

**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 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 sọ-mặt, và bất thường cấu trúc đường hô hấp trên gây hẹp đường thở khi ngủ [6]. Tác giả 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.Tư thế ngủ 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 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 ngủ [8].

Để đánh giá chính xác tình trạng OSA, hiện nay đa ký giấc ngủ (Polysomnography - PSG) 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 hô hấp nhằm đánh giá toàn diện hoạt động của cơ thể trong suốt giấc ngủ [9]. Trong đó, tư thế ngủ luôn là dữ liệu quan trọng để đưa ra kết luận về tình trạng OSA và phương pháp điều trị hợp lý. Phương pháp này được xem là tiêu chuẩn vàng trong chẩn đoán các rối loạn giấc ngủ, đặc biệt là ngưng thở khi ngủ, rối loạn vận động, động kinh và hành

vi bất thường trong giấc ngủ REM. Tuy nhiên, theo tìm hiểu của tác giả, tại Việt Nam, chi phí cho một lần thực hiện đa ký giấc ngủ (PSG) 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, thời gian chờ đặt lịch lâu, cùng 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 chế tạo ngày càng phát triển mạnh mẽ, việc thu nhỏ và tối ưu hiệu suất vi điều khiển, cảm biến, pin. Bên cạnh đó, sự phát triển vượt bậc của trí tuệ nhân tạo (AI) đóng vai trò quan trọng trong việc nâng cao hiệu quả khai thác dữ liệu cảm biến. AI không chỉ tối ưu hóa quy trình xử lý và phân tích dữ liệu mà còn thúc đẩy khả năng phân loại, phân cụm, dự đoán và đưa ra quyết định. 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 độ chính xác 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>	x
<b>Danh sách bảng</b>	xi
<b>Đặt vấn đề</b>	1
<b>Chương 1 Tổng quan OSA, tư thế ngủ và công nghệ liên quan</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 . . . . .	7
1.2. Công nghệ trong chẩn đoán OSA và phân loại tư thế ngủ . . . . .	9
1.2.1. Đa ký giấc ngủ . . . . .	10
1.2.2. Thiết bị theo dõi OSA tại nhà . . . . .	11
1.2.3. Thiết bị 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 VÀ XỬ LÝ TÍN HIỆU CẢM BIẾN</b>	<b>33</b>
2.1. Nghiên cứu, phát triển phần cứng . . . . .	33
2.1.1. Vิ xử lý . . . . .	33
2.1.2. Cảm biến . . . . .	36
2.1.3. Bluetooth năng lượng thấp . . . . .	39
2.1.4. Thiết kế mạch nguyên lý . . . . .	43
2.2. Hệ thống phần mềm . . . . .	51
2.2.1. Lập trình vi xử lý . . . . .	51
2.2.2. Hiệu chuẩn cảm biến . . . . .	57
2.2.3. Hiệu chuẩn cảm biến . . . . .	58
2.2.4. Xây dựng phần mềm ứng dụng . . . . .	61
2.2.5. Thiết kế và xây dựng hệ thống lưu trữ . . . . .	68
2.2.6. Học máy trong phân loại tư thế ngủ . . . . .	72
<b>Chương 3 KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ</b>	<b>83</b>
3.1. Hệ thống thực nghiệm . . . . .	83
3.2. Thu thập và gán nhãn dữ liệu . . . . .	88
3.3. Phân loại tư thế ngủ bằng học máy . . . . .	90
3.3.1. Phân tích dữ liệu . . . . .	90
3.3.2. Xử lý và trích xuất đặc trưng . . . . .	92
3.3.3. Kịch bản kiểm thử và lựa chọn tính năng . . . . .	93
3.3.4. Huấn luyện mô hình . . . . .	98
3.3.5. Đánh giá kết quả . . . . .	100
3.4. Triển khai trên vi điều khiển nhúng . . . . .	104
3.4.1. Quy trình triển khai mô hình . . . . .	105
3.4.2. Đánh giá hiệu suất và tài nguyên . . . . .	108



