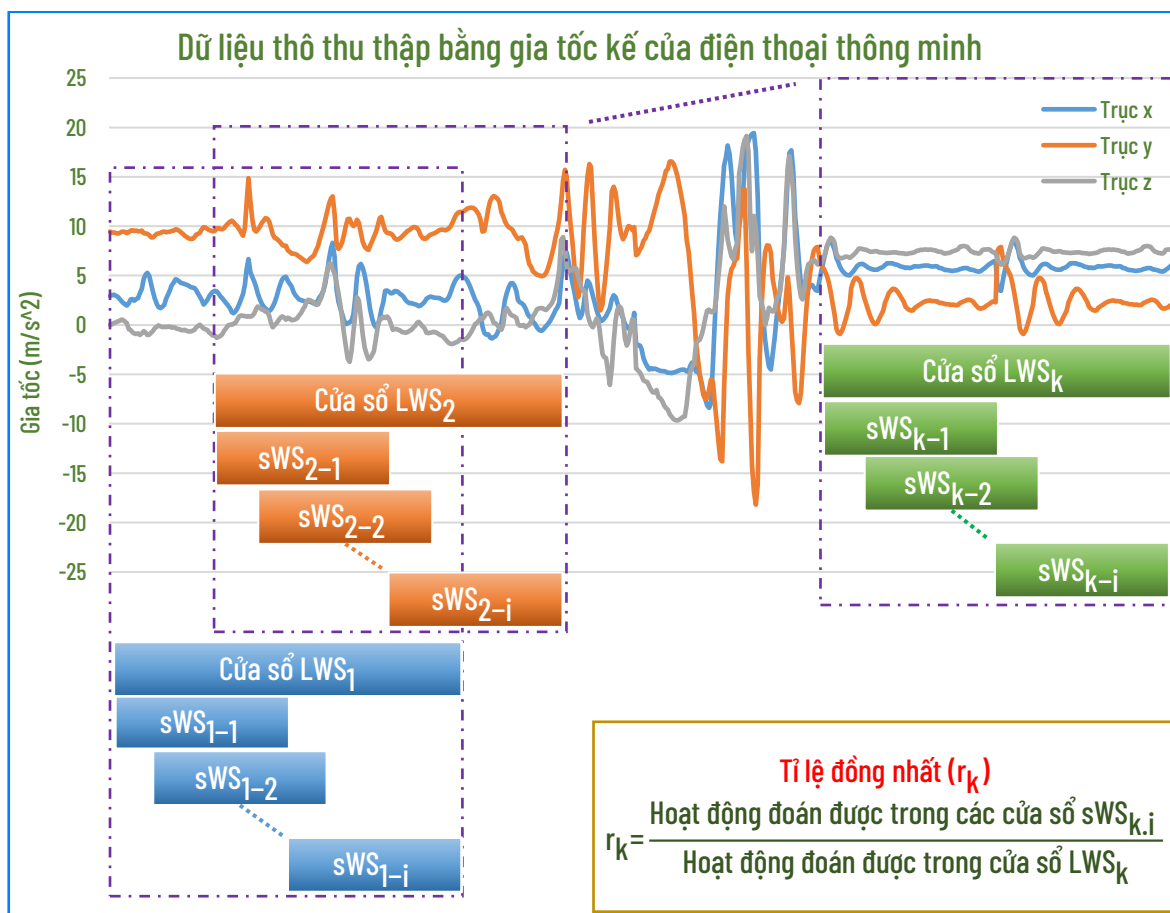
Mỗi loại hoạt động có một DWS tối ưu để lượng thông tin của đối tượng cần nhận dạng không bị bỏ sót và không bị nhiễu. Căn cứ vào sự thay đổi giá trị của gia tốc theo hoạt động của người dùng, một sự kiện ngã có thể phân thành các giai đoạn: (1) giai đoạn trước lúc xảy ra sự kiện ngã, (2) giai đoạn xảy ra ngã, và (3) giai đoạn sau khi ngã. Tùy thuộc vào mức độ nghiêm trọng của cú ngã, sau khi ngã có thể xảy ra hai trường hợp, (i) ngã xong nằm yên và (ii) ngã xong tự đứng dậy. Phân tích mẫu dữ

liệu sau khi xảy ra sự kiện ngã có thể đánh giá được mức độ nghiêm trọng của cú ngã. Hình 3.8 trình bày mẫu biểu diễn một sự kiện ngã của người đi bộ được thu thập bằng gia tốc kế của điện thoại thông minh.



Hình 3.8. Biểu diễn một sự kiện ngã điển hình của người đi bộ được thu thập bằng gia tốc kế ba trục của SP.

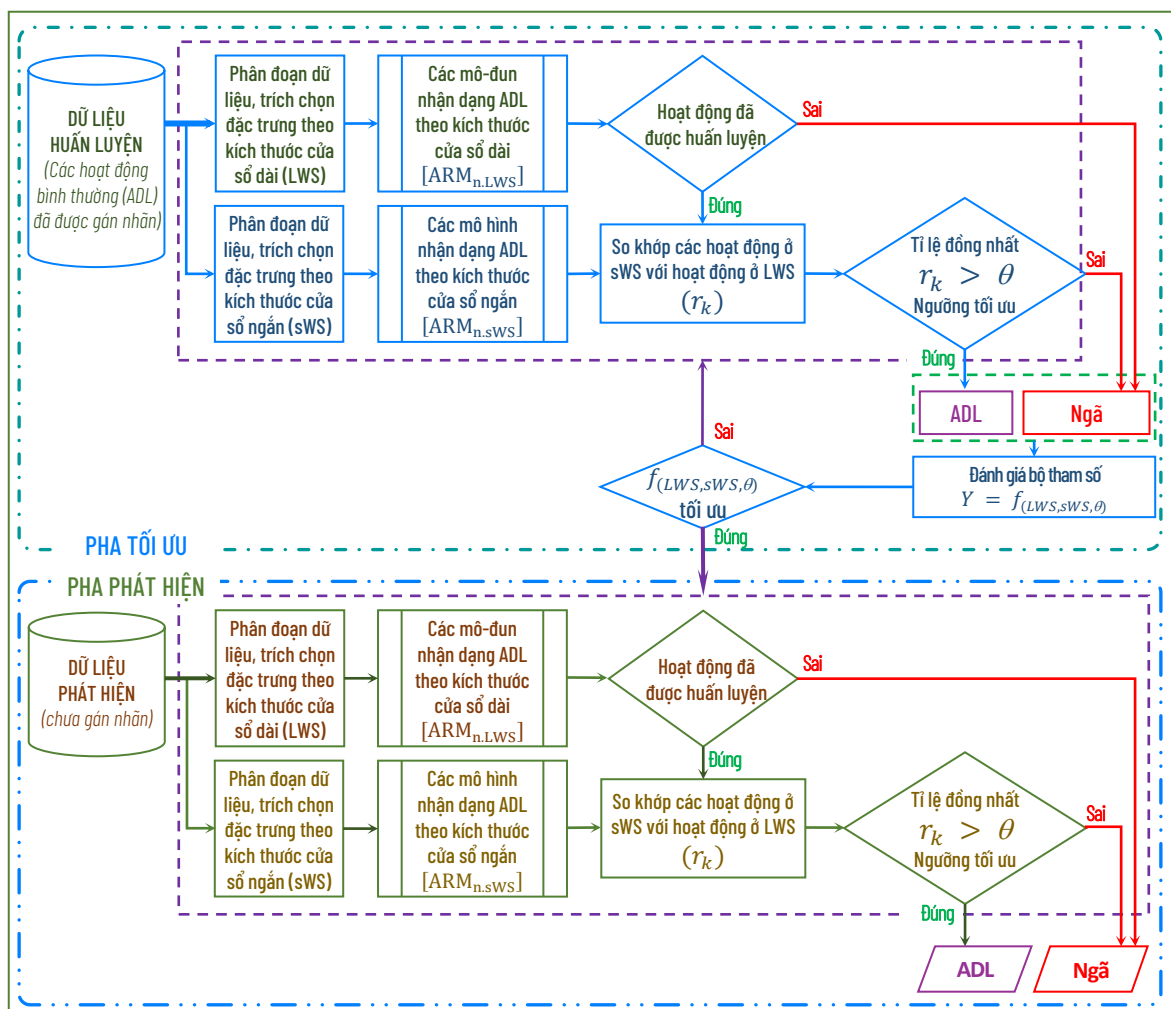
Mỗi loại hoạt động của con người có đặc điểm riêng, thời gian diễn ra cũng khác nhau. Các hoạt động thường ngày có thời gian dài, trong khi ngã thường diễn ra bất ngờ trong thời gian ngắn. Vì vậy cần được điều chỉnh kích thước cửa sổ một cách linh động để phù hợp với từng nhóm hoạt động. Với các hoạt động xảy ra nhanh như ngã, sử dụng kích thước cửa sổ ngắn (sWS) sẽ giúp trích xuất đặc trưng chính xác hơn. Ngược lại, kích thước cửa sổ dài (LWS) phù hợp với các hoạt động tuần hoàn như đi bộ, chạy, nhảy. Để giải quyết vấn đề này, NCS đề xuất phương pháp so khớp hoạt động ở nhiều cửa sổ con với cửa sổ dài để tăng cường khả năng phát hiện các sự kiện ngã và các sự kiện ngã. Ý tưởng thiết kế mô hình mô hình phát hiện các sự kiện ngã (FEDM) trong luận án này được mô tả như trong Hình 3.9.



Hình 3.9. Ý tưởng thiết kế mô hình phát hiện các sự kiện ngã (FEDM). Trong hình vẽ, LWS là kích thước cửa sổ dài, sWS là kích thước cửa sổ ngắn.

3.3.2. Mô hình phát hiện các sự kiện ngã (FEDM).

Mô hình đề xuất FEDM sử dụng kết quả từ bộ 44 đặc trưng đề xuất ở Chương 2 và thuật toán GA4RF kết hợp với kỹ thuật so khớp dữ liệu linh hoạt và kỹ thuật ngưỡng. Dữ liệu ADL đã gán nhãn được sử dụng để huấn luyện các mô-đun nhận dạng hoạt động (ARM). Các phương pháp lọc nhiễu và trích xuất đặc trưng từ dữ liệu gia tốc kế được kế thừa từ nghiên cứu trước đây [CT1]. Phương pháp xây dựng FEDM được biểu diễn ở Hình 3.10. Để tăng khả năng phát hiện các sự kiện ngã, mô hình đề xuất gồm hai giai đoạn: (1) giai đoạn tối mô hình; (2) giai đoạn phát hiện sự kiện ngã.



Hình 3.10. Mô hình tổng quan phát hiện các sự kiện ngã

3.3.2.1. Giai đoạn tối ưu tham số

Giai đoạn này được thực hiện để tìm ra bộ tham số tốt nhất cho FEDM. Bộ tham số cần tìm là $t_k = (LWS_k, sWS_k, \theta_k)$, trong đó LWS là kích thước của cửa sổ dữ liệu lớn (phù hợp với các hoạt động bình thường như đi bộ, chạy, nhảy,...), sWS là kích thước của cửa sổ dữ liệu nhỏ (phù hợp với các sự kiện ngã như ngã), và θ là ngưỡng hiệu suất của mô hình trong việc phát hiện các sự kiện ngã và bình thường. Quá trình tìm kiếm bộ tham số tối ưu được thực hiện như sau:

- (1) Lựa chọn dữ liệu: Lấy ngẫu nhiên 70% dữ liệu thô của các hoạt động thường ngày trong bộ dữ liệu MobiAct v2.0.

