

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Trần Hữu Nam**

**NGHIÊN CỨU, PHÁT TRIỂN MÔ HÌNH HỌC  
MÁY TẠI BIÊN NHĂM PHÂN LOẠI TƯ THẾ  
NGỦ**

**LUẬN VĂN THẠC SĨ**

**HÀ NỘI - 2025**

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Trần Hữu Nam

NGHIÊN CỨU, PHÁT TRIỂN MÔ HÌNH HỌC  
MÁY TẠI BIÊN NHẨM PHÂN LOẠI TƯ THẾ  
NGỦ

Ngành: Điện tử viễn thông  
Chuyên ngành: Kỹ thuật điện tử  
Mã số học viên: 23025029

LUẬN VĂN THẠC SĨ

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS. Mai Anh Tuấn

HÀ NỘI - 2025

## LỜI CAM ĐOAN

Tôi xin cam đoan: luận văn thạc sĩ “Nghiên cứu, phát triển mô hình học máy tại biên nhằm phân loại tư thế ngủ” là công trình nghiên cứu của tôi dưới sự hướng dẫn của thầy **PGS. TS. Mai Anh Tuấn** và thầy **ThS. Trần Ngọc Thái** cùng với sự tham khảo từ những tài liệu đã liệt kê trong mục Tài liệu tham khảo. Tôi không sao chép công trình nghiên cứu của cá nhân khác dưới bất kỳ hình thức nào. Nếu có tôi xin hoàn toàn chịu trách nhiệm.

Hà Nội, ngày tháng năm 2025  
Học viên

Trần Hữu Nam

## **LỜI CẢM ƠN**

Lời đầu tiên, tôi xin gửi lời cảm ơn đến thầy PGS.TS. Mai Anh Tuấn và thầy ThS. Trần Ngọc Thái vì đã tận tình hướng dẫn, truyền đạt kiến thức cho tôi trong suốt quá trình học tập và thực hiện luận văn. Tôi xin cảm ơn tập thể thầy, cô khoa “Điện tử Viễn thông”, Trường Đại học Công nghệ - ĐHQGHN, đã giảng dạy tôi trong quá trình tôi học tập tại trường. Tôi cũng xin cảm ơn các anh chị ở Bộ môn Công nghệ Vi cơ Điện tử và Kỹ thuật Máy tính đã tạo điều kiện giúp đỡ, chỉ bảo tôi trong thời gian làm luận văn. Cuối cùng, tôi xin cảm ơn bố mẹ, gia đình cũng như bạn bè, tập thể lớp K30 đã luôn đồng hành, chia sẻ và động viên tôi suốt thời gian qua.

## TÓM TẮT

### *Tóm tắt:*

Hội chứng ngưng thở khi ngủ do tắc nghẽn (Obstructive Sleep Apnea - OSA) được đặc trưng bởi sự tắc nghẽn một phần hoặc hoàn toàn của đường hô hấp trên kéo dài ít nhất 10 giây, trong khi các nỗ lực hô hấp vẫn tiếp tục thông qua cử động của lồng ngực và bụng [1]. Trong nghiên cứu [2], tác giả chỉ ra có gần 1 tỷ người mắc OSA và theo GS.BS. Dương Quý Sỹ, Việt Nam có tới 8.5% người trưởng thành mắc OSA [3]. OSA có ảnh hưởng lớn tới sức khỏe cả thể chất lẫn tinh thần đối với người mắc. Xia Wang trong nghiên cứu của mình có tới 25.760 người tham gia đã kết luận rằng: mỗi khi chỉ số ngưng-giảm thở (Apnea-Hypopnea Index - AHI) tăng thêm 10 đơn vị, nguy cơ mắc bệnh tim mạch sẽ tăng 17% [4]. Như trong nghiên cứu[5], nhóm tác giả đã phân tích dữ liệu từ 15 nghiên cứu với tổng cộng 42.924 người và kết quả cho thấy tỷ lệ mắc chứng buồn ngủ ban ngày quá mức (Excessive Daytime Sleepiness - EDS) ở bệnh nhân ngưng thở tắc nghẽn khi ngủ trên toàn cầu là 39,9%.

Hội chứng ngưng thở khi ngủ do tắc nghẽn chủ yếu do béo phì, đặc điểm hình thái vùng hàm-mặt và bất thường cấu trúc đường hô hấp trên gây hẹp đường thở khi ngủ theo nghiên cứu [6]. Tác giả của nghiên cứu cũng đưa ra thêm các yếu tố nguy cơ bổ sung gồm hút thuốc, uống rượu, nghẹt mũi ban đêm, thay đổi nội tiết tố sau mãn kinh, và di truyền, góp phần làm tăng khả năng xuất hiện và mức độ nặng của OSA. Bên cạnh đó, tư thế ngủ được cho rằng đóng vai trò quan trọng trong việc khởi phát và làm trầm trọng thêm các triệu chứng của OSA [7]. Hội chứng ngưng thở khi ngủ do tắc nghẽn tư thế (POSA) được tác giả Cartwright mô tả lần đầu tiên vào năm 1984, ở những người có các đợt ngưng thở và giảm thở xảy ra chủ yếu khi nằm ngửa [8].

Để đánh giá chính xác tình trạng OSA, đa ký giấc ngủ (Polysomnography - PSG) được sử dụng và được xem tiêu chuẩn vàng trong đánh giá giấc ngủ. Đây là phương pháp ghi đồng thời nhiều tín hiệu sinh lý gồm EEG, EOG, EMG, ECG, nồng độ oxy trong máu và các thông số có liên quan khác [9]. Trong đó, tư thế ngủ luôn là một kênh dữ liệu quan trọng để bác sĩ đưa ra kết luận về tình trạng OSA và phương pháp điều trị hợp lý. Tuy nhiên, theo tìm hiểu của tác giả, tại Việt Nam,

chi phí cho một lần thực hiện đa ký giác ngủ còn khá cao, do người bệnh phải lưu trú qua đêm tại phòng đo đạt chuẩn và được theo dõi trực tiếp bởi kỹ thuật viên. Bên cạnh đó, việc phải gắn nhiều điện cực trong khi ngủ gây nhiều bất tiện cho người bệnh, đồng thời có thể làm sai lệch hoặc gián đoạn tín hiệu ghi nhận trong quá trình đo.

Trong bối cảnh công nghệ thiết kế - chế tạo phát triển, việc thu nhỏ kích thước và nâng cao suất của vi điều khiển, cảm biến, dung lượng pin ngày càng trở nên mạnh mẽ. Kết hợp với sự phát triển vượt bậc của trí tuệ nhân tạo góp phần nâng cao hiệu quả từ khâu khai thác dữ liệu thu thập từ cảm biến đến phân tích dữ liệu và thâm trí đưa ra những hành động mà không cần sự can thiệp của con người. Đặc biệt, học máy tại biên (Edge machine learning) nổi lên để giải quyết các bài toán cần đưa ra quyết định nhanh, nhưng đòi hỏi vẫn giữ độ chính xác cao và triển khai trên các thiết bị phần cứng có tài nguyên hạn chế đi kèm mức tiêu thụ năng lượng thấp.

Trên cơ sở đó, tác giả quyết định lựa chọn đề tài là nghiên cứu, phát triển mô hình học máy tại biên nhằm phân loại tư thế ngủ với ba mục tiêu chính bao gồm: 1) Nghiên cứu, đề xuất hệ thống phần cứng phục vụ đo lường, thu thập, xử lý tín hiệu gia tốc kèm với hiệu năng phù hợp cho việc triển khai mô hình học máy tại biên trong bài toán phân loại tư thế ngủ ở người; 2) Nghiên cứu, đề xuất những mô hình học máy thích hợp trong việc phân loại tư thế ngủ ở người dựa trên những đặc trưng của tín hiệu gia tốc được trích xuất từ bộ dữ liệu cảm biến do nhóm nghiên cứu thu thập. 3) Chuẩn hóa, thực thi, đánh giá mô hình học máy trên vi điều khiển Nordic nRF52840, nhằm kiểm chứng hoạt động thực tế của hệ thống nhận dạng tư thế ngủ tại biên

**Từ khóa:** cảm biến gia tốc, học máy, ngưng thở tắc nghẽn khi ngủ, Tiny ML

# Mục lục

<b>Lời cam đoan</b>	i
<b>Lời cảm ơn</b>	ii
<b>Tóm tắt</b>	iii
<b>Danh sách hình vẽ</b>	ix
<b>Danh sách bảng</b>	x
<b>Danh mục ký hiệu, viết tắt</b>	xii
<b>Đặt vấn đề</b>	1
<b>Chương 1 TỔNG QUAN OSA, MỐI LIÊN HỆ VỚI TƯ THẾ NGỦ VÀ KĨ THUẬT SÀNG LỌC, PHÂN LOẠI</b>	4
1.1. Hội chứng ngưng thở khi ngủ . . . . .	4
1.1.1. Định nghĩa . . . . .	5
1.1.2. Nguyên nhân . . . . .	6
1.1.3. Ảnh hưởng của OSA . . . . .	7
1.1.4. Tư thế ngủ và liên hệ với OSA . . . . .	8
1.2. Công nghệ trong chẩn đoán OSA và phân loại tư thế ngủ . . . . .	10
1.2.1. Đa ký giấc ngủ . . . . .	10
1.2.2. Thiết bị theo dõi OSA tại nhà . . . . .	11
1.2.3. Kĩ thuật phân loại tư thế ngủ . . . . .	14
1.2.4. Cảm biến gia tốc trong đánh giá tư thế ngủ . . . . .	17

1.2.5. Học máy trong phân loại OSA và tư thế ngủ . . . . .	25
<b>Chương 2 HỆ THỐNG THU THẬP, XỬ LÝ TÍN HIỆU CẢM BIẾN VÀ ĐÁNH GIÁ BẰNG MÔ HÌNH HỌC MÁY</b>	<b>33</b>
2.1. Nghiên cứu, phát triển phần cứng . . . . .	33
2.1.1. Vị xử lý . . . . .	33
2.1.2. Cảm biến . . . . .	36
2.1.3. Bluetooth năng lượng thấp . . . . .	39
2.1.4. Thiết kế mạch . . . . .	43
2.2. Hệ thống phần mềm . . . . .	51
2.2.1. Lập trình phần cứng . . . . .	51
2.2.2. Phần mềm thu thập, lưu trữ . . . . .	58
2.2.3. Học máy trong phân loại tư thế ngủ . . . . .	70
<b>Chương 3 KẾT QUẢ VÀ ĐÁNH GIÁ</b>	<b>78</b>
3.1. Hệ thống thu thập, lưu trữ . . . . .	78
3.2. Thu thập và gắn nhãn dữ liệu . . . . .	81
3.3. Phân loại tư thế ngủ bằng học máy . . . . .	84
3.3.1. Phân tích dữ liệu . . . . .	84
3.3.2. Xử lý và trích xuất đặc trưng . . . . .	86
3.3.3. Lựa chọn tham số . . . . .	90
3.3.4. Kịch bản kiểm thử . . . . .	94
3.3.5. Đánh giá kết quả . . . . .	96
3.4. Triển khai mô hình lên phần cứng . . . . .	99
3.5. Chuyển đổi mô hình . . . . .	100
3.6. Triển khai tại biên . . . . .	101
3.6.1. Đánh giá hiệu suất và tài nguyên . . . . .	102

# Danh sách hình vẽ

1.1	Sự khác nhau về đường thở của người bình thường và người mắc OSA . . . . .	5
1.2	Các tư thế ngủ cơ bản của con người . . . . .	8
1.3	Xếp đường thở ở tư thế ngửa . . . . .	9
1.4	Màn hình tổng hợp tín hiệu của PSG . . . . .	11
1.5	Thiết bị Sleep Care Kit . . . . .	13
1.6	Thiết thống nệm áp suất trong nghiên cứu của Xi Xu và cộng sự . . . . .	14
1.7	Vị trí đặt cảm biến được đề xuất trong nghiên cứu [22] . . . . .	16
1.8	Vị trí đặt cảm biến gia tốc . . . . .	17
1.9	(a) Sơ đồ bố trí tổng thể của cảm biến gia tốc áp điện trở (b) Cấu trúc mặt cắt qua khối gia trọng và cấu trúc dầm (c) Cầu Wheatstone đo gia tốc trực X (d) Cầu Wheatstone đo gia tốc trực Y (e) Cầu Wheatstone đo gia tốc trực Z. [75] . . . . .	19
1.10	Cấu trúc cơ bản của cảm biến MEMS áp điện: (a) mặt trên (b) mặt dưới [76] . . . . .	20
1.11	Sơ đồ thiết kế cơ bản của cảm biến MEMS điện dung . . . . .	22
1.12	Quá trình biến đổi tín hiệu trong cảm biến điện dung. . . . .	22
2.1	Nordic NRF52840 . . . . .	34
2.2	Các tính năng Nordic nRF52840 . . . . .	35
2.4	Một số bộ kit phát triển dựa trên vi điều khiển Nordic nRF52840 . . . . .	36
2.3	Sơ đồ khối Nordic NRF52840 . . . . .	37

2.5	Bosch BMI270 . . . . .	38
2.6	Sơ đồ khối Bosch BMI270 . . . . .	39
2.7	Sơ đồ chân Bosch BMI270 . . . . .	40
2.8	Các kiểu kết nối không dây trong họ chip nRF52 . . . . .	41
2.9	Cấu trúc của GATT . . . . .	42
2.10	U-blox NINA-B306 . . . . .	44
2.11	Sơ đồ khối U-blox NINA-B306 . . . . .	45
2.12	Sơ đồ chân U-blox NINA-B306 . . . . .	46
2.13	Sơ đồ khối IC MP2322GQH, dải đầu vào 3 V - 22 V, dòng tải tối đa 1 A, hiệu suất cao, tần số chuyển mạch 1.25 MHz . . . . .	47
2.14	Sơ đồ mạch . . . . .	49
2.15	Bố trí mạch in hai mặt của bo mạch . . . . .	50
2.16	Mô hình mạch 3D . . . . .	50
2.17	Sơ đồ tóm tắt quy trình hoạt động BLE-BMI270 của đoạn mã nguồn 2.2	56
2.18	So sánh tín hiệu trực Z trước và sau lọc Kalman . . . . .	57
2.19	Kiến trúc tổng thể của hệ thống . . . . .	59
2.20	Cấu trúc kiến trúc BLoC trong ứng dụng Flutter . . . . .	61
2.21	Sơ đồ kết nối và nhận dữ liệu của ứng dụng di động . . . . .	62
2.22	Cấu trúc cây thư mục mã nguồn của máy chủ trung tâm . . . . .	68
2.23	Lưu đồ thuật toán lưu trữ dữ liệu cảm biến . . . . .	69
2.24	Phân bổ khối lượng công việc với bài toán học máy phân loại tư thế ngủ trong luận văn . . . . .	70
2.25	Minh họa mặt siêu phẳng phân tách hai lớp trong SVM . . . . .	74
2.26	Chiến lược phân loại đa lớp bằng phương pháp One-vs-One . . . . .	74
2.27	Chiến lược phân loại đa lớp bằng phương pháp One-vs-Rest . . . . .	75
3.1	Mạch hoàn chỉnh . . . . .	79

3.2	Giao diện chức năng chatbot và bộ câu hỏi tâm soá . . . . .	79
3.3	Giao diện chức năng đăng ký và đăng nhập . . . . .	80
3.4	Màn hình yêu cầu quyền truy cập Bluetooth và danh sách thiết bị đang hoạt động . . . . .	81
3.5	Màn hình hiển thị dữ liệu gia tốc ba trục với hoàng là giá trị thời gian, trục tung là giá trị cảm biến gấp lên 10 lần . . . . .	82
3.6	Dữ liệu cảm biến gia tốc được lưu trong cơ sở dữ liệu . . . . .	82
3.7	Mô phỏng thực nghiệm thực tế . . . . .	83
3.8	Biểu đồ phân bố các nhãn thu được . . . . .	84
3.9	Biểu đồ phân bố dữ liệu 3 trục theo các tư thế ngủ . . . . .	85
3.10	Phân bố giá trị trục x của 2 tập dữ liệu . . . . .	86
3.11	Phân bố của đặc trưng trung bình, độ lệch chuẩn và năng lượng trên miền thời gian của tập huấn luyện . . . . .	89
3.12	Phân bố của đặc trưng trung bình, độ lệch chuẩn trên miền tần số của tập huấn luyện . . . . .	89
3.13	Ma trận nhầm lẫn của các mô hình học máy . . . . .	92
3.14	Ý nghĩa của các đặc trưng lên mô hình . . . . .	93
3.15	Ma trận tương quan giữa các đặc trưng trích xuất. . . . .	94
3.16	Kết quả độ chính xác của 8 kịch bản . . . . .	96
3.17	Ma trận nhầm lẫn trong kịch bản 8 . . . . .	97
3.18	Kết quả kết thước mô hình của 8 kịch bản (đơn vị: kb) . . . . .	98
3.19	Mức độ ảnh hưởng của số lượng mẫu đến độ chính xác mô hình LR . .	99
3.20	Quy trình đánh giá mô hình trên MCU và đối chiếu kết quả trên PC. . .	104
3.21	Ma trận nhầm lẫn của 2 mô hình triển khai trên biên . . . . .	106
3.22	Biểu đồ mức độ ảnh hưởng của các đặc trưng lên mô hình với bài toán nằm hay không nằm . . . . .	107
3.23	Ma trận nhầm lẫn phân biệt 2 nhãn nằm và không nằm . . . . .	108

# Danh sách bảng

1.1	Phân loại mức độ OSA dựa trên chỉ số AHI . . . . .	6
1.2	Phân loại thiết bị đeo trong phát hiện OSA. tư thế ngủ . . . . .	12
1.3	So sánh đặc tính giữa ba loại cảm biến gia tốc MEMS . . . . .	24
1.4	Các bước chính trong bài toán sử dụng học máy trên biên . . . . .	26
1.5	Hiệu quả mô hình học máy và khả năng triển khai biên trong nhận diện tư thế ngủ . . . . .	31
2.1	Các nhóm chức năng chính của ứng dụng . . . . .	60
2.2	Mô tả các thư mục chính trong cấu trúc mã nguồn máy chủ trung tâm .	67
3.1	Các đặc trưng thống kê và tín hiệu được sử dụng trong phân loại tư thế ngủ	88
3.2	Kết quả lựa chọn tham số cho các mô hình học máy . . . . .	90
3.3	Các kịch bản lựa chọn và sử dụng đặc trưng trong nghiên cứu . . . . .	95
3.4	So sánh hiệu năng LR và ANN trên nRF52840 . . . . .	103
3.5	So sánh kết quả mô hình trên tập thử Python và khi triển khai trên MCU	105

# Danh mục ký hiệu, viết tắt

AHI	Apnea-Hypopnea Index - Chỉ số ngưng thở giảm thở
ANN	Artificial Neural Network - Mạng nơ-ron nhân tạo
BLE	Bluetooth Low Energy - Bluetooth năng lượng thấp
DT	Decision Tree - Cây quyết định
ET	Extra Trees - Rừng cực kì ngẫu nhiên
GATT	Generic Attribute Profile - Đặc điểm cấu hình chung
GB	Gradient Boosting - Thuật toán tăng cường
IOT	Internet of Things - Internet vạn vật
LR	Logistic Regression - Hồi quy Logistic
LSTM	Long Short-Term Memory - Bộ nhớ dài ngắn hạn
MEMS	Micro-Electro-Mechanical System - Hệ thống vi cơ điện tử

OSA	Obstructive Sleep Apnea - Hội chứng ngưng thở khi ngủ do tắc nghẽn
pOSA	Position Obstructive Sleep Apnea - Hội chứng ngưng thở khi ngủ do tắc nghẽn do tư thế
PSG	Polysomnography - Đa ký giấc ngủ
RF	Random Forest - Rừng ngẫu nhiên
SVM	Support Vector Machine - Máy véc-tơ hỗ trợ

# Đặt vấn đề

Tư thế ngủ nắm vai trò đặc biệt quan trọng trong việc duy trì sức khỏe tổng thể và chất lượng giấc ngủ, bên cạnh các yếu tố khác như thời lượng, môi trường và thói quen ngủ. Nhiều nghiên cứu đã chỉ ra rằng tư thế nằm có thể ảnh hưởng trực tiếp đến hoạt động của hệ hô hấp, tim mạch và hệ cơ - xương, đặc biệt là cột sống [10]. Trong hội chứng ngưng thở tắc nghẽn khi ngủ, tư thế ngủ được xem là một yếu tố quyết định mức độ nghiêm trọng của bệnh [11]. Vì vậy, phân loại đúng tư thế ngủ giúp góp phần đánh giá đúng đắn, tình trạng OSA và đưa ra những liệu trình điều trị phù hợp.

Các nghiên cứu hiện nay đã đang tập trung phát triển các hệ thống theo dõi tư thế ngủ dựa trên đa dạng công nghệ cảm biến. Các phương pháp này trải rộng giữa dạng tiếp xúc và không tiếp xúc, cụ thể gồm: tấm dẫn điện với các điện cực ECG đặt tại giường thu thập tín hiệu nhịp tim để phân loại tư thế ngủ [12]; cảm biến áp suất dưới nệm đo phân bố lực tiếp xúc toàn thân [13, 14]; camera hồng ngoại ghi hình ban đêm không cần chiếu sáng, nhận dạng tư thế bằng thị giác máy [15, 16].

Các thiết bị đeo sử dụng tín hiệu đo lường quán tính (Inertial Measurement Unit - IMU) được đặt tại nhiều vị trí khác nhau trên cơ thể nổi lên như cách tiếp cận nhỏ gọn, thuận tiện. Trong nghiên cứu [17], tác giả đã đề xuất hệ thống iSleepPost sử dụng cảm biến gia tốc đeo ở cổ tay có độ chính xác lên tới 85% trong việc nhận dạng các tư thế ngủ bằng các phương pháp học máy. Một thiết bị đeo khác có tên MORFEA được gắn phía trên mũi [18]. Thiết bị này tích hợp cảm biến gia tốc ba trục để phân tích chuyển động, trong đó tư thế ngủ được ước lượng dựa trên góc xoay và độ nghiêng của đầu và tham chiếu tới bảng điều kiện định sẵn. Tuy nhiên, tư thế nằm sắp không được xem xét do vị trí gắn thiết bị trên mũi. Một nghiên cứu gần đây giới thiệu hệ thống NAPPA - tã thông minh được thiết kế để theo dõi nhịp thở và tư thế của trẻ sơ sinh trong khi ngủ [19]. Thiết bị tích hợp cảm biến gia tốc và con quay hồi chuyển. Tuy nhiên, trong nghiên

cứu này, phương pháp và kết quả phát hiện tư thế ngủ chi tiết không được công bố

Các nghiên cứu sử dụng duy nhất một cảm biến gia tốc trong việc phân loại tư thế ngủ cũng cho thấy tiềm năng rất lớn. Nghiên cứu sử dụng cảm biến gia tốc dạng miếng dán cho thấy độ chính xác tổng thể đạt 99,16% khi cảm biến được gắn ở bên trái ngực, với các điều kiện định sẵn theo ba trục X, Y, Z [20]. Tuy nhiên, không có tư thế nằm sấp được ghi nhận trong nghiên cứu này. Nhóm tác giả Vũ Hoàng Diệu đã nghiên cứu phát triển thiết bị đeo sử dụng một cảm biến gia tốc duy nhất, áp dụng mô hình học sâu AnpoNet (1D-CNN + LSTM) để phân loại 12 tư thế ngủ với độ chính xác 94,67%, hỗ trợ theo dõi giấc ngủ tại nhà cho bệnh nhân trào ngược dạ dày [21]. Đối với nghiên cứu [22], tác giả sử dụng một cảm biến gia tốc đặt ở cổ, đánh giá qua ba mô hình (DT, ET, LSTM) dựa trên tần số lấy mẫu 5Hz và cửa sổ 1 giây, đạt độ chính xác > 98% (F1-score trung bình 0,945-0,975), trong đó DT có hiệu năng và mức tiêu thụ bộ nhớ tối ưu nhất. Nhóm tác giả trong nghiên cứu [23] phát triển ứng dụng SleepPos với độ chính xác tổng thể 98,2%, nhưng độ nhạy với tư thế nằm sấp chỉ đạt 38,9%.Thêm vào đó, thiết kế phần cứng còn khá lớn làm giảm tính thoải mái. Nghiên cứu trên 89 bệnh nhân sử dụng cảm biến gắn tại hõm ức tích hợp gia tốc kế 3 trục cho thấy độ chính xác 96.9-98.6% trong phát hiện tư thế ngủ, khẳng định hõm ức là vị trí tối ưu cho cảm biến đeo cổ [24]. Các kết quả này cho thấy hướng tiếp cận đơn cảm biến hoàn toàn có thể thay thế cho hệ thống đa cảm biến trong bài toán phân loại tư thế ngủ.

Tuy nhiên, các công trình hiện có chưa xem xét đầy đủ việc phân biệt trạng thái “đã nằm” hay “chưa nằm” - một yếu tố nền tảng để xác định chính xác giai đoạn bắt đầu giấc ngủ hoặc phát hiện sớm thay đổi tư thế. Đồng thời, việc lựa chọn tập đặc trưng và mô hình học máy phù hợp cho dữ liệu từ một cảm biến gia tốc duy nhất đặt tại vùng cổ - vị trí có khả năng phản ánh biến thiên tư thế tốt nhất vẫn chưa được nghiên cứu cụ thể. Tại Việt Nam cho đến nay chưa có công bố khoa học nào tập trung vào bài toán nhận dạng tư thế ngủ dựa hoàn toàn trên dữ liệu cảm biến gia tốc và được triển khai mô hình học máy trực tiếp trên phần cứng hạn chế tài nguyên. Khoảng trống này đặt ra yêu cầu cần thiết cho việc phát triển một hệ thống giám sát tư thế ngủ đơn giản, chi phí thấp, có khả năng xử lý cục bộ, hướng tới ứng dụng trong sàng lọc và hỗ trợ chẩn đoán sớm hội chứng ngưng thở tắc nghẽn khi ngủ.

Vì vậy, luận văn này tập trung làm rõ các đặc trưng của dữ liệu cảm biến gia tốc khi được đặt tại vùng cổ, đồng thời đi sâu tìm hiểu các đặc trưng dữ liệu cảm biến gia tốc và

đánh giá các mô hình học máy phù hợp nhằm tối ưu cho việc triển khai trên các thiết bị tính toán biên. Để đạt được điều đó, luận văn sẽ xây dựng một hệ thống hoàn chỉnh gồm phần cứng đeo được kích thước nhỏ gọn, chi phí thấp; phần mềm thu thập; máy chủ lưu trữ dữ liệu; cùng với mô-đun xử lý và phân loại tư thế ngủ tích hợp trực tiếp trên vi điều khiển. Mặc dù chưa tích hợp chức năng sàng lọc hội chứng ngưng thở tắc nghẽn khi ngủ, kết quả nghiên cứu này tạo nền tảng kỹ thuật tiềm năng cho các ứng dụng trong tương lai, đặc biệt trong việc hỗ trợ đánh giá nguy cơ tư thế của OSA.

Mục tiêu cụ thể của luận văn gồm: 01) Nghiên cứu, đề xuất hệ thống phần cứng phục vụ đo lường, thu thập, xử lý tín hiệu gia tốc kèm với hiệu năng phù hợp cho việc triển khai mô hình học máy tại biên trong bài toán phân loại tư thế ngủ ở người; 02) Nghiên cứu, đề xuất những mô hình học máy thích hợp trong việc phân loại tư thế ngủ ở người dựa trên những đặc trưng của tín hiệu gia tốc được trích xuất từ bộ dữ liệu cảm biến do nhóm nghiên cứu thu thập; 03) Chuẩn hóa, thực thi, đánh giá mô hình học máy trên vi điều khiển Nordic nRF52840, nhằm kiểm chứng hoạt động thực tế của hệ thống nhận dạng tư thế ngủ tại biên;

Cấu trúc luận văn được trình bày trong bốn chương chính như sau:

**Chương 1:** Tổng quan OSA, mối liên hệ với tư thế ngủ và kĩ thuật sàng lọc, phân loại.

**Chương 2:** Hệ thống thu thập, xử lý tín hiệu cảm biến và đánh giá bằng mô hình học máy.

**Chương 3:** Kết quả và đánh giá.

# **Chương 1**

# **TỔNG QUAN OSA, MỐI LIÊN HỆ VỚI TƯ THẾ NGỦ VÀ KĨ THUẬT SÀNG LỌC, PHÂN LOẠI**

Trong chương này, tác giả trình bày tổng quan về hội chứng ngưng thở khi ngủ do tắc nghẽn, phân tích mối quan hệ của tư thế ngủ đối với mức độ nghiêm trọng của OSA và dạng ngưng thở khi ngủ phụ thuộc tư thế (Position OSA - pOSA). Tiếp đó, chương tập trung làm rõ cơ sở khoa học cho việc phát triển các thiết bị theo dõi giấc ngủ, phân loại tư thế ngủ tại nhà. Cuối cùng, tác giả đưa ra xu hướng ứng dụng trí tuệ nhân tạo (AI) và điện toán biên (Edge Computing) trong thu thập, phân tích dữ liệu sinh lý để phân loại tư thế ngủ, qua đó đặt nền tảng cho các hướng nghiên cứu và triển khai kỹ thuật được trình bày trong những chương tiếp theo.

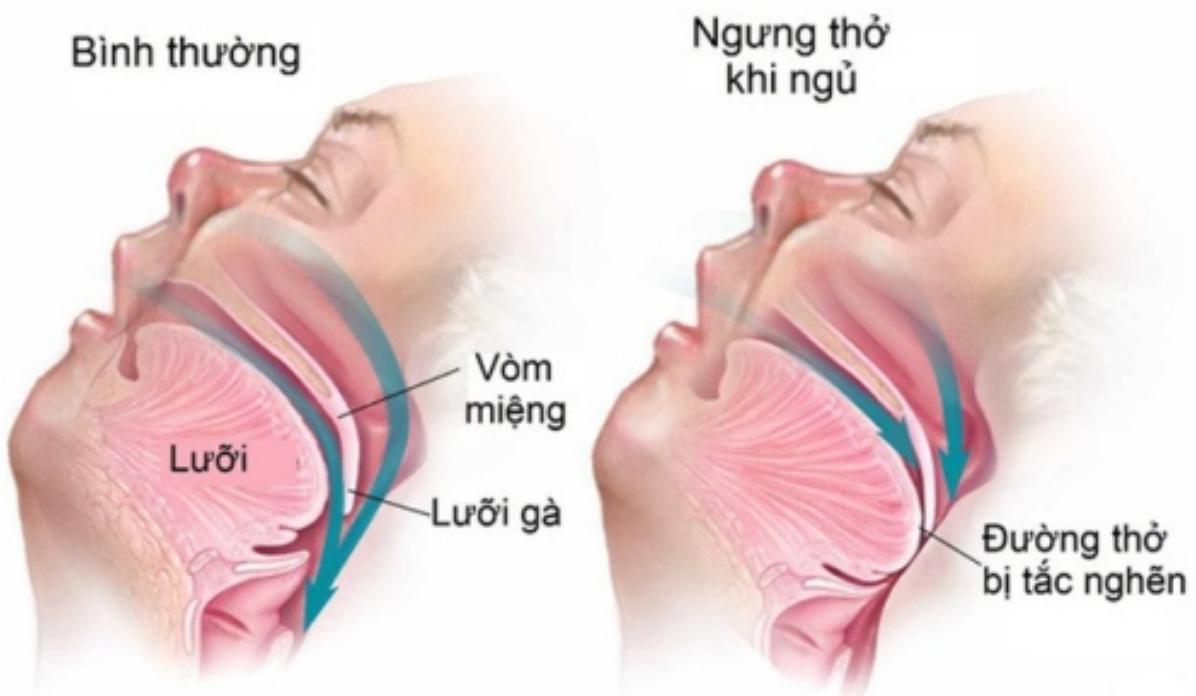
## **1.1. Hội chứng ngưng thở khi ngủ**

Trong lĩnh vực nghiên cứu các rối loạn hô hấp liên quan đến giấc ngủ, theo tiêu chuẩn của Hiệp hội Y học Giấc ngủ Hoa Kỳ (AASM) [25], ba hiện tượng hô hấp chính

cần được nhận diện bao gồm: ngưng thở (apnea), giảm thở (hypopnea), và hiện tượng kích hoạt liên quan đến nỗ lực hô hấp (Respiratory Effort-Related Arousal - RERA).

### 1.1.1. Định nghĩa

Ngưng thở được AASM định nghĩa là sự ngưng luồng khí hô hấp qua mũi và miệng trong thời gian tối thiểu 10 giây gây giảm nồng độ oxy trong máu. Các sự kiện ngưng thở có thể kéo dài đến 30 giây hoặc hơn trong những trường hợp nặng. Có ba dạng chính của hội chứng ngưng thở khi ngủ [26]: ngưng thở tắc nghẽn, ngưng thở trung ương, ngưng thở hỗn hợp. Trong đó: 01) Ngưng thở khi ngủ do tắc nghẽn OSA là sự hẹp hoặc tắc nghẽn một phần hay toàn bộ đường hô hấp trên, bao gồm vùng mũi họng, hầu họng [27]; 02) Ngưng thở khi ngủ do trung ương (Central Sleep Apnea - CSA) là tình trạng não không gửi tín hiệu đúng đắn đến các cơ kiểm soát hô hấp [28]. 03) Ngưng thở hỗn hợp (Mixed Apnea) là sự kết hợp của cả hai yếu tố. Dạng này thường xuất hiện ở những bệnh nhân OSA nặng.



Hình 1.1: Sự khác nhau về đường thở của người bình thường và người mắc OSA

Giảm thở phản ánh sự giảm một phần của lưu lượng khí qua đường hô hấp trên mà

không dẫn đến ngừng thở hoàn toàn. Theo AASM sự kiện giảm thở được xác định khi biên độ tín hiệu luồng khí thường đo bằng cảm biến áp lực mũi hoặc tín hiệu lưu lượng khí của thiết bị CPAP giảm ít nhất 30% so với giá trị nền trước sự kiện, kéo dài tối thiểu 10 giây, và đi kèm với hiện tượng giảm độ bão hòa oxy từ 3% trở lên hoặc xuất hiện kích hoạt điện não.

RERA theo AASM là sự kiện gia tăng nỗ lực hô hấp kéo dài ít nhất 10 giây, gây đánh thức khỏi giấc ngủ nhưng không đủ tiêu chí của ngưng thở hoặc giảm thở. 01) Phương pháp tiêu chuẩn để đo là đo áp lực thực quản, tuy nhiên khó áp dụng do gây khó chịu cho bệnh nhân. 02) Phương án thay thế đáng tin cậy là dùng ống thông mũi kết hợp cảm biến áp lực, cho kết quả tương đương về mặt lâm sàng. 03) RERA được tính vào chỉ số rối loạn hô hấp (Respiratory Disturbance Index - RDI).

Trong số các rối loạn hô hấp liên quan đến giấc ngủ đã đề cập, hội chứng ngưng thở khi ngủ do tắc nghẽn là dạng phổ biến nhất và có tác động sâu rộng đến sức khỏe cộng đồng. Mức độ của OSA được đánh giá dựa trên chỉ số ngưng thở giảm thở (Apnea-Hypopnea Index - AHI) bằng cách chia tổng số lần ngưng thở và giảm thở cho tổng số giờ đã ngủ, với mỗi sự kiện phải kéo dài ít nhất 10 giây Bảng 1.1 [27].

Bảng 1.1: Phân loại mức độ OSA dựa trên chỉ số AHI

AHI	Cấp độ
<5	Không mắc
5 đến 15	Nhẹ
15 đến 30	Trung bình
>30	Nặng

### 1.1.2. Nguyên nhân

Nguyên nhân của OSA đến từ đa yếu tố kết hợp giữa giải phẫu và các chức năng sinh lý của đường hô hấp trên. Các nguyên nhân thường không đồng nhất giữa các người bệnh. Về mặt giải phẫu, việc hẹp hoặc giảm độ vững của cấu trúc đường hô hấp trên là yếu tố tiên quyết. Những bất thường tại vòng họng như lưỡi to, phì đại amidan, vách mũi lệch, xương hàm nhỏ hoặc tụt về sau, cùng với phân bố mỡ vùng cổ tăng (vòng cổ lớn,

béo phì trung tâm) đều làm giảm tiết diện đường thở và tăng khả năng xẹp trong khi hít vào [6].

Về chức năng thần kinh - cơ, trong giấc ngủ, trương lực cơ duy trì độ mở của đường hô hấp trên có thể suy giảm, làm tăng xu hướng xẹp đường thở. Ngoài ra, các yếu tố nguy cơ được ghi nhận bao gồm: béo phì, nam giới, lớn tuổi, hút thuốc lá, uống rượu, mẫn kinh ở nữ giới, nghẹt mũi mạn tính, và yếu tố di truyền [29, 30, 31].

### 1.1.3. Ảnh hưởng của OSA

Theo nghiên cứu của Benjafield và cộng sự [2], ước tính có gần một tỷ người trên toàn cầu mắc OSA. Tại Việt Nam, theo GS.TS.BS. Dương Quý Sỹ<sup>1</sup>, tỷ lệ mắc OSA ở người trưởng thành chiếm khoảng 8,5% [3]. OSA được xem là một rối loạn hô hấp trong giấc ngủ có ảnh hưởng sâu rộng tới sức khỏe thể chất lẫn tinh thần và là nguyên nhân y khoa hàng đầu gây ra tình trạng buồn ngủ quá mức vào ban ngày (Excessive Daytime Sleepiness - EDS).

Một phân tích gộp được thực hiện bởi Salari và cộng sự [5] trên 15 nghiên cứu với tổng cộng 42.924 đối tượng cho thấy tỷ lệ EDS ở bệnh nhân OSA trên toàn cầu đạt 39,9%. Tình trạng buồn ngủ quá mức này làm gia tăng đáng kể nguy cơ tai nạn giao thông, suy giảm năng suất lao động, và rối loạn chức năng tình dục [32].

Hơn nữa, tình trạng giảm oxy máu liên tục tái diễn trong khi ngủ cùng với sự gián đoạn chu kỳ giấc ngủ được chứng minh có liên quan mật thiết đến nhiều bệnh lý mạn tính như suy tim, bệnh động mạch vành, rối loạn nhịp tim, gan nhiễm mỡ do rối loạn chuyển hóa, và đột quy [33, 34, 35]. Nghiên cứu quy mô lớn của Xia Wang và cộng sự với 25.760 người tham gia cho thấy, khi chỉ số AHI tăng thêm mỗi 10 đơn vị, nguy cơ mắc bệnh tim mạch tăng tương ứng 17% [4]. GS.TS.BS. Dương Quý Sỹ và cộng sự đã khảo sát 524 trẻ em mắc rối loạn tăng động (Attention Deficit Hyperactivity Disorder - ADHD) tại Bệnh viện Nhi Trung ương Việt Nam [36]. Kết quả cho thấy tỷ lệ mắc OSA ở nhóm này là 23.3%, trong đó chủ yếu ở mức độ trung bình đến nặng. Nghiên cứu cũng đồng thời xác định mối tương quan đáng kể giữa mức độ nghiêm trọng của OSA và các triệu chứng mất tập trung, tăng động, rối loạn hành vi, lo âu và trầm cảm.

---

<sup>1</sup>Giáo sư, Tiến sĩ, Bác sĩ Dương Quý Sỹ là Chủ tịch Hội Giấc ngủ Việt Nam, chuyên gia đầu ngành về rối loạn giấc ngủ và hô hấp.

#### **1.1.4. Tư thế ngủ và liên hệ với OSA**

Tư thế ngủ (Hình 1.2) giữ vai trò thiết yếu trong việc duy trì sức khỏe tổng thể và chất lượng giấc ngủ, bên cạnh các yếu tố khác như thời lượng, môi trường và thói quen ngủ. Nhiều nghiên cứu đã chỉ ra rằng tư thế nằm có thể ảnh hưởng trực tiếp đến hoạt động của hệ hô hấp, tim mạch và hệ cơ - xương, đặc biệt là cột sống [10]. Trong hội chứng ngưng thở tắc nghẽn khi ngủ, tư thế ngủ được xem là một yếu tố quan trọng quyết định mức độ nghiêm trọng của bệnh.



Hình 1.2: Các tư thế ngủ cơ bản của con người

Các nghiên cứu cho thấy, nhiều bệnh nhân OSA có tần suất ngưng thở và giảm thở cao hơn rõ rệt khi nằm ngửa so với các tư thế khác. Nguyên nhân chủ yếu là do tác động của trọng lực lên các cấu trúc mô vùng họng gây tăng khả năng bị xẹp đường thở và dẫn đến sự gia tăng của các sự kiện hô hấp bất thường như Hình 1.3. Hiện tượng này được gọi là OSA phụ thuộc tư thế (positional OSA - pOSA) [37, 38].



Hình 1.3: Xẹp đường thở ở tư thế ngửa

Để chẩn đoán pOSA, nhiều tiêu chí khác nhau đã được đề xuất, từ các phương pháp đơn giản đến phức tạp hơn. Định nghĩa kinh điển nhất do Cartwright (1984) đưa ra cho rằng bệnh nhân được xem là mắc pOSA khi chỉ số AHI ở tư thế nằm ngửa cao hơn ít nhất hai lần so với AHI ở tư thế không nằm ngửa [11]. Tiếp đó, Mador (2005) đã mở rộng định nghĩa này bằng cách bổ sung tiêu chí rằng AHI ở tư thế không nằm ngửa phải nhỏ hơn 5 nhằm tăng tính đặc hiệu cho chẩn đoán [39]. Levendowski (2015) lại đề xuất một cách tiếp cận theo tỷ lệ, trong đó pOSA được xác định khi AHI toàn bộ tư thế lớn hơn hoặc bằng 1.5 lần so với AHI ở tư thế không nằm ngửa [40]. Ngoài ra, Frank và cộng sự (2014) đã giới thiệu Amsterdam Positional Obstructive Sleep Apnea Classification (APOC) - một hệ thống phân loại dành cho các bệnh nhân có chỉ số AHI cao [41]. Theo tiêu chí APOC, bệnh nhân được xác định mắc pOSA khi có  $AHI > 5$ , đồng thời tổng thời gian ngủ ở tư thế có AHI nhỏ nhất và tư thế có AHI lớn nhất đều chiếm ít nhất 10% tổng thời gian ngủ. Hệ thống này chia bệnh nhân thành ba nhóm: APOC-I (có thể khỏi hoàn toàn nhờ thay đổi tư thế ngủ), APOC-II (không phụ thuộc tư thế), và APOC-III (phụ thuộc tư thế một phần).

Từ các nghiên cứu trên, có thể thấy việc hiểu rõ mối liên hệ giữa tư thế ngủ và sự ổn định của đường thở trên không chỉ giúp cải thiện khả năng chẩn đoán mà còn mở ra hướng điều trị không xâm lấn đầy tiềm năng - đó là liệu pháp thay đổi tư thế ngủ, nhằm giảm mức độ tắc nghẽn và cải thiện chất lượng giấc ngủ ở bệnh nhân OSA.

## **1.2. Công nghệ trong chẩn đoán OSA và phân loại tư thế ngủ**

Với sự phát triển vượt bậc của công nghệ thiết kế, chế tạo cảm biến - vi xử lý và các thuật toán học máy, việc chẩn đoán hội chứng ngưng thở khi ngủ tắc nghẽn và phân loại tư thế ngủ đã đạt được những bước tiến đáng kể. Các hệ thống hiện nay có xu hướng thu nhỏ kích thước thiết bị, mở rộng khả năng ứng dụng thực tế và giảm đáng kể chi phí triển khai, tạo tiền đề cho việc giám sát giấc ngủ cá nhân hóa ngay tại nhà.

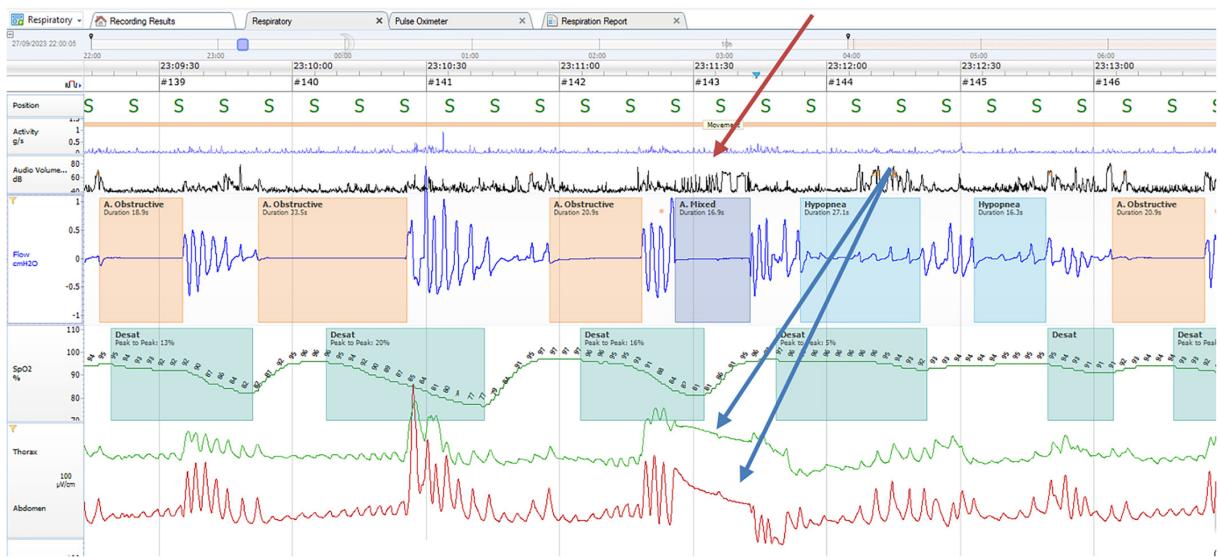
### **1.2.1. Đa ký giấc ngủ**

Phần lớn bệnh nhân OSA không tự nhận biết được các rối loạn hô hấp xảy ra trong lúc ngủ [42]. Hiện tượng này đặc biệt phổ biến ở những người sống hoặc ngủ một mình, do thiếu sự quan sát từ bên ngoài.