# Danh sách hình vẽ

1.1	Hình ảnh khác nhau về đường thở của người bình thường và người mắc OSA . . . . .	6
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 [20] . . . . .	17
1.8	Vị trí tối ưu để gắn cảm biến gia tốc . . . . .	18
1.9	(a) Sơ đồ bố trí tổng thể của cảm biến gia tốc điện trở áp (b) Cấu trúc mặt cắt qua khối trọng và 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.	20
1.10	Cấu trúc cơ bản của cảm biến MEMS áp điện: (a) mặt trên (b) mặt dưới	21
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	STM MP34DT06J . . . . .	46
2.14	Sơ đồ khối IC MP2322GQH, dải đầu vào 3V–22V, dòng tải tối đa 1A, hiệu suất cao, tần số chuyển mạch 1.25MHz (Monolithic Power Systems)	47
2.15	Sơ đồ nguyên lý . . . . .	49
2.16	Bố trí mạch in hai mặt của bo mạch: (a) lớp trên; (b) lớp dưới . . . . .	50
2.17	Mô hình mạch 3D: (a) lớp trên; (b) lớp dưới . . . . .	50
2.18	Mạch hoàn chỉnh . . . . .	51
2.19	Lưu đồ luồng gửi thông tin BLE . . . . .	56
2.20	Minh họa kết quả phân tích đường cong Allan . . . . .	57
2.21	Biểu đồ phương sai Allan của trục X . . . . .	58
2.22	Minh họa kết quả phân tích đường cong Allan . . . . .	59
2.23	Biểu đồ phương sai Allan của trục X . . . . .	60
2.24	Kết quả bộ lọc Kalman cho dữ liệu trục X của cảm biến gia tốc . . . . .	62
2.25	Cấu trúc kiến trúc BLoC trong ứng dụng Flutter . . . . .	66
2.26	Mô hình tích hợp giữa mạng cảm biến và cấu trúc dữ liệu đám mây . . . . .	68
2.27	Lưu đồ thuật toán lưu trữ dữ liệu cảm biến . . . . .	73
2.28	Phân bố thời gian thực hiện đối với dự án học máy . . . . .	74
2.29	Minh họa mặt siêu phẳng phân tách hai lớp trong SVM . . . . .	77
2.30	Chiến lược phân loại đa lớp bằng phương pháp One-vs-One . . . . .	78

2.31	Chiến lược phân loại đa lớp bằng phương pháp One-vs-Rest . . . . .	78
3.1	Hệ thống thử nghiệm: (a) vị trí đặt thiết bị cảm biến; (b) minh họa thực nghiệm thực tế trong tư thế nằm. . . . .	84
3.2	Giao diện chức năng đăng ký và đăng nhập . . . . .	85
3.3	Giao diện trang chủ . . . . .	85
3.4	Giao diện màn hình danh sách BLE và chi tiết các dịch vụ kết nối với phần cứng . . . . .	86
3.5	Giao diện chức năng chatbot và bộ câu hỏi tầm soát . . . . .	87
3.6	Giao diện hiển thị dữ liệu gia tốc ba trục . . . . .	88
3.7	Mô phỏng thực nghiệm thực tế . . . . .	89
3.8	Phân bố dữ liệu cảm biến theo ba trục $x$ , $y$ , $z$ ứng với các tư thế ngũ nhau. . . . .	91
3.9	Phân bố số lượng mẫu trong tập huấn luyện theo từng tư thế. . . . .	92
3.10	Ma trận tương quan giữa các đặc trưng trích xuất. Cường độ màu thể hiện hệ số tương quan Pearson. Màu đỏ là tương quan dương mạnh, xanh là tương quan âm mạnh, xám là không tương quan. . . . .	96
3.11	Phân tích giá trị SHAP nhằm xác định tầm quan trọng của các đặc trưng trong mô hình phân loại tư thế ngũ. Các đặc trưng từ trục $z$ chiếm ưu thế về mức ảnh hưởng đến đầu ra mô hình. . . . .	98
3.12	Ma trận nhầm lẫn (confusion matrix) của năm mô hình phân loại trong kịch bản S1. GB và RF cho kết quả chính xác cao nhất. Mô hình NN được huấn luyện trực tiếp trên dữ liệu thô. . . . .	102

# 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 biến . . . . .	26
1.5	Hiệu quả mô hình học máy và khả năng triển khai biến trong nhận diện tư thế ngủ . . . . .	31
2.1	Các nhóm chức năng chính của ứng dụng . . . . .	64
3.1	Các đặc trưng thống kê và tín hiệu được sử dụng trong phân loại tư thế ngủ	94
3.2	Các kịch bản lựa chọn và sử dụng đặc trưng trong nghiên cứu . . . . .	97
3.3	Các mô hình học máy và siêu tham số sử dụng trong nghiên cứu . . . . .	99
3.4	Độ chính xác phân loại của các mô hình trong 8 kịch bản . . . . .	100
3.5	Kích thước mô hình (KB) trong 8 kịch bản . . . . .	102
3.6	So sánh hiệu năng Logistic Regression và Neural Network trên nRF52840	109