- (2) Huấn luyện mô-đun nhận dạng theo kích thước cửa sổ dài: Dữ liệu hoạt động thường ngày (ADL) (k) được phân đoạn theo kích thước cửa sổ dài (LWS), trích chọn 44 đặc trưng (TFDHD) và huấn luyện mô-đun nhận dạng hoạt động (ARM) bằng thuật toán GA4RF. Kết quả, mỗi hoạt động thu được một mô-đun $ARM_k(LWS)$;
- (3) Huấn luyện mô-đun nhận dạng theo kích thước cửa sổ ngắn: Làm tương tự như bước thứ (2). Tuy nhiên, bước này sẽ phân đoạn dữ liệu thành các cửa sổ ngắn (sWS) thay vì cửa sổ dài như bước (2). Các mô-đun $ARM_k(LWS)$, $ARM_k(sWS)$ là thành phần chính để thiết kế mô hình FEDM;
- (4) Tìm bộ tham số tối ưu $t_k=(LWS_k, sWS_k, \theta_k)$: Phương pháp thực nghiệm thử sai nhiều lần được thực hiện để tìm;
- (5) Đánh giá bộ tham số tối ưu $t_k=(LWS_k, sWS_k, \theta_k)$: Lấy ngẫu nhiên 20% dữ liệu thô còn lại để phân đoạn, trích xuất đặc trưng theo các LWS, sWS như bước (2) và bước (3). Bộ dữ liệu này được dùng để đánh giá bộ tham số (LWS_k, sWS_k, θ_k). Mỗi cửa sổ LWS_k của chuỗi dữ liệu đầu vào khi qua các $ARM_k(LWS)$ nếu kết quả trả về là '0' nghĩa là mô hình không phát hiện được các hoạt động bình thường ($A_{k(LWS)}$) trong cửa sổ LWS_k . Nói cách khác, mô hình đã phát hiện ra các sự kiện ngã tại cửa sổ LWS_k . Ngược lại, nếu tại cửa sổ LWS_k , mô hình cho kết quả là '1', nghĩa là hoạt động ($A_{k(LWS)}$) tại cửa sổ LWS_k được nhận dạng là bình thường.

Để tránh tình trạng người dùng bị ngã nhưng mô hình dự đoán là bình thường (âm tính giả), các hoạt động được dự đoán là bình thường tiếp tục được kiểm tra thêm bằng cách phân đoạn dữ liệu tại cửa sổ LWS_k thành tập (j) cửa sổ có kích thước ngắn hơn ($sWS_{k,i}$) và được đưa qua các mô hình $ARM_{k,i}(sWS)$ để nhận dạng. Tập các hoạt động phát hiện được trong các cửa sổ $sWS_{k,i}$ là $A_{k(sWS)} = \{A_{k,i}(sWS); i = 1, \dots, j\}$.

Các hoạt động $A_{k,i}(sWS)$ đoán được trong cửa sổ $sWS_{k,i} \in LWS_k$ có thể giống hoàn toàn hoặc giống một phần với hoạt động trong cửa sổ LWS_k tương ứng. Mức độ giống

của các hoạt động trong cửa sổ ngắn với hoạt động ở cửa sổ dài được gọi là tỉ lệ đồng nhất. Tỉ lệ này được thể hiện ở Biểu thức (3.12).

$$r_k = \frac{\text{Các hoạt động đoán được trong các cửa sổ } sWS_{k,i}}{\text{Hoạt động đoán được trong các cửa sổ } LWS_k} = \frac{\sum A_{k,i(sWS)}}{A_{k(LWS)}} \quad (3.12)$$

Trong đó,

+ $A_{k(LWS)}$ là hoạt động mà mô hình phát hiện được trong cửa sổ LWS;

+ $A_{k,i(sWS)}$ là hoạt động mà mô hình phát hiện được trong cửa sổ sWS; lý tưởng, các hoạt động $A_{k,i(sWS)}$ phải giống với $A_{k(LWS)}$.

Tỉ lệ r_k càng lớn đồng nghĩa với mô hình phát hiện hoạt động ở cửa sổ sWS_{ki} giống với hoạt động ở cửa sổ LWS_k . Nếu tỉ lệ (r_k) này nhỏ hơn một ngưỡng (θ) cho trước thì hoạt động phát hiện được xem là sự kiện ngã. Dựa vào giá trị ngưỡng θ cho trước, nếu $r_k \geq \theta$ thì mô hình phát hiện đó là hoạt động bình thường, ngược lại, nếu $r_k < \theta$ thì kết luận đó là một sự kiện ngã. Bảng 3.4 là ví dụ về cách xác định tỉ lệ đồng nhất (r_k) giữa hoạt động ở các cửa sổ ngắn sWS với cửa sổ dài LWS.

Bộ tham số tối ưu (LWS, sWS, θ) được xác định bởi Biểu thức (3.13)

$$y_k = f(t_k) \quad (3.13)$$

trong đó,

$$t_k = (LWS_k, sWS_k, \theta_k) \quad (3.14)$$

Bộ tham số này được xem là tối ưu khi:

$$y_m = \max(y_k) \quad (3.15)$$

và được ký hiệu là:

$$t_m = (LWS_m, sWS_m, \theta_m) \quad (3.16)$$

trong đó, $k = 1 \div n$; n là số bộ tham số thu được trong giai đoạn huấn luyện và tối ưu.

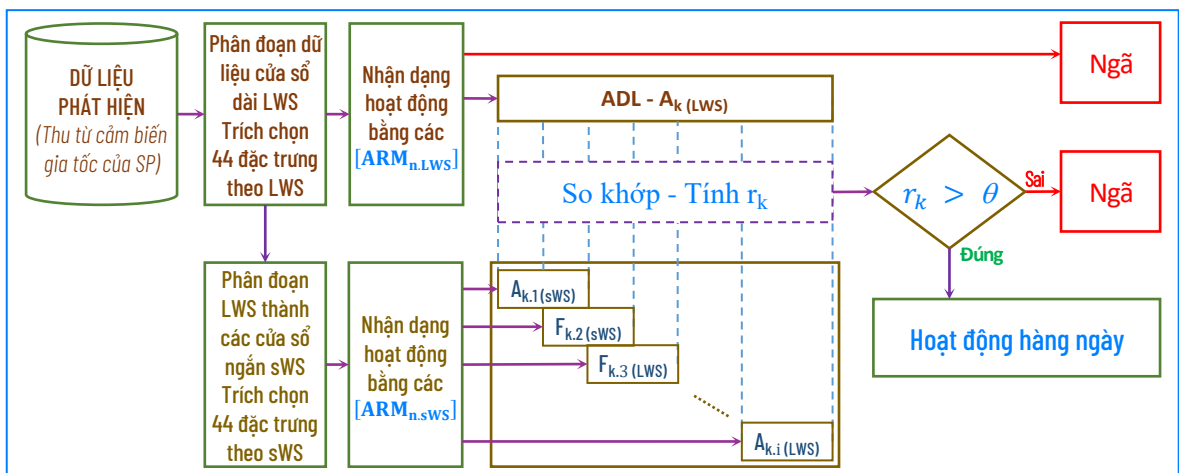
Bảng 3.4. Tỷ lệ đồng nhất (r_k) khi dự đoán hoạt động ở các sWS so với LWS; “1” mô hình đoán đúng là hoạt động bình thường, “0” mô hình đoán là sự kiện ngã.

Tập các cửa sổ ngắn ($sWS_{k,i} \in LWS_k$)								Cửa sổ dài	Tỷ lệ đồng nhất
sWS ₁	sWS ₂	sWS ₃	sWS ₄	sWS ₅	sWS ₆	sWS ₇	sWS ₈	LWS	r_k
1	1	1	0	0	1	1	1	1	$r_k = 6/8 = 0.75$
1	1	1	1	0	1	0	0	1	$r_k = 5/8 = 0.63$
1	0	0	0	0	0	1	1	1	$r_k = 3/8 = 0.38$

3.3.2.2. Giai đoạn phát hiện

Trong giai đoạn này, 10% dữ liệu còn lại (xóa nhãn) bao gồm dữ liệu về các sự kiện ngã không dùng để huấn luyện mô hình được đưa vào để đánh giá, kiểm thử mô hình. Bộ tham số tốt nhất $t_m = (LWS_m, sWS_m, \theta_m)$ thu được ở giai đoạn tối ưu được sử dụng cho giai đoạn này. Quy trình thực hiện nhận dạng và phát hiện sự kiện ngã được trình bày Hình 3.11.

Dữ liệu thu thập bằng cảm biến gia tốc được phân đoạn thành các kích thước cửa sổ dài (LWS) theo thời gian để trích chọn đặc trưng. Các mô-đun $ARM_{n,LWS}$ sẽ kiểm tra mẫu dữ liệu mỗi cửa sổ LWS để đưa ra dự đoán là bình thường hay ngã. Nếu hoạt động được phát hiện là bình thường, phân đoạn dữ liệu có kích thước LWS sẽ được chia thành các sWS. Kết quả dự đoán trong từng sWS được tổng hợp để so sánh với kết quả ở LWS, từ đó đưa ra kết luận hành vi đoán được là bình thường hay ngã.



Hình 3.11. Quy trình thực hiện nhận dạng và phát hiện sự kiện ngã

3.3.3. Thực nghiệm và đánh giá kết quả

3.3.3.1. Dữ liệu thử nghiệm

Bộ dữ liệu MobiAct 2.0 [107] sử dụng để huấn luyện, kiểm thử và đánh giá các phương pháp đề xuất trong phần này. Chi tiết về bộ dữ liệu này đã được trình bày ở Mục 2.3.1. Chính hoạt động bình thường trong bộ dữ liệu MobiAct 2.0 được sử dụng để huấn luyện các mô-đun ARM. Các hoạt động này được mô tả lại trong Bảng 3.5.

Bảng 3.5. Các mẫu hoạt động bình thường trong bộ dữ liệu MobiAct v2.0 được sử dụng để huấn luyện mô hình.

TT (i)	Nhãn	Thời gian (s)	Mô tả	Loại hoạt động
1	CHU	6 giây	Chuyển tiếp từ ngồi sang đứng (đứng lên từ ghế)	Bình thường (hàng ngày)
2	JOG	30 giây	Chạy bộ	
3	JUM	30 giây	Nhảy liên tục	
4	SCH	6 giây	Chuyển tiếp từ đứng sang ngồi (ngồi trên ghế)	
5	SIT	1 phút	Ngồi trên ghế với những cử động nhẹ nhàng	
6	STD	5 phút	Đứng với cử động nhẹ nhàng	
7	STN	10 giây	Đi xuống cầu thang (10 bậc)	
8	STU	10 giây	Đi lên cầu thang (10 bậc)	
9	WAL	5 phút	Đi bộ bình thường	
10	Fall	10 giây	Dữ liệu tổng hợp của 4 kiểu sự kiện ngã khác nhau (các sự kiện ngã) là BSC, FKL, FOL và SDL.	Ngã

3.3.3.2. Mô tả thực nghiệm

Quy trình tổ chức các thực nghiệm để tìm bộ tham số tối ưu, đánh giá tính hiệu quả của mô hình đề xuất FEDS được thực hiện như sau:

- (1) Lựa chọn dữ liệu: Lấy ngẫu nhiên 70% dữ liệu thô của chín hoạt động thường ngày trong bộ dữ liệu MobiAct v2.0 (Bảng 3.5).
- (2) Huấn luyện mô-đun nhận dạng hoạt động (ARM): Dữ liệu của hoạt động CHU được phân đoạn theo các loại cửa sổ (64, 128, 256, 512 và 1024 mẫu). Trích chọn 44 đặc trưng (TFDHD) và huấn luyện ARM bằng thuật toán GA4RF theo

từng lại cửa sổ được phân đoạn. Kết quả thu được các mô-đun $ARM_{CHU(64)}$, $ARM_{CHU(128)}$..., $ARM_{CHU(1024)}$ Làm tương tự cho tám hoạt động còn lại.