Trong điều kiện lâm sàng hiện nay, phần lớn bệnh nhân nghi ngờ mắc OSA được thăm khám bởi bác sĩ chuyên khoa Tai Mũi Họng và bác sĩ chuyên sâu về giấc ngủ. Quy trình thăm khám bao gồm khai thác lịch sử bệnh lý, đánh giá các yếu tố nguy cơ và sử dụng những thang điểm sàng lọc như Epworth Sleepiness Scale hoặc STOP-BANG - các công cụ được chấp thuận tại Việt Nam nhằm sàng lọc sớm khả năng mắc OSA. Ngoài ra, bác sĩ có thể tiến hành nội soi mũi - họng để tìm nguyên nhân gây hẹp đường thở trên, chẳng hạn như phì đại amidan, hay bất thường vùng hàm mặt. Sau đó tiến hành ghi nhận khách quan bằng đa ký giấc ngủ (Polysomnography - PSG) [43, 44]. Phương pháp PSG, khi được thực hiện dưới sự giám sát của kỹ thuật chuyên môn, được xem là tiêu chuẩn vàng trong chẩn đoán hội chứng ngưng thở khi ngủ. Đây là một kỹ thuật ghi đa kênh liên tục trong lúc ngủ suốt một đêm, cho phép theo dõi đồng thời nhiều thông số sinh lý phản ánh toàn diện hoạt động thần kinh và hô hấp trong giấc ngủ.

Một hệ thống PSG điển hình bao gồm: điện não đồ (EEG) để ghi và lưu lại hoạt động điện của não; điện cơ đồ (EMG) nhằm đo trương lực cơ; điện nhãn đồ (EOG) để xác định giai đoạn ngủ thông qua chuyển động mắt; và điện tâm đồ (ECG) để giám sát hoạt động tim. Bên cạnh đó, các cảm biến ghi nhận độ bão hòa oxy trong máu ( $SpO_2$ ), lưu lượng khí thở qua mũi và miệng, nỗ lực hô hấp thông qua chuyển động ngực và bụng

và cường độ tiếng ngáy [45, 46]. Trong số các tín hiệu này, tư thế ngủ là một thông số đặc biệt quan trọng, góp phần xác định dạng bệnh pOSA. Theo dõi liên tục tư thế cơ thể trong quá trình do PSG cho phép xác định mối tương quan giữa tư thế nằm và mức độ nghiêm trọng của rối loạn hô hấp. Các thông số thu được sau đêm ghi PSG được tổng hợp và phân tích, giúp bác sĩ đưa ra chẩn đoán chính xác về tình trạng ngưng thở khi ngủ tắc nghẽn. Hình 1.4 là màn hình phần mềm NoxA1 hiển thị nhiều tín hiệu sinh lý theo dấu thời gian, cho phép quan sát đồng thời các kênh hô hấp. Hệ thống cũng cung cấp công cụ đánh dấu tự động các sự kiện bất thường như ngưng thở và giảm thở.



Hình 1.4: Màn hình tổng hợp tín hiệu của PSG

Tuy nhiên, phương pháp PSG vẫn còn một số hạn chế về mặt chi phí bao gồm đi lại, lưu trú và thăm khám. Việc bệnh nhân bắt buộc gắn nhiều điện cực và cảm biến trên cơ thể không chỉ gây bất tiện mà còn có thể làm sai lệch hành vi ngủ tự nhiên. Điều này còn có thể làm tín hiệu bị gián đoạn trong quá trình thu thập. Theo tìm hiểu của tác giả, trong nhiều trường hợp, bác sĩ có thể đưa ra nhận định dựa trên một tập nhỏ kênh tín hiệu còn nguyên vẹn, từ đó gợi mở hướng ứng dụng các mô hình học máy phản ánh đúng đặc trưng sinh lý nhằm tự động hóa quá trình nhận dạng.

### 1.2.2. Thiết bị theo dõi OSA tại nhà

Những hạn chế về chi phí, sự bất tiện của đa ký giấc ngủ (PSG) trong điều kiện lâm sàng đã thúc đẩy sự phát triển của các thiết bị theo dõi giấc ngủ ngoài trung tâm

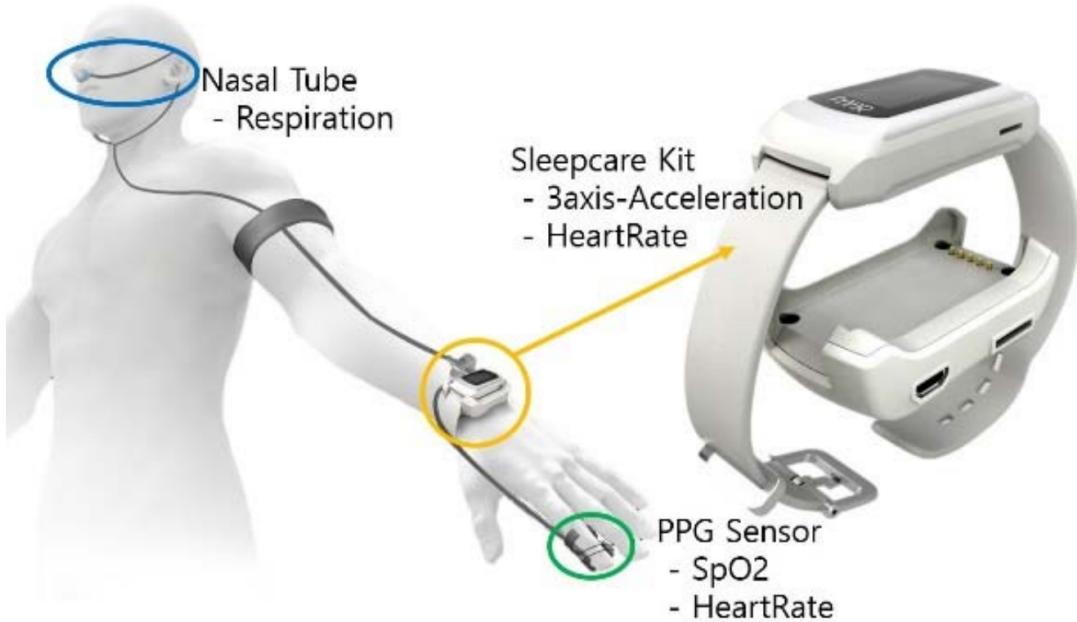
(Out-of-Center Sleep Testing Devices - OCST), hay còn gọi là thiết bị kiểm tra giấc ngủ tại nhà (Home Sleep Test - HST). Các hệ thống HST được thiết kế với mục tiêu giảm kích thước, chi phí phù hợp, thuận tiện sử dụng tại nhà. Để làm được điều đó thì các thiết bị HST cần phải giảm thiểu số lượng cảm biến nhưng vẫn đảm bảo độ chính xác chẩn đoán thông qua việc tích hợp các thuật toán xử lý tín hiệu và mô hình học máy có thể được thực thi trực tiếp trên phần cứng nhúng hoặc thông qua ứng dụng hỗ trợ trên điện thoại di động. Một trong những khung chuẩn được sử dụng rộng rãi trong quá trình phát triển các hệ thống HST là mô hình **SCOPERA**, bao gồm sáu nhóm thông số cốt lõi: *Sleep (S)* - đặc trưng giấc ngủ; *Cardiovascular (C)* - hoạt động tim mạch; *Oximetry (O)* - độ bão hòa oxy; *Effort (E)* - nỗ lực hô hấp; *Respiratory flow (R)* - lưu lượng khí hô hấp; và *Audio (A)* - âm thanh thở.

Dựa trên vị trí gắn trên cơ thể, các thiết bị HST có thể được phân thành các nhóm chính như: vòng tay, đai ngực, miếng dán, tai nghe và nhãn thông minh (Bảng 1.2).

Bảng 1.2: Phân loại thiết bị đeo trong phát hiện OSA. tư thế ngủ

<b>Loại thiết bị đeo</b>	<b>Tài liệu tham khảo</b>
Vòng tay	[47, 48, 49, 50]
Đai ngực	[47, 48, 51, 52]
Miếng dán	[48, 53, 54, 55]
Dạng khác	[56, 57, 58, 48, 59]

Nghiên cứu của Jeon và cộng sự [50] sử dụng thiết bị đeo Sleep Care Kit (Hình 1.5) gắn ở ngực để thu nhận tín hiệu nhịp tim, tín hiệu gia tốc ba trục, tín hiệu hô hấp và SpO<sub>2</sub>. Các mô hình Gaussian Naive Bayes, Artificial Neural Network và k-Nearest Neighbor được huấn luyện nhằm phân loại trạng thái hô hấp và phát hiện ngưng thở, trong đó mô hình KNN đạt độ chính xác 95% và thời gian xử lý chỉ 640 μs - đáp ứng tiêu chí chẩn đoán OSA theo thời gian thực. Tương tự, Chen và cộng sự [52] phát triển thiết bị vòng tay ghi tín hiệu phản hồi ánh sáng để phát hiện biến thiên lưu lượng máu trên 100 đối tượng tình nguyện, tối ưu về bộ nhớ, độ trễ và năng lượng, đồng thời đồng bộ dữ liệu với PSG để đảm bảo độ tin cậy, hướng tới ứng dụng giám sát dài hạn tại nhà.



Hình 1.5: Thiết bị Sleep Care Kit

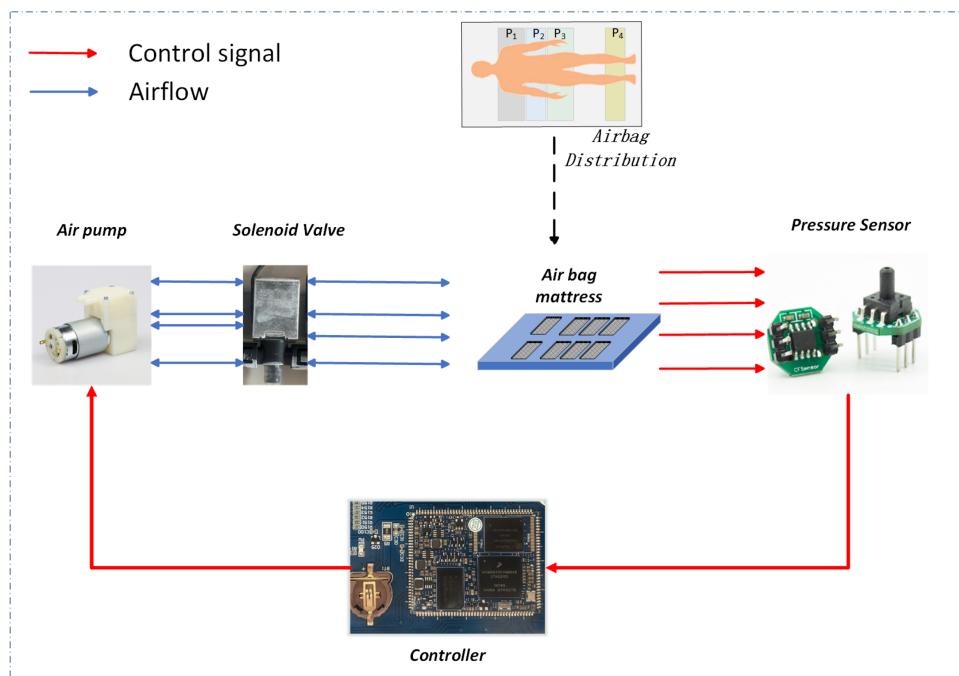
Yeo và cộng sự [53, 54] triển khai thiết bị dán *T-REX TR100A* để ghi điện tâm đồ một kênh tại vùng bụng trên. Thiết bị dán trực tiếp lên da, đảm bảo tiếp xúc ổn định và tín hiệu chính xác, giúp ghi liên tục trong thời gian dài mà không gây khó chịu cho người sử dụng. Bên cạnh đó, nghiên cứu [51] chỉ ra rằng các tín hiệu chuyển động ngực và bụng thu từ dải cảm biến gia tốc áp điện có thể được khai thác hiệu quả trong phân loại rối loạn hô hấp khi ngủ bằng mô hình SVM, đạt độ chính xác trung bình 81.8%.

Ngoài ra, Domingues và cộng sự (2024) đề xuất một mô hình mạng nơ-ron nhân tạo sử dụng dữ liệu từ SpO<sub>2</sub>, cảm biến gia tốc và âm thanh ngáy của hệ thống Biologix, cho kết quả tiệm cận độ chính xác của PSG truyền thống [60]. Hướng tiếp cận khác được Boiko và cộng sự [61] phát triển dựa trên cảm biến gia tốc đặt dưới nệm giường, ghi nhận dao động do chuyển động trong quá trình hô hấp. Hệ thống này đạt độ chính xác 94.6%, độ đặc hiệu 95.3% và độ nhạy 93.7%, khẳng định tiềm năng ứng dụng của cảm biến thụ động trong phát hiện ngưng thở khi ngủ mà không cần thiết bị đeo trực tiếp lên cơ thể.

Nhìn chung, các hệ thống HST thế hệ mới đang chuyển dịch từ kiến trúc đám mây sang kiến trúc tại biên - cho phép suy luận tại biên. Điều này không chỉ nâng cao tính riêng tư và độ tin cậy, mà còn mở đường cho các thiết bị chẩn đoán OSA cá nhân hóa, nhỏ gọn và có khả năng triển khai đại trà trong thực hành y học giấc ngủ hiện đại.

### 1.2.3. Kỹ thuật phân loại tư thế ngủ

Hiện nay, đã có nhiều nghiên cứu tập trung phát triển các hệ thống theo dõi tư thế ngủ phục vụ chẩn đoán và hỗ trợ điều trị OSA. Hầu hết các hướng tiếp cận ban đầu đều sử dụng nhiều loại cảm biến cùng lúc để bổ trợ như cảm biến áp lực, camera RGB/hồng ngoại hoặc hệ thống đeo đa điểm, nhằm ghi nhận toàn bộ chuyển động cơ thể trong khi ngủ. Chẳng hạn, Asma Channa và cộng sự [62] đã phát triển hệ thống theo dõi tư thế ngủ sử dụng hai thảm cảm biến áp lực để thu thập dữ liệu từ 13 người tham gia, với các thuật toán học máy như Weighted KNN và Linear SVM cho độ chính xác tới 98,7%. Tương tự, Xi Xu và cộng sự [63] đề xuất mô hình stacking kết hợp ba thuật toán mạnh (XGBoost, SVM, DNDT) đạt độ chính xác 94,48% trên dữ liệu áp suất từ nệm khí.



Hình 1.6: Thiết bị áp suất trong nghiên cứu của Xi Xu và cộng sự

Các nghiên cứu sử dụng camera RGB hoặc hồng ngoại cũng cho kết quả ấn tượng trong môi trường có kiểm soát. Ví dụ, Khan và cộng sự [64] ứng dụng kết hợp camera RGB và camera nhiệt, huấn luyện mô hình VGG16 và ResNet50, đạt độ chính xác 99%. Mahvash và cộng sự [65] tiếp tục mở rộng hướng nghiên cứu này bằng cách sử dụng camera hồng ngoại 2D, kết hợp học sâu với mô hình ResNet-152, cho độ chính xác 95.1%, f1-score 94.9% vượt trội hơn cảm biến PSG truyền thống. Ngoài ra, một hướng tiếp cận không tiếp xúc khác sử dụng cảm biến radar cũng được đề xuất để nhận dạng

tư thế ngủ. Kết quả cho thấy hệ thống đạt độ chính xác 91% trong nhận dạng tư thế và 83.7% trong phát hiện chuyển tư thế, với ưu điểm nổi bật về khả năng bảo vệ quyền riêng tư và độ ổn định tín hiệu. Nhóm Yunyoung Nam và cộng sự cũng đã tích hợp hệ thống thu thập, phân tích sử dụng một cảm biến gia tốc ba trục và một cảm biến áp suất để giám sát chất lượng giấc ngủ tư thế ngủ, trạng thái ngủ, giai đoạn ngủ (REM và chu kỳ giai đoạn ngủ không REM) [66]. Tại Việt Nam, nhóm nghiên cứu của GS.TS. Lê Tiến Thường<sup>2</sup> đã sử dụng cảm biến gia tốc MPU6050 cùng với vi xử lý ESP32 nhằm ghi nhận hơi thở và nhịp tim của bệnh nhân OSA thông qua rung động, và dòng chảy của động mạch và tĩnh mạch ở cổ [67].

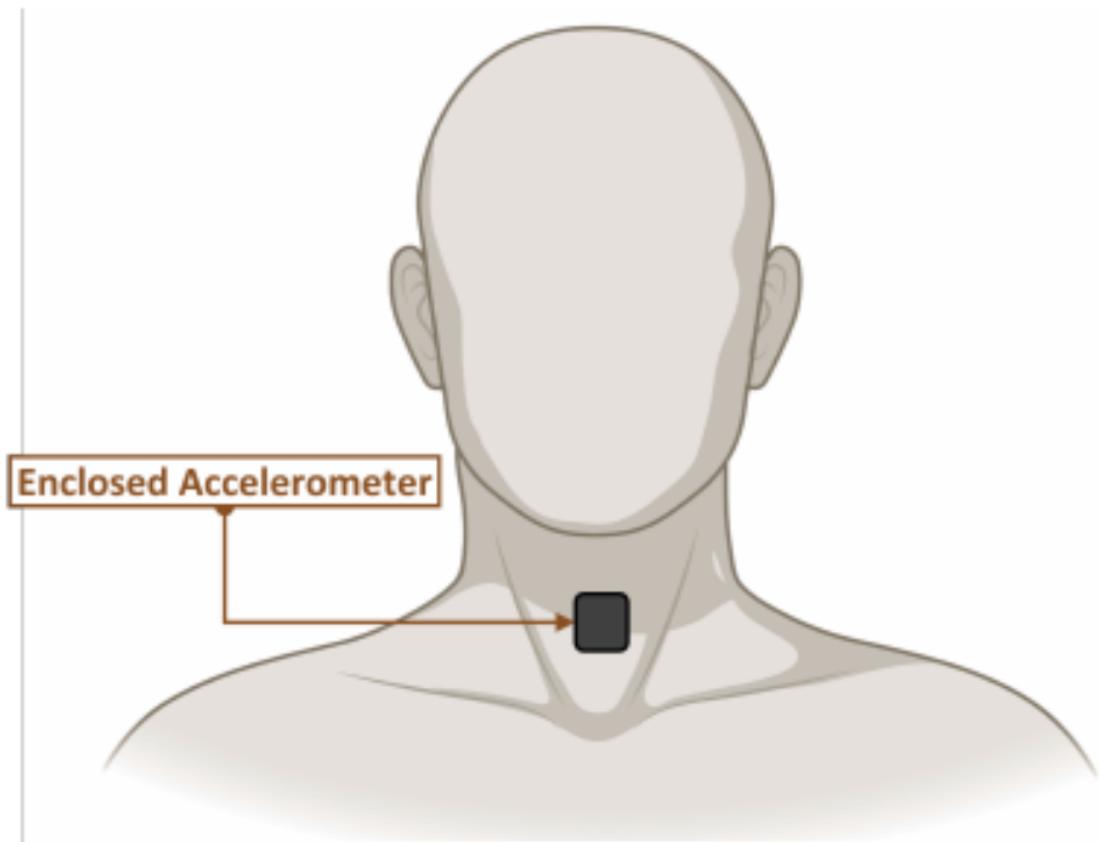
Ngoài ra, việc tận dụng cảm biến gia tốc ở ngay trên chính chiếc điện thoại cũng là giải pháp hữu hiệu [68]. Nhóm tác giả trong [23] đã báo cáo nghiên cứu đánh giá tư thế ngủ của bệnh nhân sử dụng thiết bị di động đeo ở ngực kết hợp với phần mềm trên nền tảng Android để thu thập lại dữ liệu kể cả khi tắt màn hình. Trong một nghiên cứu khác, Natale và cộng sự đã khai thác các cảm biến tích hợp sẵn trên điện thoại iPhone để ước lượng các thông số liên quan đến chất lượng giấc ngủ, bao gồm tổng thời gian ngủ, độ trễ vào giấc và hiệu quả giấc ngủ. Phương pháp tiếp cận này cho thấy tiềm năng trong việc sử dụng thiết bị di động như một công cụ theo dõi giấc ngủ tiện lợi và dễ tiếp cận, đặc biệt trong các nghiên cứu cộng đồng và ứng dụng tại nhà[69]. Đặc điểm của sử dụng tích hợp cảm biến gia tốc trên điện thoại là rất tiện lợi, sử dụng trực tiếp mà không cần phát triển phần cứng. Tuy nhiên, việc tiếp xúc điện thoại trực tiếp với cơ thể trực tiếp trong thời gian lâu cũng có gây những ảnh hưởng nhất định đến người dùng.

Thêm một xu hướng nghiên cứu gần đây nữa là tập trung vào việc tối giản phần cứng, đặc biệt là sử dụng duy nhất một cảm biến gia tốc để nhận dạng tư thế ngủ. Trong hướng này, Jeng và cộng sự [17] phát triển hệ thống iSleepPost gồm hai cảm biến gia tốc đeo ở cổ tay và ngực để ánh xạ chuyển động và nhãn tư thế, với độ chính xác trung bình 0.82%. Zhang và cộng sự [70] tiến thêm một bước khi chỉ sử dụng một cảm biến gia tốc ba trục gắn ở ngực, kết hợp bộ phân loại tuyển tính (LDA), cho phép nhận dạng bốn tư thế ngủ cơ bản và phát hiện chuyển động hiệu quả. Một số thiết kế khác như Wearable Sleep System (WSS) [71] hoặc thiết bị HST dạng miếng dán cổ [59] cũng dùng cảm biến gia tốc ba trục có độ chính xác cao (trên 97%) trong nhận dạng tư thế ngủ.

---

<sup>2</sup>Giáo sư Lê Tiến Thường - nguyên là giảng viên Trường Đại học Bách Khoa TP. Hồ Chí Minh, là chuyên gia trong lĩnh vực kỹ thuật điện tử và hệ thống nhúng, với nhiều công trình về cảm biến và trí tuệ nhân tạo ứng dụng trong y sinh.

Nghiên cứu [22], tác giả tập trung vào nhận dạng tư thế ngủ tự động bằng cảm biến gia tốc đơn gắn tại vùng cổ (Hình 1.7). Ba mô hình được thử nghiệm gồm Decision Tree (DT), Extra-Trees (ET) và LSTM Neural Network (LSTM-NN) và cho kết quả rằng mô hình DT có ưu thế về tốc độ dự đoán (0.8 ms) và dung lượng nhỏ (1.765 kb), phù hợp triển khai mô bình trên biên. Tác giả Vũ Hoàng Diệu và cộng sự [21] đã chứng minh khả năng của một hệ thống đơn cảm biến gia tốc trong nhận dạng đến 12 tư thế ngủ phục vụ bệnh nhân GERD, sử dụng mô hình AnpoNet kết hợp 1D-CNN và LSTM, đạt độ chính xác 94,67% và f1-score 92,94%. Các kết quả này cho thấy cảm biến gia tốc đơn không chỉ đảm bảo độ chính xác cao mà còn mang lại sự tiện lợi, chi phí phù hợp và khả năng triển khai thực tế.



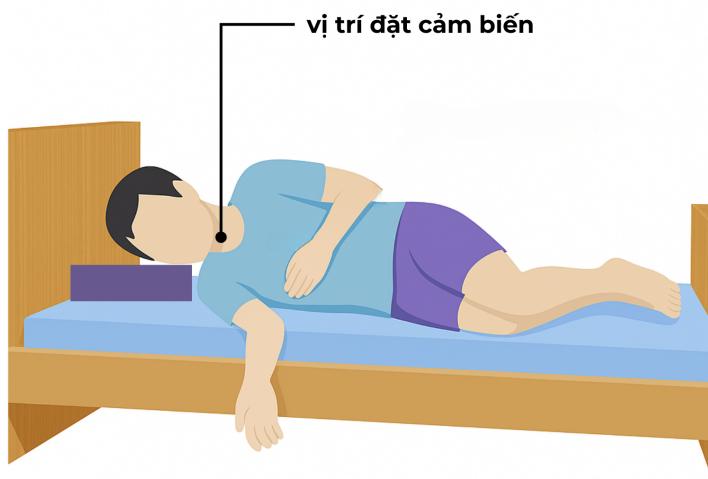
Hình 1.7: Vị trí đặt cảm biến được đề xuất trong nghiên cứu [22]

Bám sát định hướng này, luận văn đề xuất một hệ thống sử dụng duy nhất một cảm biến gia tốc gắn tại hõm ức để phát hiện các tư thế ngủ liên quan đến OSA. Vị trí đặt thiết bị ở cổ còn là tiền đề để nhóm mở rộng thêm cảm biến như luồng khí thở với mục

đích đánh giả OSA. Dữ liệu thu được sẽ được xử lý qua các bước tiền xử lý, trích xuất đặc trưng và huấn luyện mô hình học máy nhẹ, hướng tới việc phát triển thiết bị IoT nhỏ gọn, tiêu thụ năng lượng thấp, phân loại tư thế ngủ và có thể có khả năng ước lượng chỉ số AHI được trình bày ở các chương tiếp theo.

#### 1.2.4. Cảm biến gia tốc trong đánh giá tư thế ngủ

Ưu điểm nổi bật của cảm biến gia tốc nằm ở khả năng vận hành độc lập với mức tiêu thụ năng lượng thấp, đồng thời có thể tích hợp dễ dàng vào các hệ thống bảng mạch.



Hình 1.8: Vị trí đặt cảm biến gia tốc

Nhờ đó, cảm biến này đặc biệt thích hợp cho việc phát triển các thiết bị đeo thông minh có khả năng xử lý tại chỗ, nhỏ gọn, di động, tiêu thụ năng lượng thấp phù hợp với xu hướng TinyML trong lĩnh vực điện toán biên. Ngoài ra, cảm biến gia tốc không phụ thuộc vào các điều kiện môi trường như ánh sáng, nhiệt độ, sử dụng chăn, và quan trọng hơn, không xâm phạm quyền riêng tư của người dùng như các hệ thống giám sát bằng hình ảnh.

Tuy nhiên, việc đặt sai vị trí hoặc xoay lệch trục có thể dẫn đến sai lệch trong kết quả phân loại tư thế. Về mặt lựa chọn vị trí, các vùng như cổ tay hoặc trán thường chịu ảnh hưởng mạnh từ các chuyển động ngoại ý và lệch trục cơ thể, làm giảm độ đại diện của tín hiệu thu được. Ngược lại, vùng cổ được xem là vị trí lý tưởng nhờ tính ổn định hình học cao, gần trực trung tâm của cơ thể, và khả năng phản ánh trực tiếp hướng trọng lực. Những đặc điểm này giúp tín hiệu gia tốc tại vùng cổ có độ tương quan cao với tư

thế ngủ thực tế, qua đó nâng cao độ chính xác của hệ thống nhận diện tư thế trong ứng dụng y sinh. Đặc biệt, thuận lợi cho việc tích hợp thêm các cảm biến khác như cảm biến âm thanh hoặc cảm biến nhiệt độ, phục vụ cho các mục tiêu mở rộng trong các nghiên cứu tiếp theo

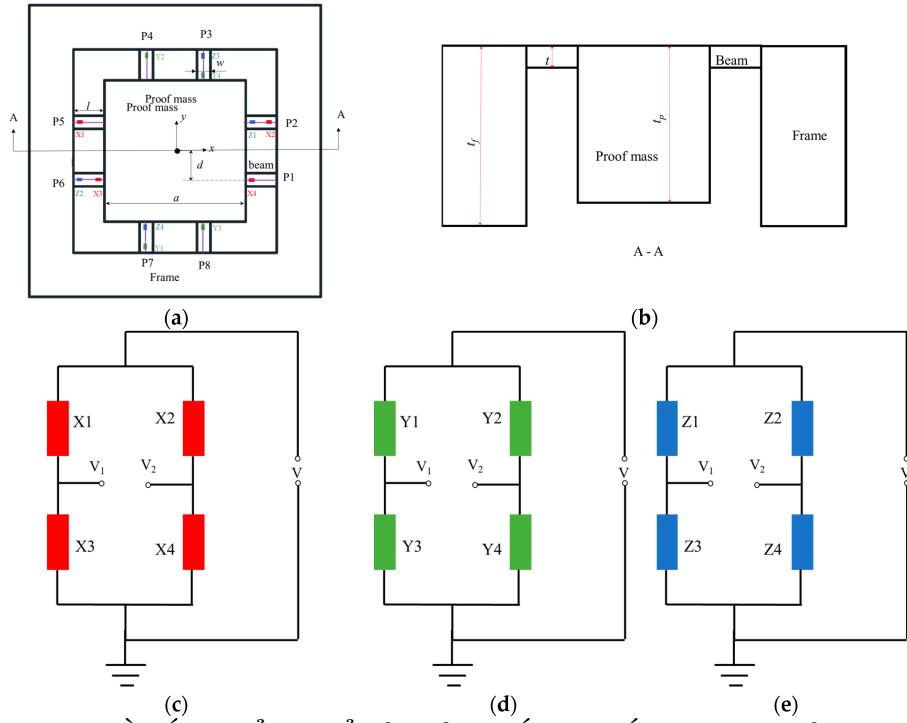
Xuất phát từ các phân tích nêu trên, nghiên cứu này lựa chọn thiết kế một thiết bị đeo tiếp xúc sử dụng cảm biến gia tốc ba trục, được gắn tại vùng cổ. Sau khi xác định được vị trí gắn tối ưu, bước tiếp theo là làm rõ cơ sở nguyên lý đo lường của cảm biến, nhằm lý giải vì sao thiết bị này có thể phản ánh chính xác tư thế cơ thể trong khi ngủ.

## **Nguyên lý cảm biến gia tốc**

Cảm biến gia tốc là một thiết bị đo lường có khả năng phát hiện và ghi nhận gia tốc theo thời gian. Với ưu điểm nhỏ gọn, tiêu thụ năng lượng thấp và chi phí hợp lý, cảm biến gia tốc được ứng dụng rộng rãi trong nhiều lĩnh vực và đặc biệt trong các thiết bị theo dõi hoạt động và giấc ngủ [72, 73].

Hiện nay, có ba dạng chính của cảm biến gia tốc vi cơ điện tử (Micro-Electro-Mechanical Systems - MEMS) là kiểu áp điện trở, kiểu điện dung và kiểu áp điện [74].

## Cảm biến gia tốc kiểu áp điện trở



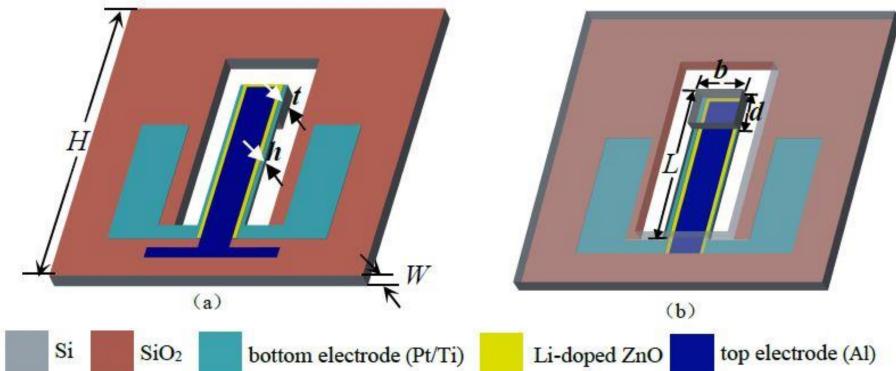
Hình 1.9: (a) Sơ đồ bố trí tổng thể của cảm biến gia tốc áp điện trở (b) Cấu trúc mặt cắt qua khối gia trọng và cấu trúc dầm (c) Cầu Wheatstone đo gia tốc trục X (d) Cầu Wheatstone đo gia tốc trục Y (e) Cầu Wheatstone đo gia tốc trục Z. [75]

Đầu tiên, cảm biến kiểu áp điện trở có cấu tạo gồm một cấu trúc dầm bằng vật liệu có tính chất có thể thay đổi điện trở, kèm theo một khối gia trọng (mass) và khung đỡ [75]. Bề mặt trên của khối gia trọng, cấu trúc dầm và khung nằm trên cùng một mặt phẳng, đảm bảo tính đối xứng cấu trúc. Khi cảm biến chịu tác động của gia tốc ngoài, ứng suất cơ học xuất hiện tại vùng đầu cấu trúc dầm, dẫn đến sự thay đổi điện trở của các phần tử này. Các điện trở được bố trí và liên kết một cách tối ưu để tạo thành ba mạch cầu Wheatstone độc lập, tương ứng với việc phát hiện các thành phần gia tốc theo ba trục vuông góc X, Y và Z,

Trong mỗi cầu Wheatstone, sự thay đổi điện trở vi sai được chuyển đổi trực tiếp thành tín hiệu điện áp đầu ra, cho phép cảm biến biến đổi biến dạng cơ học thành tín hiệu điện tương ứng.

## Cảm biến gia tốc kiểu áp điện

Cảm biến gia tốc kiểu áp điện được cấu tạo gồm hai phần chính [76]: (1) Cấu trúc dầm có một đầu cố định liên kết với khối gia trọng. Khi có gia tốc tác động, khối này tạo ra lực quán tính khiến cấu trúc dầm bị uốn cong, sinh ra ứng suất cơ học trên bề mặt dầm. Các thông số hình học cơ bản bao gồm: chiều dài dầm ( $L$ ), chiều rộng ( $b$ ), độ dày ( $h$ ), chiều rộng khối lượng ( $d$ ) và độ dày khối lượng ( $t$ ). (2) Cấu trúc áp điện gồm ba lớp vật liệu xếp chồng theo thứ tự: lớp điện cực dưới thường bằng bạch kim/titan, lớp màng áp điện thường bằng Li-doped ZnO, và lớp điện cực trên thường bằng nhôm.



Hình 1.10: Cấu trúc cơ bản của cảm biến MEMS áp điện: (a) mặt trên (b) mặt dưới [76]

Khi chịu tác dụng của gia tốc ngoài  $F$ , dầm bị biến dạng đàn hồi. Theo phân tích cơ học dầm và lý thuyết biến dạng nhỏ [77], ứng suất dọc  $\sigma_l$  xuất hiện trên bề mặt dầm theo phương trục  $x$  được biểu diễn như sau:

$$\sigma_l = \sigma_1 = \frac{6FL}{bh^2} \quad (1.1)$$

Dựa trên hiệu ứng áp điện thuận, hai bề mặt trên và dưới của màng áp điện ZnO pha tạp Li lần lượt mang điện tích dương và âm theo phương trục  $z$ . Lượng điện tích cảm ứng  $q_3$  được xác định bởi công thức:

$$q_3 = d_{31}\sigma_1 \quad (1.2)$$

Trong đó,  $d_{31}$  là hệ số áp điện, và  $\sigma_1$  là ứng suất pháp tuyến theo phương  $x$ . Theo định nghĩa, điện dung  $C$  giữa hai điện cực trên và dưới của cấu trúc áp điện được xác định bởi:

$$C = \frac{q_3}{V} \quad (1.3)$$

với  $V$  là điện áp đầu ra giữa hai điện cực của màng áp điện. Từ các quan hệ trên, điện áp đầu ra có thể được biểu diễn theo dạng:

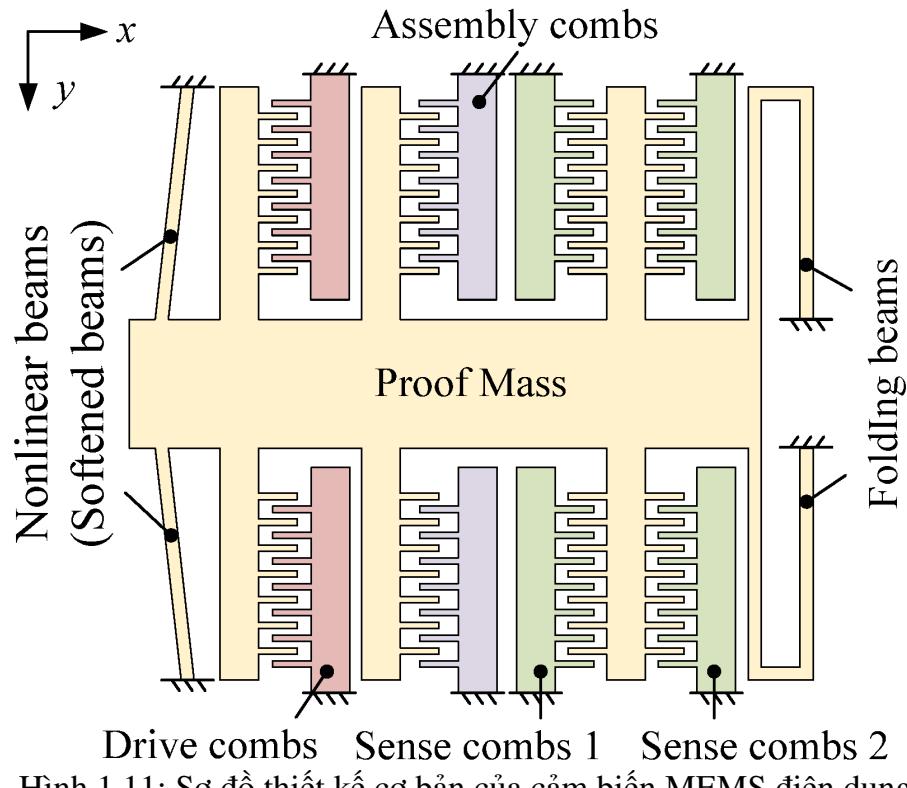
$$V = \frac{d_{31} \cdot 6mL}{bh^2 C} \cdot a \quad (1.4)$$

Kết hợp với định nghĩa độ nhạy của cảm biến, ta có biểu thức:

$$S = \frac{V}{a} = \frac{d_{31} \cdot 6mL}{bh^2 C} \quad (1.5)$$

### Cảm biến gia tốc MEMS điện dung

Cảm biến gia tốc điện dung được gồm ba khối chức năng chính: khối gia trọng, hệ đàm đòn hồi và cụm điện cực hình răng lược [78]. Khối gia trọng là phần trung tâm của cảm biến, dao động khi có gia tốc tác động. Các đàm có nhiệm vụ đỡ và liên kết khối lượng này, đồng thời cho phép chuyển động đòn hồi trong giới hạn thiết kế. Cụm điện cực hình răng lược (assembly combs) được sử dụng để cǎn chỉnh trong quá trình chế tạo, trong khi cụm điện cực hình răng lược (drive combs) tạo dao động cưỡng bức cho khối gia trọng trong các thí nghiệm đo đáp ứng. Cụm điện cực hình răng lược (sense combs) đảm nhiệm chức năng phát hiện độ dịch chuyển của khối lượng thông qua sự thay đổi điện dung giữa các ngón điện cực.



Hình 1.11: Sơ đồ thiết kế cơ bản của cảm biến MEMS điện dung

Để đo tín hiệu, cảm biến sử dụng hai cụm (sense combs 1 và sense combs 2) được mắc theo cấu hình vi sai. Hai tín hiệu điện dung biến thiên ngược pha này giúp tăng hệ số khuếch đại vi sai và loại bỏ nhiễu chung, nhờ đó cải thiện đáng kể độ nhạy và độ ổn định của cảm biến điện dung.

Tác động đầu vào  $\xrightarrow{\text{Gây ra dịch chuyển } x}$  Thay đổi điện dung ( $\Delta C$ )  $\xrightarrow{\text{Mạch xử lý tín hiệu}}$  Điện áp ra

Hình 1.12: Quá trình biến đổi tín hiệu trong cảm biến điện dung.

Khi có tác động gia tốc sẽ gây ra sự dịch chuyển cơ học  $x$  giữa các bản cực của tụ điện. Sự dịch chuyển này làm thay đổi điện dung  $\Delta C$  của cảm biến. Biến thiên điện dung  $\Delta C$  này được mạch xử lý tín hiệu chuyển đổi thành điện áp đầu ra  $V_{out}$ .

### **Nguyên lý kích thích và chuyển đổi C-V trong cảm biến điện dung MEMS**

Dựa trên nguyên lý cảm ứng điện dung vi sai, hai tụ điện cảm biến có điện dung lần lượt được xác định bởi [79]:

$$C_1 = C_0 + \Delta C, \quad C_2 = C_0 - \Delta C \quad (1.6)$$

Trong đó,  $C_0$  là điện dung xác định khi không có gia tốc tác động, và  $\Delta C$  là phần biến thiên điện dung phát sinh do dịch chuyển cơ học của khối gia trọng dưới tác dụng của gia tốc.

Tín hiệu kích thích được áp dụng lên hai tụ có dạng sóng vuông vi sai  $\pm V_p$ , gọi là điện áp kích thích. Sự dịch chuyển cơ học  $x$  gây ra biến thiên điện dung  $\Delta C$  được chuyển đổi thành tín hiệu điện thông qua mạch khuếch đại điện tích hoạt động theo nguyên lý bảo toàn điện tích:

$$Q = C_f V_{\text{out}} \quad (1.7)$$

Trong đó,  $Q$  là điện tích lũy tại nút khuếch đại,  $C_f$  là điện dung phản hồi của op-amp, và  $V_{\text{out}}$  là điện áp đầu ra của mạch.

Sau quá trình giải điều chế trong hai pha  $\varphi_1$  và  $\varphi_2$ , hai nhánh đầu ra vi sai của mạch được xác định bởi:

$$V_{\text{out}+} = V_{\text{cm}} + \frac{4\Delta C}{C_f} V_p, \quad V_{\text{out}-} = V_{\text{cm}} - \frac{4\Delta C}{C_f} V_p \quad (1.8)$$

với  $V_{\text{cm}}$  là điện áp chung,  $V_p$  là biên độ của tín hiệu kích thích.

Khi lấy hiệu hai đầu ra vi sai, điện áp đầu ra của cảm biến được xác định như sau:

$$V_{\text{out}} = V_{\text{out}+} - V_{\text{out}-} = \frac{8\Delta C}{C_f} V_p \quad (1.9)$$

Trong đó,  $\Delta C$  biểu thị mức độ biến dạng cơ học của phần tử cảm biến,  $C_f$  quyết định độ khuếch đại của mạch (giá trị  $C_f$  càng nhỏ thì độ khuếch đại càng lớn), và  $V_p$  là biên độ kích thích tỉ lệ thuận với biên độ điện áp đầu ra.

Từ công thức 1.9, có thể thấy rằng điện áp vi sai đầu ra  $V_{\text{out}}$  tỉ lệ tuyến tính với biến thiên điện dung  $\Delta C$ , tỉ lệ thuận với điện áp kích thích  $V_p$ . Dạng tổng quát của mối quan hệ này có thể được biểu diễn gọn hơn như sau:

$$V_{\text{out}} = K \cdot \frac{\Delta C}{C_f} V_p \quad (1.10)$$

với  $K = 8$  là hệ số khuếch đại hiệu dụng phụ thuộc vào cấu hình vi sai.

Bảng 1.3: So sánh đặc tính giữa ba loại cảm biến gia tốc MEMS

Tham số	Áp điện	Áp điện trở	Điện dung
<b>Đáp ứng DC</b>	Kém - không đo được tín hiệu tĩnh	Tốt - đo được tín hiệu tĩnh	Tốt - đo được tín hiệu tĩnh và chậm
<b>Trở kháng</b>	Cao	Thấp	Cao
<b>Mức tín hiệu</b>	Cao - tạo điện áp trực tiếp	Thấp - cần khuếch đại cầu Wheatstone	Trung bình - phụ thuộc vào biến thiên điện dung và mạch khuếch đại
<b>Hiệu chuẩn tĩnh</b>	Kém	Tốt	Tốt
<b>Độ nhạy</b>	Cao	Thấp	Cao
<b>Thiết kế giảm chấn</b>	Kém - khó kiểm soát tắt dần	Tốt - dễ điều chỉnh	Tốt - điều khiển được qua cấu trúc MEMS
<b>Giá thành</b>	Cao - vật liệu và chế tạo phức tạp	Thấp - tương thích công nghệ CMOS	Cao - yêu cầu mạch xử lý phức tạp
<b>Đo xung dài (<math>&gt;10</math> ms)</b>	Kém - chỉ đo tín hiệu dao động nhanh	Tốt	Tốt
<b>Độ nhạy với ứng suất lắp đặt</b>	Cao - dễ sai lệch do biến dạng	Thấp	Thấp - cấu trúc vi sai cân bằng
<b>Độ phức tạp của mạch</b>	Đơn giản - xuất điện áp trực tiếp	Đơn giản - cầu Wheatstone	Phức tạp - cần mạch khuếch đại vi sai, điều chế và giải điều chế

Dựa vào Bảng 1.3 trong khuôn khổ luận văn này, cảm biến gia tốc MEMS kiểu điện dung được lựa chọn làm phần tử cảm biến trung tâm cho hệ thống theo dõi tư thế ngủ kéo dài, trên cơ sở đánh giá đa chiều cả về đặc trưng tín hiệu, yêu cầu hệ thống, lẫn khả năng triển khai thực tế.

**Tóm lại**, việc lựa chọn cảm biến điện dung MEMS không đơn thuần là một quyết định kỹ thuật thuần túy, mà còn phản ánh một giải pháp có tính cân bằng giữa hiệu quả đo lường, khả năng tích hợp phần cứng, và chiến lược mở rộng ứng dụng lâm sàng theo

hướng chi phí phù hợp. Đây là một trong những nguyên tắc then chốt trong xu hướng đổi mới công nghệ y tế cộng đồng hiện đại - nơi mà tính khả thi triển khai và khả năng mở rộng đóng vai trò không kém phần quan trọng so với độ chính xác kỹ thuật.