# Đặt vấn đề

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 tiếp xúc và không tiếp xúc, cụ thể gồm: Tấm dán điện với các điện cực ECG đặt tại giường thu thập tín hiệu nhịp tim để phân loại tư thế ngủ [10]. Cảm biến áp suất dưới nệm đo phân bố lực tiếp xúc toàn thân [11, 12]. Camera hồng ngoại ghi hình ban đêm không cần chiếu sáng, nhận dạng tư thế bằng thị giác máy [13, 14].

Các thiết bị đeo sử dụng tín hiệu đo lường quán tính (Inertial Measurement Unit - IMU) được đặt đa vị trí trên cơ thể nổi lên như cách tiếp cận nhỏ gọn, thuận tiện. Trong nghiên cứu [15], tác giả đã đề xuất hệ thống iSleePost 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 [16]. 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, so sánh với bảng điều kiện tham chiếu định sẵn. Tuy nhiên, tư thế nằm sấp 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 trưa, tích hợp cảm biến gia tốc và con quay hồi chuyển [17]. 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 cũng cho thấy tiềm năng trong việc phân loại tư thế ngủ. 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 [18]. 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 [19]. Đối với nghiên cứu [20], 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 cây quyết định (Decision Tree) có hiệu năng và mức tiêu thụ bộ nhớ tối ưu nhất. Nhóm tác giả trong nghiên cứu [21] 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 kèm kềnh làm giảm tính thoái 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 [22].

Hầu hết các nghiên cứu về đánh giá tư thế ngủ đều đưa ra những phương pháp toàn diện sử dụng các mô hình học máy, kết hợp nhiều loại cảm biến khác nhau như cảm biến quán tính, cảm biến liên quan sinh lý, hoặc cảm biến áp suất nhằm thu thập dữ liệu đa chiều. Một số nghiên cứu đã chứng minh tính khả thi của việc sử dụng duy nhất một cảm biến gia tốc trong nhận dạng tư thế ngủ, đạt được độ chính xác cao khi kết hợp với các mô hình học máy. Các kết quả này cho thấy hướng tiếp cận đơn cảm biến hoàn toàn có thể thay thế cho hệ thống đa cảm biến truyền thống trong các 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 của vẫn chưa được nghiên cứu cụ thể.

Đặc biệt, 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 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 (OSA).

Qua đó, 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. Trên cơ sở 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 và 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 (OSA), 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 pOSA (positional OSA) dựa trên tư thế ngủ.

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, tư thế ngủ và công nghệ liên quan

**Chương 2:** Hệ thống thu thập và xử lý tín hiệu cảm biến.

**Chương 3:** Trích xuất đặc trưng và mô hình học máy.

**Chương 4:** Kết quả và đánh giá.

# **Chương 1**

## **Tổng quan OSA, tư thế ngủ và công nghệ liên quan**

Trong chương này, tác giả trình bày tổng quan về hội chứng ngưng thở khi ngủ do tắc nghẽn (OSA), phân tích ảnh hưởng 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 và 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, việc chuẩn hóa và định nghĩa chính xác các kiểu sự kiện hô hấp giúp đảm bảo tính nhất quán trong chẩn đoán, phân lớp nguy cơ và lựa chọn phương pháp điều trị phù hợp. Theo tiêu chuẩn của Hiệp hội Y học Giấc ngủ Hoa Kỳ (AASM) [23], 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).