- (3) Đánh giá mô hình dựa vào dữ liệu các hoạt động bình thường;
- (4) Đánh giá mô hình dựa vào dữ liệu các sự kiện ngã;
- (5) Lựa chọn bộ tham số tối ưu $t_m = (LWS_m, sWS_m, \theta_m)$.

3.3.3.3. Kết quả và bàn luận

a. Huấn luyện mô-đun nhận dạng hoạt động (ARM)

Để huấn luyện các mô-đun nhận dạng hoạt động (ARM), thử nghiệm này thiết lập DOR ở mức 80%. Dữ liệu thô của chín loại hoạt động bình thường (Bảng 3.5) lần lượt được phân thành năm DWS khác nhau, đó là 64 mẫu, 128 mẫu, 256 mẫu, 512 mẫu và 1024 mẫu. Từng hoạt động sẽ được xây dựng thành một bộ dữ liệu TFDHD 44 đặc trưng để huấn luyện các ARM. Kết quả thực nghiệm dựa trên chín nhãn ADL trong bộ dữ liệu MobiAct 2.0 tổng hợp trong Bảng 3.6.

Bảng 3.6. Kết quả huấn luyện các mô hình ở các WS khác nhau dựa trên chín nhãn ADL thường gặp trong bộ dữ liệu MobiAct 2.0

Hoạt động	Nhãn	F1-score (%)				
		WS = 64	WS = 128	WS = 256	WS = 512	WS = 1024
Chuyển tiếp từ ngồi sang đứng (đứng lên từ ghế)	CHU	96.70	96.60	95.20	94.80	94.20
Chạy bộ	JOG	96.50	98.50	99.50	99.90	99.90
Nhảy liên tục	JUM	98.10	99.60	99.90	99.90	99.90
Chuyển tiếp từ đứng sang ngồi (ngồi trên ghế)	SCH	97.90	97.30	97.60	96.40	96.10
Ngồi trên ghế với những cử động nhẹ	SIT	99.80	99.70	99.80	99.30	99.50
Đứng với cử động nhẹ nhàng	STD	99.10	99.70	100	100	100
Đi lên cầu thang (10 bậc)	STN	88.60	92.60	95.00	96.60	94.20
Đi xuống cầu thang (10 bậc)	STU	88.70	93.10	95.50	96.90	92.70
Đi bộ bình thường	WAL	98.30	99.30	99.60	99.90	99.90

Từ kết quả được tổng hợp trong Bảng 3.6, có thể thấy rằng hiệu suất nhận dạng các hoạt động có sự thay đổi trạng thái nhanh như ngồi chuyển sang đứng (CHU) hoặc đứng chuyển sang ngồi (SCH) giảm khi WS tăng lên. Khi sử dụng LWS hơn, các hoạt động thay đổi trạng thái nhanh có thể bị nhiễu bởi ADL khác, dẫn đến giảm hiệu suất của các mô hình phát hiện ngã như ngã. Tuy nhiên, đối với ADL như đi bộ, đứng, ngồi có tính tuần hoàn và diễn ra trong khoảng thời gian dài, dữ liệu được ghi lại nhiều hơn. Lúc này, LWS có thể làm tăng khả năng nhận dạng hoạt động bình thường của mô hình. Vì vậy, WS tối ưu là một sự kết hợp đủ ngắn để tránh nhiễu và đủ dài để mô tả đầy đủ các đặc điểm của các hoạt động.

b. Đánh giá mô hình dựa vào dữ liệu các hoạt động bình thường

Dữ liệu kiểm thử của chín ADL trong bộ dữ liệu MobiAct 2.0 được sử dụng để kiểm tra mô hình đề xuất. Như đã phân tích ở trên, dữ liệu thô được phân đoạn thành LWS (LWS) để dự đoán hoạt động bình thường của con người. Mỗi phân đoạn dữ liệu được phân loại bởi chín ARM_{LWS} như đã trình bày. Nếu cả chín $ARM_{(LWS)}$ không phát hiện ra hoạt động bình thường trên phân đoạn dữ liệu được đưa vào thì hệ thống sẽ cho kết quả hoạt động của người dùng là ngã. Ngược lại, một trong chín $ARM_{k(LWS)}$ phát hiện ra hoạt động bình thường thì phân đoạn dữ liệu này tiếp tục được chia thành các cửa sổ dữ liệu nhỏ hơn (sWS). Bộ dữ liệu này được nhận dạng bởi mô-đun $ARM_{k,i(sWS)}$ để đánh giá xem có sự ngã nào trong hoạt động bình thường này không. Kết quả thử nghiệm trên chín loại hoạt động bình thường với các DWS khác nhau được tổng hợp ở Bảng 3.7.

Tỉ lệ đồng nhất $r_k = 1.0$ tương ứng với dự đoán trên phân đoạn sWS giống với LWS, nghĩa là FEDM đã dự đoán đúng là hoạt động bình thường trong cả hai loại cửa sổ LWS và sWS. Nếu r_k càng giảm, sự khác biệt về kết quả dự đoán giữa phân đoạn dữ liệu LWS và sWS càng lớn. Nghĩa là, FEDM có thể dự đoán là hoạt động bình thường với phân đoạn dữ liệu có LWS, nhưng lại phát hiện ra các sự kiện ngã ở phân đoạn dữ liệu sWS ($sWS \in LWS$).

Bảng 3.7. Kết quả thử nghiệm FEDM với bộ dữ liệu hoạt động bình thường.

Tỉ lệ (r_k)*	LWS = 512 sWS = 256	LWS = 512 sWS = 128	LWS = 512 sWS = 64	LWS = 256 sWS = 128	LWS = 256 sWS = 64	LWS = 128 sWS = 64
0.1	96.12	96.22	95.94	93.77	93.60	89.05
0.2	96.06	96.17	95.70	93.57	93.20	88.63
0.3	95.98	96.10	95.34	93.32	92.66	88.16
0.4	95.85	95.87	94.73	93.05	91.72	87.57
0.5	95.46	95.62	94.09	92.24	90.78	86.07
0.6	95.22	95.21	92.74	91.72	89.64	84.99
0.7	94.76	94.28	91.15	91.03	87.27	83.41
0.8	94.06	92.63	87.76	90.08	84.20	81.01
0.9	92.88	90.05	80.98	88.27	78.18	77.13
1.0	90.82	80.14	54.01	83.98	61.19	69.39

* r_k càng nhỏ, kết quả dự đoán của FEDM trên các LWS và sWS càng khác biệt

Kết quả tổng hợp ở Bảng 3.7 cho thấy, với LWS = 512, sWS = 256 và $r_i = 1.0$ có kết quả phân loại chính xác các hoạt động bình thường đạt 90.82%. Khi sWS giảm xuống thì sự khác biệt về kết quả dự đoán trên LWS và sWS tăng lên, nghĩa là khả năng phát hiện các sự kiện ngã có thể xảy ra trong các hoạt động bình thường tăng lên.

c. Đánh giá mô hình dựa vào dữ liệu các sự kiện ngã

Với cách làm tương tự, thay vì sử dụng bộ dữ liệu các hoạt động bình thường, dữ liệu các sự kiện ngã được dùng để đánh giá hiệu suất của FEDM. Trong trường hợp lý tưởng, trên các phân đoạn dữ liệu có LWS, FEDM phải đoán được đây là các sự kiện ngã. Nếu FEDM đoán ra hoạt động bình thường thì phân đoạn dữ liệu LWS tiếp tục được chia nhỏ thành các sWS để FEDM dự đoán. Khi đó, nếu FEDM dự đoán ra các sự kiện ngã thì tỉ lệ đồng nhất (r_k) giữa sWS và LWS bằng không. Nghĩa là r_k càng nhỏ thì FEDM dự đoán các sự kiện ngã trên cửa sổ ngắn sWS càng chính xác. Tỉ lệ đồng nhất $r_k = 0$ tương ứng với dự đoán trên phân đoạn sWS khác hoàn toàn với dự đoán trên phân đoạn LWS. Phân đoạn dữ liệu theo LWS dự đoán hoạt ra động bình thường là sai. Phân đoạn dữ liệu theo sWS được dự đoán các sự kiện ngã là một dự

đoán đúng. Kết quả thử nghiệm FEDM trên các mẫu dữ liệu các sự kiện ngã với các DWS khác nhau được tổng hợp ở Bảng 3.8.

Bảng 3.8. Kết quả thử nghiệm FEDM với bộ dữ liệu các sự kiện ngã.

Tỉ lệ (r_k)*	LWS = 512 sWS = 256	LWS = 512 sWS = 128	LWS = 512 sWS = 64	LWS = 256 sWS = 128	LWS = 256 sWS = 64	LWS = 128 sWS = 64
0.1	90.28	89.45	88.93	96.63	95.36	97.87
0.2	90.13	88.96	88.10	95.64	93.44	96.93
0.3	89.90	88.36	87.20	94.26	92.00	96.03
0.4	89.34	87.27	86.07	93.07	90.01	95.14
0.5	87.88	86.45	85.32	90.33	88.15	92.95
0.6	87.09	85.66	84.65	88.42	85.99	91.88
0.7	86.56	83.86	83.07	86.73	82.90	90.74
0.8	85.59	80.82	80.37	85.04	80.51	89.69
0.9	84.12	76.65	73.69	82.98	77.97	88.61
1.0	79.05	70.31	67.79	80.31	75.61	85.65

* r_k càng nhỏ, kết quả dự đoán của FEDM trên các LWS và sWS càng khác biệt

d. Lựa chọn bộ tham số tối ưu

Kết quả trình bày ở Bảng 3.8 tổng hợp lại từ kết quả ở Bảng 3.6 và Bảng 3.7 để phân tích và lựa chọn bộ tham số tối ưu $t_m = (LWS_m, sWS_m, \theta_m)$. Trong Bảng 3.8, các cột NA là kết quả thực nghiệm của thuật toán đề xuất với bộ dữ liệu hoạt động bình thường. Các cột AB là kết quả dự đoán các sự kiện ngã với dữ liệu ngã không dùng để huấn luyện mô hình.

Dựa vào kết quả ở Bảng 3.8 có thể thấy LWS, sWS và tỉ lệ đồng nhất r_k tác động lớn đến hiệu suất của FEDM. Tỉ lệ r_k khi phát hiện hoạt động bình thường và ngã là hoàn toàn ngược nhau. Vì vậy, để có được sự cân bằng giữa việc phát hiện hoạt động bình thường và sự kiện ngã, có thể chọn mức ngưỡng $\theta = 0.5$. Có thể nhận thấy, nếu chọn ngưỡng θ càng thấp ($\approx r_k$ càng nhỏ) thì khả năng phát hiện các sự kiện ngã của FEDM càng tốt. Tuy nhiên, như đã trình bày, lúc này FEDM tăng khả năng nhận dạng sai các hoạt động bình thường. Vì vậy, mức ngưỡng $\theta = 0.4$, LWS = 256 mẫu,

sWS = 128 mẫu là sự lựa chọn phù hợp để làm bộ tham số tối ưu cho FEDM. Với bộ tham số $t_m = (256, 128, 0.4)$, hiệu suất phát hiện các hoạt động của con người (bao gồm cả bình thường và ngã) đều đạt trên 93%.