Nhìn chung, việc xác định tư thế ngủ để phục vụ việc chẩn đoán hội chứng ngưng thở khi ngủ đòi hỏi một hệ thống giám sát liên tục, có khả năng thu thập dữ liệu sinh lý trong thời gian dài trong môi trường sinh hoạt tự nhiên tại nhà. Những yêu cầu này không chỉ giới hạn ở độ chính xác mô hình, mà còn mở rộng tới tiêu chí về tính gọn nhẹ, tiết kiệm năng lượng, và dễ triển khai trên nền tảng vi điều khiển hoặc thiết bị đeo.

Trong bối cảnh đó, các mô hình học máy trở thành công cụ thiết yếu để khai thác dữ liệu cảm biến. Chúng không chỉ cho phép phân loại tư thế với độ chính xác cao mà còn mở rộng sang suy luận các chỉ số lâm sàng như AHI. Việc triển khai trực tiếp trên vi điều khiển theo hướng TinyML đưa giải pháp từ phòng thí nghiệm đến ứng dụng thực tiễn tại nhà, đánh dấu sự giao thoa giữa y học giấc ngủ truyền thống và trí tuệ nhân tạo ứng dụng - một lĩnh vực đang phát triển mạnh mẽ trong thập niên gần đây [48].

### **1.2.5. Học máy trong phân loại OSA và tư thế ngủ**

Cùng với lượng dữ liệu phong phú và liên tục được thu thập từ các cảm biến sinh học, các thuật toán học máy trở thành công cụ thiết yếu để phân tích và khai thác hiệu quả các dòng dữ liệu này, giúp đưa ra quyết định nhanh chóng và chính xác. Đặc biệt, trong bối cảnh các thiết bị theo dõi sức khỏe tại nhà yêu cầu kích thước nhỏ gọn, tính di động cao, và không phụ thuộc vào kết nối Internet, thì học máy tại biên (Edge Machine Learning) nổi lên như một giải pháp tối ưu vừa mang lại khả năng tính toán mạnh mẽ, vừa dễ dàng triển khai trên các phần cứng có giới hạn tài nguyên.

Chính vì những lý do đó, tác giả lựa chọn hướng nghiên cứu và phát triển học máy tại biên cho đề tài này.

Bảng 1.4: Các bước chính trong bài toán sử dụng học máy trên biên

STT	Giai đoạn	Mô tả tổng quát
1	<b>Thu thập tín hiệu</b>	Ghi nhận tín hiệu từ cảm biến, xử lý và lưu trữ.
2	<b>Tiền xử lý</b>	Lọc nhiễu, phân đoạn theo cửa sổ thời gian, thống kê dữ liệu.
3	<b>Trích xuất đặc trưng</b>	Tính toán các đặc trưng thời gian, tần số - đại diện cho nội dung sinh lý trong từng đoạn tín hiệu.
4	<b>Lựa chọn và huấn luyện mô hình</b>	Lựa chọn thuật toán học máy phù hợp với bài toán.
5	<b>Đánh giá hiệu năng</b>	Sử dụng các chỉ số đánh giá mô hình như độ chính xác (Accuracy), độ chính xác dự đoán dương (Precision), độ bao phủ (Recall), điểm F1 (F1-Score), diện tích dưới đường cong (AUC) và ma trận nhầm lẫn (Confusion Matrix)..
6	<b>Tối ưu mô hình</b>	Ứng dụng kỹ thuật cắt tỉa (pruning), lượng tử hóa (quantization) để giảm kích thước và độ phức tạp mô hình nhằm phục vụ triển khai biên.
7	<b>Triển khai thực tế</b>	Triển khai mô hình trên vi điều khiển, đánh giá và tái huấn luyện

Bảng 1.4 trình bày tổng quan quy trình triển khai một bài toán học máy. Quy trình này có thể được điều chỉnh linh hoạt tùy theo loại tín hiệu đầu vào và mục tiêu phân tích cụ thể, chẳng hạn như nhận diện tư thế ngủ, phát hiện ngưng thở hay theo dõi nhịp hô hấp. Tuy nhiên, nguyên tắc cốt lõi vẫn là đảm bảo tín hiệu đầu vào có chất lượng cao và mô hình học máy đủ nhẹ để triển khai trong thực tế.

**Phần tiếp theo tác giả sẽ tập trung phân tích quy trình triển khai nêu trên vào bài toán phân loại tư thế ngủ và đánh giá OSA.**

### Thu thập tín hiệu

Là bước đầu tiên và đóng vai trò nền tảng trong toàn bộ quy trình tiếp đó. Các tín hiệu được ghi nhận có thể bao gồm tín hiệu gia tốc ba trục để phát hiện chuyển động và

tư thế nằm; tín hiệu phản hồi quang học dùng để đo nhịp tim và độ bão hòa oxy trong máu; tín hiệu ECG phản ánh hoạt động điện học của tim; hoặc các tín hiệu hô hấp và áp lực từ cảm biến gắn trên giường, giúp xác định tư thế ngủ thông qua phân bố trọng lực và các tín hiệu bổ sung khác [80, 81]. Tùy theo dạng dữ liệu thu thập và cấu hình thiết bị, các tín hiệu này được lấy mẫu với tần số phù hợp [17, 23, 48, 59, 70, 82].

Một yêu cầu then chốt của bước này là đảm bảo tính ổn định và chất lượng dữ liệu. Các cảm biến cần được gắn cố định tại các vị trí tối ưu, chẳng hạn như vùng ngực, cổ tay hoặc vùng cổ, nhằm giảm nhiễu chuyển động và duy trì tiếp xúc ổn định trong suốt quá trình theo dõi.

Ngoài ra, hệ thống cũng cần đảm bảo khả năng lưu trữ và truyền dữ liệu hiệu quả thông qua các giao thức không dây tiết kiệm năng lượng như Bluetooth Low Energy (BLE) hoặc Wi-Fi [67, 58, 59].

Chất lượng dữ liệu thu được trong bước này ảnh hưởng trực tiếp đến hiệu quả của các mô hình học máy. Do đó, quy trình thu thập tín hiệu cần được thiết kế sao cho vừa đảm bảo phản hồi đúng đặc trưng sinh lý, vừa duy trì tính khả thi khi triển khai lâu dài trong môi trường ngoài lâm sàng.

Tuy nhiên, trong nhiều nghiên cứu hiện nay, kiến trúc phần mềm cho quá trình thu thập và lưu trữ dữ liệu vẫn chưa được mô tả một cách đầy đủ và thống nhất. Các yếu tố như:

- Kiến trúc hệ thống: ứng dụng di động, nền tảng web hoặc máy chủ trung tâm;
- Giao thức truyền thông: MQTT, HTTP;
- Phương thức lưu trữ dữ liệu: cục bộ, đám mây (cloud) hoặc cơ sở dữ liệu liên kết;
- Các vấn đề liên quan đến mã hóa, bảo mật dữ liệu sức khỏe và tuân thủ quy định chuyên ngành.

Việc thiếu chuẩn hóa trong các yếu tố trên có thể ảnh hưởng đáng kể đến khả năng mở rộng, tích hợp và ứng dụng thực tiễn của hệ thống trong môi trường chăm sóc sức khỏe tại nhà. Chính vì vậy, luận văn này tập trung giải quyết những hạn chế đó, hướng đến xây dựng một quy trình thu thập và quản lý tín hiệu sinh học thống nhất, bảo mật và sẵn sàng cho triển khai thực tế.

## Tiền xử lý

**Phân tích dữ liệu** là bước thông kê lại nhưng dữ liệu đã thu thập bao gồm các yếu tố như: độ cân bằng, độ lặp, các giá trị null/NaN, hay các thông kê khác theo yêu cầu bài toán.

**Lọc tín hiệu** là bước đầu tiên trong tiền xử lý nhằm loại bỏ nhiễu khỏi các tín hiệu sinh lý như EEG, ECG, EMG, âm thanh, SpO<sub>2</sub> và gia tốc. Các loại nhiễu thường gặp bao gồm: nhiễu điện lưới (50-60 Hz), nhiễu tần số cao và trôi đường cơ sở [83, 84]. Các kỹ thuật lọc thường được sử dụng bao gồm: bộ lọc chặn dải (notch filter) - đặc biệt là bộ lọc notch IIR bậc hai để loại bỏ nhiễu điện lưới; bộ lọc thông dải (bandpass filter) nhằm giữ lại dải tần sinh lý; và các bộ lọc làm mượt như Butterworth bậc ba, bộ lọc trung vị (median filter) hoặc bộ lọc trung bình trượt (moving average filter) [80, 81]. Trong nghiên cứu của Sheta và cộng sự [84], tác giả đã sử dụng bộ lọc notch IIR bậc hai nhằm loại bỏ nhiễu điện lưới 60 Hz khỏi tín hiệu ECG. Đây là loại nhiễu phổ biến gây khó khăn cho việc phân tích và trích xuất đặc trưng. Kết quả cho thấy tín hiệu sau lọc cho chất lượng cao hơn đáng kể và cải thiện hiệu năng của các mô hình học máy trong chẩn đoán OSA. Đối với tín hiệu gia tốc, các bộ lọc cần đảm bảo vừa loại bỏ nhiễu vừa bảo toàn các đặc trưng chuyển động. Trong đó, Kalman filter và bộ lọc biến thiên toàn phần (total variation filter) được sử dụng rộng rãi để làm mượt chuỗi tín hiệu thời gian và giảm nhiễu ngẫu nhiên [68, 85, 86, 87]. Việc lựa chọn bộ lọc phù hợp cần dựa trên đặc điểm của từng loại tín hiệu và mục tiêu phân tích cụ thể.

**Nội suy** là một bước quan trọng trong quá trình chuẩn hóa chuỗi thời gian, nhằm đảm bảo tính đồng nhất về tần số lấy mẫu giữa các kênh dữ liệu và duy trì tính toàn vẹn của đặc trưng tín hiệu. Các phương pháp nội suy có thể kể đến như spline bậc ba (cubic spline), nội suy Hermite đa thức từng đoạn (piecewise cubic Hermite - PCHIP) và nội suy tuyến tính (linear interpolation) thường được áp dụng để điều chỉnh chuỗi dữ liệu về cùng tần số chuẩn, từ đó bảo đảm đầu vào đồng bộ cho các mô hình học máy hoặc mạng nơ-ron, giúp cải thiện khả năng trích xuất đặc trưng của tín hiệu [88]. Bên cạnh đó, đối với các chuỗi tín hiệu rời rạc như nhịp RR từ ECG hoặc chuỗi SpO<sub>2</sub> thường xuyên bị gián đoạn hay mất mẫu, kỹ thuật nội suy đóng vai trò thiết yếu trong việc phục hồi dữ liệu bị thiếu (missing/null values imputation) và tái cấu trúc tín hiệu thành chuỗi liên tục. Cách tiếp cận này không chỉ nâng cao tính toàn vẹn của dữ liệu mà còn giúp mô hình học sâu khai thác được các mối quan hệ động trong miền thời gian, dẫn đến hiệu năng

suy luận tốt hơn [89]. Trong nghiên cứu của [90], cả hai loại tín hiệu gia tốc và phản hồi quang học đều được nội suy để tạo thành chuỗi thời gian đồng nhất với tần số lấy mẫu.

**Chuẩn hóa dữ liệu** giúp đưa các đặc trưng đầu vào về cùng một miền giá trị, tránh hiện tượng các đặc trưng có giá trị số lớn chi phối quá trình huấn luyện mô hình. Một số phương pháp phổ biến bao gồm: chuẩn hóa về trung bình 0 và độ lệch chuẩn 1 (Z-score normalization), đưa về khoảng 0,1 (min-max scaling), dựa trên trung vị và tứ phân vị, phù hợp với dữ liệu có nhiễu hoặc ngoại lệ(robust scaling) [59, 21, 83].

**Phân đoạn tín hiệu** là một bước thiết yếu trong xử lý tín hiệu sinh lý. Mục tiêu của bước này là chia chuỗi dữ liệu liên tục thành các đoạn thời gian ngắn cố định gọi là cửa sổ (window), giúp mô hình học máy nhận diện hiệu quả các đặc trưng biến đổi theo thời gian. Thời lượng cửa sổ thường phụ thuộc vào loại tín hiệu và mục tiêu phân tích: 30 giây cho EEG, 60 giây cho ECG và SpO<sub>2</sub>, 5 phút cho HRV, hoặc các cửa sổ trượt ngắn để phát hiện sự kiện ngắn hạn [48, 59, 89, 91]. Trong một số trường hợp, việc phân đoạn còn dựa vào các đặc điểm sinh lý như đỉnh sóng R trong ECG. Riêng đối với dữ liệu gia tốc để xác định tư thế thì cửa sổ phân đoạn là từ 0.5Hz đến 3Hz [21, 22].

Tóm lại, tiền xử không chỉ giúp nâng cao chất lượng dữ liệu mà còn đảm bảo tính nhất quán đầu vào cho hệ thống học máy. Việc lựa chọn kỹ thuật tiền xử lý cần phù hợp với đặc điểm của từng loại tín hiệu và mục tiêu phân tích cụ thể.

### Trích xuất đặc trưng

**Trích xuất đặc trưng** là một bước quan trọng nhằm chuyển đổi tín hiệu cảm biến thô thành tập hợp các đặc trưng có ý nghĩa, phản ánh đúng ý nghĩa về mặt sinh lý. Quá trình này giúp làm nổi bật các thông tin cốt lõi từ tín hiệu sinh học, đồng thời loại bỏ các yếu tố dư thừa hoặc nhiễu không mang giá trị chẩn đoán.

Trong bài toán phát hiện hội chứng ngưng thở khi ngủ (OSA) [56, 62, 82, 91, 92, 93], các đặc trưng được sử dụng phổ biến bao gồm: biên thiên nhịp tim, khoảng RR và biên độ sóng R từ tín hiệu ECG; các chỉ số thông kê, tần số và entropy từ tín hiệu SpO<sub>2</sub>; hoặc các đặc trưng phi tuyến và miền tần số như năng lượng phổ EEG. Ngoài ra, các hệ số wavelet thu được từ tín hiệu gốc có khả năng phản ánh rõ ràng các tín hiệu liên quan đến âm thanh, hình ảnh [93].

Đối với bài toán phân loại tư thế ngủ sử dụng cảm biến gia tốc đơn, các đặc trưng

thường được trích xuất trực tiếp từ miền thời gian. Trong nghiên cứu *Sleep Posture Monitoring Using a Single Neck-Situated Accelerometer* [22], các đặc trưng được tính toán hoàn toàn trong miền thời gian, bao gồm giá trị trung bình và trung vị của ba trục gia tốc (X, Y, Z) cùng với các góc nghiêng giữa các mặt phẳng XY, YZ và XZ, tạo thành tổng cộng 12 đặc trưng cho mỗi cửa sổ dữ liệu. Trong khi đó, nghiên cứu [21, 94] không trích xuất đặc trưng thủ công mà đưa trực tiếp tín hiệu gia tốc thô ( $a_x, a_y, a_z$ ) vào mô hình LSTM. Mạng học sâu này tự động học các đặc trưng với cửa sổ 20 mẫu, nhờ đó mô hình đạt độ chính xác cao mà không cần bước trích chọn đặc trưng.

**Gán nhãn** Được thực hiện nhằm liên kết các phân đoạn tín hiệu với nhãn sự kiện hô hấp tương ứng như “Apnea”, “Hypopnea”, hoặc phân mức độ nặng của OSA như “Mild”, “Moderate”, “Severe”, các nhãn liên quan đến tư thế ngủ. Việc gán nhãn đúng đắn giúp huấn luyện mô hình AI chính xác. Tùy theo thiết kế nghiên cứu, nhãn có thể được gán thủ công dựa trên chuyên gia hoặc tự động đồng bộ với thiết bị tham chiếu như camera, hệ thống PSG.

## Lựa chọn và huấn luyện mô hình

Từ các bước thu thập dữ liệu, tiền xử lý và trích xuất đặc trưng đã trình bày, có thể nhận thấy rằng các tín hiệu sinh lý trong việc chẩn đoán OSA. Đặc biệt là tín hiệu gia tốc cung cấp nguồn thông tin giá trị trong việc nhận diện tư thế ngủ. Tuy nhiên, trong khuôn khổ nghiên cứu này, tác giả không đi sâu vào phân tích chi tiết các mô hình liên quan đến hội chứng ngưng thở khi ngủ (OSA), mà tập trung vào việc xây dựng và tối ưu mô hình học máy nhằm phục vụ phân loại tư thế ngủ một cách hiệu quả và khả thi khi triển khai trên thiết bị biên. Do đó, từ phần tiếp theo, nội dung nghiên cứu sẽ tập trung vào quy trình nghiên cứu, huấn luyện và đánh giá hiệu năng của mô hình học máy cho bài toán phân loại tư thế ngủ.

Dữ liệu trong Bảng 1.5 cho thấy sự đa dạng trong cách tiếp cận bài toán nhận diện tư thế ngủ. Hầu hết các nghiên cứu đạt độ chính xác cao (trên 95%), ngay cả khi sử dụng các thuật toán đơn giản như ngưỡng định sẵn hoặc mô hình học máy truyền thống. Tuy nhiên, một điểm đáng lưu ý là chỉ duy nhất nghiên cứu của Abdulsadig và cộng sự [22] thực sự được triển khai trên phần cứng nhúng, chứng tỏ tính khả thi trong môi trường tài nguyên hạn chế. Các nghiên cứu còn lại chủ yếu dừng ở mức mô phỏng phần mềm hoặc chạy trên máy tính.

Bảng 1.5: Hiệu quả mô hình học máy và khả năng triển khai biên trong nhận diện tư thế ngủ

Tài liệu	Dữ liệu	Nhiệm vụ	Đặc trưng	Mô hình	Độ chính xác	Phần mềm	On-chip
[22]	18 đối tượng	4 tư thế ngủ	12 đặc trưng miền thời gian	DT, Extra-Trees, LSTM-NN	>98%	BLE -> laptop	Có
[17]	NM	4 tư thế ngủ + 2 tư thế ngồi và đứng	3 đặc trưng chính - trung bình (mean)	RF, SVM	80-90%	BLE -> điện thoại -> laptop	NM
[94]	561.859 mẫu	4 tư thế ngủ và không phải	dữ liệu thô	LSTM_sq	>99%	NM	NM
[71]	16 đối tượng	8 tư thế ngủ	Trung bình, phương sai của 3 trực	Ngưỡng	lên tới 99.5%	NM	NM
[95]	5 đối tượng	12 tư thế ngủ	động học khớp cổ tay, mô tả hướng quay và biên độ xoay	Ngưỡng	>99.2%	cảm biến -> PC	Không
[59]	18 đối tượng	4 tư thế ngủ	12 đặc trưng miền thời gian	Ngưỡng , Extra-Trees	>95%	cảm biến -> PC	Không
[20]	13 đối tượng	4 tư thế ngủ và không phải	trung bình giá trị giá tốc 3 trực	Ngưỡng	>96%	cảm biến -> PC	Không
[70]	7 đối tượng	4 tư thế ngủ	trung bình giá trị giá tốc 3 trực	Linear Discriminant Analysis	>99%	cảm biến -> PC	Không
[96]	15 đối tượng	4 tư thế ngủ	40 đặc trưng thời gian, tần số	DT, RF, SVM, XGBoost, AdaBoost	95.65%	cảm biến -> PC	Không

NM = Không đề cập

Đi sâu hơn, nhóm mô hình học sâu LSTM hoặc kết hợp nhiều lớp tuy đạt độ chính xác vượt trội (trên 99%) như trong nghiên cứu[94], nhưng đều chưa tích hợp trực tiếp lên vi điều khiển do dung lượng bộ nhớ lớn và nhu cầu tính toán cao. Điều này đặt ra thách thức cho việc ứng dụng chúng trong môi trường thực tế như giấc ngủ tại nhà, nơi mà năng lượng, khả năng lưu trữ và kết nối đều bị hạn chế.

Ở chiều ngược lại, các mô hình truyền thống như cây quyết định, rừng ngẫu nhiên hay ngưỡng cố định cho thấy hiệu quả không hề thua kém khi được khai thác đúng đặc trưng. Đặc biệt, việc sử dụng các đặc trưng miền thời gian dễ tính toán so với miền tần số, giúp giảm đáng kể độ phức tạp tính toán [97], [98]. Điều này rất phù hợp với yêu cầu thiết kế của TinyML: đơn giản, hiệu quả và khả thi khi triển khai thực tế trên các nền tảng phần cứng hạn chế.

Nhờ sự phát triển của các nền tảng hỗ trợ triển khai mô hình học máy như **TensorFlow Lite for Microcontrollers**, **Edge Impulse**, và **TinyML EON Compiler**, việc huấn luyện, chuyển đổi và triển khai các mô hình đã trở nên dễ tiếp cận hơn nhiều. Điều này mở ra hướng mới trong chẩn đoán y tế cá nhân hóa, thông qua thiết bị đeo thông minh hoạt động liên tục tại nhà.

Vì vậy, luận văn được trình bày theo hai giai đoạn: (i) *Đánh giá đặc trưng và mô hình trên dữ liệu thu thập - xây dựng chuỗi xử lý từ tiền xử lý, phân đoạn theo cửa sổ, trích xuất đặc trưng đại diện (miền thời gian và tần số) đến so sánh các mô hình học máy.*

Tiêu chí lựa chọn không chỉ dựa trên độ chính xác, độ chụm cho từng tư thế, mà còn cân nhắc tới kích thước mô hình, chi phí tính toán. (ii) *Triển khai on-chip (TinyML) trên vi điều khiển mục tiêu* triển khai trên vi điều khiển nRF52840, đồng thời kiểm chứng so với giai đoạn (i).

## **Chương 2**

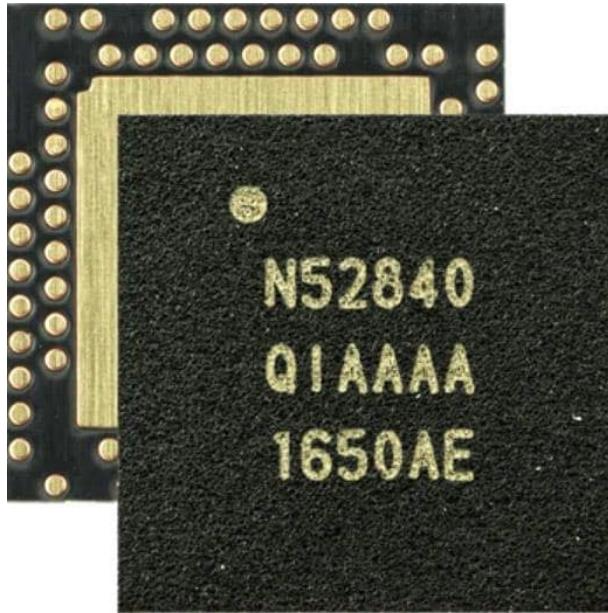
# **HỆ THỐNG THU THẬP, XỬ LÝ TÍN HIỆU CẢM BIẾN VÀ ĐÁNH GIÁ BẰNG MÔ HÌNH HỌC MÁY**

### **2.1. Nghiên cứu, phát triển phần cứng**

Trong phần này, tác giả sẽ trình bày các nội dung bao gồm: nghiên cứu, lựa chọn vi điều khiển và cảm biến phù hợp; từ đó tiến hành thiết kế sơ đồ mạch, mô phỏng mạch 3D và hoàn thiện sản phẩm.

#### **2.1.1. Vi xử lý**

Với sự phát triển vượt bậc và đa dạng của công nghệ thiết kế và chế tạo, có rất nhiều cấu hình phần cứng được nhiều nhóm tác giả lựa chọn phù hợp với các mục đích khác nhau. Trong đó, [99] các tác giả đã sử dụng máy tính đơn Raspberry Pi để phát hiện 4 tư thế ngủ với sự lấy nhãn từ video theo dõi người bệnh trong suốt quá trình lấy mẫu. Tác giả Kwasnicki và cộng sự đã sử dụng bộ xử lý công suất thấp TI MSP430 và mô-đun RF Chipcon CC2420 cho truyền thông không dây kết hợp với cảm biến gia



Hình 2.1: Nordic NRF52840

tốc 3 trục ADXL330, con quay hồi chuyển đạt được 99.5% độ chính xác [71]. Nhóm của tác giả I.Yun đã phát triển thiết bị theo dõi tư thế ngủ của trẻ nhỏ sử dụng vi xử lý ATmega328P-PU kết hợp cảm biến gia tốc ADXL335 được đặt trên bụng [55]. Trong nghiên cứu của Abdulsadig và cộng sự, hệ thống thu thập dữ liệu được xây dựng dựa trên một bo mạch tùy chỉnh tích hợp vi điều khiển nRF5232 (tích hợp Bluetooth Low Energy (BLE). Vi điều khiển này đảm nhiệm đồng thời cả việc lấy mẫu dữ liệu từ cảm biến gia tốc ba trục LIS2DH12 với tần số 100 Hz và truyền dữ liệu không dây theo thời gian thực [59, 22]. Trong nghiên cứu của tác giả Vũ Hoàng Diệu, mô-đun ESP32 được lựa chọn làm đơn vị xử lý trung tâm nhờ tích hợp bộ vi điều khiển hiệu năng cao, kết nối không dây Wi-Fi và khả năng mở rộng linh hoạt [94]. Với thiết kế nhỏ gọn, chi phí hợp lý và mức tiêu thụ điện năng thấp, ESP32 đáp ứng tốt yêu cầu của hệ thống thu thập dữ liệu tư thế ngủ theo thời gian thực. Thiết bị không chỉ cho phép truyền dữ liệu trực tiếp lên máy chủ hoặc nền tảng đám mây thông qua Wi-Fi, mà còn hỗ trợ lưu trữ cục bộ trên thẻ nhớ microSD, đảm bảo tính liên tục trong điều kiện mất kết nối mạng.

Qua phân tích các nghiên cứu trên có thể thấy rằng phần lớn các cấu hình phần cứng có kích thước vẫn còn lớn ( $> 5\text{cm}$ ), hoặc gặp giới hạn trong khả năng tích hợp mô hình học máy tại thiết bị. Do đó, sau khi nghiên cứu, tác giả quyết định lựa chọn vi điều khiển nRF52840 làm bộ điều khiển trung tâm [100] với các lý do sau:

nRF52840 tích hợp bộ giao thức không dây Bluetooth năng lượng thấp hoạt động

# Feature list

Features:	
<ul style="list-style-type: none"> <li>• Bluetooth® 5, IEEE 802.15.4-2006, 2.4 GHz transceiver           <ul style="list-style-type: none"> <li>• -95 dBm sensitivity in 1 Mbps Bluetooth low energy mode</li> <li>• -103 dBm sensitivity in 125 kbps Bluetooth low energy mode (long range)</li> <li>• -20 to +8 dBm TX power, configurable in 4 dB steps</li> </ul> </li> <li>• On-air compatible with nRF52, nRF51, nRF24L, and nRF24AP Series devices</li> <li>• Supported data rates:           <ul style="list-style-type: none"> <li>• Bluetooth 5 – 2 Mbps, 1 Mbps, 500 kbps, and 125 kbps</li> <li>• IEEE 802.15.4-2006 – 250 kbps</li> <li>• Proprietary 2.4 GHz – 2 Mbps, 1 Mbps</li> </ul> </li> <li>• Single-ended antenna output (on-chip balun)</li> <li>• 128-bit AES/ECB/CCM/AAR co-processor (on-the-fly packet encryption)</li> <li>• 4.8 mA peak current in TX (0 dBm)</li> <li>• 4.6 mA peak current in RX</li> <li>• RSSI (1 dB resolution)</li> </ul> <ul style="list-style-type: none"> <li>• Arm® Cortex®-M4 32-bit processor with FPU, 64 MHz           <ul style="list-style-type: none"> <li>• 212 EEMBC CoreMark® score running from flash memory</li> <li>• 52 µA/MHz running CoreMark from flash memory</li> <li>• Watchpoint and trace debug modules (DWT, ETM, and ITM)</li> <li>• Serial wire debug (SWD)</li> </ul> </li> <li>• Rich set of security features           <ul style="list-style-type: none"> <li>• Arm TrustZone® CryptoCell™ 310 security subsystem               <ul style="list-style-type: none"> <li>• NIST SP800-90A and SP800-90B compliant random number generator</li> <li>• AES-128 – ECB, CBC, CMAC/CBC-MAC, CTR, CCM/CCM*</li> <li>• ChaCha20/Poly1305 AEAD supporting 128- and 256-bit key size</li> <li>• SHA-1 and SHA-2 up to 256 bit</li> <li>• Keyed-hash message authentication code (HMAC)</li> <li>• RSA up to 2048-bit key size</li> <li>• SRP up to 3072-bit key size</li> <li>• ECC support for most used curves, including P-256 (secp256r1) and Ed25519/Curve25519</li> <li>• Application key management using derived key model</li> </ul> </li> <li>• Secure boot ready               <ul style="list-style-type: none"> <li>• Flash access control list (ACL)</li> <li>• Root-of-trust (RoT)</li> <li>• Debug control and configuration</li> <li>• Access port protection (CTRL-AP)</li> </ul> </li> <li>• Secure erase</li> </ul> </li> </ul> <ul style="list-style-type: none"> <li>• Flexible power management           <ul style="list-style-type: none"> <li>• 1.7 V to 5.5 V supply voltage range</li> <li>• On-chip DC/DC and LDO regulators with automated low current modes</li> <li>• 1.8 V to 3.3 V regulated supply for external components</li> <li>• Automated peripheral power management</li> <li>• Fast wake-up using 64 MHz internal oscillator</li> <li>• 0.4 µA at 3 V in System OFF mode, no RAM retention</li> <li>• 1.5 µA at 3 V in System ON mode, no RAM retention, wake on RTC</li> </ul> </li> <li>• 1 MB flash and 256 kB RAM</li> <li>• Advanced on-chip interfaces           <ul style="list-style-type: none"> <li>• USB 2.0 full speed (12 Mbps) controller</li> <li>• QSPI 32 MHz interface</li> <li>• High-speed 32 MHz SPI</li> <li>• Type 2 near field communication (NFC-A) tag with wake-on field               <ul style="list-style-type: none"> <li>• Touch-to-pair support</li> </ul> </li> <li>• Programmable peripheral interconnect (PPI)</li> <li>• 48 general purpose I/O pins</li> <li>• EasyDMA automated data transfer between memory and peripherals</li> </ul> </li> <li>• Nordic SoftDevice ready with support for concurrent multiprotocol</li> <li>• 12-bit, 200 ksps ADC – 8 configurable channels with programmable gain</li> <li>• 64 level comparator</li> <li>• 15 level low-power comparator with wake-up from System OFF mode</li> <li>• Temperature sensor</li> <li>• Four 4 channel pulse width modulator (PWM) units with EasyDMA</li> <li>• Audio peripherals – I²S, digital microphone interface (PDM)</li> <li>• Five 32-bit timers with counter mode</li> <li>• Up to four SPI masters/three SPI slaves with EasyDMA</li> <li>• Up to two I²C compatible two-wire master/slave</li> <li>• Two UART (CTS/RTS) with EasyDMA</li> <li>• Quadrature decoder (QDEC)</li> <li>• Three real-time counters (RTC)</li> <li>• Single crystal operation</li> <li>• Package variants           <ul style="list-style-type: none"> <li>• aQFN73™ package, 7 x 7 mm</li> <li>• QFN48 package, 6 x 6 mm</li> <li>• WLCSP package, 3.544 x 3.607 mm</li> </ul> </li> </ul>	

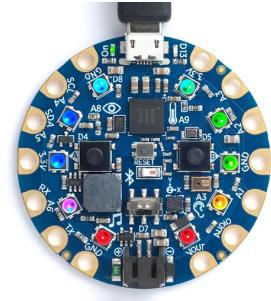
Hình 2.2: Các tính năng Nordic nRF52840

ở băng tần 2.4 GHz và bộ xử lý trung tâm Arm Cortex-M4F chạy ở xung nhịp 64 MHz, kèm bộ xử lý dấu phẩy động (FPU). Vi xử lý này được trang bị bộ nhớ 1 MB Flash và 256 kb RAM phù hợp để triển khai các mô hình cho bài toán phân loại tư thế ngủ. Ngoài khả năng hoạt động trong dải điện áp rộng từ +1.7 V đến +5.5 V (tương thích với nguồn pin và USB), nRF52840 còn cung cấp các giao tiếp ngoại vi phong phú: hai giao diện I2C, bốn SPI chủ (master), ba SPI tớ (slave), bốn kênh điều chế xung (PWM) hỗ trợ EasyDMA, cùng với bộ định thời 32-bit, phù hợp cho các ứng dụng đòi hỏi xử lý thời gian thực chính xác.

Đặc biệt, nRF52840 có một hệ sinh thái phần mềm đi kèm, bao gồm bộ công cụ phát triển (SDK) của Nordic Semiconductor và nền tảng TensorFlow Lite for Microcontrollers, giúp rút ngắn thời gian phát triển và triển khai hệ thống TinyML trực tiếp lên chip [101]. Hiện nay, nRF52840 cũng tích hợp trong nhiều bộ kit phát triển thương mại, như Adafruit Circuit Playground Bluefruit, SparkFun Pro nRF52840 Mini, Seeed Studio XIAO. Các bộ kit này cung cấp tài nguyên phần cứng và thư viện mã nguồn mở, hỗ trợ kết nối BLE, USB, GPIO, SPI, I<sup>2</sup>C, cùng khả năng lập trình trực tiếp qua môi trường Arduino IDE, PlatformIO hoặc Zephyr RTOS. Nhờ đó, nRF52840 càng trở nên phổ biến trong các nghiên cứu và dự án về IoT, thiết bị đeo thông minh và học máy tại biên.



(a) Seeed Studio XIAO

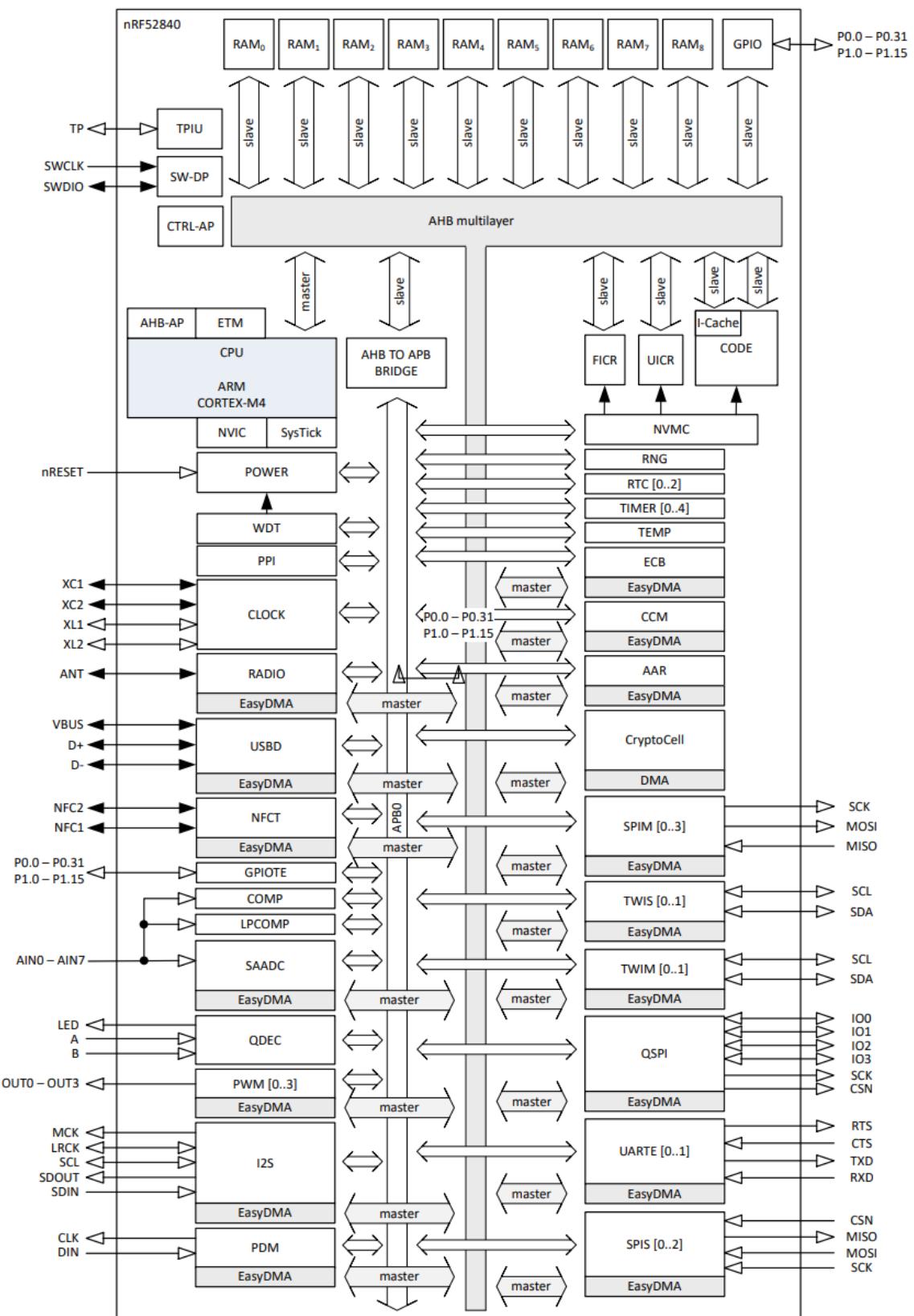


(b) Adafruit Circuit Playground Bluefruit

Hình 2.4: Một số bộ kit phát triển dựa trên vi điều khiển Nordic nRF52840

### 2.1.2. Cảm biến

Trong quá trình ngủ, các chuyển động thân thể chủ yếu là chuyển động chậm, với biên độ nhỏ và ít mang tính đột ngột [102]. Trong giai đoạn ngủ REM, cơ thể gần như bất động. Do đó, việc ghi nhận chính xác các thay đổi tư thế ngủ đòi hỏi cảm biến có độ nhạy cao. Như đã trình bày trong Chương I, các cảm biến gia tốc MEMS sử dụng



Hình 2.3: Sơ đồ khối Nordic NRF52840

nguyên lý điện dung hiện đang được ứng dụng rộng rãi trong giám sát tư thế và chuyển động khi ngủ nhờ vào đặc điểm nổi bật là tần số lấy mẫu phù hợp và đặc biệt là độ nhạy cao với mọi loại chuyển động.

Sau khi đã khảo sát, cảm biến gia tốc Bosch BMI270 được lựa chọn cho hệ thống nhận diện tư thế ngủ với các lý do sau:



Hình 2.5: Bosch BMI270

Thứ nhất, độ chính xác và dải đo linh hoạt: BMI270 tích hợp gia tốc kế ba trục độ phân giải cao 16-bit với các dải đo linh hoạt  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  và  $\pm 16g$ , cho phép ghi nhận chính xác cả chuyển động chậm và biến độ nhỏ trong khi ngủ.

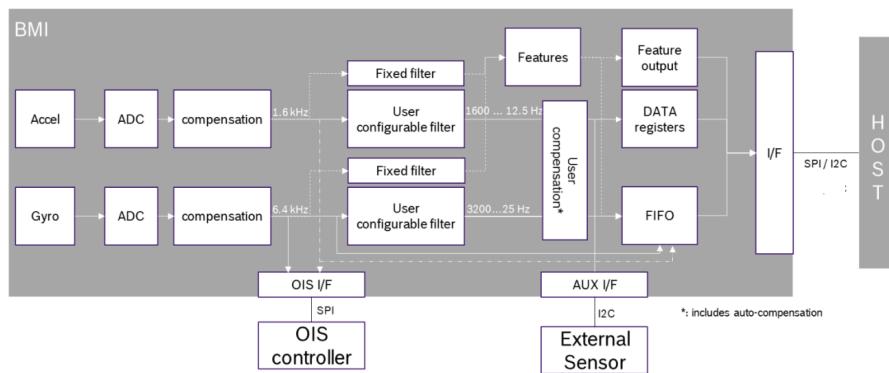
Thứ hai, mức tiêu thụ năng lượng thấp và khả năng hoạt động ổn định: cảm biến tiêu thụ trung bình chỉ khoảng  $685 \mu A$  khi hoạt động bình thường, đồng thời được trang bị bộ quản lý năng lượng tích hợp hỗ trợ nhiều chế độ tiết kiệm điện khác nhau. Điều này cho phép thiết bị hoạt động lâu dài trên nền tảng phần cứng công suất thấp mà không ảnh hưởng đến hiệu năng đo.

Thứ ba, tốc độ lấy mẫu và thời gian đáp ứng nhanh: BMI270 hỗ trợ tốc độ lấy mẫu (ODR) từ 0.7 Hz đến 1.6 kHz cho gia tốc kế, phù hợp bài toán phân loại giấc ngủ yêu cầu tần số lấy mẫu thấp (10 Hz).

Thứ tư, khả năng chống nhiễu và hiệu chuẩn tự động: cảm biến có cơ chế bù sai số

và bù độ nhạy theo thời gian thực, giúp giảm sai lệch khi hoạt động lâu dài. Tính năng bù sai lệch cho gia tốc kế đảm bảo độ ổn định cao của dữ liệu đo. Ngoài ra, cảm biến còn tích hợp thêm bộ lọc thông thấp.

Cuối cùng, tích hợp tính năng thông minh được lập trình trong các thanh ghi riêng được Google chứng nhận tương thích với hệ điều hành Wear OS và hỗ trợ phát hiện chuyển động tự động: BMI270 có khả năng phát hiện các trạng thái như phát hiện có chuyển động (any motion), không có chuyển động (no motion) hoặc di chuyển các thiết bị đeo ở cổ tay (wrist wear wakeup). Các tính năng này có thể được tận dụng để giảm tải tính toán cho bộ vi điều khiển, chỉ kích hoạt mô hình học máy khi có thay đổi tư thế đáng kể, qua đó tiết kiệm năng lượng và tăng tuổi thọ pin

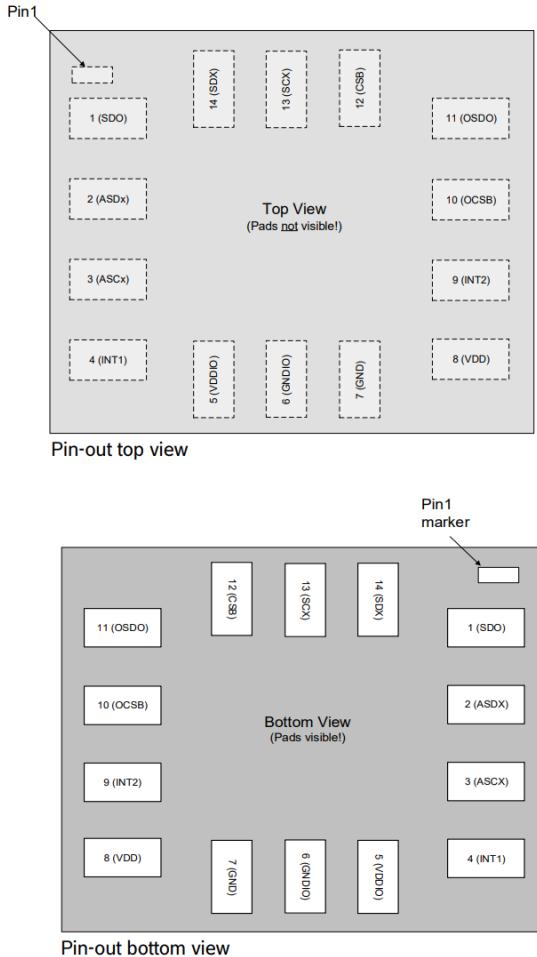


Hình 2.6: Sơ đồ khái niệm Bosch BMI270

### 2.1.3. Bluetooth năng lượng thấp

Với mục tiêu tối ưu hóa năng lượng và đảm bảo khả năng hoạt động lâu dài cho thiết bị đeo sử dụng pin, BLE được lựa chọn làm chuẩn kết nối không dây chính trong hệ thống phần cứng. So với các giao thức khác, BLE tỏ ra vượt trội nhờ mức tiêu thụ năng lượng rất thấp, tốc độ khởi tạo kết nối nhanh và khả năng tương thích rộng rãi với hầu hết các thiết bị di động hiện nay [103].

Các công nghệ truyền thông không dây khác đều bộc lộ những hạn chế đối với thiết bị phân loại tư thế ngủ. Wi-Fi, có ưu thế về băng thông, lại tiêu thụ năng lượng cao và phụ thuộc vào internet. Bluetooth cổ điển, vốn được thiết kế cho các ứng dụng truyền tải dữ liệu dung lượng lớn như âm thanh, cũng đòi hỏi năng lượng cao hơn nhiều so với BLE. ZigBee, hiệu quả trong các mạng cảm biến nhà thông minh nhờ cấu trúc lưới, lại thiếu sự hỗ trợ trên điện thoại và đòi hỏi gateway riêng.



Hình 2.7: Sơ đồ chân Bosch BMI270

Ngoài ra, BLE còn được hỗ trợ trên vi điều khiển nRF52840 đã chọn ở phần trên. Nhờ sự kết hợp này, hệ thống có thể duy trì khả năng truyền dữ liệu ổn định, tiết kiệm năng lượng, và đáp ứng tốt yêu cầu hoạt động liên tục trong quá trình giám sát tư thế ngủ.