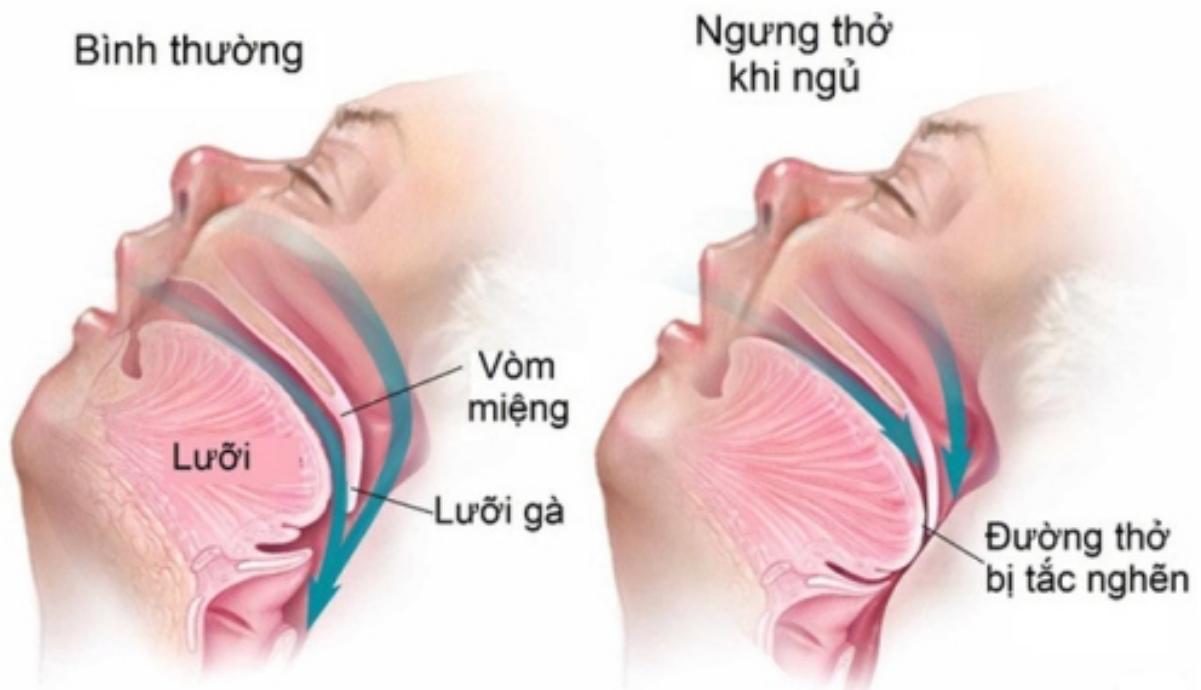
### **1.1.1. Định nghĩa**

Ngưng thở (Apnea) 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 [24]: 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 [25]; 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 [26]. 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.

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  $\geq 10$  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); RDI  $>5$  là bất thường,  $>15$  là có ý nghĩa lâm sàng.

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 [25].



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

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

AHI	Cấp độ
<5	Không mắc
5 đến 10	Nhẹ
15 đến 30	Trung bình
>30	Nặng

### 1.1.2. Nguyên nhân

Nguyên nhân của OSA đến đa yếu tố kết hợp giữa yếu tố 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, hẹp hoặc giảm độ vững của cấu trúc đường hô hấp trên là yếu tố tiên quyết. Những bất thường như vòm khẩu hẹp, lưỡi to, phì đại amidan, vách mũi lệch, xương hàm nhỏ hoặc tụt về sau, cùng 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, mãn kinh ở nữ giới, nghẹt mũi mạn tính, và yếu tố di truyền [27, 28, 29].

### **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, 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 [30].

Hơn nữa, tình trạng giảm oxy máu 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 [31, 32, 33]. 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 [34]. 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.

### **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.



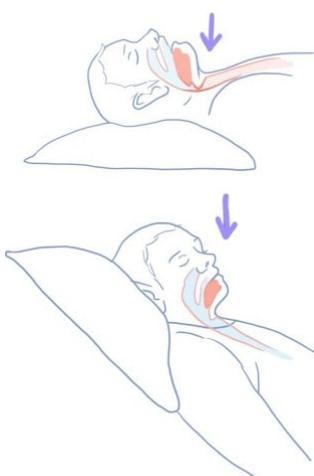
Hình 1.2: Các tư thế ngủ cơ bản của con người

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 [35]. 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.

Các nghiên cứu cho thấy, nhiều bệnh nhân OSA có tần suất ngưng thở và giảm thở cao hơn rõ rệt khi nằm ngủ so với các tư thế khác. Nguyên nhân chủ yếu là do tác động của trọng lực lên các cấu trúc mô mềm 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) [36, 37].