Bảng 3.9. Kết quả thực nghiệm hệ thống nhận dạng các mẫu chuyển động bình thường và ngã

Ngưỡng (θ)	LWS = 512 sWS = 256		LWS = 512 sWS = 128		LWS = 512 sWS = 64		LWS = 256 sWS = 128		LWS = 256 sWS = 64		LWS = 128 sWS = 64	
	NA	AE	NA	AE	NA	AE	NA	AE	NA	AE	NA	AE
0.1	96.12	90.28	96.22	89.45	95.94	88.93	93.77	96.63	93.60	95.36	89.05	97.87
0.2	96.06	90.13	96.17	88.96	95.70	88.10	93.57	95.64	93.20	93.44	88.63	96.93
0.3	95.98	89.90	96.10	88.36	95.34	87.20	93.32	94.26	92.66	92.00	88.16	96.03
0.4	95.85	89.34	95.87	87.27	94.73	86.07	93.05	93.07	91.72	90.01	87.57	95.14
0.5	95.46	87.88	95.62	86.45	94.09	85.32	92.24	90.33	90.78	88.15	86.07	92.95
0.6	95.22	87.09	95.21	85.66	92.74	84.65	91.72	88.42	89.64	85.99	84.99	91.88
0.7	94.76	86.56	94.28	83.86	91.15	83.07	91.03	86.73	87.27	82.90	83.41	90.74
0.8	94.06	85.59	92.63	80.82	87.76	80.37	90.08	85.04	84.20	80.51	81.01	89.69
0.9	92.88	84.12	90.05	76.65	80.98	73.69	88.27	82.98	78.18	77.97	77.13	88.61
1.0	90.82	79.05	80.14	70.31	54.01	67.79	83.98	80.31	61.19	75.61	69.39	85.65

Ghi chú: NA - Hoạt động bình thường; AE - Các sự kiện ngã

3.4. Kết luận chương

Trong chương này, nghiên cứu đã tập trung vào việc cải thiện hiệu suất của mô hình Random Forest trong phát hiện sự kiện ngã và xây dựng mô hình phát hiện sự kiện ngã (FEDM) dựa trên dữ liệu các hoạt động thường ngày.

Thứ nhất, luận án đã đề xuất thuật toán GA4RF, sử dụng giải thuật di truyền (GA) để tìm kiếm giá trị tối ưu cho các siêu tham số của mô hình RF. Thuật toán GA4RF đã thể hiện khả năng tìm kiếm bộ siêu tham số tối ưu một cách hiệu quả, giúp cải thiện đáng kể hiệu suất phát hiện ngã, đạt độ chính xác từ 95.23% lên 96.91% trên bộ dữ liệu MobiAct V2.0 và từ 96.16% lên 97.67% trên bộ dữ liệu UP-Fall. Khả năng tìm kiếm bộ siêu tham số tối của GA4RF ưu nhờ cấu trúc nhiễm sắc thể, không gian tìm

kiếm và hàm thích nghi được thiết kế theo hướng mới phù hợp với đặc thù của các bộ dữ liệu ngã.

Thứ hai, luận án đã phát triển mô hình FEDM nhằm nhận diện các sự kiện ngã bằng cách phân tích dữ liệu các hoạt động bình thường hàng ngày. Mô hình này kết hợp các kỹ thuật trích xuất đặc trưng hiệu quả từ Chương 2, thuật toán GA4RF tối ưu ở Chương 3, phương pháp so khớp dữ liệu linh hoạt và kỹ thuật ngưỡng để phân biệt hiệu quả giữa các hoạt động bình thường và sự kiện ngã. Mô hình đề xuất đã đạt được hiệu suất phát hiện các hoạt động thường ngày là 93.05% và các sự kiện ngã là 93.07% ở mức ngưỡng $\theta = 0.4$, LWS = 256 mẫu, sWS = 128 mẫu. Kết quả cho thấy mô hình FEDM có khả năng phát hiện ngã tốt, đặc biệt là khi dữ liệu về ngã khan hiếm và khó thu thập. Các nội dung và kết quả nghiên cứu của chương này đã được công bố ở các công trình nghiên cứu [CT2], [CT3], [CT4] và [CT5].

KẾT LUẬN VÀ KIẾN NGHỊ

Luận án đã đề xuất một số giải pháp nhằm tăng cường hiệu quả của hệ thống phát hiện sự kiện ngã của con người. Các giải pháp đề xuất được thiết kế để tận dụng các thiết bị sẵn có như điện thoại, hướng tới việc tạo ra một phần mềm miễn phí và dễ dàng tiếp cận cho nhiều người. Những đóng góp chính của luận án bao gồm:

- + Đề xuất một phương pháp trích xuất đặc trưng mới từ dữ liệu gia tốc 3 trục. Phương pháp này kết hợp các đặc trưng từ miền thời gian, miền tần số, cùng với các tham số định hướng và tham số Hjorth. Việc kết hợp đa dạng các loại đặc trưng này giúp tăng cường khả năng nhận dạng các sự kiện ngã một cách hiệu quả và đã được chứng minh có hiệu suất tốt hơn các nghiên cứu trước đó bằng thực nghiệm trên cùng bộ dữ liệu.
- + Đề xuất thuật toán GA4RF, thể hiện khả năng tìm kiếm bộ siêu tham số tối ưu cho Random Forest nhờ cấu trúc nhiệm sắc thể, không gian tìm kiếm và hàm thích nghi được thiết kế theo hướng mới, phù hợp với đặc điểm của các bộ dữ liệu ngã.
- + Đề xuất mô hình phát hiện sự kiện ngã (FEDM) dựa trên ưu điểm của thuật toán GA4RF đã tối ưu, cùng với kỹ thuật so khớp dữ liệu linh hoạt và kỹ thuật ngưỡng. FEDM có thể phân biệt hiệu quả giữa các hoạt động bình thường (dữ liệu huấn luyện) và các sự kiện ngã (dữ liệu mới), mang lại khả năng phát hiện ngã chính xác ngay cả khi dữ liệu về sự kiện ngã khan hiếm.

Mặc dù các đề xuất và kết quả của luận án đã đạt được các mục tiêu đề ra, NCS nhận thấy luận án còn một số hạn chế cần được khắc phục trong tương lai:

- + Chưa thu thập được bộ dữ liệu về hoạt động của người Việt Nam và mới thử nghiệm trên hai bộ dữ liệu khác nhau.

- + Chưa thử nghiệm với nhiều phương pháp khác và chưa so sánh với các kỹ thuật lựa chọn đặc trưng tự động như phân tích thành phần chính (PCA), phân tích thành phần độc lập (ICA), giảm số chiều phi tuyến (t-SNE), ...
- + Chưa khám phá nhiều giải pháp tối ưu khác ngoài giải thuật di truyền như Tối ưu hóa Bayesian, Hyperband, Adam, ...
- + Phần mềm phát hiện sự kiện ngã mới chỉ ở dạng nguyên mẫu, cần được phát triển và thử nghiệm thực tế.

Để tiếp tục phát triển và hoàn thiện hệ thống phát hiện sự kiện ngã, đề xuất các hướng nghiên cứu sau:

- + Xây dựng bộ dữ liệu về hoạt động của người Việt Nam và đa dạng hóa nguồn dữ liệu để đánh giá toàn diện hơn hiệu quả của mô hình.
- + Thử nghiệm các phương pháp trích xuất đặc trưng khác nhau, đặc biệt là các kỹ thuật lựa chọn đặc trưng tự động để tìm ra bộ đặc trưng tối ưu cho từng bộ dữ liệu cụ thể.
- + Khám phá và áp dụng các kỹ thuật tối ưu mô hình tiên tiến hơn như Bayesian Optimization, Hyperband, hoặc các biến thể của giải thuật di truyền.
- + Xây dựng mô hình với dữ liệu thời gian thực, đồng thời hoàn thiện phần mềm phát hiện sự kiện ngã, tích hợp thêm các tính năng như cảnh báo khẩn cấp, theo dõi sức khỏe liên tục và triển khai thử nghiệm thực tế trên các đối tượng người dùng mục tiêu.
- + Nghiên cứu kết hợp với các loại cảm biến khác như cảm biến nhịp tim, cảm biến vị trí cũng như tiếp cận vấn đề theo hướng học chuyển giao để nâng cao độ chính xác và độ tin cậy của hệ thống phát hiện sự kiện ngã.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

- [CT1]. **Hong-Lam Le**, Duc-Nhan Nguyen, Nguyen Thi Hau, Ha-Nam Nguyen (2022). “A Novel Feature Set Extraction Based on Accelerometer Sensor Data for Improving the Fall Detection System”, *Electronics*, 11(7), 1030. (Tập chí thuộc danh mục trong Scopus, SCIE (WoS) Q2)
- [CT2]. **Hong-Lam Le**, Duc-Nhan Nguyen, Ha-Nam Nguyen (2021). “The Novel Method of Pedestrian Fall Detection Based on PSO and RF Using Accelerometer Data”, In *2021 International Conference on System Science and Engineering (ICSSE)* (pp. 111–115). IEEE. (Kỷ yếu thuộc danh mục Scopus)
- [CT3]. **Hong-Lam Le**, Duc-Nhan Nguyen, Ngo-Thi-Thu-Trang, Ha-Nam Nguyen (2022). “Applying PSO to improve the pedestrian fall detection rate using wearable sensor data”, In *2022 24th International Conference on Advanced Communication Technology (ICACT)* (pp. 453–458). IEEE. (Kỷ yếu thuộc danh mục Scopus)
- [CT4]. **Hong-Lam Le**, Thanh-Tuoi Le, Vu Thi-Thu-Hien, Doan-Hieu Tran, Dinh Van Chau, Ngo-Thi-Thu-Trang (2023). “A Survey on the Impact of Hyperparameters on Random Forest Performance using Multiple Accelerometer Datasets”, *International Journal of Computers and their Applications (ISCA)*, Vol. 30, No. 4, 2020, 351–361. (Tập chí thuộc danh mục Scopus)
- [CT5]. **Lê Hồng Lam**, Lê Tiến Hiếu, Hà Huy Công, Bùi Xuân Vinh, Phạm Thành Công, Nguyễn Đức Nhân, Đinh Văn Châu, Nguyễn Hà Nam (2023). “Mô hình mới sử dụng kỹ thuật so khớp để phát hiện hành vi bất thường của con người.” *Kỷ yếu Hội nghị khoa học Quốc gia lần thứ XVI (FAIR-2023)*. 757–765.
- [ĐT1]. Nguyễn Hà Nam, **Lê Hồng Lam**, Nguyễn Đức Nhân (2022). “Nghiên cứu xây dựng hệ thống phát hiện và cảnh báo hành vi bất thường của bệnh nhân/người già dựa trên phân tích dữ liệu cảm biến trong điện thoại thông minh”. *Nhiệm vụ KH&CN Đại học Quốc gia Hà Nội (QG 19.47)*.