### Các thành phần, chức năng

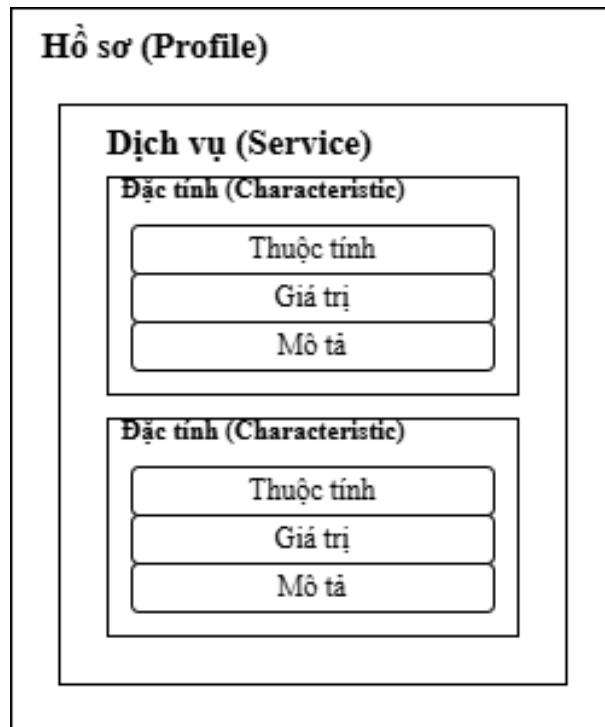
BLE là giao thức kết nối không dây được thiết kế chuyên biệt cho các ứng dụng năng lượng thấp, hoạt động ở băng tần ISM 2.4 GHz, hỗ trợ thông lượng ứng dụng lên đến 1.4 Mbps [104]. BLE hiện được hỗ trợ phổ biến trên hầu hết các hệ điều hành như iOS, Android, macOS, Windows và Linux, cũng như trong các thiết bị di động hiện đại.

Về mặt bảo mật, BLE tích hợp các cơ chế mã hóa và xác thực nhằm đảm bảo tính bí

	nRF52805	nRF52810	nRF52811	nRF52820	nRF52832	nRF52833	nRF52840	nRF5340
Bluetooth 5.3	X	X	X	X	X	X	X	X
Bluetooth 2 Mpbs	X	X	X	X	X	X	X	X
Bluetooth Long Range			X	X		X	X	X
Bluetooth Direction Finding			X	X		X		X
Bluetooth LE Audio								X
Bluetooth mesh			X	X	X	X		X
Thread		X	X		X	X	X	X
Zigbee			X		X	X		X
Matter							X	X

Hình 2.8: Các kiểu kết nối không dây trong họ chip nRF52

mật, toàn vẹn của dữ liệu. Bluetooth 5 là bước phát triển đột phá tiếp theo kể từ khi BLE được giới thiệu trong chuẩn Bluetooth 4.0, mang đến hàng loạt cải tiến đáng kể giúp mở rộng phạm vi ứng dụng và nâng cao hiệu suất hệ thống. Quan trọng hơn, chế độ này còn giúp giảm đáng kể mức tiêu thụ năng lượng. Từ đó kéo dài thời gian hoạt động của thiết bị hoặc cho phép sử dụng các nguồn năng lượng nhỏ hơn [105]. Đặc biệt, chế độ Long Range mở rộng đáng kể phạm vi truyền thông của BLE, cho phép các thiết bị duy trì kết nối trong toàn bộ không gian trong nhà.



Hình 2.9: Cấu trúc của GATT

BLE tổ chức logic giao tiếp dựa trên mô hình hồ sơ thuộc tính tổng quát Generic Attribute Profile - Đặc điểm cấu hình chung (GATT). GATT quy định cách hai thiết bị BLE trao đổi dữ liệu thông qua các đơn vị logic: dịch vụ (services) và đặc tính (characteristics). Giao thức nền tảng là Attribute Protocol (ATT). Mỗi dịch vụ, đặc tính được định danh bằng mã định danh duy nhất (UUID) 16-bit hoặc 128-bit, với quyền truy cập như chỉ đọc, chỉ ghi, hoặc hỗ trợ thông báo.

Một điểm quan trọng trong mô hình GATT là tính kết nối độc quyền: tại một thời điểm, thiết bị ngoại vi chỉ có thể duy trì một kết nối duy nhất với thiết bị trung tâm. Khi kết nối được thiết lập, thiết bị ngừng quảng cáo, điều này hạn chế khả năng kết nối đồng thời từ nhiều thiết bị.

Ngoài ra, vi xử lý nRF52840 còn hỗ trợ Bluetooth Mesh, cho phép thiết lập mang lưới nhiều nút. Mỗi nút trong mạng có thể đóng vai trò chuyển tiếp, cho phép dữ liệu lan truyền đến các vùng rộng hơn theo mô hình phân tán. Điều này phù hợp với các ứng dụng IoT quy mô đa cảm biến đặt tại nhiều nơi khác nhau.

#### **2.1.4. Thiết kế mạch**

Sau khi đã lựa chọn được vi điều khiển, cảm biến của hệ thống phần cứng phân loại tư thế ngủ. Trong phần này, tác giả sẽ trình bày về thiết kế sơ đồ nguyên lý cho mạch thông qua những câu hỏi: Các thành phần trong cần thiết để phục vụ cho bài toán phân loại tư thế ngủ là gì? Chúng được kết nối với nhau như thế nào? và chi tiết từng thành phần đó là gì?

Từ đó, tác giả từng bước nghiên cứu và hoàn thiện thiết kế sơ đồ mạch.

##### **Mục tiêu**

Bám sát vào yêu cầu đã trình bày từ các phần trước là phần cứng nhỏ gọn, tính di động cao, tiêu thụ năng lượng ít, giá thành phù hợp tác giả đã nghiên cứu và đưa ra hệ thống cần các thành phần sau: 01) Khối điều khiển: có chức năng xử lý toàn bộ logic trong mạch, kết nối được với các thiết bị khác như điện thoại thông qua BLE và đặc biệt, đủ hiệu năng để triển khai các mô hình học máy. 02) Khối cảm biến: có nhiệm vụ thu thập dữ liệu sinh lý bao gồm 2 cảm biến chính là gia tốc (đã được trình bày các phần trên) và thêm cảm biến âm thanh để phục vụ bài toán xác định tiếng ngáy của nhóm. 03) Khối nguồn: có nhiệm vụ ổn áp về đúng dải điện áp tương thích, lọc nhiễu, bảo vệ dòng 04) Khối nạp và gỡ lỗi: có nhiệm vụ nạp mã chương trình vào vi điều khiển 05) Khối hiển thị: có nhiệm vụ đưa ra thông báo khi có phát hiện bất thường. Kích thước của mạch là hình tròn, có đường kính không quá 4 cm. Đây chính là những mục tiêu điểm tiên quyết để coi là đạt được mục tiêu trong phần này.

##### **Khối điều khiển**

Với yêu cầu quan trọng đối với vi điều khiển nRF52840 được chọn hoàn toàn có thể đáp ứng các tiêu chí đã nêu trong phần 2.1.1. để làm vi điều khiển của mạch.

Tuy nhiên để đáp ứng đủ các yêu cầu của khôi điều khiển thì vẫn đang thiếu các thành phần như khôi thu phát tín hiệu vô tuyến bao gồm ăng ten, mạch phôi hợp trở khánh, bộ dao động tinh thể và bộ đếm định thời nhằm duy trì hoạt động ổn định trong các chế độ năng lượng thấp.

Sau khi tìm hiểu, tác giả quyết định lựa chọn module U-blox NINA-B306 để đảm nhiệm các nhiệm vụ của khôi điều khiển [106].

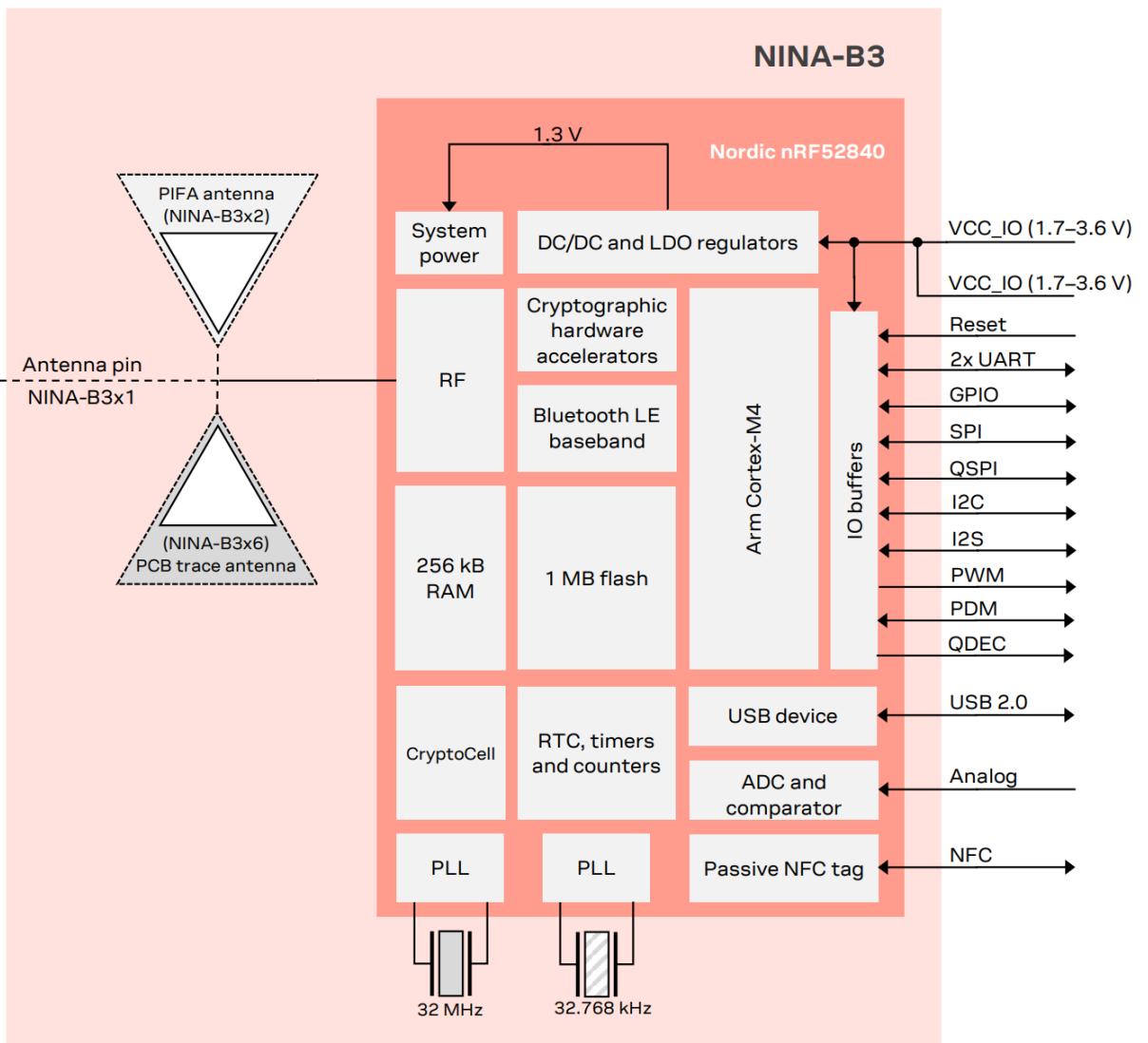


Hình 2.10: U-blox NINA-B306

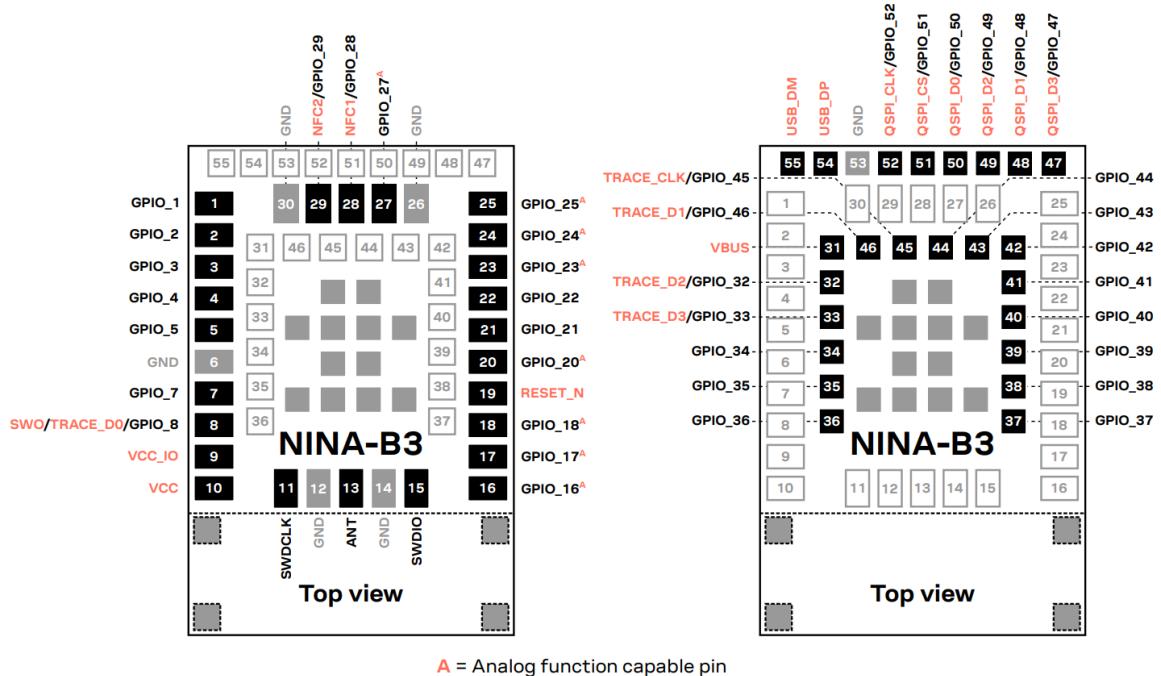
Mô-đun NINA-B306 được phát triển dựa trên vi điều khiển Nordic nRF52840, tích hợp ăng-ten PCB nội bộ được cấp phép từ Abracon ngay trên bảng mạch của mô-đun Hifnh 2.11. Ngoài ra, NINA-B306 được thiết kế với dải điện áp hoạt động chuẩn hóa từ 1.7 V đến 3.6 V, thấp hơn so với giới hạn tối đa 5.5 V của vi điều khiển nRF52840, nhằm tối ưu hiệu suất năng lượng.

Sơ đồ chân thể hiện cấu trúc bố trí tối ưu với hơn 50 chân GPIO đa chức năng, bao gồm các kênh analog, các đường truyền debug (SWD, TRACE) và các chân nguồn được phân bổ đối xứng, góp phần tăng cường độ ổn định tín hiệu và giảm nhiễu điện từ

Hình 2.12.



Hình 2.11: Sơ đồ khối U-blox NINA-B306



Hình 2.12: Sơ đồ chân U-blox NINA-B306

## Khối cảm biến

Như đã đề cập chi tiết ở phần 2.1.2., cảm biến gia tốc Bosch BMi270 được lựa chọn để thu thập tín hiệu gia tốc cho bài toán phân loại tư thế ngủ.

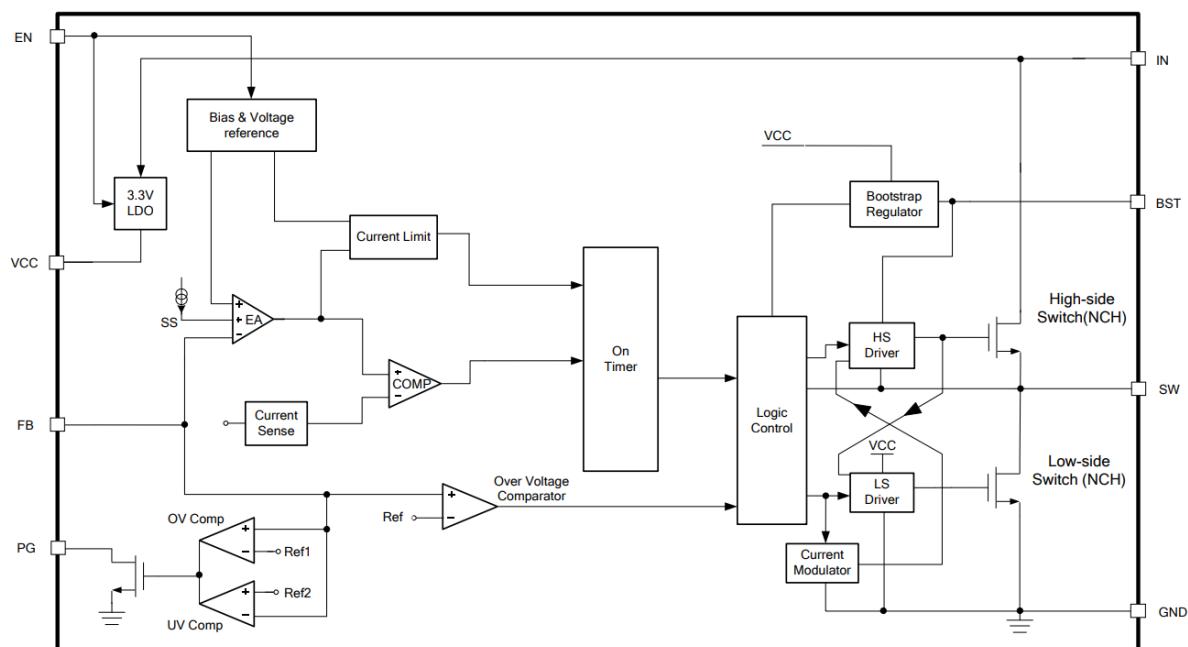
Ngoài ra, việc hướng tới việc xác định chứng ngưng thở khi ngủ OSA thì nhóm có bổ sung thêm cảm biến âm thanh STMicroelectronics MP34DT06J. Cảm biến MP34DT06J là microphone MEMS kỹ thuật số định hướng đa hướng có kích thước siêu nhỏ gọn và tiêu thụ năng lượng thấp. Ngoài ra MP34DT06J là cảm biến số có tỷ lệ tín hiệu trên nhiễu (SNR) 64 dB và độ nhạy  $-26 \text{ dBFS} \pm 1 \text{ dB}$ , đảm bảo chất lượng thu âm ổn định. Thiết bị được đóng gói theo chuẩn SMD top-port, có che chắn EMI (Electromagnetic Interference), và được đảm bảo hoạt động ổn định trong dải nhiệt độ mở rộng từ  $-40^\circ\text{C}$  đến  $+85^\circ\text{C}$  và kèm thêm điện áp đầu vào 1.8V hoàn toàn phù hợp với các thông số của vi điều khiển.

Tuy nhiên, trong khuôn khổ luận văn này tác giả không đề cập sâu vào cảm biến này để tập trung tối đa vào dữ liệu cảm biến gia tốc cho bài toán phân loại tư thế ngủ.

## **Khối nguồn, nạp và gỡ lỗi, hiển thị**

Ba khối chức năng này được thiết kế nhằm đảm bảo sự ổn định và khả năng vận hành tin cậy của toàn bộ hệ thống phần cứng.

Đối với khói nguồn, hệ thống được xác định có hai nguồn cung cấp chính gồm nguồn từ cổng USB và nguồn từ pin Li-Polymer. Điện áp đầu vào của toàn mạch được xác định ở mức 3.3 V để tương thích với các linh kiện và vi điều khiển. Sau khi nghiên cứu, tác giả sử dụng bộ chuyển đổi hạ áp (step-down converter) tích hợp mạch điều khiển MP2322GQH, cho phép giảm điện áp đầu vào từ 5 V (USB) hoặc 3.7 V (pin Li-Polymer) xuống mức ổn định 3.3 V với hiệu suất cao.



Hình 2.13: Sơ đồ khói IC MP2322GQH, dải đầu vào 3 V - 22 V, dòng tải tối đa 1 A, hiệu suất cao, tần số chuyển mạch 1.25 MHz

Bên cạnh đó, nguồn USB còn được bổ sung các thành phần hỗ trợ như diode bảo vệ ngược dòng, tụ lọc nhiễu cao tần và bù tín hiệu, cùng điện trở xả tĩnh điện, nhằm đảm bảo an toàn và giảm nhiễu điện từ trong quá trình vận hành.

Đối với khói hiển thị, hệ thống sử dụng bốn đèn LED khác màu bố trí tại rìa của mạch, có nhiệm vụ thông báo kết quả tư thế được phân loại, trạng thái hoạt động của hệ thống như nguồn, kết nối, truyền dữ liệu hoặc lỗi, giúp người dùng và kỹ sư dễ dàng theo dõi tình trạng thiết bị.

Với khối nạp và gỡ lỗi, tác giả triển khai hai phương thức nạp bao gồm: (1) Qua Serial Wire Debug (SWD); (2) Cơ chế Bootloader thông qua kết nối USB. Sự kết hợp này giúp nâng cao tính linh hoạt và khả năng mở rộng trong quá trình phát triển phần mềm, đồng thời giảm thiểu thời gian thử nghiệm.

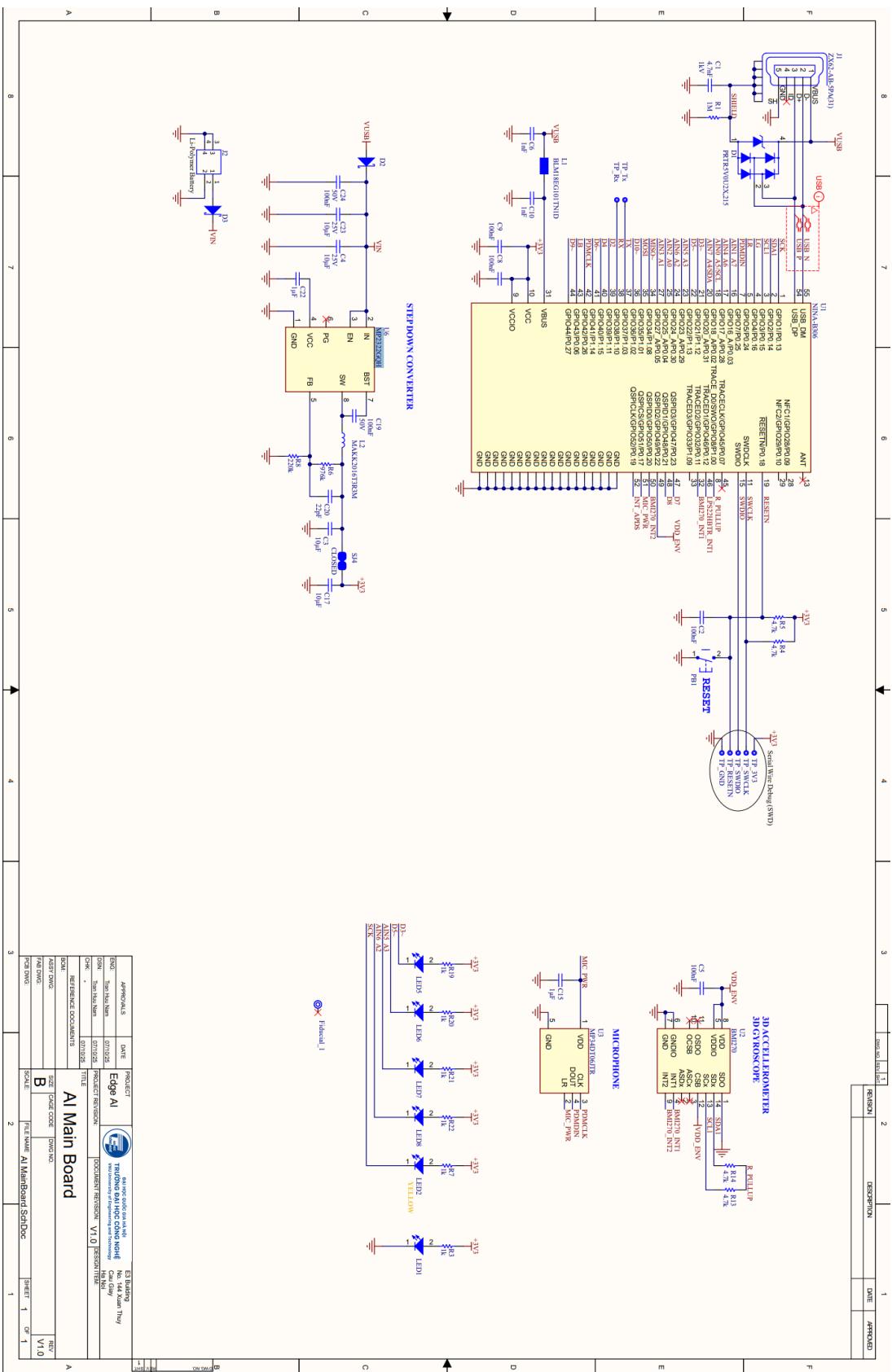
## Sơ đồ mạch

Sau khi xác định các linh kiện, các yêu cầu về điện áp, kiểu dữ liệu đầu vào, đầu ra của các thành phần kể trên, tác giả tiến hành vẽ sơ đồ mạch thông qua phần mềm Altium Designer.

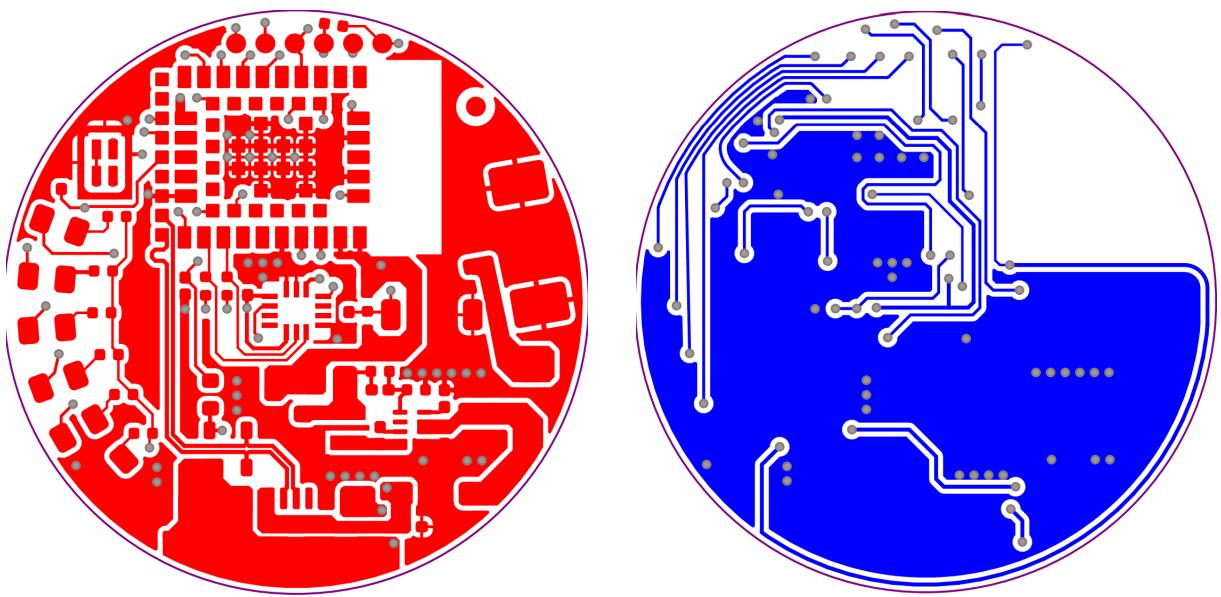
Mạch được thiết kế hình tròn kích thước đường kính 3 cm, 2 lớp đúng với định hướng của nhóm.

Kết quả được thể hiện tại Hình 2.14

Sau khi hoàn thiện thiết kế, tác giả thực hiện đặt in mạch, hàn lắp các linh kiện theo sơ đồ mạch, và tiến hành kiểm tra hoạt động của các khối chức năng trên bo mạch để đánh giá tính ổn định và độ tin cậy của hệ thống.



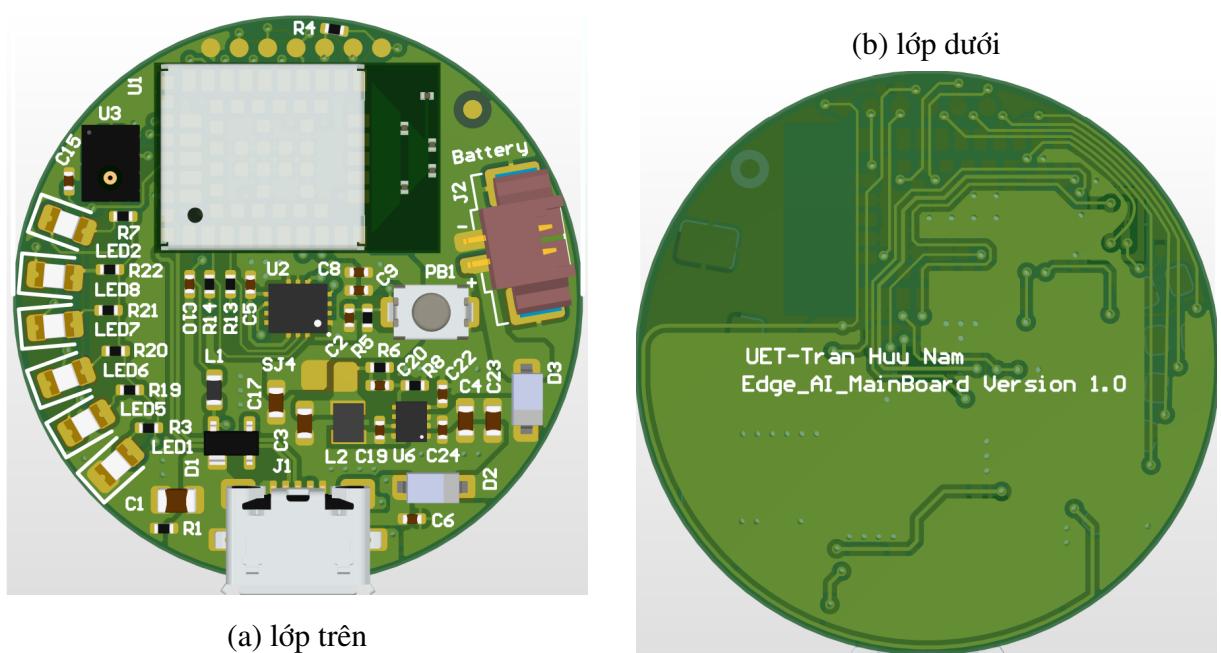
Hình 2.14: Sơ đồ mạch



(a) lớp trên

(b) lớp dưới

Hình 2.15: Bố trí mạch in hai mặt của bo mạch



Hình 2.16: Mô hình mạch 3D

## 2.2. Hệ thống phần mềm

Phần này trình bày tổng quan kiến trúc hệ thống phần mềm bao gồm: lập trình phần cứng để đọc giá trị cảm biến, cấu hình BE ; phát triển ứng dụng di động làm cầu nối giữa phần cứng và hệ thống đám mây, cùng với máy chủ và cơ sở dữ liệu lưu trữ dữ liệu. Nội dung cũng đề cập đến các yêu cầu chức năng, phi chức năng và thiết kế hệ thống ở mức cao nhằm đảm bảo khả năng triển khai thực tế và mở rộng trong tương lai.

### 2.2.1. Lập trình phần cứng

Ở giai đoạn này, tác giả tiến hành lập trình phần cứng bằng cách sử dụng các framework và thư viện do hãng cung cấp. Cách tiếp cận này cho phép khai thác hầu hết các phương thức tương tác với phần cứng, cộng đồng hỗ trợ lớn đồng thời giúp rút ngắn đáng kể thời gian phát triển và giảm thiểu sai sót trong quá trình lập trình. Tuy nhiên, để có thể kiểm soát toàn diện các chức năng của hệ thống cũng như tối ưu hóa mức tiêu thụ năng lượng, trong tương lai tác giả dự định sẽ chuyển sang phương pháp lập trình ở mức thanh ghi.

#### Cấu hình vi xử lý và cảm biến

Bước đầu tiên để giao tiếp với phần cứng, tác giả lựa chọn phương án thông qua USB và Bootloader. Bootloader của Arduino Nano được nạp vào vi điều khiển bằng giao thức Serial Wire Debug (SWD), cho phép thiết lập môi trường lập trình ban đầu và hỗ trợ việc tải chương trình trực tiếp từ máy tính thông qua cổng USB.

Việc đọc các giá trị cảm biến thông qua thư viện SparkFun BMI270 Arduino Library [107]. Đây là thư viện mã nguồn mở do SparkFun phát triển đóng vai trò như một lớp trùu tượng, cho phép vi điều khiển giao tiếp trực tiếp với cảm biến Bosch BMI270. Thư viện cung cấp đầy đủ các hàm để cấu hình dải đo, tần số lấy mẫu và chế độ năng lượng tiêu thụ.

Mã nguồn 2.1: Cấu hình cảm biến qua thư viện SparkFun\_BMI270\_Arduino\_Library

1

2      Wire1.begin();

3

```

4   if (imu.beginI2C(BMI2_I2C_PRIM_ADDR, Wire1) != BMI2_OK) {
5     Serial.println("Error: BMI270 not detected!");
6     while (1);
7   }
8
9   struct bmi2_sens_config config;
10  config.type = BMI2_ACCEL;
11  config.cfg.acc.odr = BMI2_ACC_ODR_12_5HZ;
12  config.cfg.acc.range = BMI2_ACC_RANGE_2G;
13  config.cfg.acc.bwp = BMI2_ACC_NORMAL_AVG4;
14  config.cfg.acc.filter_perf = BMI2_POWER_OPT_MODE;
15
16  if (imu.setConfig(config) == BMI2_OK)
17    Serial.println("Accelerometer configured successfully!");
18  else
19    Serial.println("Failed to configure accelerometer!");
20
21  if (imu.getConfig(&config) == BMI2_OK) {
22    Serial.println("Current Accelerometer Config:");
23    Serial.print(" Range setting = ");
24    switch (config.cfg.acc.range) {
25      case BMI2_ACC_RANGE_2G: Serial.println("$\\pm$2g"); break;
26      case BMI2_ACC_RANGE_4G: Serial.println("$\\pm$4g"); break;
27      case BMI2_ACC_RANGE_8G: Serial.println("$\\pm$8g"); break;
28      case BMI2_ACC_RANGE_16G: Serial.println("$\\pm$16g"); break;
29      default: Serial.println("Unknown"); break;
30    }
31
32    Serial.print(" ODR setting = ");
33    switch (config.cfg.acc.odr) {
34      case BMI2_ACC_ODR_0_78HZ: Serial.println("0.78 Hz"); break;
35      case BMI2_ACC_ODR_1_56HZ: Serial.println("1.56 Hz"); break;
36      case BMI2_ACC_ODR_3_12HZ: Serial.println("3.12 Hz"); break;
37      case BMI2_ACC_ODR_6_25HZ: Serial.println("6.25 Hz"); break;
38      case BMI2_ACC_ODR_12_5HZ: Serial.println("12.5 Hz"); break;
39      case BMI2_ACC_ODR_25HZ: Serial.println("25 Hz"); break;

```

```
40     case BMI2_ACC_ODR_50HZ: Serial.println("50 Hz"); break;
41     case BMI2_ACC_ODR_100HZ: Serial.println("100 Hz"); break;
42     default: Serial.println("Unknown"); break;
43 }
44 } else {
45     Serial.println();
46 }
```

---

Phần cấu hình được thể hiện trong mã nguồn 2.1 Việc gọi hàm `imu.beginI2C()` nhằm thiết lập kênh truyền I<sup>2</sup>C và xác nhận sự hiện diện của thiết bị. Tần số lấy mẫu được đặt ở 12,5 Hz và dải đo ở mức ±2g nhằm tối ưu độ phân giải cho các dao động nhỏ khi ngủ ở người. Việc thiết lập cấu hình qua hàm `setConfig()` và xác thực lại bằng `getConfig()` thể hiện nguyên tắc kiểm chứng hai chiều nhằm chắc chắn các cấu hình đã được đặt trên phần cứng.

## Cấu hình BLE

Quá trình khởi tạo, kết nối và truyền lên thiết bị trung tâm bao gồm các bước sau: bao gồm: tạo bản tin quảng bá (advertising), thiết lập cơ chế kết nối/ngắt kết nối, cũng như định nghĩa UUID của dịch vụ và đặc tính theo mô hình GATT.

Mã nguồn 2.2: Chương trình thiết lập kết nối, truyền dữ liệu qua BLE

---

```
1 const char* deviceServiceUuid = "19b10000-e8f2-537e-4f6c-d104768a1214";
2 const char* deviceServiceCharacteristicUuid =
3   "19b10001-e8f2-537e-4f6c-d104768a1214";
4
5 BLEService accelerometerService(deviceServiceUuid);
6 BLECharacteristic accelerometerCharacteristic(
7   deviceServiceCharacteristicUuid,
8   BLERead | BLEWrite | BLENotify,
9   9, 3
10 );
11
12 void setup() {
13   Serial.begin(9600);
14   Serial.println("Started");
15   Wire1.begin();
16
17   if (!BLE.begin()) {
18     Serial.println("- Starting Bluetooth Low Energy module failed!");
19     while (1);
20   }
21
22   BLE.setLocalName("Master_2025_BLE");
23   BLE.setDeviceName("Master_2025_BLE");
24   BLE.setAdvertisedService(accelerometerService);
25
26   accelerometerService.addCharacteristic(accelerometerCharacteristic);
27   BLE.addService(accelerometerService);
28   accelerometerCharacteristic.canSubscribe();
29   accelerometerCharacteristic.subscribed();
```

```

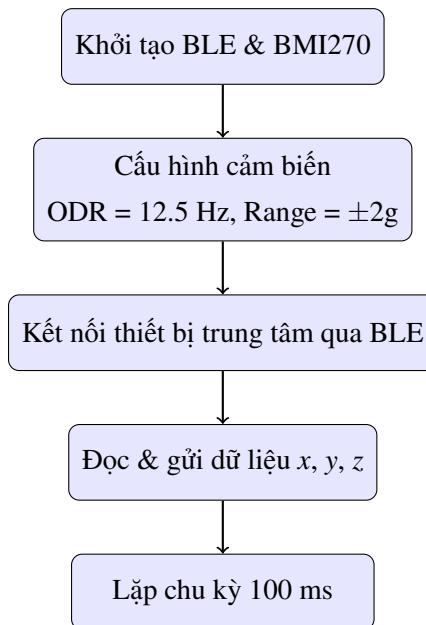
30
31     BLE.advertise();
32
33     Serial.println("Nano 33 BLE (Peripheral Device)");
34
35     **SETUP accelerometer**
36 }
37
38 static float x, y, z;
39
40 void loop() {
41     BLEDevice central = BLE.central();
42     Serial.println("- Discovering central device..."); 
43     delay(500);
44
45     if (central) {
46         Serial.println("* Connected to central device!");
47         Serial.print("* Device MAC address: ");
48         Serial.println(central.address());
49         while (central.connected()) {
50             imu.getSensorData();
51             x = imu.data.accelX;
52             y = imu.data.accelY;
53             z = imu.data.accelZ;
54
55             Serial.print(x);
56             Serial.print('\t');
57             Serial.print(y);
58             Serial.print('\t');
59             Serial.println(z);
60
61             uint8_t accelData[9] = {
62                 (x >= 0) ? 1 : 0,
63                 abs((int)x),
64                 abs((int)(x * 100) % 100),
65                 (y >= 0) ? 1 : 0,

```

```

66     abs((int)y),
67     abs((int)(y * 100) % 100),
68     (z >= 0) ? 1 : 0,
69     abs((int)z),
70     abs((int)(z * 100) % 100)
71 };
72
73     accelerometerCharacteristic.writeValue(accelData,
74         sizeof(accelData));
75     delay(100);
76 }
77 }
```

---



Hình 2.17: Sơ đồ tóm tắt quy trình hoạt động BLE-BMI270 của đoạn mã nguồn 2.2

Thứ nhất, khởi tạo BLE, kích hoạt truyền thông BLE với tên hiển thị là Master\_2025\_BLE, thiết lập cấu trúc dịch vụ GATT với hai UUID cho service và characteristic. Trong đó characteristic được phép đọc, ghi và phát tín hiệu.

Thứ hai, khởi tạo cảm biến BMI270 và cấu hình.

Thứ ba, tạo kết nối với thiết bị trung tâm, khi tín hiệu quảng bá được phản hồi. Sau

khi kết nối thành công, phần cứng lấy được thông tin của thiết bị trung tâm ở đây là ứng dụng điện thoại. Đây một bước xác thực cần thiết để chỉ cho phép thiết bị trung tâm nào được lập trình sẵn.

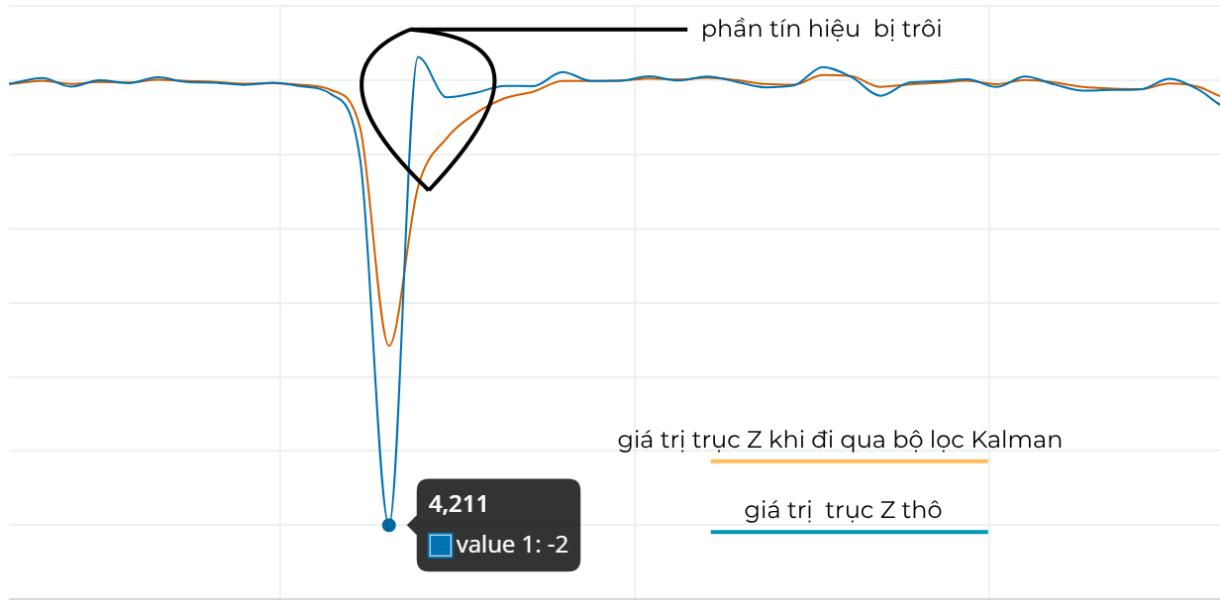
Thứ tư, thu nhận dữ liệu cảm biến ba trục (x, y, z) từ BMI270. Các giá trị gia tốc được đọc và chuẩn hóa trước khi truyền. Việc chuẩn hóa ở đây gồm tách dầu và phần nguyên - phần thập phân để giảm kích thước gói tin mà vẫn đảm bảo độ chính xác khi tái lập lại.

Cuối cùng, truyền dữ liệu định dạng byte qua BLE thông qua characteristic đã định nghĩa. Mỗi gói tin gồm chín byte chứa đầy đủ thông tin về hướng và biên độ gia tốc trên ba trục, được gửi chu kỳ 100 ms, bảo đảm tốc độ lấy mẫu ổn định (10 Hz) phục vụ phân tích tư thế trong thời gian thực.

Việc chuẩn hóa 5 giai đoạn này có ưu điểm có thể bóc tách cho mục đích khác nhau, tối ưu cho từng giai đoạn. Việc này để bảo code mạch lạc và dễ bảo trì.

## Lọc nhiễu bằng bộ lọc Kalman

Trong phần phân tích các bộ lọc cũng như từ thực nghiệm tác giả thấy, khi có những chuyển động đột ngột gia tốc cao có gây ra sai lệch giá trị thu được.



Hình 2.18: So sánh tín hiệu trục Z trước và sau bộ lọc Kalman

Hình 2.18 trình bày so sánh tín hiệu gia tốc thu được trên trục Z giữa hai trạng thái:

dữ liệu thô và dữ liệu sau khi qua bộ lọc Kalman. Có thể quan sát thấy, trong vùng được khoanh tròn, tín hiệu thô (đường màu xanh) xuất hiện một giá trị đột biến âm với biên độ lớn, thể hiện hiện tượng trôi tín hiệu tạm thời do nhiễu cảm biến hoặc rung động ngoài mong muốn.

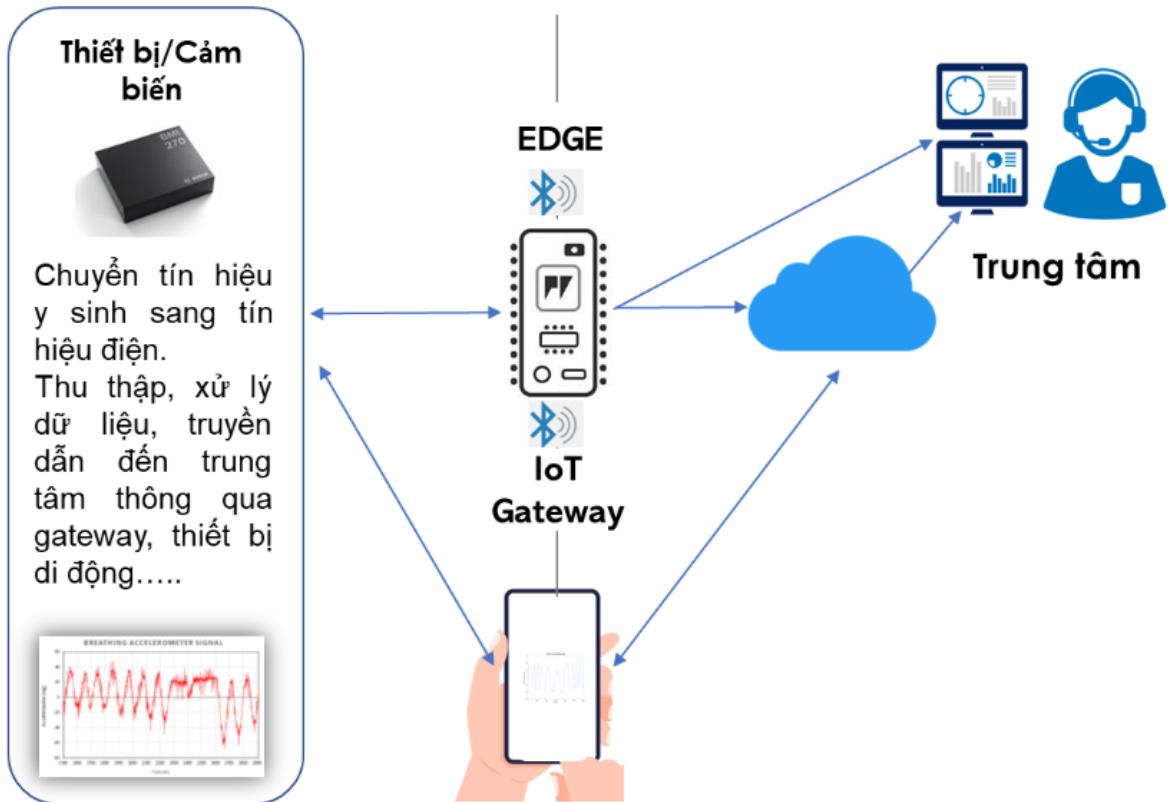
Sau khi áp dụng bộ lọc Kalman, đường tín hiệu (đường màu cam) trở nên mượt hơn và ổn định, đồng thời loại bỏ được phần lớn sai lệch ngắn hạn. Kết quả này chứng tỏ bộ lọc Kalman không chỉ thực hiện vai trò giảm nhiễu mà còn có khả năng ước lượng trạng thái thật của hệ thống thông qua mô hình dự đoán - cập nhật liên tục. Việc lọc này đặc biệt quan trọng trong bối cảnh dữ liệu thu từ cảm biến gia tốc dùng cho nhận diện tư thế ngủ, bởi vì sai số đột biến dù chỉ trong một khoảng ngắn cũng có thể dẫn đến phân loại sai tư thế trong mô hình học máy.