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 ngủ cao hơn ít nhất hai lần so với AHI ở tư thế không nằm ngủ [38]. Tiếp đó, Mador (2005) đã mở rộng định nghĩa này bằng cách bổ sung tiêu chí rằng AHI ở tư thế không nằm ngủ 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 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 khởi hoàn toàn nhờ thay đổi tư thế ngủ), APOC-II (không phụ thuộc tư thế), và APOC-III (phụ thuộc tư thế một phần).



Hình 1.3: Xếp đường thở ở tư thế ngủ

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 chế tạo cảm biến, vi xử lý và các thuật toán học máy, việc chẩn đoán hội chứng ngưng thở khi ngủ tắc nghẽn (OSA) 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**

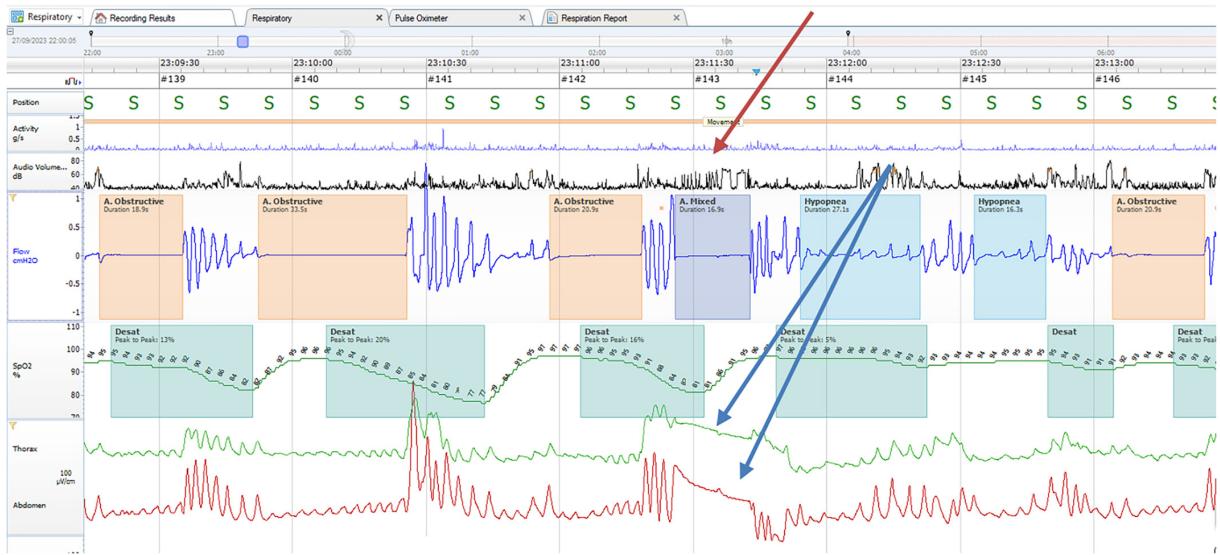
Phần lớn bệnh nhân OSA không tự nhận biết được các rối loạn hô hấp xảy ra trong lúc ngủ [42]. Hiện tượng này đặc biệt phổ biến ở những người sống hoặc ngủ một mình, do thiếu sự quan sát từ bên ngoài.

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 bệnh sử, 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 Tai Mũi Họng để tìm nguyên nhân gây hẹp đường thở trên, chẳng hạn như phì đại amidan, hay bất thường vùng hàm mặt.

Các đánh giá thường bắt đầu bằng khảo sát giấc ngủ toàn diện, bao gồm thu thập bệnh sử liên quan đến các triệu chứng đặc trưng, 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 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, áp lực đường thở qua mũi, và cường độ tiếng ngáy [45, 46]. 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 chuỗi thời gian, cho phép quan sát đồng thời các kênh hô hấp. Hệ thống cung cấp công cụ đánh dấu tự động và thủ công các sự kiện bất thường

như ngưng thở và giảm thở.



Hình 1.4: Màn hình tổng hợp tín hiệu của PSG

Trong số các tín hiệu này, tư thế ngủ là một thông số đặc biệt quan trọng, góp phần xác định thể bệnh positional OSA (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.