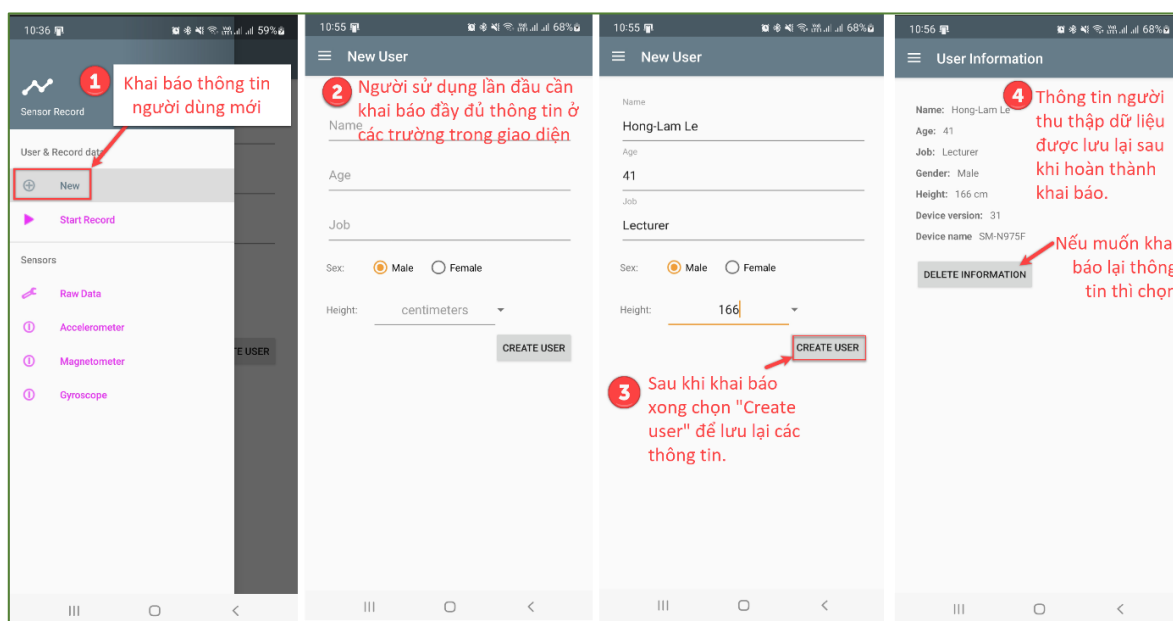
Phụ lục: XÂY DỰNG ỨNG DỤNG DEMO

1. Ứng dụng thu thập dữ liệu

Ứng dụng này được cài đặt lên SP qua file apk. Phương pháp cài đặt giống như nhiều ứng dụng khác dành cho điện thoại chạy hệ điều hành Android. Giao diện của ứng dụng thu thập dữ liệu lần đầu tiên khi mở lên có dạng như Hình PL 1. Nếu lần đầu tiên người dùng sử dụng ứng dụng, cần phải khai báo thông tin để quá trình gán nhãn và phân loại đối tượng sau này được chính xác hơn. Sau đây là phần hướng dẫn chi tiết tuần tự các bước để thu thập dữ liệu cho người mới sử dụng app lần đầu.

1.2. Khai báo thông tin của người thu thập dữ liệu,

Trình tự khai báo thông tin và quy trình thực hiện được tóm tắt như Hình PL 1.



Hình PL 1. Trình tự khai báo thông tin người dùng lần đầu sử dụng phần mềm

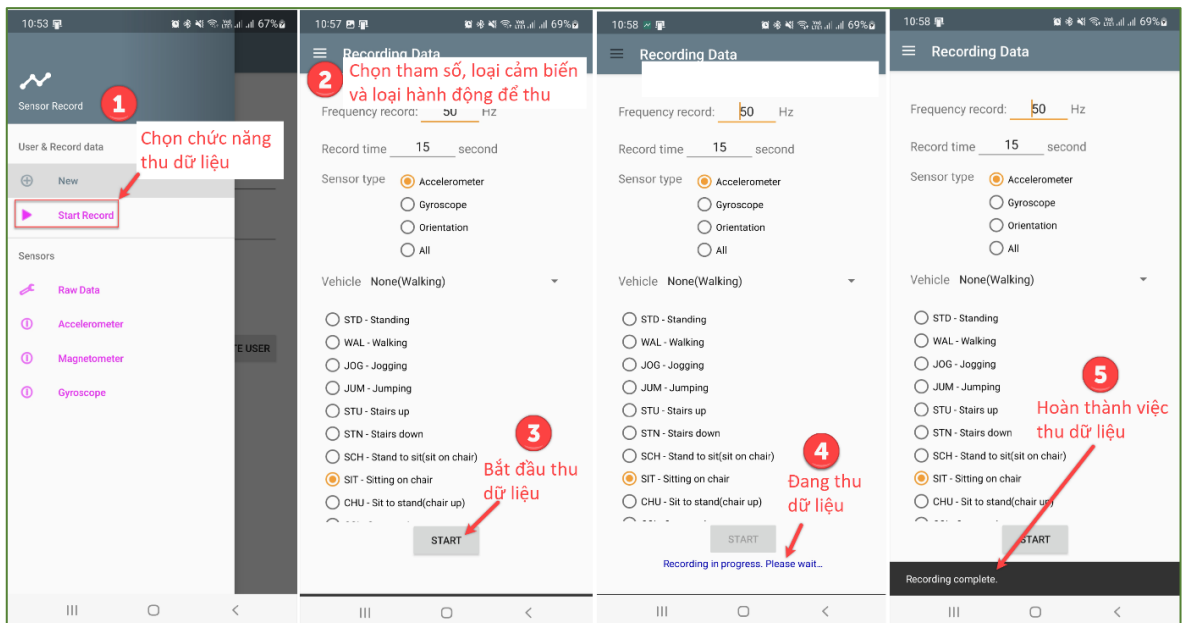
- 1 Chọn “new” để truy cập giao diện khai báo thông tin người dùng
- 2 Chọn “Name” để khai tên người dùng, chọn “Age” để khai tuổi người dùng, chọn “Job” để khai nghề nghiệp người dùng, chọn “Sex” để khai giới tính và chọn “Height” để khai báo chiều cao. Các trường Age, Job, Sex và Height là những

thông tin quan trọng. Việc khai báo đúng sự thật sẽ giúp quá trình phân tích, phân loại dữ liệu sẽ có nhiều thuận lợi và chính xác.

- 3 Sau khi khai báo xong thông tin, người dùng chọn “ CREATE USER” để lưu lại các thông tin.
- 4 Thông tin của người thu dữ liệu đã được lưu lại. Nếu người dùng muốn khai báo lại hoặc người khác sử dụng cùng thiết bị thì chọn “DELETE INFORMATION” để thực hiện lại từ bước 1.

1.3. Thu thập dữ liệu

Trình tự khai báo thông tin và quy trình thực hiện được tóm tắt như Hình PL 2.



Hình PL 2. Trình tự thực hiện thu dữ liệu.

- 1 Chọn “Start Record” để truy cập giao diện cài đặt thông số, loại cảm biến và loại hoạt động cần thu thập.

- 2 Cách lựa chọn các thông số để thu như sau:

Frequency record (Hz): Tần số ghi dữ liệu (số bản ghi dữ liệu mỗi giây), tham số này được giới hạn trong khoảng [1, 100]Hz

Record time (s): Thời gian cửa sổ ghi (độ dài thời gian thực hiện ghi dữ liệu), tham số này được giới hạn trong khoảng [1, 300] giây. Lưu ý, người dùng không thể kết

thức sớm thời gian ghi, chỉ có thể ghi đủ trong khoảng thời gian đã nhập từ trước.

Sensor type: Loại cảm biến sử dụng để thu dữ liệu. Mặc định, cảm biến gia tốc được sử dụng để thu dữ liệu người dùng.

Vehicle: Loại hoạt động muốn thu thập, trường hợp thu thập các hoạt động của người đi bộ nên chọn chức năng “Non (Walking)”. Các kiểu hoạt động của người đi bộ được thiết kế sẵn một danh sách để người dùng lựa chọn. Chẳng hạn như trong giao diện ❷ ở Hình PL 2, hoạt động ngồi trên ghế (SIT) của người dùng được lựa chọn để ghi lại thông tin bằng cảm biến gia tốc.

❸ Sau khi lựa chọn xong các tham số người dùng ấn nút “START” để bắt đầu quá trình ghi dữ liệu. Ngay sau khi ấn nút “START”, người dùng đặt điện thoại vào vị trí túi áo hoặc túi quần, sau đó thực hiện hoạt động tương ứng với lựa chọn cho tới khi hết thời gian đã cài đặt. Trong thời gian ghi, người dùng không thể thao tác với ứng dụng, cần tránh việc sửa đổi thông số khi đang ghi dữ liệu để không làm sai lệch thông tin.

❹ Giao diện hiển thị quá trình thu thập dữ liệu đang được thực hiện.

❺ Giao diện hiển thị hoàn tất quá trình thu thập dữ liệu.

1.4. Kịch bản thu thập dữ liệu

Thiết bị sử dụng: Điện thoại di động android

Các vị trí đặt thiết bị: Túi áo, túi quần, cầm tay, trong túi hoặc ba lô

Nhóm người thử nghiệm (dự kiến): 15 nam, 10 nữ (Độ tuổi 20-30)

❶ Thu thập các hoạt động cơ bản hàng ngày của con người:

Bảng PL 1. Mô tả các hoạt động cơ bản hàng ngày của con người để thu thập

Hoạt động	Nhãn	Mô tả
Đứng	STD	Đứng với với cử động nhẹ nhàng
Đi bộ	WAL	Đi bộ bình thường
Chạy	JOG	Chạy bộ

Hoạt động	Nhãn	Mô tả
Nhảy	JUM	Nhảy liên tục
Lên cầu thang	STU	Đi lên cầu thang (10 bậc)
Xuống cầu thang	STN	Đi xuống cầu thang (10 bậc)
Đứng sang ngòai	SCH	Chuyển tiếp từ đứng sang ngòai (ngòai trên ghế)
Ngòai trên ghế	SIT	Ngòai trên ghế với những cử động nhẹ nhàng
Ngòai sang đứng	CHU	Chuyển tiếp từ ngòai sang đứng (đứng lên từ ghế)
Nằm	LYI	Nằm nghỉ

❷ Thu thập các sự kiện ngã của con người để thu thập:

Bảng PL 2. Mô tả các sự kiện ngã của con người để thu thập

Hoạt động	Nhãn	Mô tả
Ngã về phía trước	FOL	Ngã về phía trước khi đang đứng, sử dụng tay để chống.
Ngã về phía trước	FKL	Ngã về phía trước khi đang đứng, đầu gối tiếp xúc đầu tiên.
Ngã ngửa	BSC	Ngã ngửa trong khi đang ngòai trên ghế
Ngã sang 1 bên	SDL	Ngã sang một bên khi đang đứng, gập chân

1.5. Chi tiết kịch bản

❶ Kịch bản thu thập dữ liệu ADL

Bảng PL 3. Các quy ước để thu thập những hoạt động cơ bản hàng ngày

Hoạt động	Số người thực hiện	Thời lượng mỗi bản ghi	Số lần thực hiện	Mô tả
Đứng	25	3 phút	2	Đứng yên tại một chỗ, có thể đọc sách hoặc thực hiện một việc gì đó bằng tay, chân không di chuyển
Đi bộ	25	3 phút	2	Đi bộ với tốc độ thông thường theo lộ trình cho trước
Chạy	25	30 giây	3	Chạy chậm với tốc độ đều đặn
Nhảy	25	30 giây	3	Nhảy liên tục trên đường phẳng