Ngoài ra, vùng sai lệch này cũng cho thấy hạn chế của tín hiệu thô khi cảm biến được gắn trên cơ thể người, nơi dao động nhỏ do hô hấp, chuyển động tự nhiên hoặc thay đổi điểm đặt có thể gây ra nhiễu. Việc sử dụng Kalman giúp hiệu chỉnh, đảm bảo giá trị đo được bám sát quy đạo thật của chuyển động. Tuy nhiên, bộ lọc Kalman có thể làm giảm giá trị biên tại các đỉnh. Vấn đề này được tác giả xem xét không ảnh hưởng nhiều đến toàn bộ bài toán phân loại tư thế ngủ. Lý do là vì bộ đặc trưng không sử dụng trực tiếp các giá trị đỉnh này mà là một cửa sổ mẫu và các thông kê trong cửa sổ đó.

### 2.2.2. Phần mềm thu thập, lưu trữ

Phần mềm trong nghiên cứu này không chỉ đơn thuần là công cụ trực quan hóa dữ liệu cảm biến, mà còn được thiết kế như một mắt xích trọng yếu trong toàn bộ quy trình từ thu thập, truyền tải, lưu trữ, cho đến huấn luyện và triển khai mô hình học máy. Cách tiếp cận này đảm bảo rằng dữ liệu thu nhận từ môi trường thực tế được xử lý nhất quán, có khả năng tái sử dụng và dễ dàng tích hợp với các hệ thống khác.

Hình 2.19 mô tả tổng quan các thành phần cần thiết trong luận văn và kết nối giữa chúng. Để đưa ra kết quả phân loại tư thế ngủ dữ liệu từ cảm biến có thể có 2 hướng đi. Thứ nhất, dữ liệu từ cảm biến thông qua vi điều khiển đẩy lên ứng dụng di động (BLE) sau đó được lưu trữ hoặc được suy luận trên máy chủ đám mây rồi trả về kết quả tại ứng dụng di động. Thứ hai, dữ liệu từ cảm biến sau khi được đọc từ vi điều khiển thì được suy luận ngay tại biên và hiển thị kết quả thông qua các con đèn led. Cả 2 cách đều được



Hình 2.19: Kiến trúc tổng thể của hệ thống

tác giả nghiên cứu và triển khai trong luận văn này.

## Ứng dụng di động

Được coi như 1 cổng kết nối (gateway) trong điện toán đám mây có nhiệm vụ gom các tín hiệu từ cảm biến gia tốc hoặc có thể mở rộng đa cảm biến khác, hiển thị chúng theo thời gian thực và chuyển lên máy chủ đám mây.

Dựa trên yêu cầu bài toán, tác giả xác định được các yêu cầu của phần mềm như Bảng 2.1. Ngoài ra, các yêu cầu phi chức năng cũng được tác giả chú ý như là bảo mật thông tin, thời gian phản hồi hệ thống.

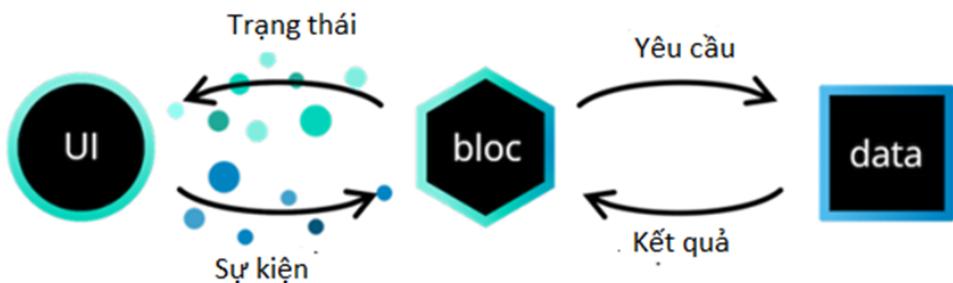
Ứng dụng di động được viết bằng ngôn ngữ DART trên nền tảng Flutter. Vì sao lại chọn Dart mà không phải là các ngôn ngữ thuận tay như Java hay Swift? Nói chung, tác giả chọn Dart trên nền tảng Flutter lý do chính là có thể tạo ra hai phiên bản trên Android và IOS cùng một bộ mã chương trình. Ngoài ra, việc khởi tạo dự án nhanh, cú pháp thân thiện với người lập trình hơn, kèm cộng đồng đông đảo làm giảm đáng kể thời gian

Bảng 2.1: Các nhóm chức năng chính của ứng dụng

Nhóm	Chức năng	Mô tả
Xác thực	Đăng ký	Người dùng đăng ký thông tin và xác thực qua email, sđt
	Đăng nhập	Sử dụng tài khoản mật khẩu đã được đăng ký để đăng nhập vào hệ thống
	Quên mật khẩu	Lấy lại tài khoản của người dùng
Ứng dụng	Kết nối Bluetooth	Kết nối ứng dụng với phần cứng thông qua BLE
	Chọn cảm biến	Lựa chọn cảm biến như (gia tốc, âm thanh)
	Câu hỏi STOPBANG, Epworth	Trả lời câu hỏi STOP BANG để đánh giá OSA và hiển thị lịch sử các lần trả lời
	Theo dõi thông tin từ cảm biến	Hiển thị dữ liệu thời gian thực thông qua các biểu đồ
	Chatbot y tế	Chatbot xây dựng trên bộ câu hỏi 2000 câu về y học giấc ngủ và công nghệ RAG

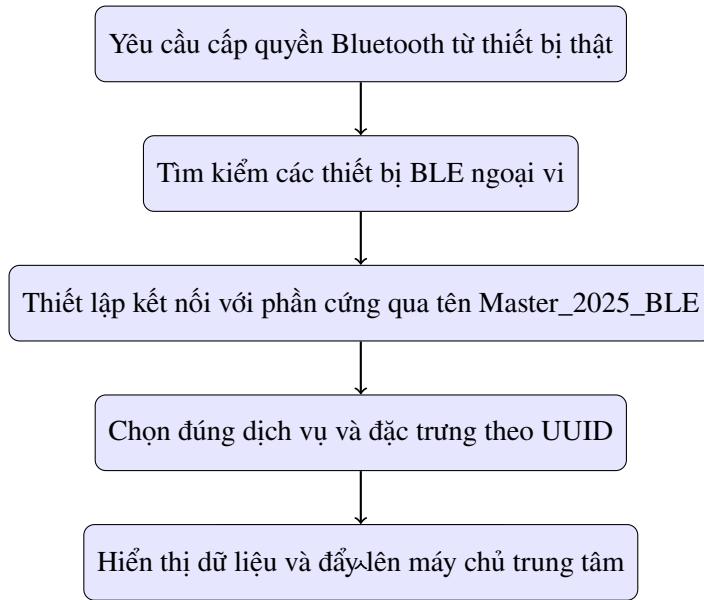
phát triển.

Để ứng dụng dễ kiểm thử và nâng cấp, tác giả tìm hiểu và lựa chọn nguyên tắc thiết kế BLoC. Nó hoạt động dựa trên nguyên tắc nhận sự kiện đầu vào từ giao diện gồm trạng thái và dữ liệu đầu vào, xử lý trong khối BLOC như gọi lên máy chủ trung tâm và trả lại dữ liệu. Qua đó, luồng dữ liệu trở nên rõ ràng. Cấu trúc tổng thể của kiến trúc BLoC gồm ba lớp chính được mô tả trong Hình 2.20.



Hình 2.20: Cấu trúc kiến trúc BLoC trong ứng dụng Flutter

Mã nguồn của toàn bộ ứng dụng di động tác giả sẽ để dưới phần mục lục. Còn trong khuôn khổ luận văn này, tác giả sẽ giải thích chức năng chính là kết nối phần cứng, hiển thị và chuyển dữ liệu lên máy chủ trung tâm.



Hình 2.21: Sơ đồ kết nối và nhận dữ liệu của ứng dụng di động

Sau khi tìm hiểu, tác giả chọn thư viện flutter\_blue\_plus làm nền tảng để kết nối BLE từ ứng dụng di động đến phần cứng [108]. Nó hỗ trợ các phương thức gồm quét, kết nối, đọc/ghi đặc tính và nhận thông báo từ thiết bị BLE trên Android, iOS, macOS, Linux và web. Ngoài ra nó không có thêm phụ thuộc ngoài Flutter, và được thiết kế cho vai trò BLE trung tâm (không hỗ trợ Bluetooth cổ điển như HC-05). Plugin miễn phí cho cá nhân, tổ chức nhỏ, và phi lợi nhuận; các công ty trên 50 nhân viên cần mua bản quyền để sử dụng thương mại.

### Mã nguồn 2.3: Mã nguồn hàm quét và kết nối các thiết bị BLE

---

```

1
2 Future onScanPressed() async {
3     try {
4         // 'withServices' is required on iOS for privacy purposes,
5         // ignored on android.
6         var withServices = [Guid("180f")]; // Battery Level Service
7         _systemDevices = await
8             FlutterBluePlus.systemDevices(withServices);
9     } catch (e) {
10         Snackbar.show(ABC.b, prettyException("System Devices Error:", e),
11                     success: false);
12         print(e);

```

```

11 }
12 try {
13     await FlutterBluePlus.startScan(
14         timeout: const Duration(seconds: 15),
15         webOptionalServices: [
16
17     ],
18 );
19 } catch (e) {
20     Snackbar.show(ABC.b, prettyException("Start Scan Error:", e),
21         success: false);
22     print(e);
23 }
24 if (mounted) {
25     setState(() {});
26 }
27 }

28

29 Future onStopPressed() async {
30     try {
31         FlutterBluePlus.stopScan();
32     } catch (e) {
33         Snackbar.show(ABC.b, prettyException("Stop Scan Error:", e),
34             success: false);
35         print(e);
36     }
37 }

38

39 void onConnectPressed(BluetoothDevice device) {
40     device.connectAndUpdateStream().catchError((e) {
41         Snackbar.show(ABC.c, prettyException("Connect Error:", e),
42             success: false);
43     });
44     MaterialPageRoute route = MaterialPageRoute(
45         builder: (context) => DeviceScreen(device: device),
46         settings: RouteSettings(name: '/DeviceScreen'));

```

```
47     Navigator.of(context).push(route);  
48 }
```

---

Mã nguồn 2.3 trình bày quy trình quét và kết nối thiết bị ngoại vi BLE trong Flutter dựa trên thư viện flutter\_blue\_plus. Hàm onScanPressed() mục đích quét thiết bị đang khả dụng thông qua dịch vụ mức pin (Battery Level) có UUID là 180F. Khi này BLoc nhận trạng thái quét, thực hiện quét và trả về danh sách mảng các thiết bị bao gồm tên hiển thị, địa chỉ MAC. Hàm onStopPressed() dừng quá trình quét, còn onConnectPressed() thiết lập kết nối với thiết bị được chọn và chuyển đến màn hình điều khiển chi tiết.

Mã nguồn 2.4 trình bày cách ứng dụng hoạt động khi người dùng chọn đúng BLE của phần cứng. Cụ thể, StreamBuilder được sử dụng để lắng nghe các thay đổi của device.services, cho phép hệ thống phản ứng tức thời khi thiết bị BLE truyền lên các dịch vụ khả dụng. Quá trình so khớp UUID giữa dịch vụ thực tế đã cài đặt ở phần cứng và hằng số định nghĩa (ACCLEROMETER\_SERVICE, ACCLEROMETER\_CHARACTION) trên ứng dụng di động cho phép xác thực logic nhằm đảm bảo rằng ứng dụng chỉ giao tiếp với dịch vụ và đặc tính cảm biến hợp lệ.

Sau khi đã khớp các thành phần BLE, ứng dụng di động có thêm nút ấn "lấy thông tin" khi này ứng dụng di động sẽ gửi 1 yêu cầu theo dõi đến phần cứng. Như đoạn mã 2.2 vi điều khiển lắng nghe được hành động theo dõi sẽ lập tức gửi các gói tin lên lại ứng dụng di động theo cấu trúc đã xác lập.

**Đến đây, tác giả đã hoàn thành quy trình kết nối và nhận dữ liệu từ phần cứng.** Tiếp theo, tác giả sẽ trình bày việc dữ liệu được đóng gói và chuyển lên phần máy chủ trung tâm.

Ngoài dữ liệu 3 trục x, y, z, tác giả xác định tập dữ liệu cần lưu thêm thời gian tạo và giá trị ID của người đang dùng. Việc này đảm bảo việc minh bạch, dễ đối chiếu và phân tích, mô phỏng lại giá trị cảm biến theo chuỗi thời gian. Mã nguồn 2.5 mô tả việc tối ưu hóa đẩy dữ liệu HTTP theo lô giúp giảm tải phía máy chủ trung tâm. Điều này tăng tính mở rộng lên hàng trăm thiết bị hoặc giá trị cảm biến khác đẩy lên cùng 1 thời điểm.

Một điểm đáng lưu ý nữa, tác giả kết hợp cùng GS.TS. Dương Quý Sỹ phát triển thêm tính năng **chatbot y học giắc ngủ** dựa trên kỹ thuật **Retrieval-Augmented Gener-**

#### Mã nguồn 2.4: Tập lệnh để tìm kiếm dịch vụ cảm biến

---

```
1 StreamBuilder<List<BluetoothService>>(
2     stream: device.services,
3     initialData: [],
4     builder: (c, snapshot) {
5         if (snapshot.data!.length > 0) {
6             isService = true;
7         }
8         BluetoothService serviceAcclerometer;
9         if (snapshot.data == null || snapshot.data!.length == 0) {
10             return Text("Please contact customer Service");
11         }
12         for (int i = 0; i < snapshot.data!.length; i++) {
13             if (snapshot.data![i].uuid.toString() ==
14                 Constants.ACCLEROMETER_SERVICE) {
15                 accelerometerService = snapshot.data![i];
16             }
17         }
18         if (accelerometerService == null) {
19             return Text("Please contact customer Service");
20         }
21         for (int i = 0;
22             i < accelerometerService!.characteristics.length;
23             i++) {
24             print(accelerometerService!.characteristics[i].uuid);
25             if (accelerometerService!.characteristics[i].uuid
26                 .toString() ==
27                 Constants.ACCLEROMETER_CHARACTION) {
28                 accelerometerCharactis =
29                     accelerometerService!.characteristics[i];
30             }
31         }
32     });
});
```

---

### Mã nguồn 2.5: Cấu trúc gói tin gửi lên máy chủ

---

```
1  {
2      "value": "
3          0.88%0.66%0.99@2025-01-01T00:00:00.000+07:00/
4          0.88%0.66%0.99@2025-01-01T00:00:01.125+07:00/
5          0.88%0.66%0.99@2025-01-01T00:00:02.532+07:00",
6      "customer": "62a5f5672ad9c724ef117d76"
7 }
```

---

**ation (RAG).** Chatbot này được xây dựng từ cơ sở dữ liệu gồm hơn 2000 câu hỏi và câu trả lời chuyên sâu liên quan đến giấc ngủ được biên tập bởi GS.TS Dương Quý Sỹ, bao gồm cả tài liệu lâm sàng, nghiên cứu khoa học và các hướng dẫn thực hành. Người dùng có thể đặt câu hỏi tự nhiên như “Tôi có nên lo nếu ngủ ngáy liên tục?” hoặc “STOP-BANG > 5 có ý nghĩa gì?”, và chatbot sẽ phản hồi dựa trên kiến thức được truy xuất từ tài liệu nền và được tổng hợp lại bằng mô hình ngôn ngữ.

Từ những phân tích nêu trên, tác giả chuẩn hóa được các bước cơ bản để từ dữ liệu ở phần cứng đến lưu trữ trên máy chủ đám mây. Đây cũng chính là các bước của hệ thống điện toán đám mây khi mọi dữ liệu, mô hình tính toán sẽ cùng một nơi trên 1 hệ thống phần cứng mạnh mẽ.

## Máy chủ đám mây

Trong phần này, tác giả sẽ trình bày các thành phần trong hệ thống máy chủ đám mây cùng với các chúng phối hợp với nhau để đưa ra phản hồi lại cho ứng dụng di động. Đầu tiên, tác giả nghiên cứu và đề xuất ba thành phần chính cho luận văn bao gồm hai phần chính: máy chủ xử lý, bộ nhớ đệm và cơ sở dữ liệu.

Để giảm thời gian phát triển, toàn bộ hệ thống máy chủ được triển khai trên hạ tầng của Amazon Web Services (AWS) . **(i) Máy chủ xử lý.** Có nhiệm vụ mở các kết nối (API) ra bên ngoài, nhận yêu cầu từ ứng dụng di động. Tác giả lựa chọn ngôn ngữ Typescript (TS) trên nền tảng NodeJS với các lý do: 1) TS cung cấp khả năng định nghĩa kiểu dữ liệu cho biến, hàm. Các lỗi đó sẽ được phát hiện trong quá trình biên dịch ngay trước khi đến quá trình chạy (run time); 2) NodeJS có nhiều thư viện hỗ trợ bảo mật,

tương tác với các thành phần khác và khả năng khởi tạo nhanh dự án. Về mặt cấu trúc mã nguồn, hệ thống được tổ chức theo mô hình phân lớp như thể hiện trong Hình 2.22.

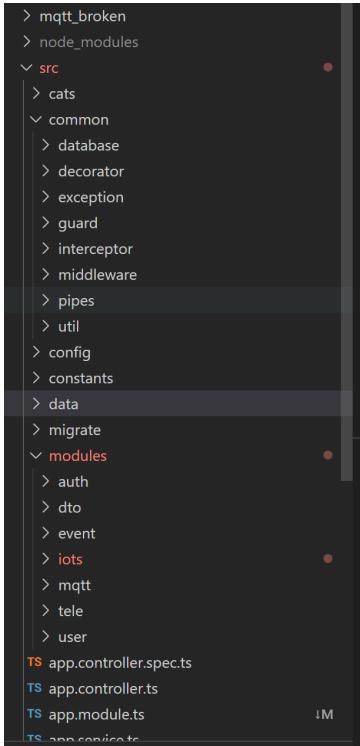
Bảng 2.2: Mô tả các thư mục chính trong cấu trúc mã nguồn máy chủ trung tâm

<b>Thư mục</b>	<b>Chức năng và nội dung chính</b>
<code>mqtt_brokers</code>	Chứa mã nguồn khởi tạo và quản lý máy chủ MQTT, chịu trách nhiệm giao tiếp theo mô hình publish-subscribe với các thiết bị đầu cuối.
<code>src/common</code>	Bao gồm các mô-đun dùng chung như xác thực người dùng, quản lý kết nối cơ sở dữ liệu, và chuẩn hóa phản hồi lỗi hệ thống.
<code>src/config</code>	Lưu trữ các thông số cấu hình nội bộ, bao gồm biến môi trường, thông tin kết nối và khóa truy cập nội bộ giữa các dịch vụ.
<code>src/migrate</code>	Chứa lịch sử các lần thay đổi cấu trúc cơ sở dữ liệu nhằm đảm bảo khả năng truy vết, phục hồi và đồng bộ khi triển khai.
<code>src/modules</code>	Tập hợp toàn bộ các mô-đun nghiệp vụ, nơi thực hiện các chức năng xử lý chính và điều phối hoạt động của hệ thống.

Kiến trúc trên không chỉ đảm bảo tính tách biệt giữa các tầng xử lý mà còn giúp nâng cao khả năng bảo trì, kiểm thử và mở rộng trong tương lai.

(ii) **Cơ sở dữ liệu** Với yêu cầu để lưu trữ dữ liệu người dùng cần rõ ràng cấu trúc, ràng buộc cao và dữ liệu cảm biến chủ yếu là ghi 1 lần theo chuỗi thời gian thì nên chọn cơ sở nào? Sau khi tìm hiểu, tác giả sẽ kiến trúc lưu trữ được thiết kế theo hướng lai, kết hợp cả NoSQL (MongoDB) và SQL (Postgres), nhằm tận dụng thế mạnh riêng của từng loại hình.

Với đặc thù của dữ liệu cảm biến là mang tính phi cấu trúc, có tần suất cập nhật gần như không và yêu cầu truy xuất theo chuỗi thời gian. Hệ thống lựa chọn sử dụng **MongoDB** làm nền tảng lưu trữ chính cho nhóm dữ liệu này. MongoDB thể hiện ưu thế vượt trội nhờ mô hình lưu trữ dạng tài liệu, cho phép linh hoạt mở rộng cấu trúc dữ liệu



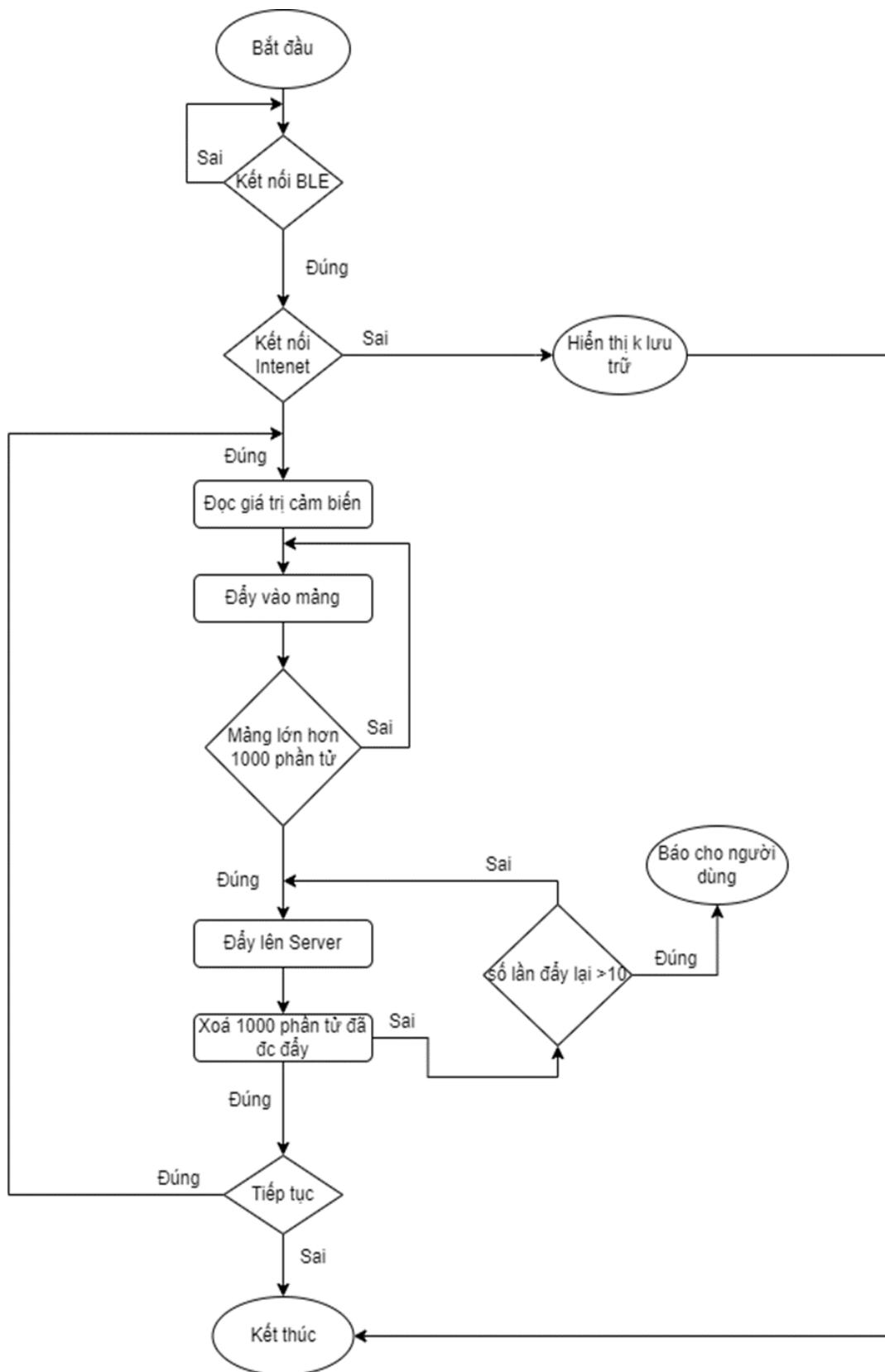
Hình 2.22: Cấu trúc cây thư mục mã nguồn của máy chủ trung tâm

mà không cần định nghĩa cột hàng cố định. Bên cạnh đó, cơ chế *indexing* tối ưu giúp tăng tốc độ truy vấn đối với dữ liệu thời gian, đồng thời hỗ trợ các phép lọc, nhóm và tổng hợp hiệu quả trong môi trường có khối lượng dữ liệu lớn.

Ngược lại, **PostgreSQL** được triển khai nhằm quản lý các dữ liệu có cấu trúc ổn định và đòi hỏi tính toàn vẹn quan hệ cao, bao gồm các thông tin định danh người dùng, mật khẩu được mã hoá, kết quả bảng điểm STOP-BANG, thang điểm buồn ngủ ban ngày Epworth, chỉ số BMI, cùng tiền sử bệnh nền. Cách tiếp cận này không chỉ tăng cường khả năng kiểm soát logic nghiệp vụ ở tầng dữ liệu mà còn là yêu cầu tiên quyết để hệ thống tuân thủ các tiêu chuẩn về **an toàn và bảo mật thông tin y tế**.

Lưu đồ thuật toán lưu trữ dữ liệu được thể hiện trong Hình 2.23, bao gồm hai nhánh xử lý. (i) Khi người dùng không có kết nối mạng, hệ thống vẫn duy trì liên kết BLE và hiển thị tín hiệu thời gian thực; dữ liệu được lưu tạm ở phía thiết bị thay vì đẩy lên cloud. (ii) Khi người dùng đã đăng nhập và có Internet, ứng dụng chuyển sang chế độ đồng bộ theo lô: mỗi 1000 mẫu được gộp thành một gói để gửi lên máy chủ. Cơ chế này sẽ làm giảm số lần yêu cầu từ ứng dụng lên máy chủ.

Do lý do độ dài trong khuôn khổ luận văn, tác giả không trình bày hết toàn bộ các

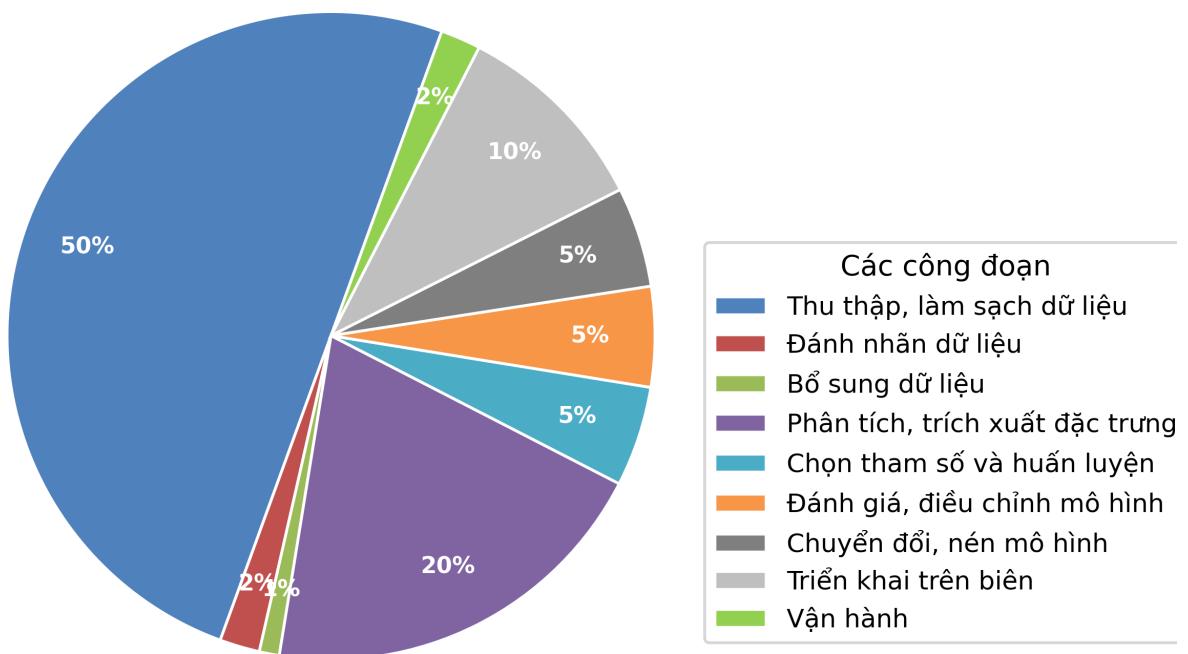


Hình 2.23: Lưu đồ thuật toán lưu trữ dữ liệu cảm biến

nghiệp vụ của máy chủ trung tâm. Phần code sẽ được tác giả chú thích tại mục lục.

### 2.2.3. Học máy trong phân loại tư thế ngủ

Học máy đóng vài trò đưa ra kết quả dựa trên các tín hiệu gia tốc thu được.



Hình 2.24: Phân bổ khối lượng công việc với bài toán học máy phân loại tư thế ngủ trong luận văn

Tổng quan về các bước xây dựng hệ thống học máy cho bài toán phân loại tư thế ngủ đã được trình bày tại Chương 1. Sau khi đánh giá, tác giả lựa chọn 4 mô hình học máy truyền thống là rừng ngẫu nhiên (Random forest - RF), máy vec-tơ hỗ trợ (Support vector machine - SVM), Hồi quy logistic (Logistic regression - LR), thuật toán tăng cường Gradient Boosting (GB) và một mô hình mạng nơ-ron nhân tạo (Artificial Neural Network - ANN).

Việc lựa chọn này, được tác giả lựa chọn dựa trên hai yếu tố là khả năng khái và độ phức tạp thuật toán. Về khả năng khái quát, các mô hình được lựa chọn phản ánh những hướng tiếp cận tiêu biểu trong học máy truyền thống, bao gồm cả mô hình tuyến tính, phi tuyến và mô hình học sâu cơ bản. Cụ thể, Rừng ngẫu nhiên và Gradient Boosting là hai đại diện tiêu biểu của nhóm mô hình ensemble dựa trên cây quyết định, có khả năng

mô hình hóa các quan hệ phi tuyến phức tạp và khắc phục hiện tượng quá khớp cao. Hồi quy logistic giúp giải thích trực quan mối quan hệ giữa các đặc trưng đầu. Máy véc-tơ hỗ trợ được lựa chọn vì có khả năng tạo ranh giới phân lớp rõ ràng trong không gian đặc trưng. Bên cạnh đó, Mạng nơ-ron nhân tạo được đưa vào nhằm khảo sát khả năng trừu tượng hóa sâu của dữ liệu cảm biến, nơi mối quan hệ giữa đặc trưng và nhãn phân loại có thể phi tuyến và phức tạp hơn so với các mô hình cổ điển.

Xét theo yếu tố độ phức tạp, tác giả cân nhắc việc lựa chọn các mô hình sao cho đảm bảo cân bằng giữa hiệu năng và chi phí tính toán. Đặc biệt, mô hình ANN được thiết kế với hai lớp ẩn nhằm đánh giá sơ bộ hiệu quả của mô hình học sâu trước khi mở rộng sang các kiến trúc phức tạp hơn như mạng tích chập (Convolutional Neural Network - CNN), mạng hồi tiếp (Recurrent Neural Network - RNN) hay mạng bộ nhớ dài ngắn hạn (Long Short-Term Memory - LSTM).

## Hồi quy Logistic

Đây là thuật toán học máy có giám sát được thiết kế cho các tác vụ phân loại dựa trên các đặc trưng đầu vào. Về mặt cấu trúc, hồi quy Logistic làm việc dựa trên nguyên tắc của hàm sigmoid - một hàm phi tuyến chuyển đầu vào của nó thành xác suất thuộc về một trong hai lớp đối với bài toán phân loại nhị phân.

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{với } z = \mathbf{w}^T \mathbf{x} + b \quad (2.1)$$

Trong đó,  $\mathbf{w}$  là vector trọng số,  $\mathbf{x}$  là vector đặc trưng đầu vào, và  $b$  là hệ số điều chỉnh (bias). Khi xác suất này vượt quá một ngưỡng nhất định (thông thường là 0.5), mô hình sẽ gán nhãn cho  $\mathbf{x}$  là thuộc lớp dương; ngược lại, nếu nhỏ hơn ngưỡng, mẫu được phân loại vào lớp âm.

Mặc dù đơn giản và dễ triển khai, hồi quy logistic nguyên thủy chỉ phù hợp với các bài toán phân loại nhị phân. Để mở rộng cho bài toán phân loại đa lớp, có thể sử dụng biến thể **Softmax Regression**, trong đó mô hình ước lượng xác suất đầu ra theo phân phối softmax:

$$P(y = j | \mathbf{x}) = \frac{e^{\mathbf{w}_j^\top \mathbf{x}}}{\sum_{k=1}^K e^{\mathbf{w}_k^\top \mathbf{x}}} \quad (2.2)$$

Trong đó,  $K$  là tổng số lớp và  $\mathbf{w}_j$  là vector trọng số tương ứng với lớp  $j$ . Cách tiếp cận này cho phép mô hình tính toán đồng thời xác suất cho tất cả các lớp và lựa chọn lớp có xác suất cao nhất làm đầu ra.

Ngoài phương pháp softmax, một chiến lược mở rộng khác thường được sử dụng là **One-vs-Rest (OvR)**. Trong chiến lược này, mô hình sẽ huấn luyện  $K$  bộ phân loại nhị phân độc lập, mỗi bộ phân biệt một lớp cụ thể với phần còn lại của tập dữ liệu. Khi dự đoán, mẫu dữ liệu mới được đưa vào cả  $K$  mô hình, và lớp có xác suất dự đoán cao nhất sẽ được chọn làm kết quả cuối cùng.

Trong bài toán phân loại tư thế ngũ, LR được kì vọng sẽ tối ưu về mặt chi phí tính toán và kích thước mô hình để phù hợp với các bài toán triển khai tại biên.

## Máy vec-tơ hỗ trợ

Đây là một thuật toán học có giám sát, đặc biệt hiệu quả cho các bài toán phân loại nhị phân với biên ranh giới rõ ràng [109]. Ý tưởng chính của SVM là tìm kiếm một **mặt siêu phẳng (hyperplane)** trong không gian đặc trưng để phân chia các điểm dữ liệu thành hai lớp sao cho biên phân cách giữa các lớp là lớn nhất.

Trong không gian hai chiều, mặt siêu phẳng tương ứng với một đường thẳng; trong không gian ba chiều, đó là một mặt phẳng; và trong không gian nhiều chiều hơn, nó là một siêu mặt phẳng tổng quát. SVM chọn mặt siêu phẳng sao cho khoảng cách từ nó đến các điểm dữ liệu gần nhất của mỗi lớp - gọi là **support vectors** là tối đa.

Trong không gian hai chiều, khoảng cách từ một điểm có tọa độ  $(x_0, y_0)$  đến đường thẳng có phương trình

$$w_1x + w_2y + b = 0$$

được xác định bởi công thức:

$$d = \frac{|w_1x_0 + w_2y_0 + b|}{\sqrt{w_1^2 + w_2^2}} \quad (2.3)$$

Tương tự, trong không gian ba chiều, khoảng cách từ một điểm  $(x_0, y_0, z_0)$  đến mặt phẳng có phương trình

$$w_1x + w_2y + w_3z + b = 0$$

được tính bằng:

$$d = \frac{|w_1x_0 + w_2y_0 + w_3z_0 + b|}{\sqrt{w_1^2 + w_2^2 + w_3^2}} \quad (2.4)$$

Dấu của biểu thức ( $w_1x_0 + w_2y_0 + w_3z_0 + b$ ) cho biết điểm  $(x_0, y_0, z_0)$  nằm về phía nào của mặt phẳng đang xét. Những điểm có giá trị dương nằm về cùng một phía (phía dương), trong khi những điểm có giá trị âm nằm về phía còn lại (phía âm). Các điểm nằm trên chính mặt phẳng này sẽ làm cho tử số bằng không, tức là khoảng cách bằng 0.

Khái niệm này có thể được **tổng quát hóa lên không gian nhiều chiều**. Giả sử ta có một điểm (vector)  $\mathbf{x}_0 \in \mathbb{R}^d$  và một siêu mặt phẳng có phương trình [110]:

$$\mathbf{w}^\top \mathbf{x} + b = 0$$

thì khoảng cách vuông góc từ điểm  $\mathbf{x}_0$  đến siêu mặt phẳng này được cho bởi:

$$d = \frac{|\mathbf{w}^\top \mathbf{x}_0 + b|}{\|\mathbf{w}\|_2} \quad (2.5)$$

trong đó:

$$\|\mathbf{w}\|_2 = \sqrt{\sum_{i=1}^d w_i^2}$$

với vector trọng số  $\mathbf{w}$ , và  $d$  là số chiều của không gian.

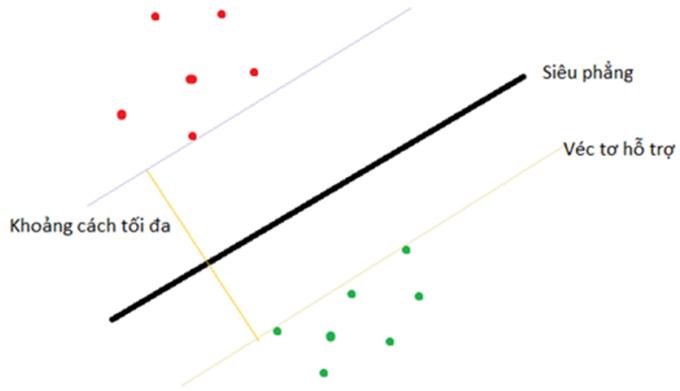
Hình 2.25 minh họa khái niệm mặt siêu phẳng và các máy hỗ trợ trong không gian hai chiều.

Với mặt phẳng phân chia như trên, khoảng cách được tính là khoảng cách gần nhất từ một điểm tới mặt phẳng đó (bất kể điểm thuộc hai lớp nào):

$$\text{margin} = \min_n \frac{y_n (\mathbf{w}^\top \mathbf{x}_n + b)}{\|\mathbf{w}\|_2}$$

Bài toán tối ưu trong SVM chính là bài toán tìm  $\mathbf{w}$  và  $b$  sao cho *margin* này đạt giá trị lớn nhất:

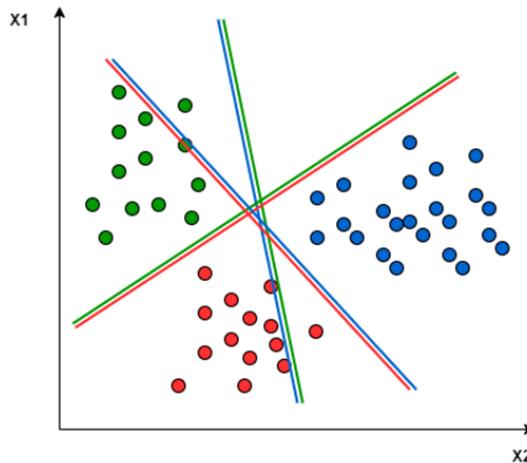
$$(\mathbf{w}, b) = \arg \max_{\mathbf{w}, b} \left\{ \min_n \frac{y_n (\mathbf{w}^\top \mathbf{x}_n + b)}{\|\mathbf{w}\|_2} \right\}$$



Hình 2.25: Minh họa mặt siêu phẳng phân tách hai lớp trong SVM

Thêm vào đó, SVM có thể mở rộng cho các bài toán không tuyến tính thông qua việc sử dụng các hàm kernel, chẳng hạn như **Gaussian RBF kernel** hoặc **polynomial kernel**, giúp ánh xạ dữ liệu vào không gian mới nơi mà việc phân tách tuyến tính trở nên khả thi.

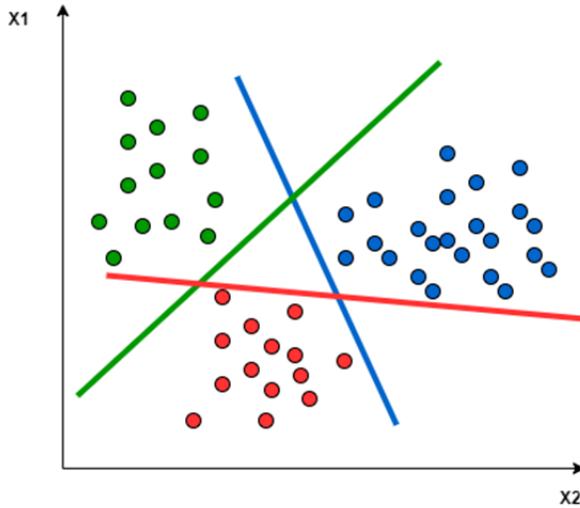
Để mở rộng cho các bài toán phân loại đa lớp, có thể áp dụng hai kỹ thuật phổ biến: **one-vs-one** và **one-vs-rest**, được minh họa trong Hình 2.26 và Hình 2.27.



Hình 2.26: Chiến lược phân loại đa lớp bằng phương pháp One-vs-One

**One-vs-One (OvO):** Trong phương pháp này, một mô hình SVM được huấn luyện cho mỗi cặp lớp. Với  $K$  lớp, tổng cộng  $\frac{K(K-1)}{2}$  mô hình con được huấn luyện. Mỗi mô hình học cách phân biệt giữa hai lớp cụ thể và bỏ qua các lớp còn lại. Trong quá trình dự đoán, một cơ chế bỏ phiếu (voting) được sử dụng để xác định lớp cuối cùng.

**One-vs-Rest (OvR):** Phương pháp này huấn luyện một mô hình cho mỗi lớp, trong



Hình 2.27: Chiến lược phân loại đa lớp bằng phương pháp One-vs-Rest

đó mô hình học cách phân biệt giữa một lớp cụ thể và phần còn lại. Với  $K$  lớp, ta có  $K$  mô hình. Trong quá trình suy luận, mô hình đưa ra xác suất hoặc độ tin cậy, và lớp có giá trị cao nhất sẽ được chọn.

Cả hai chiến lược OvO và OvR đều giúp mở rộng SVM từ mô hình phân loại nhị phân thành phân loại đa lớp hiệu quả, nhưng mỗi phương pháp đều có ưu và nhược điểm riêng về thời gian huấn luyện, độ phức tạp tính toán và hiệu năng phân loại.

## Rừng ngẫu nhiên

Đây là một mô hình học có giám sát thuộc nhóm thuật toán tổ hợp (ensemble learning), được xây dựng dựa trên nền tảng của **Cây quyết định (Decision Tree)** [111]. Khác với việc sử dụng một cây quyết định duy nhất truyền thống, rừng ngẫu nhiên xây dựng một tập hợp gồm nhiều cây quyết định độc lập, mỗi cây học trên một phần khác nhau của dữ liệu và không sử dụng toàn bộ tập thuộc tính. Dự đoán cuối cùng của mô hình được xác định thông qua cơ chế biểu quyết hoặc trung bình hoá (trong bài toán hồi quy).

Ý tưởng chính của rừng ngẫu nhiên nhằm giảm thiểu hiện tượng **quá khớp (overfitting)** thường gặp trong câu quyết định đơn lẻ. Khi xây dựng một cây quyết định mà không giới hạn độ sâu, cây có xu hướng học thuộc hoàn toàn dữ liệu huấn luyện, dẫn đến khả năng tổng quát kém trên tập kiểm thử. RF khắc phục điều này bằng cách đưa vào hai

cơ chế ngẫu nhiên chính:

**1) Lấy mẫu bootstrap:** Mỗi cây được huấn luyện trên một tập con của dữ liệu ban đầu, được chọn ngẫu nhiên có lặp lại (bootstrap sampling). Như vậy, một phần dữ liệu được bỏ qua, làm tăng tính đa dạng giữa các cây.

**2) Lựa chọn ngẫu nhiên tập thuộc tính:** Tại mỗi nút phân chia của cây, chỉ một tập con ngẫu nhiên của các thuộc tính được xem xét để chọn điểm chia tốt nhất. Điều này làm giảm sự tương quan giữa các cây trong rừng.

Ngoài ra, việc tổng hợp kết quả của nhiều cây giúp giảm phương sai, cải thiện khả năng tổng quát hóa. Nhờ đó, RF đạt được sự ổn định kể cả với các bộ dữ liệu có nhiều hoặc mất cân bằng.