Hoạt động	Số người thực hiện	Thời lượng mỗi bản ghi	Số lần thực hiện	Mô tả
Lên cầu thang	25	15 giây	3	Đi lên cầu thang liên tục với tốc độ bình thường (khoảng 15 bậc)
Xuống cầu thang	25	15 giây	3	Đi xuống cầu thang liên tục với tốc độ bình thường (khoảng 15 bậc)
Đứng sang ngồi	25	5 giây	3	Chuyển tiếp từ đứng sang ngồi (ngồi trên ghế)
Ngồi trên ghế	25	1 phút	2	Ngồi trên ghế với những cử động nhẹ nhàng, có thể đọc sách.
Ngồi sang đứng	25	5 giây	3	Chuyển tiếp từ ngồi sang đứng (đứng lên từ ghế)
Nằm	25	3 phút	2	Nằm nghỉ, có thể nằm ngửa hoặc nằm nghiêng nhưng không cử động nhiều

❷ Kịch bản sự kiện ngã

Bảng PL 4. Các quy ước để thu thập những sự kiện ngã

Hoạt động	Số người thực hiện	Thời lượng mỗi bản ghi	Số lần thực hiện	Mô tả
Ngã về phía trước	5 nam, 3 nữ	05 giây	3	Ngã chúi đầu ra phía trước
Ngã về phía trước	5 nam, 3 nữ	05 giây	3	Ngã khụy gối
Ngã ngửa	5 nam, 3 nữ	05 giây	3	Ngã ngồi ra phía sau hoặc trượt chân ngã ngồi
Ngã sang 1 bên	5 nam, 3 nữ	05 giây	3	Ngã nghiêng người

Nhóm người thực hiện được trang bị thiết bị di động có khả năng thu thập dữ liệu và thực hiện theo nhóm các hoạt động đã được liệt kê ở Bảng PL 3 và PL 4. Để đảm bảo dữ liệu đầu ra chất lượng, tất cả các yếu tố bao gồm bên trong và bên ngoài đều phải được tuân thủ (các yếu tố tự nhiên, thời gian,...). Tần số lấy mẫu mặc định là 50Hz.

2. Truy cập dữ liệu lưu trữ sau khi thu

Sau khi người dùng thực hiện quá trình thu thập theo kịch bản. Mặc định, dữ liệu được lưu trữ ở bộ nhớ trong của SP như mô tả ở Hình PL 3.

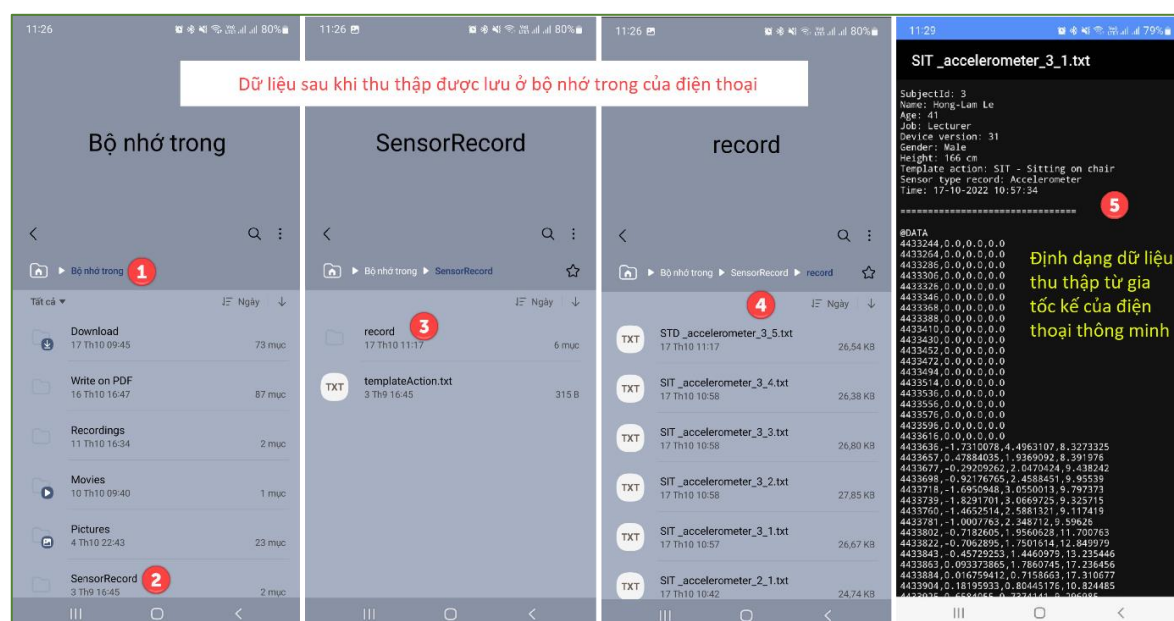
Vị trí các bạn ghi:

❶ Bộ nhớ trong \ ❷ SensorRecord \ ❸ record \

❹ Các file dữ liệu sau khi thu thập được đặt tên theo quy tắc:

<Nhãn hoạt động>_<Loại cảm biến được dùng>_<Người thu>_<lần thu thứ>.txt

❺ Định dạng chi tiết dữ liệu của một bản ghi.



Hình PL 3. Vị trí lưu trữ dữ liệu sau khi được thu thập.

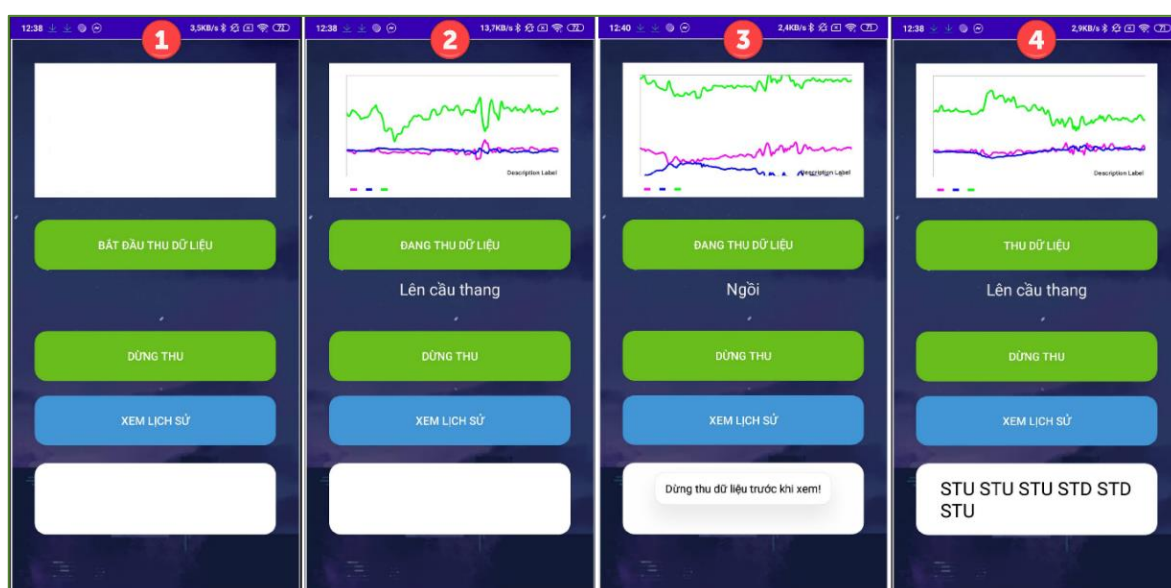
3. Thử nghiệm phần mềm giám sát

Các kết quả như trình bày ở Mục 3.3.3.3.d được sử dụng để xây dựng phần mềm giám sát hoạt động của người dùng. Phần mềm giám được phát triển ở dạng nguyên mẫu để đánh giá hiệu suất phát hiện các sự kiện ngã trong thực tế. Phần mềm xây dựng trên hệ điều hành mã nguồn mở phổ biến nhất hiện nay là Android.

Phần mềm giám sát được xây dựng dựa vào phương pháp xử lý luồng dữ liệu và sử dụng mô hình FEDM như đã trình bày ở Chương 3. Từ đó có thể phát hiện hoạt động nguy hiểm và cảnh báo bằng âm thanh và với các thiết bị được kết nối thông qua các API của ứng dụng. Dữ liệu được thu thập từ cảm biến gia tốc là dạng dữ liệu liên tục theo thời gian với tần số lấy mẫu 50Hz. Tuy nhiên, việc thu thập dữ liệu để đưa vào

hàm trích xuất đặc trưng và xử lý không đảm bảo luồng dữ liệu được liên tục. Vì vậy, kỹ thuật luồng (thread) để xử lý song song quá trình tính toán đã được sử dụng.

Ứng dụng thu thập dữ liệu và phát hiện hoạt động người dùng được phát triển bằng ngôn ngữ Java trên nền tảng IDE Android Studio. Việc phát triển ứng dụng bằng ngôn ngữ Java mang lại rất nhiều lợi thế bởi java được mọi người ngầm định là một ngôn ngữ lập trình cơ bản nhất, có nhiều công cụ hỗ trợ và các công cụ phát triển. Ngoài ra ngôn ngữ Java còn cho phép tận dụng và tái tạo các mô-đun cho việc mở rộng và phát triển ứng dụng trong tương lai, việc sử dụng java để phát triển ứng dụng hiện nay là dễ dàng và hoàn toàn hợp lí. Giao diện phần mềm nhóm đang phát triển được minh họa ở Hình PL 4.



Hình PL2. 4. Giao diện phần mềm phát hiện các sự kiện ngã.

Để tiến hành tiến giám sát các hoạt động của người dùng theo thời gian thực bằng SP, người hỗ trợ cần khởi động ứng dụng và thực hiện như sau:

u Người dùng/người giám sát bấm vào nút “BẮT ĐẦU THU DỮ LIỆU” để điện thoại tiến hành thu thập, phân tích và dự đoán các hoạt động đang diễn ra của người sử dụng.

- v Sau 3s kể từ lúc hoạt động của người dùng diễn ra, ứng dụng sẽ phân tích, dự đoán và trả về kết quả dự đoán. Cửa sổ biểu diễn tín hiệu hiển thị dạng sóng, cửa sổ hiển thị kết quả bằng chuỗi ký tự và âm thanh mô tả hoạt động đoán được. Nếu xảy ra hoạt động có nguy cơ ngã, phần mềm sẽ gửi bản tin cảnh báo đến số điện thoại của người chăm sóc.
- w Nếu muốn xem lại chuỗi các hoạt động của người cần theo dõi đã diễn ra trong thời gian giám sát, người hỗ trợ cần bấm vào nút “DỪNG THU”. Nếu không ấn “DỪNG THU” sẽ không xem được lịch sử hoạt động.
- ④ Sau khi bấm vào nút “DỪNG THU”, để xem lại lịch sử chuỗi các hoạt động của người dùng, người hỗ trợ cần bấm vào nút “XEM LỊCH SỬ”. Tất cả các hoạt động đã diễn ra được mã hóa thành các nhãn như mô tả ở Bảng PL 1 và Bảng PL 2.

TÀI LIỆU THAM KHẢO

1. Falls. (n.d.). Retrieved March 10, 2023, from <https://www.who.int/news-room/fact-sheets/detail/falls>
2. Inouye, S. K., Brown, C. J., & Tinetti, M. E. (2009). Medicare nonpayment, hospital falls, and unintended consequences. *New England Journal of Medicine*, *360*(23), 2390–2393.
3. Orces, C. H., & Alamgir, H. (2014). Trends in fall-related injuries among older adults treated in emergency departments in the USA. *Injury prevention*.
4. Fleming, J., & Brayne, C. (2008). Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. *Bmj*, *337*.
5. Rescio, G., Leone, A., & Siciliano, P. (2018). Supervised machine learning scheme for electromyography-based pre-fall detection system. *Expert Systems with Applications*, *100*, 95–105.
6. WHO. (n.d.). *Falls*. World Health Organization. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/falls>
7. Ferrari, A., Micucci, D., Mobilio, M., & Napoletano, P. (2021). Trends in human activity recognition using smartphones. *Journal of Reliable Intelligent Environments*, *7*(3), 189–213. <https://doi.org/10.1007/s40860-021-00147-0>
8. Xu, J., He, Z., & Zhang, Y. (2019). CNN-LSTM combined network for IoT enabled fall detection applications. In *Journal of Physics: Conference Series* (Vol. 1267, p. 012044). IOP Publishing.
9. Lai, K., Yanushkevich, S. N., Shmerko, V., & Hou, M. (2021). Capturing causality and bias in human action recognition. *Pattern Recognition Letters*, *147*, 164–171.
10. Kraft, D., Srinivasan, K., & Bieber, G. (2020). Deep learning based fall detection algorithms for embedded systems, smartwatches, and IoT devices using accelerometers. *Technologies*, *8*(4), 72.
11. Chen, Z., Jiang, C., & Xie, L. (2019). A Novel Ensemble ELM for Human Activity Recognition Using Smartphone Sensors. *IEEE Transactions on Industrial Informatics*, *15*(5), 2691–2699. Presented at the IEEE Transactions on Industrial Informatics. <https://doi.org/10.1109/TII.2018.2869843>

12. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., & Peñafort-Asturiano, C. (2019). UP-fall detection dataset: A multimodal approach. *Sensors*, *19*(9), 1988.
13. Chatzaki, C., Pediaditis, M., Vavoulas, G., & Tsiknakis, M. (2017). Human Daily Activity and Fall Recognition Using a Smartphone's Acceleration Sensor. In C. Röcker, J. O'Donoghue, M. Ziefle, M. Helfert, & W. Molloy (Eds.), *Information and Communication Technologies for Ageing Well and e-Health* (pp. 100–118). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-62704-5_7
14. Mahfuz, S., Isah, H., Zulkernine, F., & Nicholls, P. (2018). Detecting irregular patterns in IoT streaming data for fall detection. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 588–594). IEEE.
15. Tsinganos, P., & Skodras, A. (2017). A smartphone-based fall detection system for the elderly. In *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis* (pp. 53–58). IEEE.
16. Hoàng Phê. (2017). *Từ điển Tiếng Việt*. Công ty CP Văn hóa Văn Lang: Hồng Đức.
17. Cẩm nang MSD. (n.d.). Ngã ở Người lớn tuổi. *MSD manual - Phiên bản dành cho chuyên gia*. Retrieved from <https://www.msmanuals.com/vi-vn/chuyen-gia/nguoi-lao-khoa/ngã-ở-người-lớn-tuổi/ngã-ở-người-lớn-tuổi?query=té%20ngã>
18. Choudhury, N. A., & Soni, B. (2023). In-depth analysis of design & development for sensor-based human activity recognition system. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-023-16423-5>
19. Hussain, F., Hussain, F., Ehatisham-ul-Haq, M., & Azam, M. A. (2019). Activity-aware fall detection and recognition based on wearable sensors. *IEEE Sensors Journal*, *19*(12), 4528–4536.
20. Azmat, U., Ghadi, Y. Y., Shloul, T. al, Alsuhibany, S. A., Jalal, A., & Park, J. (2022). Smartphone sensor-based human locomotion surveillance system using multilayer perceptron. *Applied Sciences*, *12*(5), 2550.

21. Hassan, M. M., Gumaei, A., Aloï, G., Fortino, G., & Zhou, M. (2019). A smartphone-enabled fall detection framework for elderly people in connected home healthcare. *IEEE Network*, 33(6), 58–63.
22. Saha, A., Rajak, S., Saha, J., & Chowdhury, C. (2022). A Survey of Machine Learning and Meta-heuristics Approaches for Sensor-based Human Activity Recognition Systems. *Journal of Ambient Intelligence and Humanized Computing*, 1–28.
23. Casilari, E., Lora-Rivera, R., & García-Lagos, F. (2020). A study on the application of convolutional neural networks to fall detection evaluated with multiple public datasets. *Sensors*, 20(5), 1466.
24. Singh, A., Rehman, S. U., Yongchareon, S., & Chong, P. H. J. (2020). Sensor technologies for fall detection systems: A review. *IEEE Sensors Journal*, 20(13), 6889–6919.
25. Ramachandran, A., & Karuppiah, A. (2020). A survey on recent advances in wearable fall detection systems. *BioMed research international*, 2020.
26. Alazrai, R., Mowafi, Y., & Hamad, E. (2015). A fall prediction methodology for elderly based on a depth camera. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 4990–4993). IEEE.
27. Almaslukh, B., Artoli, A. M., & Al-Muhtadi, J. (2018). A robust deep learning approach for position-independent smartphone-based human activity recognition. *Sensors*, 18(11), 3726.
28. Shakya, S. R., Zhang, C., & Zhou, Z. (2018). Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data. *Int. J. Mach. Learn. Comput*, 8(6), 577–582.
29. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 1–58.
30. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
31. Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5–32.
32. Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278–282). IEEE.

33. Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2), 123–140.
34. Kukreja, H., Bharath, N., Siddesh, C., & Kuldeep, S. (2016). An introduction to artificial neural network. *Int J Adv Res Innov Ideas Educ*, 1, 27–30.
35. Xiao, G., Li, K., & Li, K. (2017). Reporting l most influential objects in uncertain databases based on probabilistic reverse top-k queries. *Information Sciences*, 405, 207–226.
36. Manwani, N., & Sastry, P. S. (2011). Geometric decision tree. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(1), 181–192.
37. Gehrke, J., Ganti, V., Ramakrishnan, R., & Loh, W.-Y. (1999). BOAT—optimistic decision tree construction. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of Data* (pp. 169–180).
38. Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.
39. Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., & Hall, M. (2002). Multiclass alternating decision trees. In *European Conference on Machine Learning* (pp. 161–172). Springer.
40. Yadav, C., Wang, S., & Kumar, M. (2013). Algorithm and approaches to handle large Data-A Survey. *arXiv preprint arXiv:1307.5437*.
41. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Routledge.
42. Quinlan, J. R. (1996). Improved use of continuous attributes in C4. 5. *Journal of artificial intelligence research*, 4, 77–90.
43. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1), 1–13.
44. Tague-Sutcliffe, J. (1992). The pragmatics of information retrieval experimentation, revisited. *Information processing & management*, 28(4), 467–490.
45. Lewis, D. D., Yang, Y., Russell-Rose, T., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr), 361–397.

46. Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Random k-labelsets for multilabel classification. *IEEE transactions on knowledge and data engineering*, 23(7), 1079–1089.
47. Pillai, I., Fumera, G., & Roli, F. (2017). Designing multi-label classifiers that maximize F measures: State of the art. *Pattern Recognition*, 61, 394–404.
48. Price, L. R. (2016). *Psychometric methods: Theory into practice*. Guilford Publications.
49. Nooruddin, S., Islam, M. M., Sharna, F. A., Alhetari, H., & Kabir, M. N. (2021). Sensor-based fall detection systems: a review. *Journal of Ambient Intelligence and Humanized Computing*, 1–17.
50. Igual, R., Medrano, C., & Plaza, I. (2015). A comparison of public datasets for acceleration-based fall detection. *Medical engineering & physics*, 37(9), 870–878.
51. Pannurat, N., Thiemjarus, S., & Nantajeewarawat, E. (2017). A hybrid temporal reasoning framework for fall monitoring. *IEEE Sensors Journal*, 17(6), 1749–1759.
52. Shi, J., Chen, D., & Wang, M. (2020). Pre-impact fall detection with CNN-based class activation mapping method. *Sensors*, 20(17), 4750.
53. Wu, X., Cheng, L., Chu, C.-H., & Kim, J. (2019). Using deep learning and smartphone for automatic detection of fall and daily activities. In *International Conference on Smart Health* (pp. 61–74). Springer.
54. Le, T. M., Van Tran, L., & Dao, S. V. T. (2021). A Feature Selection Approach for Fall Detection Using Various Machine Learning Classifiers. *IEEE Access*, 9, 115895–115908.
55. Santoyo-Ramón, J. A., Casilari, E., & Cano-García, J. M. (2021). A study of one-class classification algorithms for wearable fall sensors. *Biosensors*, 11(8), 284.
56. Al Nahian, M. J., Ghosh, T., Al Banna, M. H., Aseeri, M. A., Uddin, M. N., Ahmed, M. R., ... Kaiser, M. S. (2021). Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features. *IEEE Access*, 9, 39413–39431.

57. Chatzaki, C., Pediaditis, M., Vavoulas, G., & Tsiknakis, M. (2017). Human Daily Activity and Fall Recognition Using a Smartphone's Acceleration Sensor.
58. Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*. Springer Science & Business Media.
59. Bala, D., & Waliullah, G. M. (2020). Efficient Classification Techniques of Human Activities from Smartphone Sensor Data Using Machine Learning Algorithms. *International Journal of Knowledge Based Computer Systems*, 8. Retrieved from https://www.academia.edu/download/67799374/201_1.pdf
60. Panhwar, M., Shah, S. M. S., Shah, S. M. Z. S., Shah, S. M., & Chowdhry, B. S. (2017). Smart phone based fall detection using auto regression modeling in a non-restrictive setting. *Indian journal of science and technology*, 10(5), e111274–e111274.
61. Perry, J. T., Kellog, S., Vaidya, S. M., Youn, J.-H., Ali, H., & Sharif, H. (2009). Survey and evaluation of real-time fall detection approaches. In *2009 6th International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET)* (pp. 158–164). IEEE.
62. Chelli, A., & Pätzold, M. (2019). A machine learning approach for fall detection and daily living activity recognition. *IEEE Access*, 7, 38670–38687.
63. Wang, H., Li, M., Li, J., Cao, J., & Wang, Z. (2016). An improved fall detection approach for elderly people based on feature weight and Bayesian classification. In *2016 IEEE International Conference on Mechatronics and Automation* (pp. 471–476). IEEE.
64. He, J., Bai, S., & Wang, X. (2017). An unobtrusive fall detection and alerting system based on Kalman filter and Bayes network classifier. *Sensors*, 17(6), 1393.
65. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46–61.
66. De Leonardis, G., Rosati, S., Balestra, G., Agostini, V., Panero, E., Gastaldi, L., & Knaflitz, M. (2018). Human Activity Recognition by Wearable Sensors: Comparison of different classifiers for real-time applications. In *2018 IEEE international symposium on medical measurements and applications (memea)* (pp. 1–6). IEEE.