## Gradient boosting

Đây là một phương pháp học có giám sát cũng thuộc nhóm thuật toán tổ hợp (ensemble learning) [112]. Khác với Random Forest khi các cây được xây dựng song song và độc lập, Gradient Boosting xây dựng mô hình theo từng bước lặp, mỗi cây tiếp theo được huấn luyện để sửa lỗi còn lại từ mô hình trước đó. Cụ thể, tại mỗi vòng lặp  $t$ , mô hình hiện tại  $F_t(x)$  được cập nhật bằng cách cộng thêm một cây mới  $h_t(x)$  được huấn luyện để xấp xỉ gradient âm của hàm mất mát:

$$F_{t+1}(x) = F_t(x) + \gamma h_t(x) \quad (2.6)$$

Trong đó,  $\gamma$  là hệ số học (learning rate), điều chỉnh mức đóng góp của cây mới vào tổng thể mô hình.

Một trong những đặc điểm quan trọng của GB là khả năng tối ưu hóa trực tiếp một hàm mất mát bất kỳ, chẳng hạn như hàm log-loss trong bài toán phân loại, hoặc hàm bình phương sai số trong bài toán hồi quy. Nhờ đó, GB thường đạt độ chính xác rất cao, đặc biệt trên các bài toán với dữ liệu có quan hệ phi tuyến và có nhiều đặc trưng tương tác phức tạp.

Tuy nhiên, Gradient Boosting cũng có những hạn chế rõ rệt. Do các cây được xây dựng tuân tự phụ thuộc lẫn nhau, GB thường mất nhiều thời gian huấn luyện hơn so với

Random Forest. Hơn nữa, mô hình nhạy cảm với nhiều và dữ liệu nhiều sẽ dễ dàng bị mô hình học theo, dẫn đến hiện tượng quá khớp hơn.

## Mạng nơ-ron nhân tạo

Đây thuộc lớp mô hình học sâu mô phỏng cấu trúc hoạt động của hệ thần kinh sinh học, trong đó các nơ-ron nhân tạo (artificial neurons) được tổ chức thành nhiều lớp (layers) và kết nối với nhau qua các trọng số (weights) [113].

Trong nghiên cứu này, tác giả sử dụng kiến trúc đơn giản nhất gồm 3 lớp: lớp đầu vào (input layer), hai lớp ẩn (hidden layers) và lớp đầu ra (output layer).

Mỗi nơ-ron trong lớp ẩn thực hiện một tổ hợp tuyến tính giữa các đầu vào, sau đó áp dụng một hàm kích hoạt phi tuyến như hàm ReLU (Rectified Linear Unit):

$$f(x) = \max(0, x) \quad (2.7)$$

Đầu ra của mạng được tính thông qua lan truyền tiến, và mô hình được huấn luyện bằng cách tối thiểu hóa một hàm mất mát.

Ưu điểm chính của mạng nơ-ron là khả năng học các quan hệ phi tuyến phức tạp và tự động trích xuất đặc trưng từ dữ liệu. Khác với các mô hình tuyến tính như LR hoặc SVM, ANN có thể biểu diễn các ranh giới phân lớp không tuyến tính và phù hợp với các bài toán đa tín hiệu cảm biến, đa cấu trúc.

Tuy nhiên, trong khuôn khổ của nghiên cứu này, mục tiêu chính không chỉ là đạt độ chính xác tối đa mà còn là đánh giá ảnh hưởng của các đặc trưng trên miền thời gian và miền tần số đối với hiệu suất của mô hình học máy nên tác giả quyết định chọn LR, SVM, RF, GB, NN để tiến hành thử nghiệm. Còn tương lai, tác giả sẽ mở rộng với các thuật toán phức tạp hơn để đánh giá toàn diện hơn đối với bài toán phân loại tư thế ngủ.

Đến đây, tác giả đã trình bày đầy đủ phương pháp xây dựng hệ thống, bao gồm cả phần cứng, phần mềm và các mô hình học máy được lựa chọn phù hợp với bài toán phân loại tư thế ngủ. Phương pháp đề xuất được đánh giá một cách toàn diện, từ cơ sở lý thuyết đến triển khai thực tế, nhằm đảm bảo tính khả thi và hiệu quả trong ứng dụng. Trên cơ sở đó, tác giả tiến hành phát triển, thử nghiệm và trình bày kết quả cụ thể trong chương tiếp theo.

## **Chương 3**

# **KẾT QUẢ VÀ ĐÁNH GIÁ**

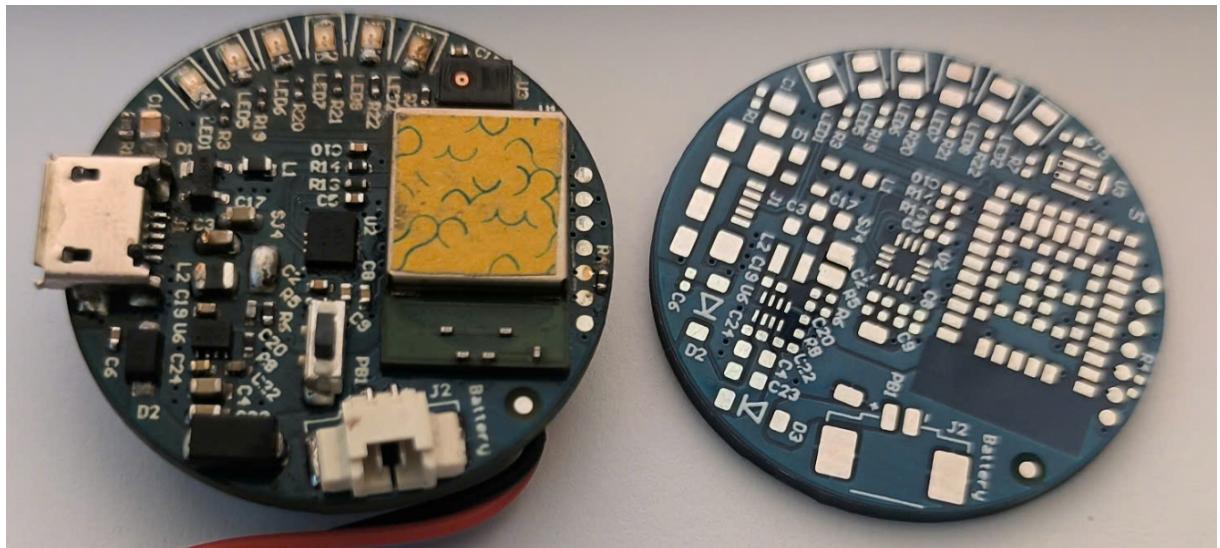
Trong phần này, tác giả trình bày những kết quả đã đạt được dựa trên các cơ sở lý thuyết trong hai chương đã trình bày phần trước. Kết quả sẽ được trình bày đúng theo trình tự mục tiêu của luận văn: Đầu tiên là kết quả nghiên cứu, phát triển phần cứng. Thứ hai, kết quả đánh giá dữ liệu cảm biến bằng học máy. Cuối cùng là kết quả triển khai trên biên.

### **3.1. Hệ thống thu thập, lưu trữ**

Hình 3.1 là kết quả mạch đã hoàn thiện. Mạch đã được nhóm kiểm tra hoạt động đúng theo yêu cầu các thành phần theo như thiết kế. Nạp bootloader và tiến hành nạp code cho vi điều khiển qua USB.

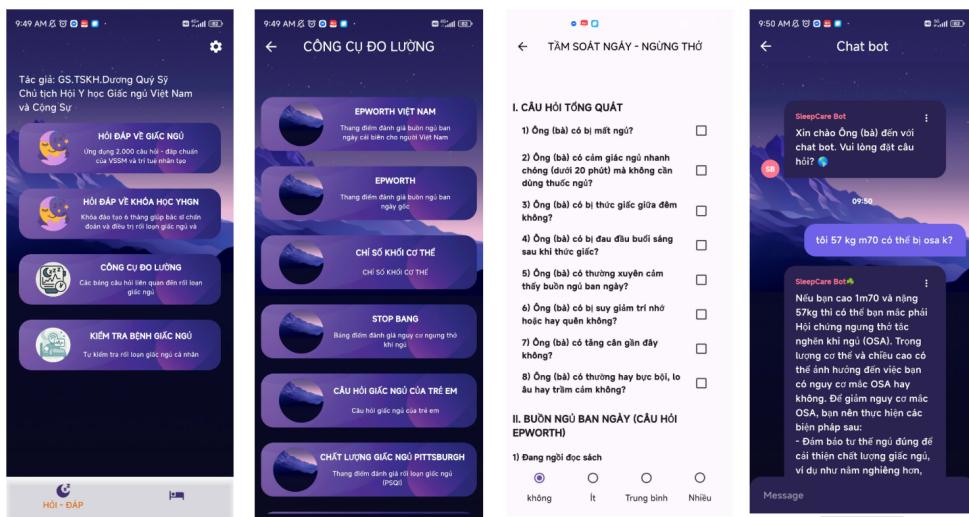
Ngoài cách cấp nguồn trực tiếp thông qua USB, tác giả còn gắn thêm viên pin lipo có dải điện áp trung bình là 3.7 V và tối thiểu là 3V có kích thước là 20 mm \* 20 mm \* 5 mm tại mạch sau của mạch để thuận lợi hơn trong quá trình lấy mẫu.

Song song với việc phát triển phần cứng, phần mềm cũng được hoàn thành với đầy đủ các tính năng như yêu cầu trước đó đưa ra bao gồm cả ứng dụng di động và máy chủ đám mây. Dựa trên phản hồi từ nhóm nghiên cứu, tư vấn khoa học của Thầy PGS.TS. Mai Anh Tuấn, tư vấn y khoa của Thầy GS.TS. Dương Quý Sỹ, ứng dụng được thiết kế với tiêu chí giao diện thân thiện, thao tác đơn giản và tính năng tập trung vào mục tiêu thử nghiệm.



Hình 3.1: Mạch hoàn chỉnh

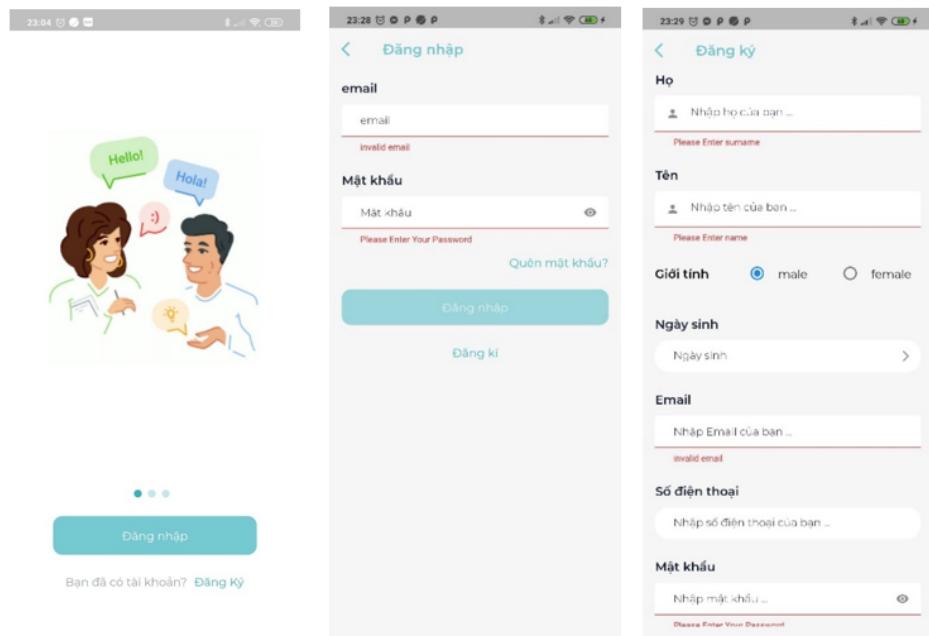
Sau khi cài đặt, người dùng có thể đăng nhập hoặc đăng ký tài khoản thông qua giao diện như được thể hiện trong Hình 3.3. Với người dùng mới, quá trình đăng ký yêu cầu xác thực địa chỉ email nhằm đảm bảo bảo mật và hỗ trợ tính năng khôi phục tài khoản. Hình 3.2 minh họa giao diện của các chức năng, Chatbot hỗ trợ người dùng trong quá trình sàng lọc nguy cơ ngưng thở khi ngủ và cung cấp thông tin về chất lượng giấc ngủ.



Hình 3.2: Giao diện chức năng chatbot và bộ câu hỏi tâm soátr

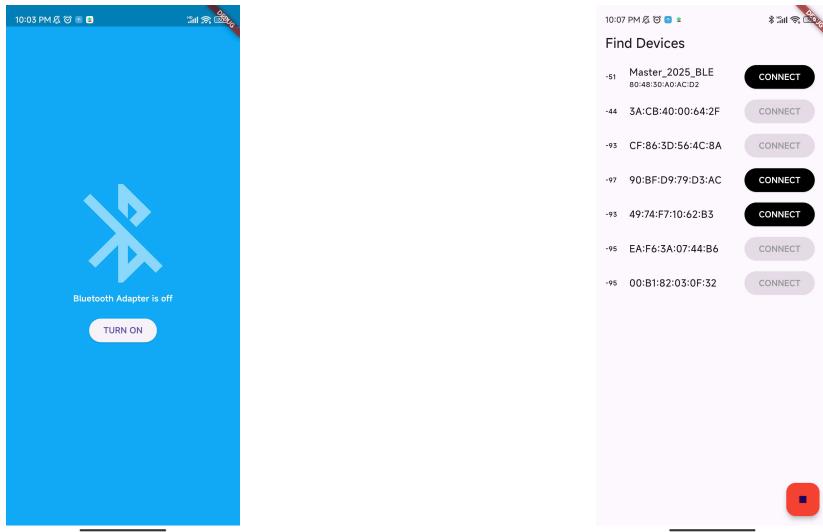
Hình ?? thể hiện danh sách BLE có thể kết nối và dịch vụ kết nối với phần cứng đã được nhắc tới bên trên. Khi người chọn vào BLE có tên là MASTER\_2025\_BLE ứng

dụng di động xuất thêm màn hình chọn cảm biến thu thập. Đây cũng có thể coi là bước chọn đúng UUID của dịch cụ GATT đã được cấu hình sẵn ở phần cứng.



Hình 3.3: Giao diện chức năng đăng ký và đăng nhập

Khi chọn đọc giá trị cảm biến, ứng dụng di động sẽ tự động truy cập vào đặc tính trong dịch vụ. Hình 3.5 minh họa giao diện hiển thị giá trị cảm biến theo thời gian thực và Hình 3.6 thể hiện giá trị cảm biến đã được lưu vào cơ sở dữ liệu. Ngoài ra, dữ liệu về tư thế hiện tại và số lượng tư thế ngủ được tính toán dựa vào suy luận kết quả mô hình đã huấn luyện trên máy chủ trung tâm.



Hình 3.4: Màn hình yêu cầu quyền truy cập Bluetooth và danh sách thiết bị đang hoạt động

### 3.2. Thu thập và gắn nhãn dữ liệu

Trong phần này, tác giả trình bày chi tiết phương pháp thu thập dữ liệu, các kịch bản thực nghiệm, cũng như quy trình xử lý và trích xuất đặc trưng để phục vụ cho việc huấn luyện các mô hình học máy trong bài toán nhận diện tư thế ngủ.

Tổng cộng 25 tình nguyện viên đã được tuyển chọn tham gia vào quá trình thu thập dữ liệu, với độ tuổi dao động từ 10 đến 60, trong đó độ tuổi phổ biến là 24. Nhóm tình nguyện viên bao gồm cả nam và nữ, được lựa chọn với tiêu chí đa dạng về giới tính và độ tuổi nhằm tăng tính đại diện và khách quan cho bộ dữ liệu.

Trong kịch bản đầu tiên (gọi là **thu thập có giám sát**), mỗi tình nguyện viên được hướng dẫn gắn thiết bị cảm biến vào vùng cổ bằng băng keo y tế hai mặt 3M, sau đó đăng nhập vào ứng dụng di động với tài khoản cá nhân đã đăng ký. Dưới sự giám sát trực tiếp của tác giả, mỗi người tham gia sẽ lần lượt thực hiện các tư thế ngủ cơ bản (nằm ngửa, nằm sấp, nghiêng trái, nghiêng phải) trong thời gian tối thiểu 2 phút cho mỗi tư thế Hình 3.7. Mỗi tư thế được lặp lại ít nhất hai lần để đảm bảo tính lặp lại và ổn định của tín hiệu. Sau khi xác minh rằng dữ liệu cảm biến đã được lưu trữ đầy đủ trên hệ thống (kiểm tra trên MongoDB và giao diện ứng dụng), quá trình thu thập dữ liệu từ một tình nguyện viên được xem là hoàn tất.

Bên cạnh đó, để mô phỏng điều kiện thực tế khi sử dụng thiết bị trong sinh hoạt



Hình 3.5: Màn hình hiển thị dữ liệu gia tốc ba trục với hoảng là giá trị thời gian, trực tung là giá trị cảm biến gấp lên 10 lần

```

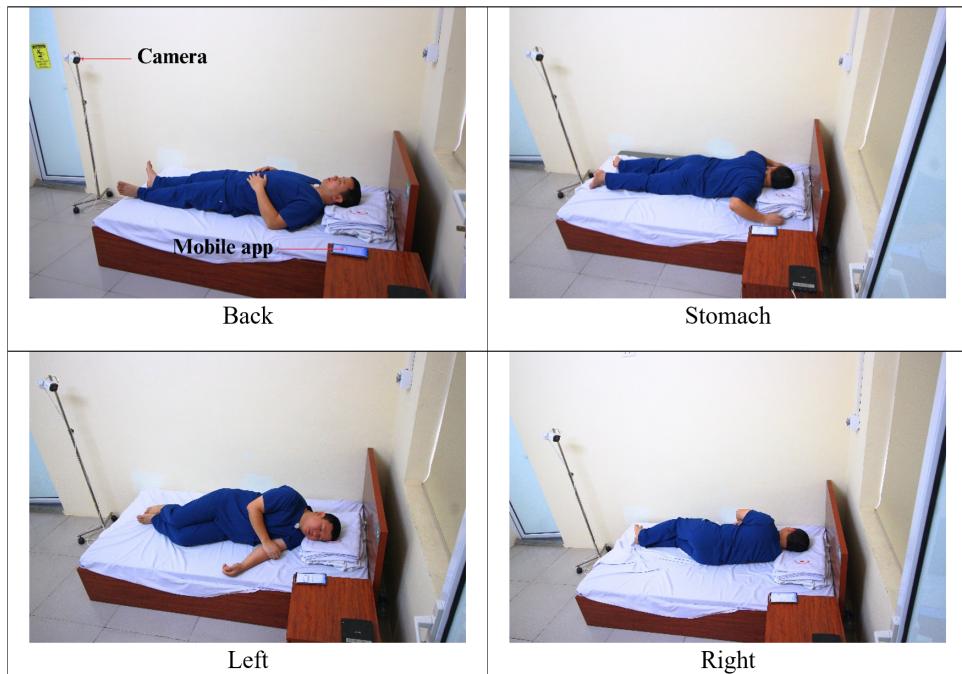
_id: ObjectId('690b5e69e3f6ee57221333c8')
user : 1
value : "-0.15, 0.97, -0.14"
createdAt : 2025-11-05T14:24:50.606+00:00
__v : 0

_id: ObjectId('690b5e69e3f6ee57221333c9')
user : 1
value : "-0.17, 0.91, -0.25"
createdAt : 2025-11-05T14:24:50.692+00:00
__v : 0

_id: ObjectId('690b5e69e3f6ee57221333ca')
user : 1
value : "-0.15, 0.97, -0.14"
createdAt : 2025-11-05T14:24:50.693+00:00
__v : 0

```

Hình 3.6: Dữ liệu cảm biến gia tốc được lưu trong cơ sở dữ liệu



Hình 3.7: Mô phỏng thực nghiệm thực tế

ban đêm, tác giả đã tự thực hiện kịch bản thứ hai (**thu thập trong giấc ngủ tự nhiên**). Trong kịch bản này, thiết bị được gắn vào cổ trước khi đi ngủ và ghi nhận dữ liệu liên tục trong suốt một đêm. Song song đó, một camera cố định được lắp đặt phía trên giường để ghi hình toàn bộ quá trình ngủ, từ đó hỗ trợ gán nhãn chính xác theo thời gian. Dữ liệu trong giai đoạn này được xử lý và đồng bộ thủ công giữa tín hiệu cảm biến và video để loại bỏ các đoạn có chuyển động hoặc sai lệch nhãn.

Mặc dù phương pháp thu thập trong môi trường tự nhiên sát với điều kiện sử dụng thực tế, nhưng đòi hỏi nhiều công sức xử lý hậu kỳ và khó kiểm soát chất lượng dữ liệu đầu vào. Theo ý kiến tư vấn từ các chuyên gia trong lĩnh vực y học giấc ngủ, phương pháp thu thập có giám sát (phương pháp 1) vẫn được ưu tiên do khả năng kiểm soát tốt, đảm bảo dữ liệu cân bằng giữa các tư thế, đồng thời vẫn duy trì được mức độ tương thích cao với điều kiện thực tế khi triển khai ứng dụng theo dõi tại nhà.

Sau quá trình thu thập, bộ dữ liệu huấn luyện bao gồm tổng cộng **158.750 mẫu** hợp lệ sau khi đã lọc nhiều và loại bỏ các phiên ghi nhận không đạt yêu cầu của 25 tình nguyện viên. Dữ liệu kiểm thử được loại bỏ các hoạt động không thuộc tư thế ngủ và gắn nhãn dựa trên camera. Kết quả là bộ dữ liệu kiểm thử gồm **64.258 mẫu**.

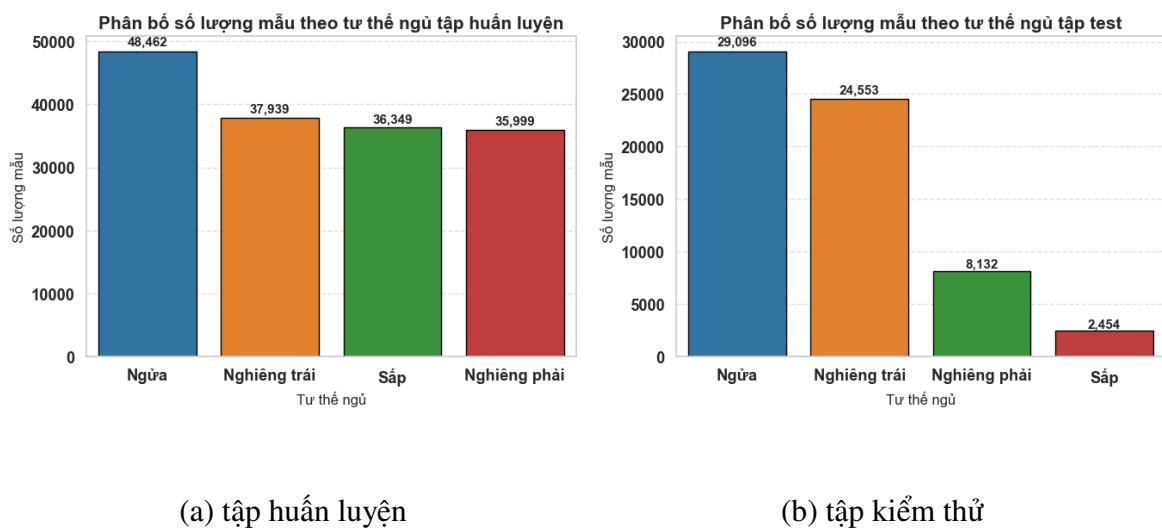
Tất cả dữ liệu thu thập từ các tình nguyện viên và tác giả đều được xuất ra định

dang CSV, bao gồm thông tin thời gian (timestamp), giá trị cảm biến trên ba trục  $x$ ,  $y$ ,  $z$ , và nhãn tư thế tương ứng (nếu có). Dữ liệu này được sử dụng làm đầu vào cho quá trình trích xuất đặc trưng và huấn luyện mô hình học máy.

### 3.3. Phân loại tư thế ngủ bằng học máy

#### 3.3.1. Phân tích dữ liệu

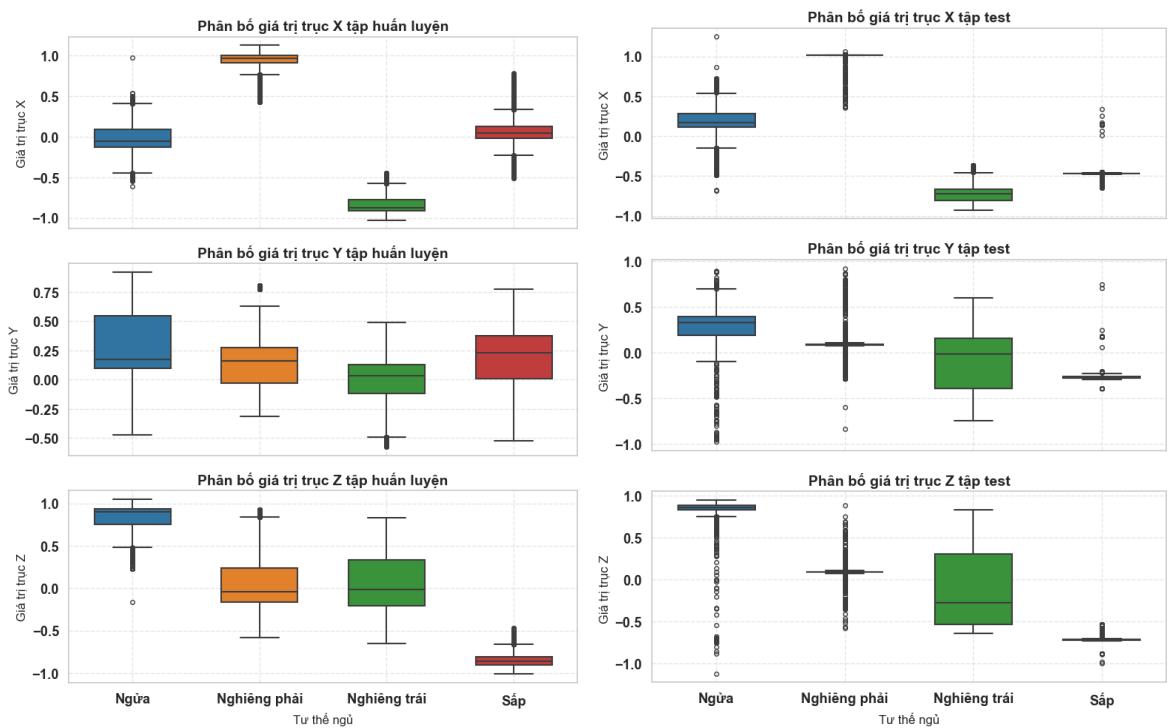
Sau khi hoàn thành quá trình thu thập dữ liệu của hai kịch bản, tác giả tiến hành phân tích các giá trị cảm biến của ba trục. Đầu tiên phân tích độ cân bằng dữ liệu như Hình 3.8.



Hình 3.8: Biểu đồ phân bố các nhãn thu được

Hình 3.8 cho thấy phân bố dữ liệu của tập huấn luyện tương đối cân bằng. Còn đối với tập kiểm thử, tư thế nằm ngửa và nghiêng trái chiếm đa số. Qua đó, cho thấy sự phản ánh dữ liệu đúng như dự kiến của tác giả. Với bộ huấn luyện, các nhãn sẽ giữ ở mức cân bằng để tránh tình trạng quá khớp. Còn ở tập kiểm thử, các nhãn phản ánh đúng sinh lý của cá nhân khi ngủ, sẽ thường có những tư thế nổi bật hẳn về số lượng do thói quen.

Hình 3.9 (a) trình bày phân tích chi tiết phân bố tín hiệu cảm biến theo ba trục gia tốc ứng với bốn tư thế ngủ cơ bản của tập huấn luyện thu từ 25 tình nguyện viên. Ở trục  $x$ , các phân bố tương đối biệt lập, giữa hai tư thế nằm ngửa và nằm sấp, cũng như giữa nghiêng trái và nghiêng phải. Điều này cho thấy trục  $x$  có khả năng phân biệt tư thế tốt.

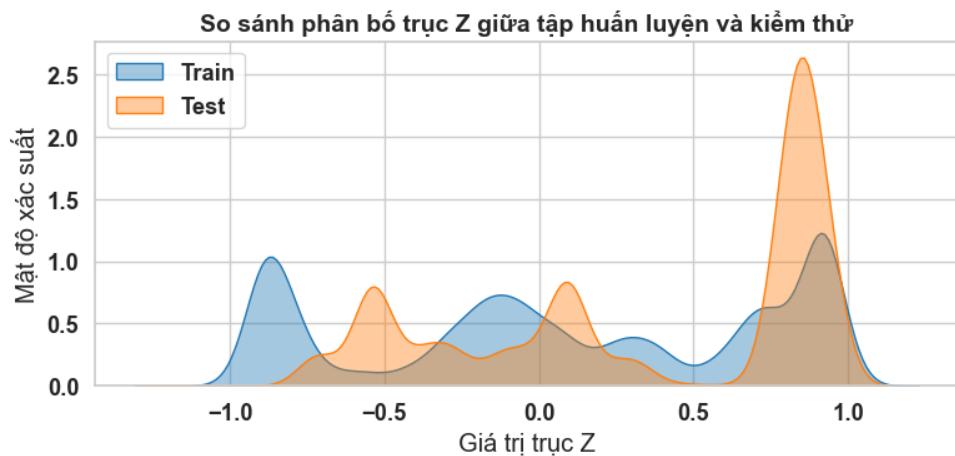


(a) tập huấn luyện

(b) tập kiểm thử

Hình 3.9: Biểu đồ phân bố dữ liệu 3 trục theo các tư thế ngủ

Trục y thể hiện mức độ tương đồng giữa các tư thế, dẫn đến khả năng tách biệt thấp và ít giá trị trong việc xác định tư thế ngủ. Đôi với trục z, có thể quan sát được sự phân tách rõ ràng giữa tư thế nằm nghiêng và các tư thế dọc (nằm ngửa và nằm sấp), chứng tỏ vai trò quan trọng của trục z trong phân loại tư thế. Còn đối với Hình 3.9 (b), các phân bố về trung bình cũng tương đồng với tập huấn luyện cho thấy sự phản ánh đúng phương pháp lấy mẫu mang tính đúng đắn. Tuy nhiên, giải phân phối của tập kiểm thử tương đối rộng, nguyên nhân được suy đoán là do các tín hiệu nhiễu chưa được sàng lọc kỹ ở bước làm sạch dữ liệu và gắn nhãn. Đây là cũng trường hợp không hiếm gặp trong khi ngủ thực tế cả đêm, con người có thể có các hoạt động khác như là đi vệ sinh hay là uống nước. Đây cũng là thách thức tác giả cần giải quyết phát hiện đang nằm ngủ hay đang không nằm.



Hình 3.10: Phân bố giá trị trục x của 2 tập dữ liệu

Hình 3.10 chỉ ra phân bố giá trị trục z của hai tập dữ liệu: huấn luyện (xanh) và kiểm thử (cam). Cho thấy, mật độ đỉnh ở hai tập dữ liệu khá tương đồng. Điều này cũng được chính minh ở phần trên. Tuy nhiên, có một số điểm khác biệt tại giá trị -0.5 của tập kiểm thử. Giá trị này rơi chủ yếu vào tư thế nằm nghiêng trái và nằm sấp. Từ đây, tác giả phỏng đoán sẽ có nhầm lẫn trong suy luận giữa 2 tư thế này tại tập kiểm thử.

### 3.3.2. Xử lý và trích xuất đặc trưng

Dữ liệu cảm biến thu thập được trước tiên được xử lý khử nhiễu bằng phương pháp hiệu chỉnh điểm gốc, bằng cách lấy hiệu giữa giá trị hiện tại và giá trị tham chiếu ban đầu trên ba trục x, y, và z. Sau đó, tín hiệu được chia thành các cửa sổ thời gian có độ dài 2 giây, với mức chồng lấn 50% giữa các cửa sổ. Chỉ những cửa sổ dữ liệu có nhãn nhất

quán trong toàn bộ thời gian mới được giữ lại để huấn luyện mô hình. Các cửa sổ chứa nhãn không đồng nhất (nhiều hơn một nhãn) hoặc có biểu hiện chuyển động bất thường sẽ bị loại bỏ khỏi quá trình xử lý tiếp theo.

### **Đặc trưng miền thời gian (T1)**

Dữ liệu cảm biến gia tốc là dữ liệu chuỗi thời gian, do đó các đặc trưng miền thời gian đóng vai trò rất quan trọng trong nhận diện tư thế ngủ. Trong nghiên cứu này, tác giả trích xuất tổng cộng 40 đặc trưng thống kê cho mỗi cửa sổ dữ liệu, trên cả ba trục  $x$ ,  $y$ ,  $z$ . Các đặc trưng bao gồm giá trị trung bình, độ lệch chuẩn, độ lệch tuyệt đối trung bình, giá trị lớn nhất, nhỏ nhất, hiệu số lớn-nhỏ nhất, trung vị, độ lệch tuyệt đối trung vị, khoảng từ phân vị, số lượng giá trị âm/dương, số lượng giá trị lớn hơn trung bình, số đỉnh tín hiệu, độ lệch, độ nhọn, năng lượng tín hiệu, gia tốc tổng hợp và tổng độ lớn tín hiệu. Các đặc trưng này được lựa chọn dựa trên tính dễ tính toán, hiệu quả phân tách tư thế và khả năng triển khai trên vi điều khiển.

### **Đặc trưng miền tần số (F1)**

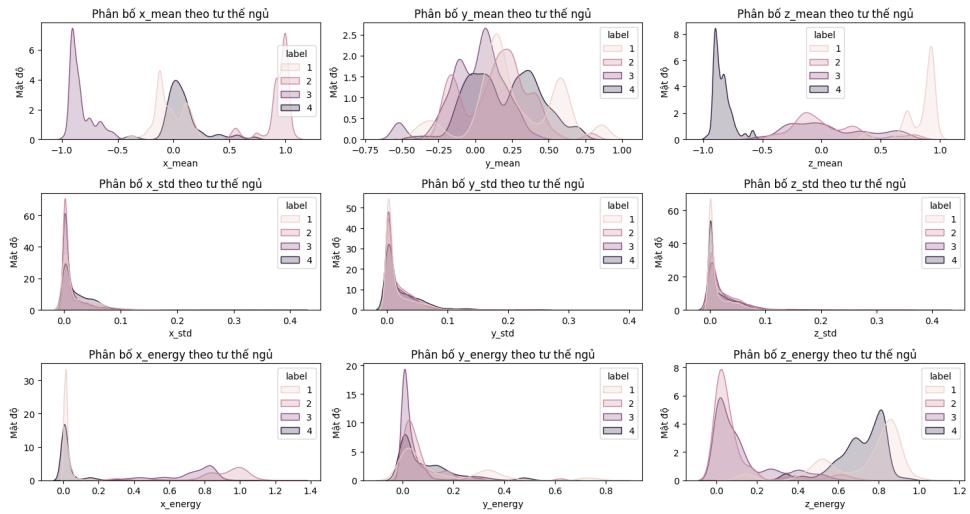
Để khai thác thông tin trong miền tần số, tác giả sử dụng Biến đổi Fourier Nhanh (FFT) để chuyển đổi dữ liệu từ miền thời gian sang miền tần số. Từ các cửa sổ tín hiệu sau biến đổi, 29 đặc trưng thống kê được tính toán, bao gồm các đặc trưng tương tự như trong miền thời gian: trung bình, độ lệch chuẩn, độ lệch tuyệt đối, giá trị cực đại - cực tiểu, trung vị, khoảng từ phân vị, số đỉnh, độ lệch, độ nhọn, năng lượng tín hiệu,... Ngoài ra, hai đặc trưng kết hợp là gia tốc tổng hợp trung bình và tổng độ lớn tín hiệu (SMA) cũng được duy trì trong miền tần số để phục vụ so sánh với miền thời gian.

Việc sử dụng đồng thời các đặc trưng từ cả hai miền thời gian và tần số giúp tăng khả năng mô tả đặc trưng cho mô hình học máy, từ đó nâng cao hiệu quả phân loại tư thế ngủ trong các điều kiện khác nhau.

Hình 3.11 đưa ra những ý nghĩa của ba đặc trưng miền thời gian. Các đặc trưng trung bình và năng lượng của 2 trục  $x$  và  $z$  cho phân bố mật độ tách biệt để phân biệt nhóm tư thế nghiêng trái - nghiêng phải và ngửa - sấp. Đối với trục  $y$ , theo đánh giá các phân bộ sẽ đa số tập trung thành cũng giống như đặc trưng độ lệnh chuẩn trên miền thời gian và các hai giá trị đặc trưng trên miền tần số Hình 3.12. Qua phân tích trên, tác giả

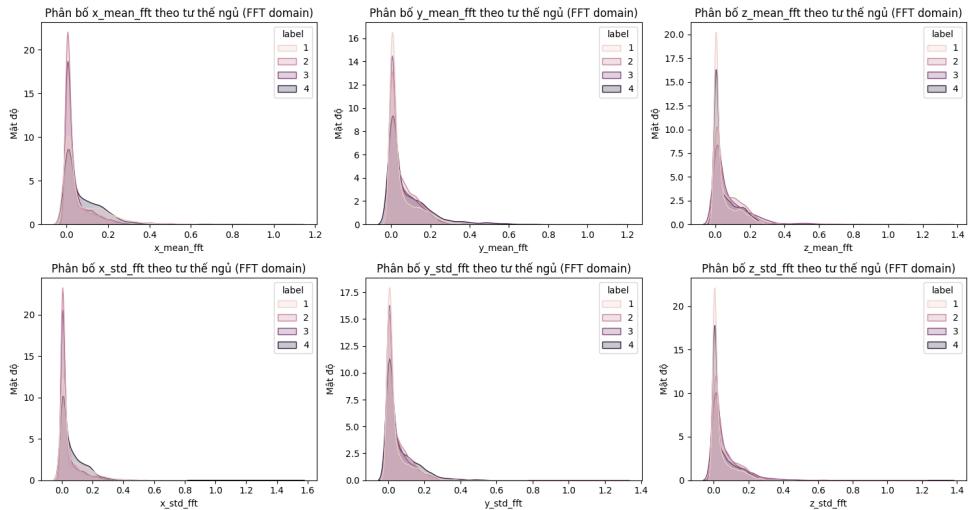
Bảng 3.1: Các đặc trưng thống kê và tín hiệu được sử dụng trong phân loại tư thế ngủ

<b>Đặc trưng</b>	<b>Mô tả / Công thức</b>
Giá trị trung bình	$\mu_s = \frac{1}{n} \sum_{i=1}^n S_i$
Độ lệch chuẩn	$\sigma_s = \sqrt{\frac{1}{n} \sum_{i=1}^n (S_i - \mu_s)^2}$
Độ lệch tuyệt đối trung bình	$AAD = \frac{1}{n} \sum_{i=1}^n  S_i - \mu_s $
Giá trị nhỏ nhất	$\min(s) = \min(S_1, S_2, \dots, S_n)$
Giá trị lớn nhất	$\max(s) = \max(S_1, S_2, \dots, S_n)$
Hiệu số lớn nhất - nhỏ nhất	$\max(s) - \min(s)$
Trung vị	$\text{Median}(s) = \text{median}(S_1, S_2, \dots, S_n)$
Độ lệch tuyệt đối trung vị	$MAD = \frac{1}{n} \sum_{i=1}^n  S_i - \text{Median}(s) $
Khoảng tứ phân vị	$IQR = \text{percentile}(75) - \text{percentile}(25)$
Số giá trị âm	$\#(S_i < 0)$
Số giá trị dương	$\#(S_i > 0)$
Số giá trị lớn hơn trung bình	$\#(S_i > \mu_s)$
Số đỉnh (local maxima)	Số lượng điểm cực đại cục bộ trong chuỗi tín hiệu
Độ lệch (Skewness)	$\frac{1}{n\sigma_s^3} \sum_{i=1}^n (S_i - \mu_s)^3$
Độ nhọn (Kurtosis)	$\frac{1}{n\sigma_s^4} \sum_{i=1}^n (S_i - \mu_s)^4$
Năng lượng tín hiệu	$\sum_{i=1}^n S_i^2$
Gia tốc tổng hợp trung bình	$\frac{1}{n} \sum_{i=1}^n \sqrt{x_i^2 + y_i^2 + z_i^2}$
Tổng độ lớn tín hiệu (SMA)	$\frac{1}{n} \sum_{i=1}^n ( x_i  +  y_i  +  z_i )$



Hình 3.11: Phân bố của đặc trưng trung bình, độ lệnh chuẩn và năng lượng trên miền thời gian của tập huấn luyện

càng tin tưởng về luận điểm rằng với tư thế ngủ, các đặc trưng miền thời gian sẽ có nhiều ý nghĩa hơn và trong đó cũng sẽ có những nhóm thực sự nổi bật để có thể phân loại tư thế ngủ.



Hình 3.12: Phân bố của đặc trưng trung bình, độ lệnh chuẩn trên miền tần số của tập huấn luyện

### 3.3.3. Lựa chọn tham số

Sau khi xác định được 69 đặc trưng cả miền thời gian và tần số, tác giả sẽ chạy qua các vòng lặp với các dải tham số cài đặt sẵn với 5 mô hình đã lựa chọn LR, RF, SVM, GB và ANN như Bảng 3.2.

Bảng 3.2: Kết quả lựa chọn tham số cho các mô hình học máy

Mô hình	Số tổ hợp	Lần thử	Tham số tối ưu	Xác thực chéo	Độ chính xác
<b>RF</b>	288	864	n_estimators=20, max_depth=10, min_samples_split=2, min_samples_leaf=1, max_features='sqrt', bootstrap=True	<b>98.82</b>	<b>99.45</b>
<b>LR</b>	54	162	C=5, max_iter=10, penalty='l2', solver='newton-cg'	96.55	96.30
<b>SVM</b>	216	648	C=1, kernel='linear', degree=2, gamma='scale', decision_function_shape='ovo'	96.62	94.06
<b>GB</b>	216	648	n_estimators=10, learning_rate=0.1, max_depth=4, max_features='sqrt', subsample=0.8	<b>97.51</b>	<b>99.51</b>
<b>NN</b>	32	96	neurons_1=8, neurons_2=8, learning_rate=0.01, batch_size=16, epochs=20	98.01	95.83

Bảng 3.2 cho thấy mô hình RF và GB được đánh giá về mặt độ chính xác là ổn định nhất cả quá trình xác thực chéo và tập kiểm tra. Hai mô hình SVM và LR có độ chính xác thấp nhất trong 5 mô hình nhưng vẫn cho độ chính xác > 94%. Mô hình NN được xây dựng không thông qua 69 đặc trưng mà dùng trực tiếp giá trị gia tốc thô với cửa sổ 10 mẫu (30 giá trị x, y, z) để huấn luyện cho kết quả ẩn tượng đối với tập huấn luyện

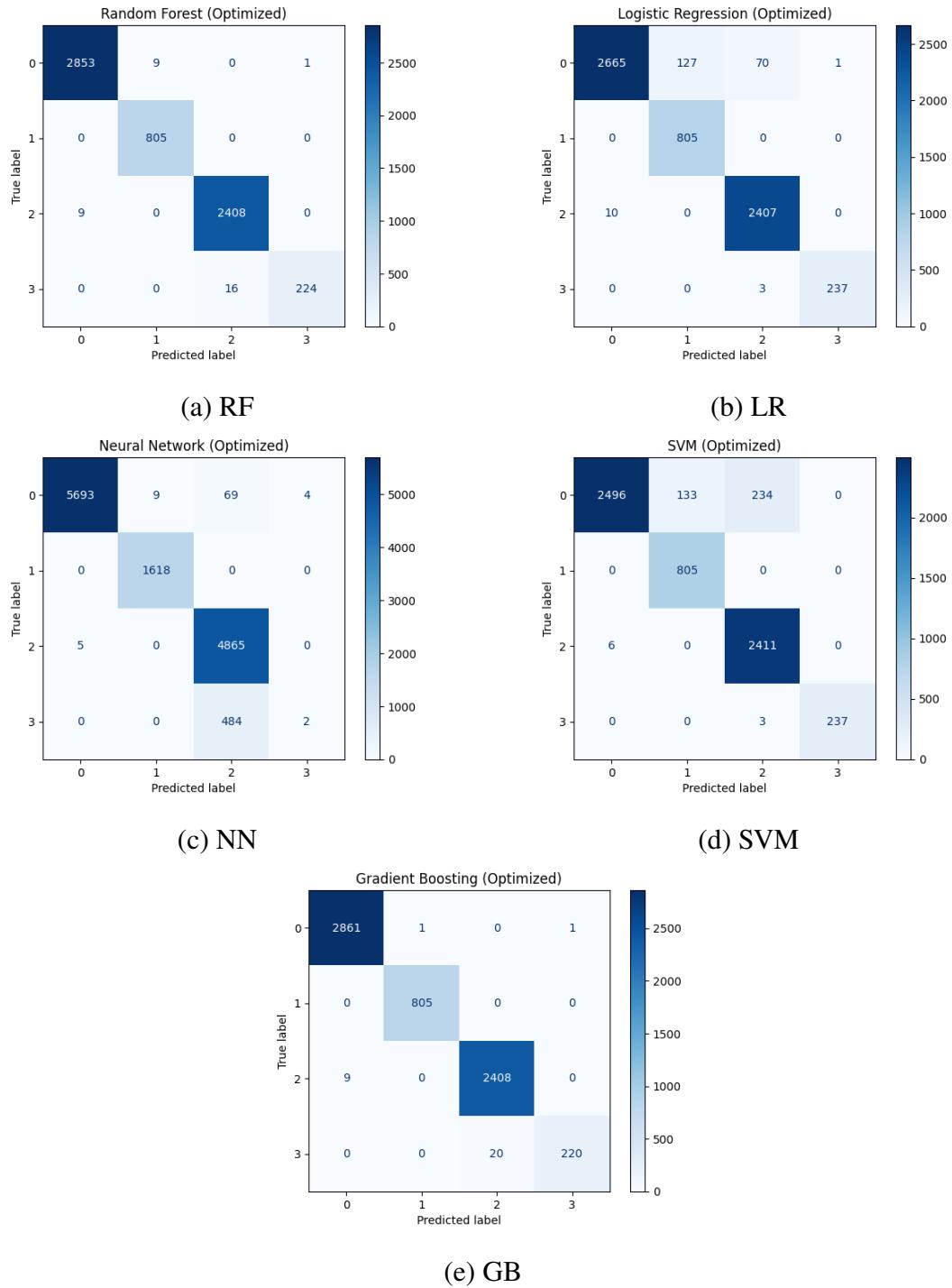
nhưng bị suy giảm so với tập kiểm tra. Hình 3.13 cũng chỉ ra rằng, đối với mô hình NN, hai tư thế nghiêng phải và ngửa đang bị suy luận sai. Việc này cũng được tác giả phân tích biểu đồ phân bố đối với trực z bên trên. Ngoài ra, SVM có những sự nhầm lẫn ở hai tư thế là nghiêng trái và nghiêng phải.