67. Putra, I., & Vesilo, R. (2019). Genetic-algorithm-based feature-selection technique for fall detection using multi-placement wearable sensors. In *Advances in Body Area Networks I* (pp. 319–332). Springer.
68. El-Maaty, A. M. A., & Wassal, A. G. (2018). Hybrid GA-PCA feature selection approach for inertial human activity recognition. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1027–1032). IEEE.
69. Nguyen, T. D., Huynh, T.-T., & Pham, H.-A. (2018). An improved human activity recognition by using genetic algorithm to optimize feature vector. In *2018 10th International Conference on Knowledge and Systems Engineering (KSE)* (pp. 123–128). IEEE.
70. Zhang, J., Sun, G., Sun, Y., Dou, H., & Bilal, A. (2020). Hyper-parameter optimization by using the genetic algorithm for upper limb activities recognition based on neural networks. *IEEE Sensors Journal*, *21*(2), 1877–1884.
71. Batool, M., Jalal, A., & Kim, K. (2019). Sensors technologies for human activity analysis based on SVM optimized by PSO algorithm. In *2019 International Conference on Applied and Engineering Mathematics (ICAEM)* (pp. 145–150). IEEE.
72. Ning, Y., Zhang, S., Xiong, W., Li, G., & Zhao, G. (2019). Human action recognition algorithm based on dbpso-svm classifier. In *2019 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)* (pp. 1–5). IEEE.
73. Pham, C., Diep, N. N., & Phuong, T. M. (2013). A wearable sensor based approach to real-time fall detection and fine-grained activity recognition. *Journal of Mobile Multimedia*, 015–026.
74. Tuấn, L. N., Pham, C., & Văn, T. N. (2020). Phát hiện hoạt động bất thường của người bằng mạng học sâu tích chập kết hợp mạng bộ nhớ dài ngắn: *Research and Development on Information and Communication Technology*, 35–43. <https://doi.org/10.32913/mic-ict-research-vn.v2020.n1.925>
75. Pham, C. (2015). MobiRAR: Real-time human activity recognition using mobile devices. In *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)* (pp. 144–149). IEEE. Retrieved from <https://ieeexplore.ieee.org/abstract/document/7371773/>

76. Nguyen, D.-A., Pham, C., Argent, R., Caulfield, B., & Le-Khac, N.-A. (2023). Towards a New Multi-tasking Learning Approach for Human Fall Detection. In N. T. Nguyen, H. Le-Minh, C.-P. Huynh, & Q.-V. Nguyen (Eds.), *The 12th Conference on Information Technology and Its Applications* (pp. 50–61). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36886-8_5
77. Pham, C., Nguyen, L., Nguyen, A., Nguyen, N., & Nguyen, V.-T. (2021). Combining skeleton and accelerometer data for human fine-grained activity recognition and abnormal behaviour detection with deep temporal convolutional networks. *Multimedia Tools and Applications*, 80(19), 28919–28940. <https://doi.org/10.1007/s11042-021-11058-w>
78. Ha, T. V., Nguyen, H., Huynh, S. T., Nguyen, T. T., & Nguyen, B. T. (2022). Fall Detection Using Multimodal Data. In *International Conference on Multimedia Modeling* (pp. 392–403). Springer.
79. Tran, D. N., & Phan, D. D. (2016). Human activities recognition in android smartphone using support vector machine. In *2016 7th international conference on intelligent systems, modelling and simulation (isms)* (pp. 64–68). IEEE.
80. Tran, T.-H., Le, T.-L., Hoang, V.-N., & Vu, H. (2017). Continuous detection of human fall using multimodal features from Kinect sensors in scalable environment. *Computer methods and programs in biomedicine*, 146, 151–165.
81. Kaenampornpan, M., Anuchad, T., & Supaluck, P. (2011). Fall detection prototype for Thai elderly in mobile computing era. In *The 8th Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand-Conference 2011* (pp. 446–449). IEEE.
82. Kose, M., Incel, O. D., & Ersoy, C. (2012). Online human activity recognition on smart phones. In *Workshop on mobile sensing: from smartphones and wearables to big data* (Vol. 16, pp. 11–15).
83. Anguita, D., Ghio, A., Oneto, L., Parra Perez, X., & Reyes Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning* (pp. 437–442).

84. Ronao, C. A., & Cho, S.-B. (2015). Deep convolutional neural networks for human activity recognition with smartphone sensors. In *International Conference on Neural Information Processing* (pp. 46–53). Springer.
85. Hassan, M. M., Gumaei, A., Aloï, G., Fortino, G., & Zhou, M. (2019). A smartphone-enabled fall detection framework for elderly people in connected home healthcare. *IEEE Network*, 33(6), 58–63.
86. Harari, Y., Shawen, N., Mummidisetty, C. K., Albert, M. V., Kording, K. P., & Jayaraman, A. (2021). A smartphone-based online system for fall detection with alert notifications and contextual information of real-life falls. *Journal of neuroengineering and rehabilitation*, 18(1), 1–13.
87. Gia, T. N., Sarker, V. K., Tcareno, I., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2018). Energy efficient wearable sensor node for IoT-based fall detection systems. *Microprocessors and Microsystems*, 56, 34–46.
88. Figueiredo, I. N., Leal, C., Pinto, L., Bolito, J., & Lemos, A. (2016). Exploring smartphone sensors for fall detection. *mUX: the journal of mobile user experience*, 5(1), 1–17.
89. Ferrari, A., Micucci, D., Mobilio, M., & Napoletano, P. (2019). Hand-crafted Features vs Residual Networks for Human Activities Recognition using Accelerometer. In *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)* (pp. 153–156). Presented at the 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT). <https://doi.org/10.1109/ISCT.2019.8901021>
90. Figo, D., Diniz, P. C., Ferreira, D. R., & Cardoso, M. P. (2010). Preprocessing techniques for context recognition from accelerometer data, 645–662. <https://doi.org/10.1007/s00779-010-0293-9>
91. Shoaib, M., Bosch, S., Incel, O., Scholten, H., & Havinga, P. (2015). A Survey of Online Activity Recognition Using Mobile Phones. *Sensors*, 15(1), 2059–2085. <https://doi.org/10.3390/s150102059>
92. Chatzaki, C., Pediaditis, M., Vavoulas, G., & Tsiknakis, M. (2017). Human Daily Activity and Fall Recognition Using a Smartphone’s Acceleration Sensor. In C. Röcker, J. O’Donoghue, M. Ziefle, M. Helfert, & W. Molloy (Eds.), *Information and Communication Technologies for Ageing Well and e-*

- Health* (pp. 100–118). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-62704-5_7
93. Veltink, P. H., Bussmann, H. J., De Vries, W., Martens, W. J., & Van Lummel, R. C. (1996). Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Transactions on Rehabilitation Engineering*, 4(4), 375–385.
 94. Van Hees, V. T., Gorzelniak, L., Dean León, E. C., Eder, M., Pias, M., Taherian, S., ... Horsch, A. (2013). Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity. *PloS one*, 8(4), e61691.
 95. Twomey, N., Diethe, T., Fafoutis, X., Elsts, A., McConville, R., Flach, P., & Craddock, I. (2018). A comprehensive study of activity recognition using accelerometers. In *Informatics* (Vol. 5, p. 27). MDPI. Retrieved from <https://www.mdpi.com/2227-9709/5/2/27>
 96. Climent-Pérez, P., & Florez-Revuelta, F. (2022). Privacy-preserving human action recognition with a many-objective evolutionary algorithm. *Sensors*, 22(3), 764.
 97. Gao, G., Li, Z., Huan, Z., Chen, Y., Liang, J., Zhou, B., & Dong, C. (2021). Human behavior recognition model based on feature and classifier selection. *Sensors*, 21(23), 7791.
 98. Anguita, D., Ghio, A., Oneto, L., Parra Perez, X., & Reyes Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 437–442).
 99. Fujiki, Y. (2010). iPhone as a physical activity measurement platform. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (pp. 4315–4320). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1753846.1754146>
 100. Ortiz, J. L. R. (2014). *Smartphone-based human activity recognition* (PhD Thesis). Universitat Politècnica de Catalunya (UPC).
 101. Andreas, A. (2006). *Digital signal processing: Signals, systems, and filters*. McGraw-Hill, New York.

102. Banos, O., Galvez, J.-M., Damas, M., Pomares, H., & Rojas, I. (2014). Window size impact in human activity recognition. *Sensors*, *14*(4), 6474–6499.
103. Janidarmian, M., Roshan Fekr, A., Radecka, K., & Zilic, Z. (2017). A comprehensive analysis on wearable acceleration sensors in human activity recognition. *Sensors*, *17*(3), 529.
104. Lara, O. D., & Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, *15*(3), 1192–1209.
105. Putra, I. P. E. S., & Vesilo, R. (2017). Window-size impact on detection rate of wearable-sensor-based fall detection using supervised machine learning. In *2017 IEEE Life Sciences Conference (LSC)* (pp. 21–26). IEEE.
106. Hjorth, B. O. (1970). Technical contributions eeg analysis based on time domain properties, 306–310.
107. Vavoulas, G., Chatzaki, C., Malliotakis, T., Pediaditis, M., & Tsiknakis, M. (2016). The mobiaact dataset: Recognition of activities of daily living using smartphones. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health* (Vol. 2, pp. 143–151). SCITEPRESS.
108. Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, *9*(3), e1301.
109. Daviran, M., Maghsoudi, A., Ghezlbash, R., & Pradhan, B. (2021). A new strategy for spatial predictive mapping of mineral prospectivity: Automated hyperparameter tuning of random forest approach. *Computers & Geosciences*, *148*, 104688.
110. Mohapatra, N., Shreya, K., & Chinmay, A. (2020). Optimization of the random forest algorithm. In *Advances in Data Science and Management: Proceedings of ICDSM 2019* (pp. 201–208). Springer.
111. Kelkar, K. M., & Bakal, J. W. (2020). Hyper Parameter Tuning of Random Forest Algorithm for Affective Learning System. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1192–1195). IEEE.

112. Ge, D.-M., Zhao, L.-C., & Esmaeili-Falak, M. (2022). Estimation of rapid chloride permeability of SCC using hyperparameters optimized random forest models. *Journal of Sustainable Cement-Based Materials*, 0(0), 1–19. <https://doi.org/10.1080/21650373.2022.2093291>
113. Boehmke, B., & Greenwell, B. (2019). *Hands-on machine learning with R*. Chapman and Hall/CRC.
114. Dufour, J.-M., & Neves, J. (2019). Chapter 1 - Finite-sample inference and nonstandard asymptotics with Monte Carlo tests and R. In H. D. Vinod & C. R. Rao (Eds.), *Handbook of Statistics* (Vol. 41, pp. 3–31). Elsevier. <https://doi.org/10.1016/bs.host.2019.05.001>
115. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).
116. Liashchynskiy, P., & Liashchynskiy, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. *arXiv preprint arXiv:1912.06059*.
117. Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer.
118. Sastry, K., Goldberg, D. E., & Kendall, G. (2014). Genetic algorithms. *Search methodologies: Introductory tutorials in optimization and decision support techniques*, 93–117.
119. Ankalaki, S., & Thippeswamy, M. N. (2023). A novel optimized parametric hyperbolic tangent swish activation function for 1D-CNN: application of sensor-based human activity recognition and anomaly detection. *Multimedia Tools and Applications*, 1–31.
120. Lai, K., Yanushkevich, S. N., Shmerko, V., & Hou, M. (2021). Capturing causality and bias in human action recognition. *Pattern Recognition Letters*, 147, 164–171.
121. Nguyen, D.-A., Pham, C., Argent, R., Caulfield, B., & Le-Khac, N.-A. (2024). Model and Empirical Study on Multi-tasking Learning for Human Fall Detection. *Vietnam Journal of Computer Science*, 1–14. <https://doi.org/10.1142/S2196888824500040>