Bên cạnh đó, tác giả cũng phân tích thêm các đặc trưng ảnh hưởng như thế nào đến các mô hình có tham số tốt nhất (không có ANN vì lý do ANN không dùng 69 đặc trưng) Hình 3.14. Đúng với nhận định từ trước, các tính năng miền thời gian chiếm nhiều ý nghĩa lên mô hình đặc biệt là đặc trưng được trích xuất từ hai trực x, y và các đặc trưng tổng hợp từ 3 trực. Đặc trưng miền tần số đa phần đứng vị trí cuối cho thấy không có nhiều ý nghĩa đối với các tham số đã chọn.

Hình 3.14 (a) và (d) khá tương đồng về mặt ý nghĩa của các đặc trưng trên lên mô hình RF và GB. Đây cũng thể xem là do 2 mô hình đều có cơ chế xây dựng dựa trên các cây quyết định. Hình 3.14 (b) chỉ ra LR thì lại ưu tiên hơn các đặc trưng được tổng hợp từ ba trực. Còn đối với SVM Hình 3.14 (b), đã xác định rõ các véc-tơ hỗ trợ nên đang tính các tính năng còn lại đều có giá trị 0.

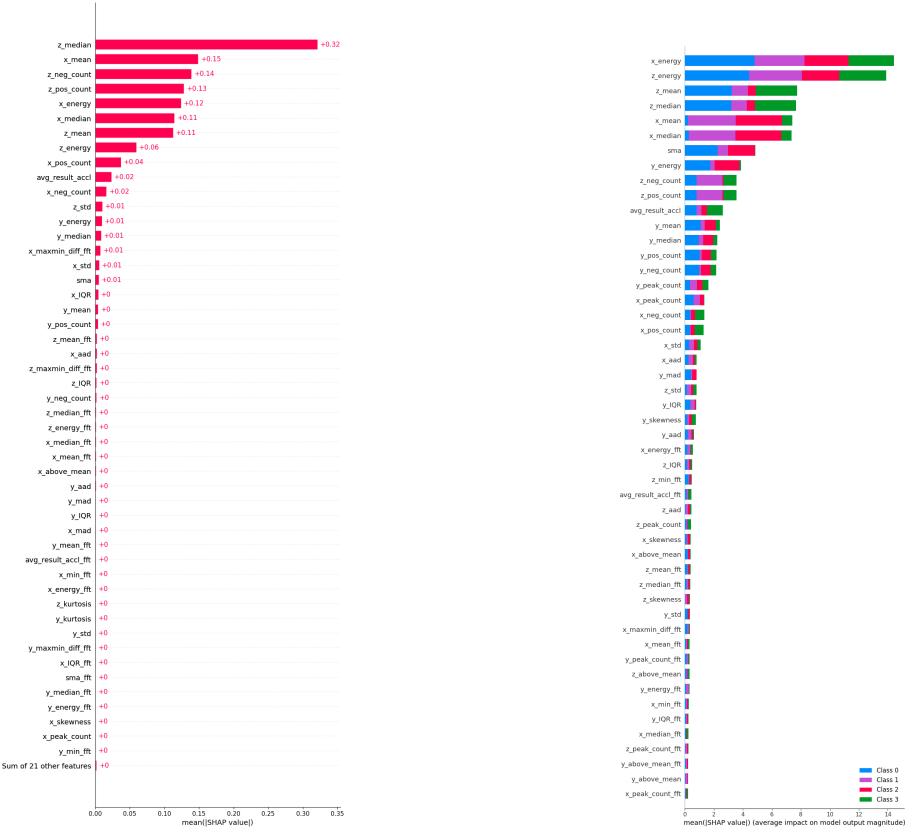
Ngoài ra, tác giả cũng nhận thấy có 15 đặc trưng luôn xuất hiện vị trí đầu của tất cả các mô hình bao gồm: "z\_median", "z\_mean", "x\_mean", "x\_energy", "z\_pos\_count", "z\_neg\_count", "x\_median", 'z\_energy', 'avg\_result\_accl', 'x\_neg\_count', 'z\_std', 'x\_pos\_count', 'y\_median'.

Để hiểu sâu hơn về ý nghĩa các đặc trưng này, tác giả xây dựng 8 kịch bản để đánh giá. Việc này không chỉ có ý nghĩa về mặt phân tích đặc trưng và còn muốn làm rõ hơn về mặt kích thước mô hình để phù hợp triển khai lên biên.



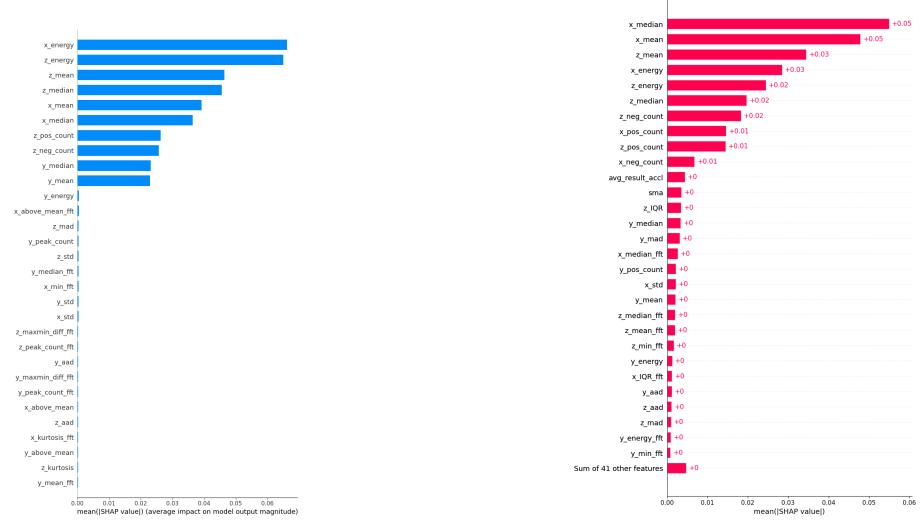
Hình 3.13: Ma trận nhầm lẫn của các mô hình học máy

*Ghi chú:* Nhãn 0 - tư thế ngửa; 1 - nghiêng phải; 2 - nghiêng trái; 3 - sấp.



(a) RF

(b) LR



(c) SVM

(d) GB

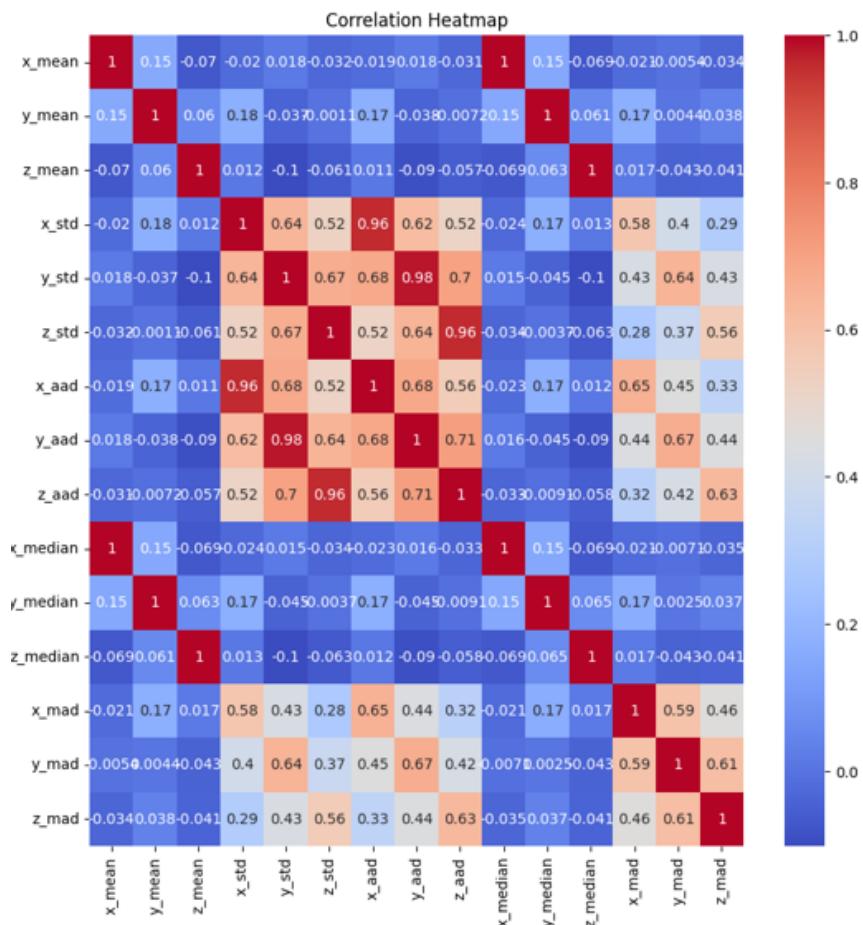
Hình 3.14: Ý nghĩa của các đặc trưng lên mô hình

### 3.3.4. Kịch bản kiểm thử

Lựa chọn đặc trưng là một bước quan trọng trong quá trình xây dựng mô hình học máy, giúp giảm chiều dữ liệu, cải thiện hiệu quả huấn luyện, rút ngắn thời gian tính toán và hạn chế hiện tượng quá khớp. Nguyên lý chung là các đặc trưng hiệu quả phải có mối tương quan cao với mục tiêu là phát hiện tư thế ngử.

Phân tích ma trận tương quan (Hình 3.15) đã cho thấy một số cặp đặc trưng có mức tương quan rất cao, điển hình như  $x_{\text{std}}$  và  $x_{\text{aad}}$  ( $r = 0.98$ ), hay  $y_{\text{std}}$  và  $y_{\text{aad}}$  ( $r = 0.68$ ). Điều này gợi ý rằng có thể loại bỏ một phần các đặc trưng trùng lặp nhằm giảm độ phức tạp mô hình mà vẫn giữ được thông tin cốt lõi.

Kết hợp với phân tích ở phần trên tác giả chọn lựa được 8 kịch bản bao gồm thay đổi 3 thông số chính là: độ dài cửa sổ - độ phủ, đặc trưng miền và số lượng đặc trưng Bảng 3.3.



Hình 3.15: Ma trận tương quan giữa các đặc trưng trích xuất.

Bảng 3.3: Các kịch bản lựa chọn và sử dụng đặc trưng trong nghiên cứu

Kịch bản	Mô tả
1	Dùng cửa sổ 20 mẫu (50% độ phủ), với toàn bộ đặc trưng.
2	Dùng cửa sổ 20 mẫu (50% độ phủ), với toàn bộ đặc trưng trong miền thời gian.
3	Dùng cửa sổ 20 mẫu (50% độ phủ), với toàn bộ đặc trưng trong miền tần số.
4	Dùng cửa sổ 20 mẫu (50% độ phủ), với đặc trưng miền thời gian, loại bỏ các đặc trưng có tương quan $> 95\%$ .
5	Dùng cửa sổ 20 mẫu (50% độ phủ), với 15 đặc trưng quan trọng nhất theo giá trị đứng đầu.
6	Dùng cửa sổ 30 mẫu (50% độ phủ), với 15 đặc trưng đứng đầu.
7	Dùng cửa sổ 10 mẫu (50% độ phủ), với 15 đặc trưng đứng đầu.
8	Dùng cửa sổ 20 mẫu (25% độ phủ), với 15 đặc trưng đứng đầu.

### 3.3.5. Đánh giá kết quả

Sau khi xác định các kịch bản, tác giả tiến hành huấn luyện trên cùng một máy tính và đưa ra kết quả về độ chính xác Hình 3.16.

Cho thấy, kịch bản 3 (màu đỏ) là toàn bộ các đặc trưng trên miền tần số hoàn toàn không phù hợp để phân loại tư thế ngủ. Điều này cũng được chứng minh tại ở phần trên. Các mô hình RF và GB luôn cho độ chính xác cao nhất và ổn định nhất ở toàn bộ các bộ dữ liệu huấn luyện. Các kịch bản 5, 6, 7, 8 đưa trên chỉ 15 đặc trưng có vị trí đầu hầu như không thay đổi độ chính xác nhiều. Điều này cho thấy, việc lựa chọn này phù hợp đối với bài toán tư thế ngủ. Trong kịch bản 6, có độ chính xác cao nhất lên tới 99.84% đối với mô hình GB với bộ dữ liệu cửa sổ 30 mẫu. Kịch bản 7 có kết quả ẩn tượng nhất. Trong đó, độ chính xác các mô hình đều được cải thiện rõ rệt ngay cả mô hình LR và SVM. Điều này thực sự có ý nghĩa vì độ rộng cửa sổ có ảnh hưởng trực tiếp đến việc tính toán, thời gian phản hồi kết quả trên biên. Với NN như đánh giá trong phần lựa chọn tham số, khả năng tự trích xuất tính năng từ chỉ 2 lớp ẩn cũng đã cho kết quả đến 98%, nhưng giảm nhẹ ở tập kiểm tra 95.5%.

Kịch bản	RF	LR	SVM	GB	ANN
1	99,45	96,36	94,06	<b>99,51</b>	
2	99,38	96,62	93,93	<b>99,68</b>	
3	43,37	46,81	<b>49,19</b>	44,95	
4	98,61	96,71	94,10	<b>99,37</b>	
5	<b>99,46</b>	96,57	94,86	<b>99,67</b>	<b>95,58</b>
6	98,46	95,57	94,62	<b>99,84</b>	
7	98,97	97,15	98,04	<b>99,57</b>	
8	<b>99,09</b>	96,55	96,56	<b>99,45</b>	

Hình 3.16: Kết quả độ chính xác của 8 kịch bản

Hình 3.17 minh họa ma trận nhầm lẫn giữa các mô hình. Qua đó cho thấy việc sử dụng các đặc trưng vị trí đầu cho kết quả thật sự hiệu quả. Đây cũng là điểm mới luận



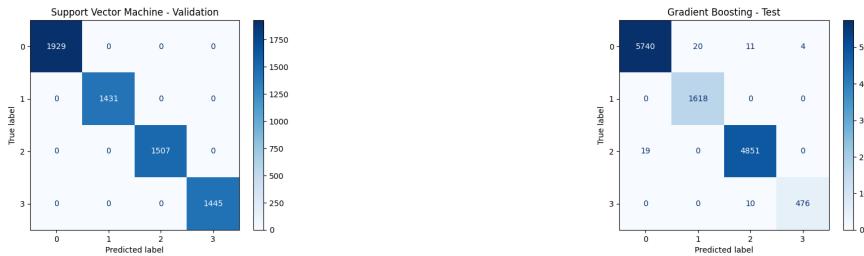
(a) RF trên tập huấn luyện

(b) RF trên tập kiểm thử



(c) SVM trên tập huấn luyện

(d) SVM trên tập kiểm thử



(e) GB trên tập huấn luyện

(f) GB trên tập kiểm thử



(g) LR trên tập huấn luyện

(h) LR trên tập kiểm thử

Hình 3.17: Ma trận nhầm lẫn trong kịch bản 8

vẫn tìm ra để triển khai bài toán phân loại tư thế ngủ lên biên.

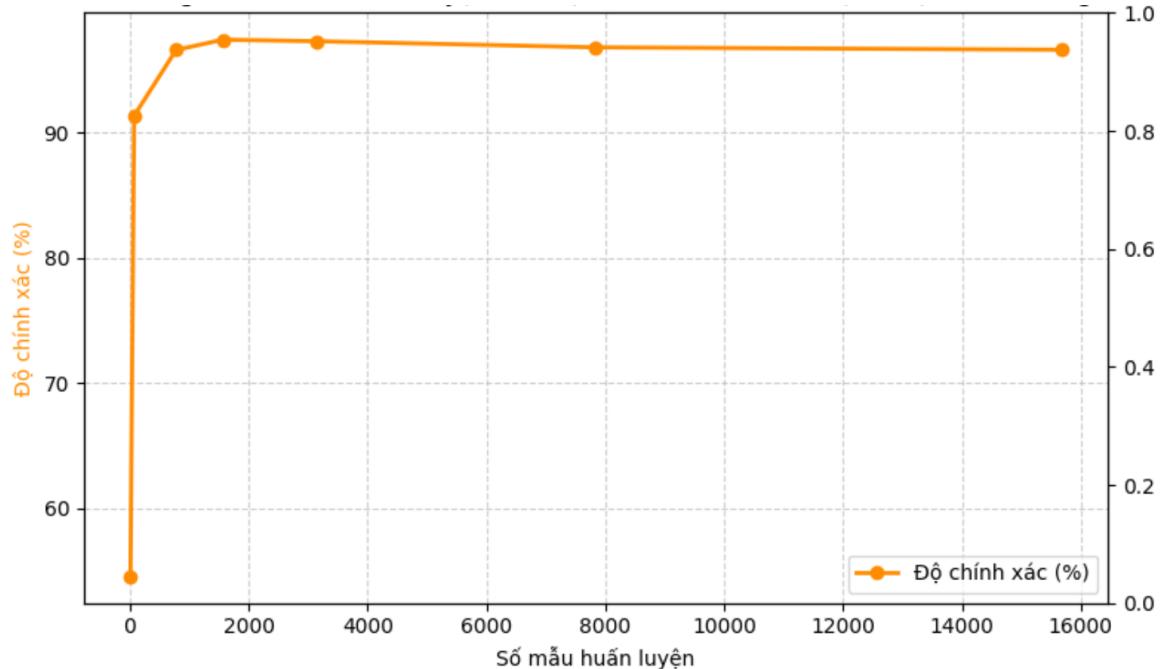
Kịch bản	RF	LR	SVM	KB	ANN
1	128,35	6,79	129,19	133,58	
2	86,73	4,63	51,73	135,31	
3	2.183,68	3,91	4.033,97	140,75	
4	169,71	4,22	46,48	136,15	
5	48,10	2,90	11,19	130,39	
6	37,79	2,90	9,79	130,59	
7	66,85	2,90	12,60	130,51	
8	48,10	2,90	11,19	130,39	31,94

Hình 3.18: Kết quả kết thước mô hình của 8 kịch bản (đơn vị: kb)

Hình 3.18 cho cái nhìn toàn diện về kích thước mô hình của 8 kịch bản. Đầu tiên dễ nhận thấy là ở kịch bản 3, hai mô hình RF và SVM không hội tụ được với bộ tham số đã tìm ra dẫn đến kích thước mô hình tăng vọt. Mô hình GB có kích thước hầu như không thay đổi với tất cả kịch bản, nhưng luôn nằm trong nhóm lớn nhất. Ở các kịch bản 5, 6, 7, 8 cũng thể hiện rõ giảm số lượng đặc trưng không nhiều ý nghĩa xuống cũng đồng thời giảm được kích thước mô hình nhưng độ chính xác hầu như vẫn giữ nguyên. Dễ dàng nhìn thấy kịch bản thứ nhất dùng tất cả các tính năng và kịch bản 7, tất cả kích thước mô hình ít nhất giảm đi 2 lần, riêng SVM đạt tới 10 lần cho thấy việc lựa chọn đúng đặc trưng thực sự có ý nghĩa mặt tính toán. Riêng đối với mô hình NN, do chỉ đánh giá với 1 kịch bản nhưng cũng cho thấy tiềm năng mạnh mẽ khi chỉ với kích thước 31.94 kb đã cho độ chính xác ở tập huấn luyện lên tới 98%.

Tất cả những đánh giá trên, tác giả quyết định lựa chọn đại diện 2 mô hình là LR - truyền thống và ANN - học sâu để tiến hành thử nghiệm triển khai trên biên. Lựa chọn LR mà không phải 3 mô hình còn lại có lý do đầu tiên là kích thước ổn định, nhỏ nhất và có độ chính xác lên tới 97%. Việc này sẽ phù hợp với phần cứng của nhóm. Còn NN, lý do tác giả muốn kiểm tra cả học sâu học động hiệu quả hay không trên môi trường tại biên. Ngoài ra, trong tương lai ngoài phân loại tư thế ngủ, phần cứng của nhóm còn phải gánh thêm những bài toán khác với kích thước mô hình, kích thước dữ liệu đầu vào cho

mô hình lớn hơn rất nhiều lần như tín hiệu audio. Vậy nên LR sẽ giúp san sẻ tài nguyên phân cứng, NN sẽ làm tiền đề để phát triển mô hình cho bài toán khác tại biên.



Hình 3.19: Mức độ ảnh hưởng của số lượng mẫu đến độ chính xác mô hình LR

Ngoài ra, tác giả còn phân tích thêm mức độ ảnh hưởng của số lượng mẫu đến mô hình Hình 3.19. Biểu đồ cho thấy, đường cong độ chính xác tăng vọt từ khi bắt đầu đến ngưỡng khoảng 2000 mẫu dữ liệu. Sau đó, đường có xu hướng đi ngang. Từ biểu đồ này phần nào cũng xác định được số lượng mẫu tác giả thu thập từ 25 tình nguyện viên đã đủ đối với bài toán phân loại tư thế ngủ. Ngoài ra, nó còn cho thấy tiềm năng để giảm tiếp kích thước mô hình bằng cách giảm số lượng mẫu huấn luyện.

### 3.4. Triển khai mô hình lên phần cứng

Trong phần này, tác giả sẽ trình bày kết quả đạt được khi triển khai hai mô hình đã chọn lên phần cứng của nhóm.

### 3.5. Chuyển đổi mô hình

Đầu tiên, tác giả sẽ chuẩn hóa từ mô hình trên máy chủ trung tâm thành định dạng mô hình trên vi điều khiển. Đối với mô hình LR, sau khi mô hình LR được huấn luyện và hiệu chỉnh, toàn bộ tham số của mô hình (bao gồm trọng số và hệ số chêch) cùng với thông số chuẩn hóa (scaler) được lưu trữ lại bằng thư viện Joblib<sup>1</sup>. Cơ chế này sử dụng để trích xuất toàn bộ tham số huấn luyện (bao gồm ma trận trọng số, vector bias, và các giá trị cực tiểu - cực đại của scaler) dưới dạng hằng số ngôn ngữ C.

Sau khi mô hình ANN được huấn luyện, việc chuyển đổi mô hình sang định dạng triển khai thực tế trên thiết bị biên trở thành một bước bắt buộc. Mô hình này ở định dạng mặc định (.h5) chứa nhiều thành phần không cần thiết cho quá trình suy luận. Hơn nữa, các định dạng này yêu cầu môi trường Python<sup>2</sup> và thư viện TensorFlow<sup>3</sup> - vốn không khả thi đối với các nền tảng vi điều khiển có giới hạn tài nguyên nghiêm ngặt về bộ nhớ (RAM = 256 KB) và bộ nhớ lưu trữ (Flash = 1 MB).

Sau khi, mô hình ANN được huấn luyện và hiệu chỉnh. Sau đó, mô hình được chuyển đổi sang định dạng **TensorFlow Lite (TFLite)** bằng thư viện TFLiteConverter, giúp nén và tối ưu mô hình cho thiết bị nhúng.

Thư viện TFLiteConverter cung cấp nhiều thuộc tính điều chỉnh quan trọng ảnh hưởng trực tiếp đến chất lượng suy luận cũng như khả năng thu gọn mô hình. Các thuộc tính như optimizations cho phép lựa chọn các chiến lược tối ưu hóa toàn cục; representative\_dataset đóng vai trò thiết yếu trong lượng tử hóa số nguyên; Thư viện cũng cung cấp tùy chọn các phép toán tùy chỉnh, loại bỏ metadata; Phương thức trung tâm của lớp là convert(), nơi toàn bộ đồ thị TensorFlow được ánh xạ sang mô hình TFLite dưới dạng chuỗi byte.

Trong phạm vi luận văn này, kích thước mô hình mạng nơ-ron nhân tạo (ANN) sau huấn luyện chỉ vào khoảng 35 kB, cho thấy chưa cần áp dụng các kỹ thuật tối ưu hóa

---

<sup>1</sup>Joblib là một thư viện trong Python, được thiết kế tối ưu cho việc tuần tự hoá và lưu trữ các đối tượng dữ liệu lớn như mô hình học máy, tham số, và bộ chuẩn hóa

<sup>2</sup>Python là một ngôn ngữ lập trình bậc cao, mã nguồn mở, hỗ trợ mạnh mẽ cho các ứng dụng khoa học dữ liệu, trí tuệ nhân tạo và học máy

<sup>3</sup>TensorFlow là thư viện mã nguồn mở do Google phát triển, hỗ trợ xây dựng và huấn luyện các mô hình học sâu. Trong nghiên cứu này, TensorFlow được sử dụng để huấn luyện mô hình mạng nơ-ron nhân tạo và chuyển đổi sang định dạng .tflite để triển khai trên vi điều khiển.

mạnh như cắt tỉa trọng số hay lượng tử hóa từ định dạng float32 xuống float8. Vì vậy, tác giả chỉ sử dụng TFLiteConverter với mục tiêu loại bỏ những thành phần không thiết yếu, qua đó bảo đảm mô hình giữ nguyên độ chính xác trong khi vẫn đáp ứng yêu cầu triển khai trên thiết bị biên.

Mô hình sau chuyển đổi được lưu thành tệp nhị phân `position.tflite`, đồng thời kích thước được kiểm tra để đánh giá khả năng lưu trữ trên vi điều khiển. Cuối cùng, tệp này được chuyển đổi sang dạng mảng nhị phân (`position_model.h`). Toàn bộ quy trình này nhằm tối ưu mô hình và giúp hệ thống thực hiện suy luận trực tiếp trên vi điều khiển ARM Cortex-M4.

### 3.6. Triển khai tại biên

Trong bước triển khai thực tế, mô hình Logistic Regression (LR) được ánh xạ trực tiếp sang mã C/C++ thông qua ba thành phần chính: (i) tệp `model.h` lưu trữ toàn bộ tham số huấn luyện (trọng số, hệ số bias, giá trị chuẩn hoá); (ii) tệp `predict.h` định nghĩa cơ chế suy luận bằng phép tính tuyến tính kết hợp Softmax; và (iii) chương trình chính điều khiển cảm biến, trích xuất đặc trưng, chuẩn hoá và gọi hàm dự đoán. Cách tổ chức này giúp mô hình hoạt động độc lập hoàn toàn trên vi điều khiển mà không cần bất kỳ thư viện học máy ngoài nào.

Thực nghiệm cho thấy toàn bộ quá trình suy luận được thực hiện chỉ với các phép toán cơ bản (*cộng, nhân, căn bậc hai, hàm mũ*), tiêu tốn rất ít tài nguyên tính toán và đạt thời gian xử lý ở mức micro giây cho mỗi cửa sổ dữ liệu. Điều này khẳng định tính phù hợp của LR trong bối cảnh triển khai trên thiết bị nhúng có bộ nhớ và công suất xử lý hạn chế.

Từ góc độ khoa học, việc triển khai LR theo cách này minh chứng rằng một mô hình học máy có thể được rút gọn thành tập tham số tĩnh và tái hiện chính xác trên chip, đồng thời duy trì độ chính xác ở mức chấp nhận được cho ứng dụng giám sát sức khoẻ thời gian thực. Chiến lược **tối giản mô hình** này là minh chứng rõ rệt cho tính khả thi của Edge AI: đảm bảo độ trễ thấp, tính riêng tư dữ liệu và khả năng vận hành bền vững trong môi trường hạn chế tài nguyên.

Khác với LR, mô hình Neural Network (NN) được triển khai trên chip thông qua

định dạng **TensorFlow Lite Micro (TFLM)**. Khối MicroInterpreter trong TFLM chịu trách nhiệm đọc mô hình, thực thi các toán tử (Dense, ReLU, Softmax), và trả về xác suất dự đoán cho từng lớp tư thế. Kết quả cuối cùng được xác định bằng cách chọn lớp có xác suất cao nhất.

Ý nghĩa khoa học của phương thức triển khai này nằm ở chỗ: thay vì trích xuất đặc trưng thủ công như với LR, NN có khả năng học trực tiếp từ dữ liệu thô, từ đó giảm thiểu sự phụ thuộc vào các bước tiền xử lý. Mặc dù chi phí tính toán cao hơn, NN có ưu thế trong việc nắm bắt quan hệ phi tuyến phức tạp giữa các tín hiệu.

Như vậy, hai mô hình LR và NN phản ánh hai chiến lược bổ sung cho nhau: LR tối giản, phù hợp khi ưu tiên tốc độ và tài nguyên, trong khi NN khai thác tối đa dữ liệu thô nhờ khả năng biểu diễn phi tuyến, phù hợp cho các ứng dụng đòi hỏi độ chính xác cao và tính khái quát.

### 3.6.1. Đánh giá hiệu suất và tài nguyên

Các thí nghiệm được tiến hành trên vi điều khiển nRF52840 và được so sánh trong điều kiện triển khai thực tế. Ngoài ra, để đảm bảo tính khách quan và khả năng tái lập, toàn bộ quá trình huấn luyện mô hình, biên dịch chương trình và nạp xuống vi điều khiển được thực hiện trên một máy tính xách tay có cấu hình phần cứng như sau:

- **Bộ xử lý:** Intel(R) Core(TM) Ultra 5 226V, tốc độ xung nhịp 2.10 GHz.
- **Bộ nhớ trong:** 16.0 GB DDR5, tốc độ 8533 MT/s.
- **Ổ lưu trữ:** SSD dung lượng 954 GB (trong đó đã sử dụng 188 GB).
- **Card đồ họa:** Intel(R) Arc(TM) 130V, dung lượng bộ nhớ đồ họa 8 GB (VRAM khả dụng 128 MB cho hệ thống).

Việc mô tả chi tiết môi trường phần cứng nhằm giúp đảm bảo tính minh bạch, đồng thời cung cấp thông tin tham chiếu để các nghiên cứu sau có thể so sánh hoặc tái hiện kết quả. Trong bối cảnh học máy nhúng, hiệu năng biên dịch và tốc độ nạp chương trình phụ thuộc không chỉ vào kiến trúc của vi điều khiển mà còn chịu ảnh hưởng từ cấu hình máy tính host, do đó việc báo cáo chi tiết hệ thống thử nghiệm là cần thiết.

Bảng 3.4 trình bày sự so sánh chi tiết hai mô hình khi triển khai trên vi điều khiển nRF52840. Ngoài các tiêu chí về dung lượng bộ nhớ và độ phức tạp, kết quả thực nghiệm còn chỉ ra sự khác biệt rất lớn về **hiệu năng suy luận**. Cụ thể, mô hình LR đạt thời gian xử lý trung bình  $501 \mu s$  trên mỗi mẫu, trong khi NN mất tới  $8000 \mu s$ , tức là chậm hơn khoảng 16 lần. Đây là một yếu tố quan trọng, đặc biệt trong các ứng dụng thời gian thực nơi độ trễ (latency) quyết định khả năng đáp ứng của hệ thống.

Bảng 3.4: So sánh hiệu năng LR và ANN trên nRF52840

Tiêu chí	Logistic Regression (LR)	Neural Network (NN)
Dung lượng Flash sử dụng	115,208 bytes (11%)	363,520 bytes (36%)
Dung lượng RAM sử dụng	46,632 bytes (17%)	60,672 bytes (23%)
Thời gian upload	4.9 s (29 pages)	15 s (89 pages)
Thời gian suy luận	$501 \mu s$	$8000 \mu s$
Kích thước mô hình	2.9 kb	23 kb

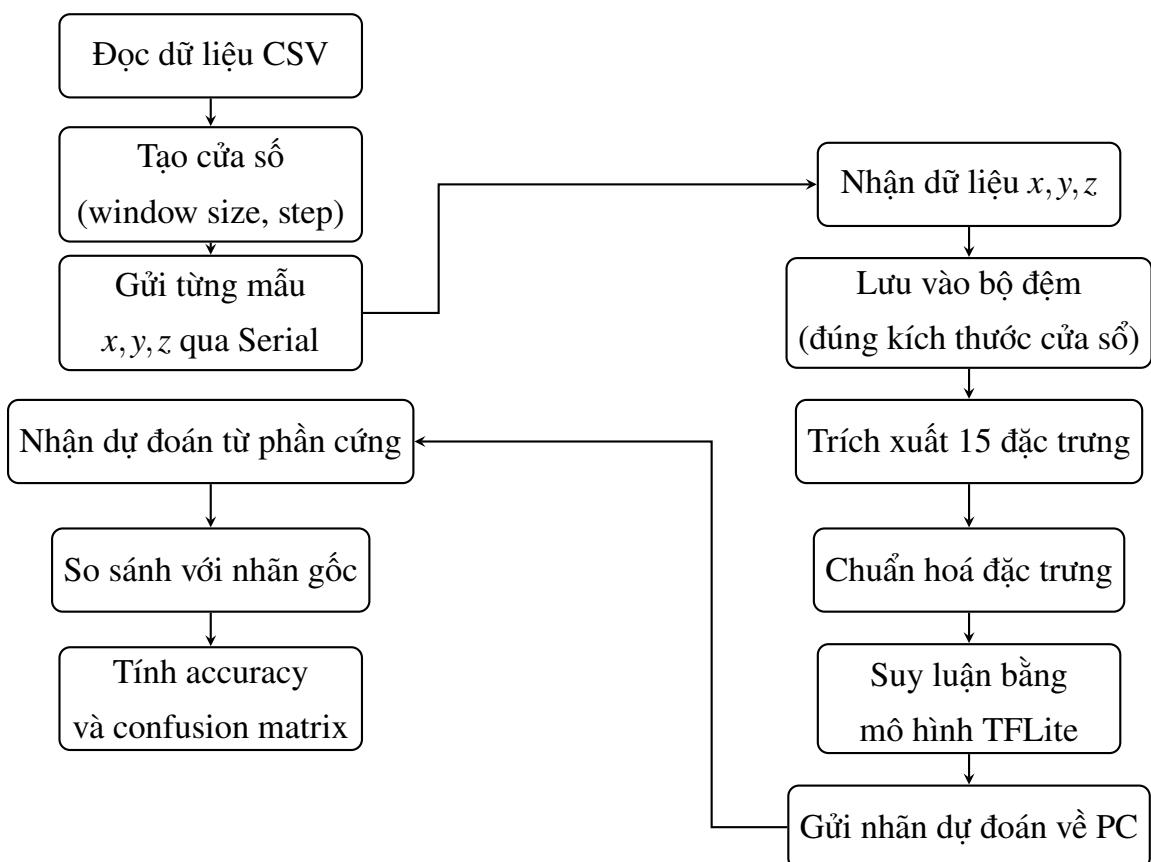
Kết quả cho thấy LR có ưu thế vượt trội về tốc độ và mức tiêu thụ tài nguyên, điều này khiến LR đặc biệt thích hợp cho các hệ thống nhúng giới hạn phần cứng và yêu cầu phản hồi tức thời (ví dụ: phân loại tư thế ngủ). Tuy nhiên, NN lại có khả năng khai thác tốt hơn các quan hệ phi tuyến giữa dữ liệu cảm biến, giúp tăng độ chính xác trong các bài toán phức tạp như nhận diện chuyển động liên tục hoặc phân loại đa trạng thái.

Điểm mấu chốt ở đây là **sự đánh đổi giữa tốc độ và khả năng biểu diễn**: LR nhanh và nhẹ, nhưng hạn chế về năng lực mô hình; trong khi NN tiêu tốn nhiều tài nguyên hơn, song mang lại độ chính xác cao và khả năng tổng quát hóa mạnh mẽ. Do đó, việc lựa chọn mô hình không mang tính tuyệt đối, mà cần được cân nhắc dựa trên mục tiêu ứng dụng: *ưu tiên thời gian thực và tiết kiệm năng lượng* (chọn LR) hoặc *ưu tiên độ chính xác và khả năng mở rộng* (chọn NN).

Để đánh giá độ chính xác của 2 mô hình đã triển khai trên biên, tác giả thiết kế mô hình để truyền toàn bộ dữ liệu tập kiểm tra vào phần cứng và nhận lại giá trị suy luận. Quy trình đánh giá mô hình trên vi điều khiển được tổ chức theo hai nhánh chính: luồng xử lý trên MCU và luồng kiểm thử trên máy tính. Phía máy tính chịu trách nhiệm tải dữ liệu gia tốc trong tệp CSV, tạo các cửa sổ trượt kích thước xác định, và lần lượt truyền từng giá trị (x,y,z) của mỗi cửa sổ xuống MCU thông qua giao tiếp Serial. Mỗi khi MCU

nhận đủ N mẫu, thiết bị thực thi toàn bộ pipeline suy luận gồm trích xuất 15 đặc trưng, chuẩn hoá các đặc trưng theo min-max, và đưa vào mô hình đã được chuyển đổi sang định dạng TFLite. Kết quả phân lớp được trả về máy tính dưới dạng một số nguyên đại diện cho tư thế ngử.

Trên máy tính, mỗi dự đoán nhận được được so sánh với nhãn gốc của cửa sổ tương ứng để tính toán độ chính xác tổng thể, đồng thời xây dựng ma trận nhầm lẫn và bộ chỉ số đánh giá như precision, recall và F1-score. Cách tiếp cận này cho phép mô phỏng chính xác điều kiện triển khai thực tế trên phần cứng, từ đó kiểm định mức độ ổn định của mô hình sau khi chuyển đổi và tích hợp vào vi điều khiển ARM Cortex-M4.



Hình 3.20: Quy trình đánh giá mô hình trên MCU và đối chiếu kết quả trên PC.

Kết quả thực nghiệm Bảng 3.5 cho thấy sự khác biệt rõ rệt giữa Logistic Regression (LR) và mô hình Neural Network (NN) khi đánh giá trên cùng một tập dữ liệu gồm 12.749 cửa sổ tín hiệu. LR vẫn đạt độ chính xác 97.58%, trong khi NN chỉ đạt 89.73%. Sự chênh lệch này không chỉ phản ánh ưu thế của LR trong điều kiện dữ liệu được chuẩn hóa tốt, mà còn làm nổi bật tính ổn định của LR khi chuyển sang môi trường triển khai

Bảng 3.5: So sánh kết quả mô hình trên tập thử Python và khi triển khai trên MCU

Mục đánh giá	Python Benchmark	MCU thực nghiệm
Tổng số cửa sổ	12 749	12 749
Accuracy	<b>0.9758</b>	<b>0.8973</b>
Ma trận nhầm lẫn	$\begin{bmatrix} 5495 & 190 & 86 & 4 \\ 0 & 1618 & 0 & 0 \\ 18 & 0 & 4852 & 0 \\ 0 & 0 & 11 & 475 \end{bmatrix}$	$\begin{bmatrix} 4950 & 215 & 606 & 4 \\ 0 & 1618 & 0 & 0 \\ 0 & 0 & 4870 & 0 \\ 0 & 0 & 484 & 2 \end{bmatrix}$
Precision (1-4)	1.00, 0.89, 0.98, 0.99	1.00, 0.88, 0.82, 0.33
Recall (1-4)	0.95, 1.00, 1.00, 0.98	0.86, 1.00, 1.00, 0.00
F1-score (1-4)	0.97, 0.94, 0.99, 0.98	0.92, 0.94, 0.90, 0.01

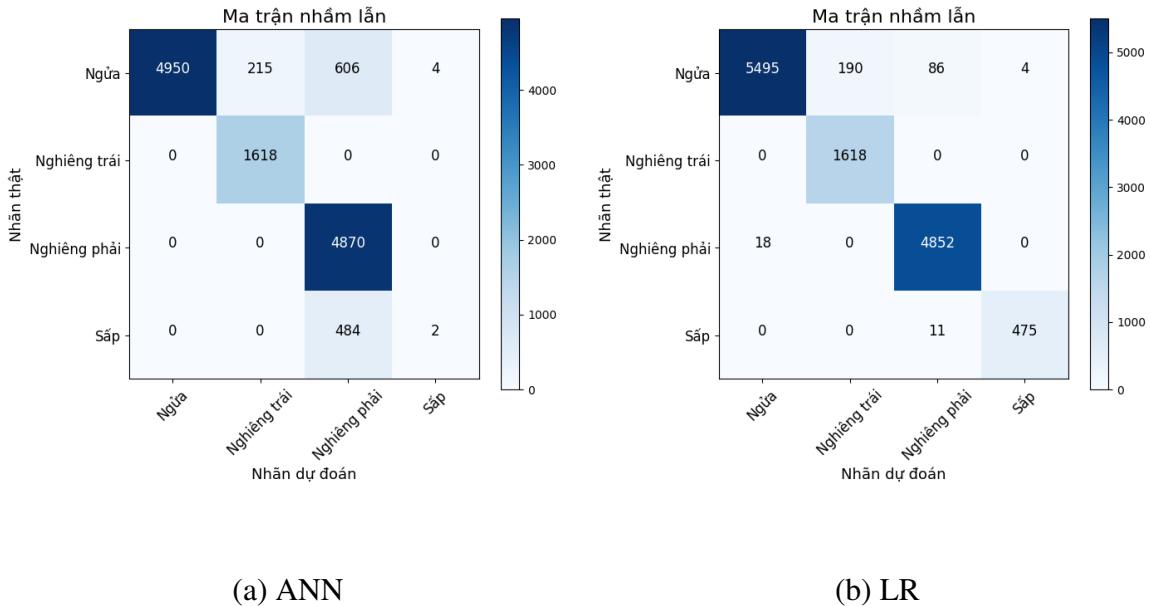
thực, vốn chứa nhiều bất định và sai số đặc trưng của thiết bị biên.

Ở mô hình LR, các chỉ số precision, recall và f1-score đều duy trì ở mức cao cho cả bốn tư thế. Đặc biệt, tư thế sắp - vốn là tư thế dễ nhất và ít xuất hiện - vẫn được nhận dạng với recall 0.98 và f1-score 0.98. Ma trận nhầm lẫn của LR cho thấy các sai lệch chủ yếu mang tính cục bộ và không gây phá vỡ biên phân lớp. Điều này cho thấy mô hình tuyến tính của LR, mặc dù đơn giản về mặt cấu trúc, lại tạo ra mặt phân tách tương đối bền vững đối với các biến động nhỏ trong tín hiệu gia tốc.

Ngược lại, mô hình NN thể hiện sự suy giảm đáng kể về hiệu năng khi triển khai. Mặc dù recall của tư thế 2 và 3 vẫn ở mức tuyệt đối, mô hình gần như thất bại hoàn toàn với tư thế sắp: recall bằng 0. Điều này phản ánh 1 phần độ chính xác đã bị suy hao trong quá trình convert và triển khai trên biên.

Sự khác biệt giữa hai mô hình đặc biệt rõ trong ma trận nhầm lẫn Hình 3.21. Trong khi LR duy trì cấu trúc phân lớp gần như lý tưởng ở cả bốn tư thế, NN xuất hiện nhiều sai lệch nghiêm trọng giữa tư thế ngửa và nghiêng phải, và đặc biệt là không thể nhận dạng tư thế sắp. Điều này cho thấy LR phù hợp hơn cho tín hiệu từ cảm biến gia tốc gắn trên cơ thể.

Từ các kết quả này có thể khẳng định rằng, trong bối cảnh triển khai trên các hệ thống TinyML, LR tỏ ra vượt trội so với NN không chỉ vì mô hình nhỏ gọn hơn mà còn



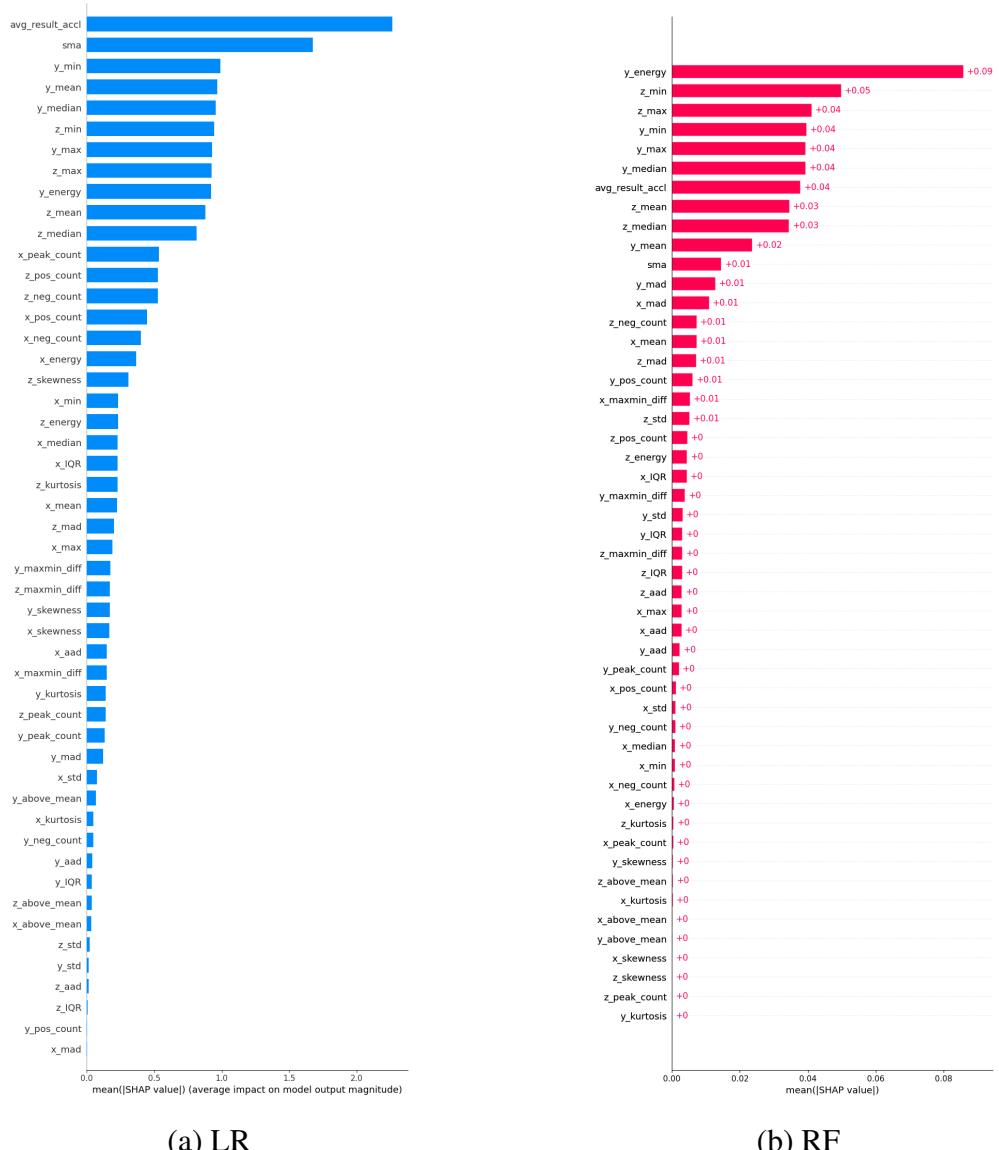
Hình 3.21: Ma trận nhầm lẫn của 2 mô hình triển khai trên biên

vì khả năng duy trì độ ổn định và độ tin cậy khi đối mặt với các biến thiên tín hiệu thực. Điều này cũng có lập luận rằng ưu tiên lựa chọn mô hình tuyến tính tinh gọn, thay vì mạng nơ-ron quy mô nhỏ, là một hướng đi có lợi thế rõ rệt trong các ứng dụng theo dõi tư thế ngủ trên thiết bị đeo chi phí thấp.

## Bài toán phát hiện nằm hoặc không nằm

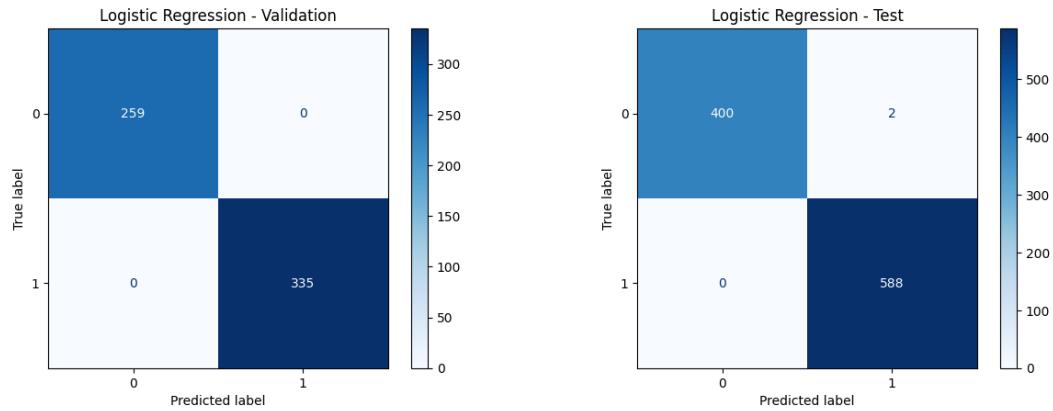
Sau khi triển khai thực tế trên biên, tác giả nhận thấy rằng, hiện nay hệ thống tại biên đang chỉ nhận dạng được 4 nhãn lần lượt ngửa, nghiêng phải, nghiêng trái, sấp chưa đủ đáp ứng thực tế rằng khi ngủ người bệnh có thể đứng lên đi lại hoặc uống nước. Qua đó tác giả mở rộng thêm bài toán xác định nằm hay không nằm. Đây là bài toán sàng lọc trước khi đến bài toán xác định tư thế ngủ. Qua đó tác giả thu thập thêm 8500 mẫu dữ liệu các hoạt động như ngồi, đi lại, uống nước. Tập các tư thế ngủ tác giả chọn lọc 11306 mẫu đủ các 4 tư thế được quy về 1 nhãn là nằm. Tác giả áp dụng nguyên các bước để đánh giá tư thế ngủ bao gồm độ dài cửa sổ 1s, độ phủ 50%, 69 đặc trưng trích xuất. Qua đó mô hình cho kết quả lên tới 100% phân loại 2 nhãn là nằm và không nằm và có kích thước mô hình là 3.8 kb.

Sau khi đánh giá mức độ ảnh hưởng của đặc trưng lên mô hình như Hình 3.22, tác giả chọn ra 6 đặc trưng: 'avg\_result\_accl', 'y\_energy', 'y\_min', 'y\_mean', 'z\_min',



Hình 3.22: Biểu đồ mức độ ảnh hưởng của các đặc trưng lên mô hình với bài toán nằm hay không nằm

'z\_max' để huấn luyện lại mô hình LR. Kết quả độ chính xác có giảm 1 chút nhưng kích thước mô hình còn 1.9 kb Hình 3.23. Điều này hoàn toàn phù với tiêu chí để trải khai đa mô hình trên biên.

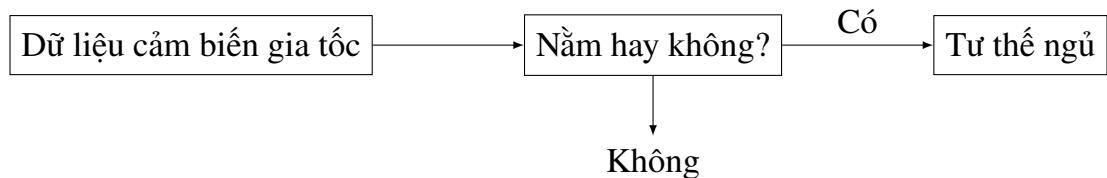


(a) dựa trên toàn bộ đặc trưng

(b) dựa trên 6 tính năng vị trí đầu

Hình 3.23: Ma trận nhầm lẫn phân biệt 2 nhãn nằm và không nằm

Qua đó bài toán phân loại trên biên được nâng cấp thành 3 bước chính:



Việc này khẳng định để thực sự đưa bài toán này vào thực tế, còn rất nhiều bước cần phải hiệu chỉnh để giải quyết các trường hợp thực tế. Tuy nhiên, các kết quả đưa ra cho thấy giải quyết phần nào bài toán phân loại tư thế ngủ tại biên cùng với 1 quy trình từ cảm biến đến mô hình trên biên. Với quy trình này, tác giả tin có thể giải quyết được hầu như các trường hợp phát sinh khác và hướng đến bài toán phân loại OSA.

# KẾT LUẬN

Trong giai đoạn nghiên cứu trước, các mô hình học máy như Logistic Regression, Random Forest, Support Vector Machine và Gradient Boosting đã được áp dụng để phân loại tư thế ngủ với độ chính xác cao. Đồng thời, một mô hình mạng nơ-ron (ANN) cũng đã được triển khai thành công trên phần cứng của nhóm phát triển, chứng minh tính khả thi của việc chạy mô hình trực tiếp trên phần cứng nhúng.

Kèm với hai mô hình đã chạy ổn định trên thiết bị phần cứng của nhóm thiết kế, tác giả tự đánh giá các kết quả này đã hoàn thành mục tiêu của đề tài đề ra từ trước.

Trong tương lai, tác giả đánh giá trên các mô hình học máy khác với bài toán tư thế ngủ để xác định được mô hình phù hợp nhất. Ngoài ra, việc các đặc trưng kết hợp cũng là một định hướng của nhóm. Không dừng lại ở đó, nhóm sẽ tối ưu thêm phần cứng, thêm các cảm biến để hướng tới bài toán phân loại OSA. Phần mềm sẽ được bổ sung các tính năng y học để trở thành công cụ cá nhân hóa.

# Tài liệu tham khảo

- [1] Epstein LJ, Kristo D, Strollo PJ Jr, Friedman N, Malhotra A, Patil SP, Ramar K, Rogers R, Schwab RJ, Weaver EM **and** Weinstein MD; Adult Obstructive Sleep Apnea Task Force of the American Academy of Sleep Medicine. Clinical guideline for the evaluation, management and long-term care of obstructive sleep apnea in adults. **in***J Clin Sleep Med*: 5.3 (june 2009), **pages** 263–276.
- [2] A. V. Benjafield, N. T. Ayas, P. R. Eastwood, R. Heinzer, M. S. M. Ip, M. J. Morrell, C. M. Nunez, S. R. Patel, T. Penzel, J. L. Pépin, P. E. Peppard, S. Sinha, S. Tufik, K. Valentine **and** A. Malhotra. Estimation of the global prevalence and burden of obstructive sleep apnoea: a literature-based analysis. **in***The Lancet Respiratory Medicine*: 7.8 (august 2019), **pages** 687–698. DOI: 10.1016/S2213-2600(19)30198-5.
- [3] S. Duong-Quy, K. Dang Thi Mai, N. Tran Van, H. Nguyen Xuan Bich, T. Huu-Huy, F. Chalumeau, A. T. Dinh-Xuan, F. Soyez **and** F. Martin. Étude de la prévalence du syndrome d’apnées obstructives du sommeil au Vietnam [Study about the prevalence of the obstructive sleep apnoea syndrome in Vietnam]. French. **in***Revue des Maladies Respiratoires*: 35.1 (january 2018), **pages** 14–24. DOI: 10.1016/j.rmr.2017.10.006.
- [4] Xia et al. Wang. Obstructive sleep apnea and risk of cardiovascular disease and all-cause mortality: A meta-analysis of prospective cohort studies. **in***International Journal of Cardiology*: 169.3 (), **pages** 207–214.
- [5] Salari N, Gholami A, Jalili F, Zereshki E, Heidarian P, Shohaimi S **and** Mohammadi M. Global Prevalence of Excessive Daytime Sleepiness in Patients with Obstructive Sleep Apnea: A Systematic Review and Meta-Analysis. **in***Indian J Otolaryngol Head Neck Surg*: 77.1 (january 2025), **pages** 216–223. DOI: 10.

- 1007/s12070-024-05146-0. eprint: Epub2024Oct29. URL: <https://doi.org/10.1007/s12070-024-05146-0>.
- [6] T. Young, J. Skatrud **and** P. E. Peppard. Risk Factors for Obstructive Sleep Apnea in Adults. **in***JAMA*: 291.16 (2004), **pages** 2013–2016. DOI: 10.1001/jama.291.16.2013.
  - [7] A. Menon **and** M. Kumar. Influence of body position on severity of obstructive sleep apnea: a systematic review. **in***ISRN Otolaryngology*: (2013), **page** 670381. DOI: 10.1155/2013/670381.
  - [8] Rosalind Dymond Cartwright. Effect of Sleep Position on Sleep Apnea Severity. **in***Sleep*: 7.2 (1984), **pages** 110–114. DOI: 10.1093/sleep/7.2.110.
  - [9] L. C. Markun **and** A. Sampat. Clinician-Focused Overview and Developments in Polysomnography. **in***Current Sleep Medicine Reports*: 6.4 (2020), **pages** 309–321. DOI: 10.1007/s40675-020-00197-5.
  - [10] D. Cary, A. Jacques **and** K. Briffa. Examining relationships between sleep posture, waking spinal symptoms and quality of sleep: A cross sectional study. **in***PLoS One*: 16.11 (November 2021). Erratum in: *PLoS One*. 2024 Jul 2;19(7):e0306662. doi: 10.1371/journal.pone.0306662., e0260582. DOI: 10.1371/journal.pone.0260582.
  - [11] R. D. Cartwright. Effect of Sleep Position on Sleep Apnea Severity. **in***Sleep*: 7.2 (1984), **pages** 110–114. ISSN: 0161-8105. DOI: 10.1093/sleep/7.2.110. URL: <https://doi.org/10.1093/sleep/7.2.110>.
  - [12] H. J. Lee, S. H. Hwang, S. M. Lee, Y. G. Lim **and** K. S. Park. Estimation of body postures on bed using unconstrained ECG measurements. **in***IEEE Journal of Biomedical and Health Informatics*: 17.6 (2013), **pages** 985–993. DOI: 10.1109/JBHI.2013.2252911.
  - [13] X. Nam, Y. Kim **and** J. Lee. Sleep monitoring based on a tri-axial accelerometer and a pressure sensor. **in***Sensors*: 16.5 (2016), **page** 750.
  - [14] Moein Enayati, Marjorie Skubic, James M. Keller, Mihail Popescu **and** Nasibeh Zanjirani Farahani. Sleep Posture Classification Using Bed Sensor Data and Neural Networks. **in**(2018): **pages** 461–465. DOI: 10.1109/EMBC.2018.8512436.

- [15] S. M. Mohammadi, S. Kouchaki, S. Khan, D.-J. Dijk, A. Hilton **and** K. Wells. Two-step deep learning for estimating human sleep pose occluded by bed covers. **in(july 2019): pages** 3115–3118.
- [16] A. Y.-C. Tam, B. P.-H. So, T. T.-C. Chan, A. K.-Y. Cheung, D. W.-C. Wong **and** J. C.-W. Cheung. A blanket accommodative sleep posture classification system using an infrared depth camera: A deep learning approach with synthetic augmentation of blanket conditions. **inSensors:** 21.16 (**august 2021**), **page** 5553.
- [17] Hu. C.-J.& Wu. D. A Jeng. P.-Y. Wang. L.-C. A Wrist Sensor Sleep Posture Monitoring System: An Automatic Labeling Approach. **inSensors:** (2021). DOI: /10.3390/s21010258.
- [18] A. Manoni, F. Loreti, V. Radicioni, D. Pellegrino, L. De Torre, A. Gumiero, D. Halicki, P. Palange **and** F. Irrera. A new wearable system for home sleep apnea testing, screening, and classification. **inSensors:** 20.24 (**december 2020**), **page** 7014.
- [19] J. Ranta, E. Ilén, K. Palmu, J. Salama, O. Roienko **and** S. Vanhatalo. An openly available wearable, a diaper cover, monitors infant's respiration and position during rest and sleep. **inActa Paediatrica:** 110.10 (**october 2021**), **pages** 2766–2771.
- [20] H. Yoon, S. Hwang, D. Jung, S. Choi, K. Joo, J. Choi, Y. Lee, D.-U. Jeong **and** K. Park. Estimation of sleep posture using a patch-type accelerometer based device. **in(august 2015): pages** 4942–4945.
- [21] Vu HD, Tran DN, Pham HH, Pham DD, Can KL, Dao TH **and** Tran DT. Human sleep position classification using a lightweight model and acceleration data. **inSleep Breath:** 29.1 (**february 2025**), **page** 95. DOI: 10.1007/s11325-025-03247-w. URL: <https://doi.org/10.1007/s11325-025-03247-w>.
- [22] R.S. Abdulsadig **and** E. Rodriguez-Villegas. Sleep posture monitoring using a single neck-situated accelerometer: A proof-of-concept. **inIEEE Access:** 11 (2023), **pages** 17693–17706. DOI: 10.1109/ACCESS.2023.3246266.
- [23] Montserrat JM & Jané R Ferrer-Lluis I Castillo-Escario Y. SleepPos App: An Automated Smartphone Application for Angle Based High Resolution Sleep Position Monitoring and Treatment. **inSensors:** (2021). DOI: 10.3390/s21134531.

- [24] W. Kukwa, T. Lis, J. Łaba, R. B. Mitchell **and** M. Młyńczak. Sleep Position Detection with a Wireless Audio-Motion Sensor—A Validation Study. *inDiagnostics*: 12.5 (2022), **page** 1195. DOI: 10.3390/diagnostics12051195.
- [25] Richard B. Berry, Reena Budhiraja, Daniel J. Gottlieb, David Gozal, Conrad Iber, Vishesh K. Kapur, Carole L. Marcus, Reena Mehra, Sairam Parthasarathy, Stuart F. Quan, Susan Redline, Kingman P. Strohl, Sally L. Davidson Ward, Mary M. Tangredi **and** American Academy of Sleep Medicine. Rules for Scoring Respiratory Events in Sleep: Update of the 2007 AASM Manual for the Scoring of Sleep and Associated Events. Deliberations of the Sleep Apnea Definitions Task Force of the American Academy of Sleep Medicine. *inJournal of Clinical Sleep Medicine*: 8.5 (2012), **pages** 597–619. ISSN: 1550-9389. DOI: 10.5664/jcsm.2172. URL: <https://doi.org/10.5664/jcsm.2172>.
- [26] S. Duong-Quy, Hoang Nguyen Huu, Dinh Hoang-Chau-Bao, Duc-Si Tran, Lien Nguyen-Thi-Hong, Thai Nguyen-Duy, Tram Tang-Thi-Thao, Chandat Phan, Khue Bui, Quan Vu, Thu Nguyen-Ngoc-Phuong, Vinh Nguyen, Huong Le-Thi-Minh **and** Timothy Craig. Personalized Medicine and Obstructive Sleep Apnea. *inJournal of Personalized Medicine*: 12 (**december** 2022), **page** 2034. DOI: 10.3390/jpm12122034.
- [27] Jayne C Carberry & Danny J Eckert Amal M Osman Sophie G Carter. Obstructive sleep apnea: current perspectives. *inNature and Science of Sleep*: (2016). DOI: 10.2147/NSS.S124657.
- [28] Danny J. Eckert, Amy S. Jordan, Prabhjyot Merchia **and** Atul Malhotra. Central Sleep Apnea: Pathophysiology and Treatment. *inChest*: 131.2 (**february** 2007), **pages** 595–607. ISSN: 0012-3692. DOI: 10.1378/chest.06.2287.
- [29] L. Spicuzza, D. Caruso **and** G. Di Maria. Obstructive sleep apnoea syndrome and its management. *inTherapeutic Advances in Chronic Disease*: 6.5 (**september** 2015), **pages** 273–285. DOI: 10.1177/2040622315590318.
- [30] Osman AM, Carter SG, Carberry JC **and** Eckert DJ. Obstructive sleep apnea: current perspectives. *inNat Sci Sleep*: 10 (**january** 2018), **pages** 21–34. DOI: 10.2147/NSS.S124657. URL: <https://doi.org/10.2147/NSS.S124657>.

- [31] Amal Osman, Sophie Carter, Jayne Carberry **and** Danny Eckert. Obstructive sleep apnea: current perspectives. *inNature and Science of Sleep*: 10 (january 2018), **pages** 21–34. DOI: 10.2147/NSS.S124657.
- [32] W. W. Flemons **and** W. Tsai. Quality of life consequences of sleep-disordered breathing. *inJournal of Allergy and Clinical Immunology*: 99.2 (1997), S750–S756. DOI: 10.1016/s0091-6749(97)70123-4.
- [33] J. Wright, R. Johns, I. Watt, A. Melville **and** T. Sheldon. Health effects of obstructive sleep apnoea and the effectiveness of continuous positive airways pressure: a systematic review of the research evidence. *inBMJ*: 314.7084 (1997), **pages** 851–858. URL: <http://www.ncbi.nlm.nih.gov/pubmed/9093094>.
- [34] Zinchuk AV, Jeon S, Koo BB, Yan X, Bravata DM, Qin L **and** Yaggi HK. Polysomnographic phenotypes and their cardiovascular implications in obstructive sleep apnoea. *inThorax*: 73.5 (2018), **pages** 472–480. DOI: 10.1136/thoraxjnl-2017-210431. URL: <https://doi.org/10.1136/thoraxjnl-2017-210431>.
- [35] T. Young, P. Peppard, M. Palta, K. M. Hla, L. Finn, B. Morgan **and** J. Skatrud. Population based study of sleep disordered breathing as a risk factor for hypertension. *inArchives of Internal Medicine*: 157 (1997), **pages** 1746–1752.
- [36] Mai Nguyen-Thi-Phuong, Mai Nguyen-Thi-Thanh, Thuy Nguyen-Thi-Dieu **and** S. Duong-Quy. Prevalence of Obstructive Sleep Apnea in Vietnamese Children with Attention Deficit Hyperactivity Disorder (ADHD). *inPulmonary therapy*: 11 (january 2025). DOI: 10.1007/s41030-024-00286-8.
- [37] Raphael Heinzer, Nicolas J. Petitpierre, Helena Marti-Soler **and** José Haba-Rubio. Prevalence and characteristics of positional sleep apnea in the HypnoLaus population-based cohort. *inSleep Medicine*: 48 (2018), **pages** 157–162. ISSN: 1389-9457. DOI: <https://doi.org/10.1016/j.sleep.2018.02.011>. URL: <https://www.sciencedirect.com/science/article/pii/S1389945718300704>.
- [38] Al Oweidat K, Toubasi AA, Al-Iede M, Hasuneh MM **and** Al-Sayegh TN. Comparing the diagnostic value of the positional obstructive sleep apnea definitions. *inRespir Med*: 212 (june 2023), **page** 107227. DOI: 10.1016/j.rmed.2023.107227. eprint: Epub2023Mar31. URL: <https://doi.org/10.1016/j.rmed.2023.107227>.

- [39] M. J. Mador, T. J. Kufel, U. J. Magalang, S. K. Rajesh, V. Watwe **and** B. J. Grant. Prevalence of Positional Sleep Apnea in Patients Undergoing Polysomnography. *inChest*: 128.4 (2005), **pages** 2130–2137. ISSN: 0012-3692. DOI: 10 . 1378 / chest . 128 . 4 . 2130. URL: <https://doi.org/10.1378/chest.128.4.2130>.
- [40] D. J. Levendowski, B. Veljkovic, S. Seagraves **and** P. R. Westbrook. Capability of a Neck Worn Device to Measure Sleep/Wake, Airway Position, and Differentiate Benign Snoring from Obstructive Sleep Apnea. *inJournal of Clinical Monitoring and Computing*: 29.1 (2015), **pages** 53–64. ISSN: 1387-1307. DOI: 10 . 1007 / s10877 - 014 - 9569 - 3. URL: <https://doi.org/10.1007/s10877-014-9569-3>.
- [41] Michaël Frank, Madeline J. L. Ravesloot, J. P. van Maanen, Evert Verhagen, J. de Lange **and** Nico de Vries. Positional OSA Part 1: Towards a Clinical Classification System for Position-Dependent Obstructive Sleep Apnoea. *inSleep and Breathing*: 19.2 (june 2014), **pages** 473–480. ISSN: 1520-9512. DOI: 10 . 1007 / s11325 - 014 - 1022 - 9. URL: <https://doi.org/10.1007/s11325-014-1022-9>.
- [42] D. J. Gottlieb **and** N. M. Punjabi. Diagnosis and Management of Obstructive Sleep Apnea: A Review. *inJAMA*: 323.14 (april 2020), **pages** 1389–1400. DOI: 10 . 1001/jama . 2020 . 3514.
- [43] Johnson J Goyal M. Obstructive Sleep Apnea Diagnosis and Management. *inMo Med*: (2017).
- [44] Medical Advisory Secretariat. Polysomnography in patients with obstructive sleep apnea: an evidence-based analysis. *inOntario Health Technology Assessment Series*: 6.13 (2006). Epub 2006 Jun 1, **pages** 1–38.
- [45] Sampa Markun L.C. Clinician-Focused Overview and Developments in Polysomnography. *inCurr Sleep Medicine*: (2020). DOI: <https://doi.org/10.1007/s40675-020-00197-5>.
- [46] C.A. Kushida, M.R. Littner, T. Morgenthaler, C.A. Alessi, D. Bailey, J. Jr Coleman, L. Friedman, M. Hirshkowitz, S. Kapen, M. Kramer, T. Lee-Chiong, D.L. Loube, J. Owens, J.P. Pancer **and** M. Wise. Practice parameters for the indications for polysomnography and related procedures: an update for 2005. *inSleep*: 28.4 (april 2005), **pages** 499–521. DOI: 10 . 1093/sleep/28 . 4 . 499.

- [47] N.T. Tran, H.N. Tran **and** A.T. Mai. A wearable device for at-home obstructive sleep apnea assessment: State-of-the-art and research challenges. *inFrontiers in Neurology*: 14 (**february** 2023), **page** 1123227. DOI: 10 . 3389 / fneur . 2023 . 1123227.
- [48] Ainhoa Osa-Sanchez, Javier Ramos-Martinez-de-Soria, Amaia Mendez-Zorrilla, Ibon Oleagordia Ruiz **and** Begonya Garcia-Zapirain. Wearable Sensors and Artificial Intelligence for Sleep Apnea Detection: A Systematic Review. *inJournal of Medical Systems*: 49.1 (2025), **page** 66. ISSN: 1573-689X. DOI: 10 . 1007 / s10916 - 025 - 02199 - 8. URL: <https://doi.org/10.1007/s10916-025-02199-8>.
- [49] Qiang Shen, Xinhui Yang, Lei Zou, Yufei Wang, Wenzhong Li, Li Li **and** Yan-hong Liu. Multitask Residual Shrinkage Convolutional Neural Network for Sleep Apnea Detection Based on Wearable Bracelet Photoplethysmography. *inIEEE Internet of Things Journal*: 9.24 (2022), **pages** 25207–25222. ISSN: 2327-4662. DOI: 10 . 1109 / JIOT . 2022 . 3195777. URL: <https://doi.org/10.1109/JIOT.2022.3195777>.
- [50] Jeon Y, Heo K **and** Kang SJ. Real-Time Sleep Apnea Diagnosis Method Using Wearable Device without External Sensors. *in*(2020): **pages** 1–5. DOI: 10 . 1109 / PerComWorkshops48775 . 2020 . 9156119. URL: <https://doi.org/10.1109/PerComWorkshops48775.2020.9156119>.
- [51] Yin-Yan Lin, Hau-Tieng Wu, Chi-An Hsu, Po-Chiun Huang, Yuan-Hao Huang **and** Yu-Lun Lo. Sleep Apnea Detection Based on Thoracic and Abdominal Movement Signals of Wearable Piezoelectric Bands. *inIEEE Journal of Biomedical and Health Informatics*: 21.6 (2017), **pages** 1533–1545. DOI: 10 . 1109 / JBHI . 2016 . 2636778.
- [52] Ting Chen, Jing Zhang, Zhaojian Xu, Yanjun Liu, Xinyu Yu, Yufei Wang, Li Li **and** Yanhong Liu. Energy-Efficient Sleep Apnea Detection Using a Hyperdimensional Computing Framework Based on Wearable Bracelet Photoplethysmography. *inIEEE Transactions on Biomedical Engineering*: (2024). ISSN: 0018-9294. DOI: 10 . 1109 / TBME . 2024 . 3377270. URL: <https://doi.org/10.1109/TBME.2024.3377270>.

- [53] Minhee Yeo, Heejin Byun, Jaehyuk Lee, Jihyun Byun, Hyung Yoon Rhee, Won Shin **and** Hyungjoon Yoon. Robust Method for Screening Sleep Apnea with Single-Lead ECG Using Deep Residual Network: Evaluation with Open Database and Patch-Type Wearable Device Data. *inIEEE Journal of Biomedical and Health Informatics*: 26.11 (2022), **pages** 5428–5438. ISSN: 2168-2194. DOI: 10.1109/JBHI.2022.3203560. URL: <https://doi.org/10.1109/JBHI.2022.3203560>.
- [54] Minhee Yeo, Heejin Byun, Jaehyuk Lee, Jihyun Byun, Hyung Yoon Rhee, Won Shin **and** Hyungjoon Yoon. Respiratory Event Detection During Sleep Using Electrocardiogram and Respiratory Related Signals: Using Polysomnogram and Patch-Type Wearable Device Data. *inIEEE Journal of Biomedical and Health Informatics*: 26.2 (2022), **pages** 550–560. ISSN: 2168-2194. DOI: 10.1109/JBHI.2021.3098312. URL: <https://doi.org/10.1109/JBHI.2021.3098312>.
- [55] M. Kim I. Yun J. Jeung. Ultra-low power wearable infant sleep position sensor. *inSensors*: 7 (2020).
- [56] Miguel Espinosa, Pedro Ponce, Arturo Molina, Vicente Borja, Martha Torres **and** Mario Rojas. Advancements in Home-Based Devices for Detecting Obstructive Sleep Apnea: A Comprehensive Study. *inSensors (Basel, Switzerland)*: 23 (december 2023). DOI: 10.3390/s23239512.
- [57] R.Nisha Aurora Naresh M.Punjabi **and** Susheel P. Patil. Home Sleep Testing for Obstructive Sleep Apnea. *inChest*: (2013). DOI: 10.1378/chest.12-2699.
- [58] Ahmet & Sumbul Yuzer. A Novel Wearable Real-Time Sleep Apnea Detection System Based on the Acceleration Sensor. *inJ Clin Sleep Med*: (2019).
- [59] Rawan S. Abdulsadig, Sukhpreet Singh, Zaibaa Patel **and** Esther Rodriguez-Villegas. Sleep Posture Detection Using an Accelerometer Placed on the Neck. *in(2022)*: **pages** 2430–2433. DOI: 10.1109/EMBC48229.2022.9871300.
- [60] D.M. Domingues, P.R. Rocha, A.C.M.V. Miachon **and others**. Sleep prediction using data from oximeter, accelerometer and snoring for portable monitor obstructive sleep apnea diagnosis. *inScientific Reports*: 14 (2024), **page** 24562. DOI: 10.1038/s41598-024-75935-8. URL: <https://doi.org/10.1038/s41598-024-75935-8>.

- [61] Andrei Boiko, Maksym Gaiduk, Ralf Seepold **and** Natividad Martínez Madrid. Accelerometer based system for unobtrusive sleep apnea detection. *inProcedia Computer Science*: 225 (**december** 2023), **pages** 1592–1600. DOI: 10.1016/j.procs.2023.10.148.
- [62] M. Yousuf A. Channa **and** N. Popescu. Machine Learning Algorithms for Posture Identification of Obstructive Sleep Apnea Patients using IoT Solutions. *inIEEE Access*: (2019). DOI: 10.1109/ACCESS.2019.2920025.
- [63] Xu X, Mo Q, Wang Z, Zhao Y **and** Li C. Classification of Sleeping Position Using Enhanced Stacking Ensemble Learning. *inEntropy (Basel)*: 26.10 (**september** 2024), **page** 817. DOI: 10.3390/e26100817. URL: <https://doi.org/10.3390/e26100817>.
- [64] Awais Khan, Chomyong Kim, Jung-Yeon Kim, Ahsan Aziz **and** Yunyoung Nam. Sleep Posture Classification Using RGB and Thermal Cameras Based on Deep Learning Model. *inCMES - Computer Modeling in Engineering and Sciences*: 140.2 (2024), **pages** 1729–1755. ISSN: 1526-1492. DOI: <https://doi.org/10.32604/cmes.2024.049618>. URL: <https://www.sciencedirect.com/science/article/pii/S1526149224000109>.
- [65] Sara Mahvash Mohammadi, Shirin Enshaeifareb, Adrian Hiltongc, Derk-Jan Dijk **and** Kevin Wells. Transfer Learning for Clinical Sleep Pose Detection Using a Single 2D IR Camera. *inIEEE Transactions on Neural Systems and Rehabilitation Engineering*: PP (**december** 2020), **pages** 1–1. DOI: 10.1109/TNSRE.2020.3048121.
- [66] Lee J Nam Y Kim Y. Sleep Monitoring Based on a Tri-Axial Accelerometer and a Pressure Sensor. *inSensors (Basel)*: (2016). DOI: 10.3390/s16050750.
- [67] Le Tien TUAN Vu Minh; THUONG. Accelerometer-Based Model Acquiring Data on Sleep Apnea Symptoms. *inIssue on Information and Communications Technology*: (2021). DOI: <https://doi.org/10.31130/ict-ud.2021.130..>
- [68] Xiao Sun, Li Qiu, Yibo Wu, Yeming Tang **and** Guohong Cao. SleepMonitor: Monitoring Respiratory Rate and Body Position During Sleep Using Smartwatch. *inProceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*: 1.3 (**september** 2017), **pages** 1–22. DOI: 10.1145/3130969. URL: <https://doi.org/10.1145/3130969>.

- [69] Vincenzo và cộng sự Natale. Monitoring sleep with a smartphone accelerometer. *inSleep and Biological Rhythms*: (2012).
- [70] G Zhang Z.; Yang. Monitoring cardio-respiratory and posture movements during sleep: What can be achieved by a single motion sensor. *inIEEE International Conference*: (2015).
- [71] Crossvà cộng sự Kwasnicki R.M. A lightweight sensing platform for monitoring sleep quality and posture: a simulated validation study. *inEur J Med Res*: 7 (2018). DOI: <https://doi.org/10.1186/s40001-018-0326-9>.
- [72] Didik R. Santoso, Sukir Maryanto **and** Ahmad Nadhir. Application of Single MEMS-accelerometer to Measure 3-axis Vibrations and 2-axis Tilt-Angle Simultaneously. *inTELKOMNIKA (Telecommunication Computing Electronics and Control)*: 13 (**june** 2015), **page** 442. DOI: [10.12928/telkomnika.v13i2.1490](https://doi.org/10.12928/telkomnika.v13i2.1490).
- [73] M. Szermer **and** J. Nazdrowicz. Study on Comb-Drive MEMS Acceleration Sensor Used for Medical Purposes: Monitoring of Balance Disorders. *inElectronics*: 14.15 (2025), **page** 3033. DOI: [10.3390/electronics14153033](https://doi.org/10.3390/electronics14153033). URL: <https://doi.org/10.3390/electronics14153033>.
- [74] S. Shi, L. Ma, K. Kang, J. Zhu, J. Hu, H. Ma, Y. Pang **and** Z. Wang. High-Sensitivity Piezoelectric MEMS Accelerometer for Vector Hydrophones. *inMicromachines (Basel)*: 14.8 (**august** 2023), **page** 1598. DOI: [10.3390/mi14081598](https://doi.org/10.3390/mi14081598).
- [75] J. Song, C. He, R. Wang, C. Xue **and** W. Zhang. A Mathematical Model of a Piezo-Resistive Eight-Beam Three-Axis Accelerometer with Simulation and Experimental Validation. *inSensors*: 18.11 (2018), **page** 3641. DOI: [10.3390/s18113641](https://doi.org/10.3390/s18113641). URL: <https://doi.org/10.3390/s18113641>.
- [76] S. Li, X. Zhao, Y. Bai, Y. Li, C. Ai **and** D. Wen. Fabrication Technology and Characteristics Research of the Acceleration Sensor Based on Li-Doped ZnO Piezoelectric Thin Films. *inMicromachines*: 9.4 (2018), **page** 178. DOI: [10.3390/mi9040178](https://doi.org/10.3390/mi9040178).
- [77] J. Y. Hao **and** W. Bai. Mechanics analysis of cantilever beam with uniform load. *inAdvanced Materials Research*: 675 (2013), **pages** 98–102.

- [78] C. Wang, Y. Hao, Z. Sun, L. Zu, W. Yuan **and** H. Chang. Design of a Capacitive MEMS Accelerometer with Softened Beams. **in***Micromachines*: 13.3 (2022), **page** 459. DOI: 10.3390/mi13030459.
- [79] Rakesh Tirupathi **and** Sougata Kar. A Differential Output Switched Capacitor based Capacitive Sensor Interfacing Circuit. **in**(october 2018): **pages** 0565–0569. DOI: 10.1109/TENCON.2018.8650389.
- [80] A. Roebuck, V. Monasterio, E. Gederi, M. Osipov, J. Behar, A. Malhotra, T. Penzel **and** G. D. Clifford. A Review of Signals Used in Sleep Analysis. **in***Physiological Measurement*: 35.1 (january 2014), R1–R57. DOI: 10.1088/0967-3334/35/1/R1.
- [81] Q. Pan, D. Brulin **and** E. Campo. Current Status and Future Challenges of Sleep Monitoring Systems: Systematic Review. **in***JMIR Biomedical Engineering*: 5.1 (2020), e20921. DOI: 10.2196/20921. URL: <https://doi.org/10.2196/20921>.
- [82] Nguyen & Vo và cộng sự Sang Vu & Thang. Human Activity Recognition and Monitoring Using Smartphones. **in***International Conference on Biomedical Engineering in Vietnam*: (2015).
- [83] M. Rossi, D. Sala, D. Bovio **and others**. SLEEP-SEE-THROUGH: Explainable Deep Learning for Sleep Event Detection and Quantification From Wearable Somnography. **in***IEEE Journal of Biomedical and Health Informatics*: 27 (2023), **pages** 3129–3140. DOI: 10.1109/JBHI.2023.3267087. URL: <https://doi.org/10.1109/JBHI.2023.3267087>.
- [84] Alaa Sheta, Hamzah Turabieh, Tamer Taher, Johnson Too, Majdi Mafarja, Md. Shamim Hossain **and** Salim R. Surani. Diagnosis of Obstructive Sleep Apnea from ECG Signals Using Machine Learning and Deep Learning Classifiers. **in***Applied Sciences*: 11.14 (2021), **page** 6622. ISSN: 2076-3417. DOI: 10.3390/app11146622. URL: <https://doi.org/10.3390/app11146622>.
- [85] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. **in***Transactions of the ASME–Journal of Basic Engineering*: 82.Series D (1960).

- [86] J. Cuadrado, F. Michaud, U. Lugrís **and** M. Pérez Soto. Using Accelerometer Data to Tune the Parameters of an Extended Kalman Filter for Optical Motion Capture: Preliminary Application to Gait Analysis. *inSensors*: 21.2 (2021), **page** 427. DOI: 10 . 3390 / s21020427. URL: <https://doi.org/10.3390/s21020427>.
- [87] Reza Abbasi Kesbi **and** Alireza Nikfarjam. Denoising MEMS Accelerometer Sensors Based on L2-Norm Total Variation Algorithm. *inElectronics Letters*: 53 (**january** 2017). DOI: 10 . 1049/el . 2016 . 3811.
- [88] M. Lepot, J.-B. Aubin **and** F. H. L. R. Clemens. Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment. *inWater*: 9.10 (2017), **page** 796. DOI: 10 . 3390/w9100796. URL: <https://doi.org/10.3390/w9100796>.
- [89] Lei Zou **and** Guang Liu. Multiscale Bidirectional Temporal Convolutional Network for Sleep Apnea Detection Based on Wearable Photoplethysmography Bracelet. *inIEEE Journal of Biomedical and Health Informatics*: 28 (2024), **pages** 1331–1340. ISSN: 2168-2194. DOI: 10 . 1109/JBHI . 2023 . 3335658. URL: <https://doi.org/10.1109/JBHI.2023.3335658>.
- [90] Mark Olsen, Jamie M. Zeitzer, Richard N. Richardson, Nicholas Gutierrez, Colin Clarke, Nikhil Naik, Daryl Robinson **and** Deepak Khosla. A Deep Transfer Learning Approach for Sleep Stage Classification and Sleep Apnea Detection Using Wrist-Worn Consumer Sleep Technologies. *inIEEE Transactions on Biomedical Engineering*: (2024), **pages** 1–12. ISSN: 0018-9294. DOI: 10 . 1109/TBME . 2024 . 3378480. URL: <https://doi.org/10.1109/TBME.2024.3378480>.
- [91] Minh Long Hoang, Guido Matrella **and** Paolo Ciampolini. Metrological evaluation of contactless sleep position recognition using an accelerometric smart bed and machine learning. *inSensors and Actuators A: Physical*: 385 (2025), **page** 116309. ISSN: 0924-4247. DOI: <https://doi.org/10.1016/j.sna.2025.116309>. URL: <https://www.sciencedirect.com/science/article/pii/S0924424725001153>.
- [92] Thanuja Pavani và cộng sự Uday Sanku & Satti. Classifying Human Activities using Machine Learning and Deep Learning Techniques. *inarxiv*: (2019).

- [93] Kandala.N.V.P.S. Rajesh, Ravindra Dhuli **and** T. Sunil Kumar. Obstructive sleep apnea detection using discrete wavelet transform-based statistical features. **in***Computers in Biology and Medicine*: 130 (2021), **page** 104199. ISSN: 0010-4825. DOI: <https://doi.org/10.1016/j.combiomed.2020.104199>. URL: <https://www.sciencedirect.com/science/article/pii/S0010482520305308>.
- [94] H.-D. Vu, D.-N. Tran, K.-L. Can, T.-H. Dao, D.-D. Pham **and** D.-T. Tran. ?Enhancing sleep postures classification by incorporating acceleration sensor and LSTM model? **in***Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*: Hanoi, Vietnam, 2023, **pages** 661–665. DOI: [10.1109/SSP53291.2023.10208083](https://doi.org/10.1109/SSP53291.2023.10208083).
- [95] O. Elnaggar, R. Arelli, F. Coenen **and others**. An interpretable framework for sleep posture change detection and postural inactivity segmentation using wrist kinematics. **in***Scientific Reports*: 13 (2023), **page** 18027. DOI: [10.1038/s41598-023-44567-9](https://doi.org/10.1038/s41598-023-44567-9).
- [96] Chunhua He, Zewen Fang, Shuibin Liu, Heng Wu, Xiaoping Li, Yangxing Wen **and** Juze Lin. A smart detection method for sleep posture based on a flexible sleep monitoring belt and vital sign signals. **in***Heliyon*: 10.11 (2024), e31839. ISSN: 2405-8440. DOI: <https://doi.org/10.1016/j.heliyon.2024.e31839>. URL: <https://www.sciencedirect.com/science/article/pii/S2405844024078708>.
- [97] E. Gomes, L. Bertini, W.R. Campos, A.P. Sobral, I. Mocaiber **and** A. Copetti. Machine learning algorithms for activity-intensity recognition using accelerometer data. **in***Sensors*: 21 (2021), **page** 1214. DOI: [10.3390/s21041214](https://doi.org/10.3390/s21041214).
- [98] W. Souza **and** K. Rajamohan. Human activity recognition using accelerometer and gyroscope sensors. **in***International Journal of Engineering and Technology*: 9 (2017), **pages** 1171–1179. DOI: [10.21817/ijet/2017/v9i2/170902134](https://doi.org/10.21817/ijet/2017/v9i2/170902134).
- [99] A. Crivello và cộng sự P. Barsocchi M. Bianchini. An unobtrusive sleep monitoring system for the human sleep behaviour understanding. **in***IEEE International Conference*: 7 (2016).
- [100] Nordic Semiconductor. *nRF52840 - Advanced Bluetooth 5, Thread and Zigbee multiprotocol SoC*. <https://www.nordicsemi.com/Products/nRF52840>. Accessed: 2025-07-28. 2025.

- [101] Nordic Semiconductor. *Tiny Machine Learning Now More Accessible for Non-Expert Developers*. <https://blog.nordicsemi.com/getconnected/tiny-machine-learning-now-more-accessible-for-non-expert-developers>. Accessed: 2025-11-07. 2021.
- [102] G. Wang, Q. Li, L. Wang, W. Wang, M. Wu **and** T. Liu. Impact of Sliding Window Length in Indoor Human Motion Modes and Pose Pattern Recognition Based on Smartphone Sensors. **in***Sensors*: 18.6 (2018), **page** 1965. DOI: 10.3390/s18061965. URL: <https://doi.org/10.3390/s18061965>.
- [103] Mahmood Al-Shareeda, Murtaja Ali, Selvakumar Manickam **and** Shankar Karuppayah. Bluetooth low energy for internet of things: review, challenges, and open issues. **in***Indonesian Journal of Electrical Engineering and Computer Science*: 31 (**august** 2023), **pages** 1182–1189. DOI: 10.11591/ijeecs.v31.i2.pp1182-1189.
- [104] *Bluetooth® Core 5.0 - Feature Enhancements*. [https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth\\_5-FINAL.pdf](https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_5-FINAL.pdf). Version 1.1.1, Revision Date 13 January 2025. 2019.
- [105] *Bluetooth® Core Specification Version 5.3 Feature Enhancements*. Available at <https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-version-5-3-feature-enhancements/>.
- [106] u-blox AG. *NINA-B3 Series - Stand-alone Bluetooth® 5 Low Energy Modules Data Sheet (Document No. UBX-17052099)*. [https://content.u-blox.com/sites/default/files/NINA-B3\\_DataSheet\\_UBX-17052099.pdf](https://content.u-blox.com/sites/default/files/NINA-B3_DataSheet_UBX-17052099.pdf).
- [107] SparkFun Electronics. *SparkFun BMI270 Arduino Library- Driver for the Bosch BMI270 6-DoF IMU*. [https://github.com/sparkfun/SparkFun\\_BMI270\\_Arduino\\_Library](https://github.com/sparkfun/SparkFun_BMI270_Arduino_Library). Accessed: 2025-11-09. 2023.
- [108] *flutter\_blue\_plus - Bluetooth Low Energy plugin for Flutter*. Accessed: 2025-11-09. URL: [https://pub.dev/packages/flutter\\_blue\\_plus](https://pub.dev/packages/flutter_blue_plus).
- [109] Corinna Cortes **and** Vladimir Vapnik. Support-Vector Networks. **in***Machine Learning*: 20.3 (1995), **pages** 273–297. ISSN: 0885-6125. DOI: 10.1007/BF00994018. URL: <https://doi.org/10.1007/BF00994018>.

- [110] *Bài 19: Support Vector Machine*. <https://machinelearningcoban.com/2017/04/09/smv/>. Accessed: 2025-11-09. **april** 2017.
- [111] Leo Breiman. Random Forests. **in***Machine Learning*: 45.1 (2001), **pages** 5–32. DOI: 10.1023/A:1010933404324. URL: <https://doi.org/10.1023/A:1010933404324>.
- [112] Tianqi Chen **and** Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. **in**(2016): **pages** 785–794. DOI: 10.1145/2939672.2939785. URL: <https://doi.org/10.1145/2939672.2939785>.
- [113] Anil K. Jain, Jianchang Mao **and** K. M. Mohiuddin. Artificial neural networks: a tutorial. **in***Computer*: 29.3 (march 1996), **pages** 31–44. DOI: 10.1109/2.485891.