Tuy nhiên, phương pháp PSG vẫn tồn tại nhiều hạn chế trong thực tế. Việc bệnh nhân bắt buộc phải lưu trú qua đêm tại cơ sở y tế, 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. Chi phí cao, thời gian chờ đợi dài, tín hiệu hay bị gián đoạn hoặc nhiễu cũng là các trở ngại đáng kể. Kinh nghiệm thực tế cho thấy, 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 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 tiên tiến và mô hình học máy. Nhờ đó, các thiết bị có khả năng tự động thu nhận, phân tích và phân loại các thông số sinh lý đặc trưng cho hội chứng ngưng thở khi ngủ, 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

Loại thiết bị đeo	Tài liệu tham khảo
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

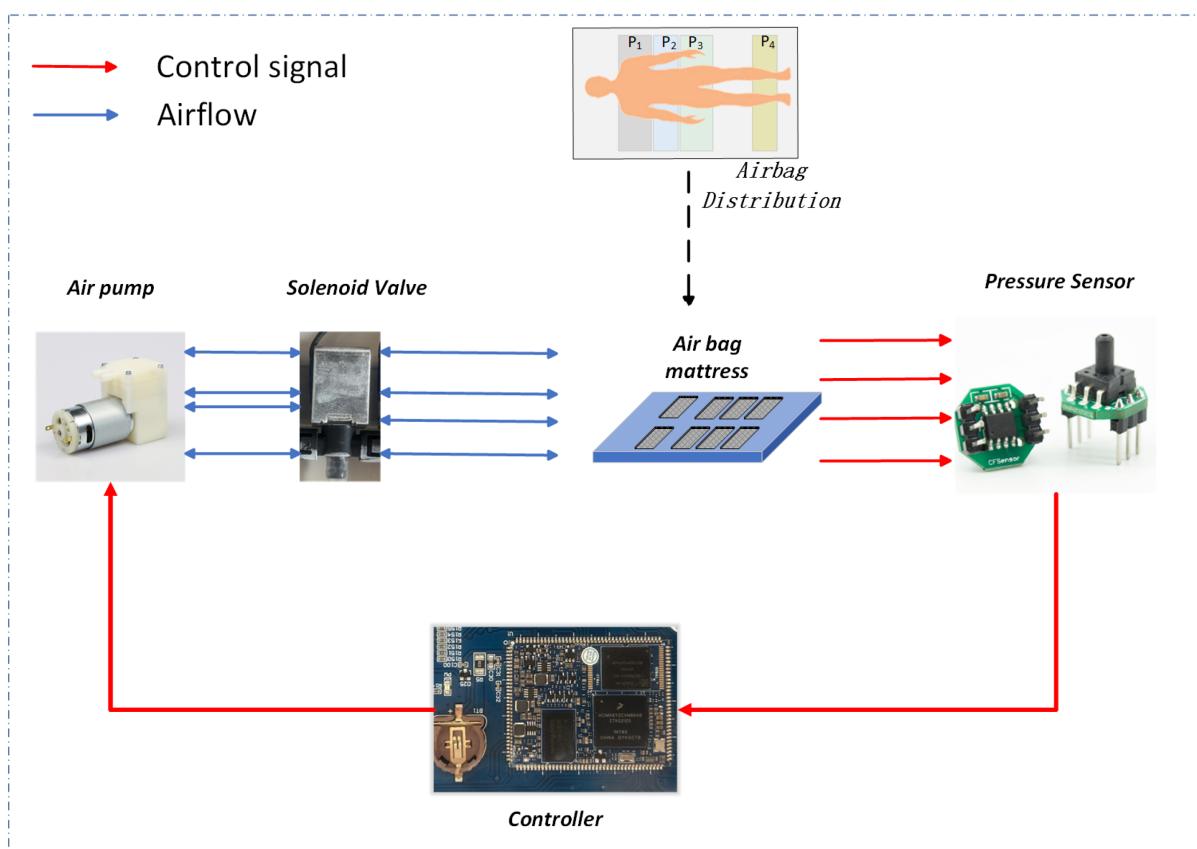
Ngoài ra, Domingues và cộng sự (2024) đề xuất một mô hình mạng nơ-ron nhân tạo sử dụng dữ liệu từ SpO<sub>2</sub>, cảm biến gia tốc và âm thanh ngáy của hệ thống Biologix, cho kết quả tiệm cận độ chính xác của PSG truyền thống [60]. Hướng tiếp cận khác được Boiko và cộng sự [61] phát triển dựa trên cảm 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, tích hợp khả năng học máy. Điều này không chỉ nâng cao tính riêng tư và độ tin cậy của dữ liệu, 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. Thiết bị 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 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 thống nệm 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

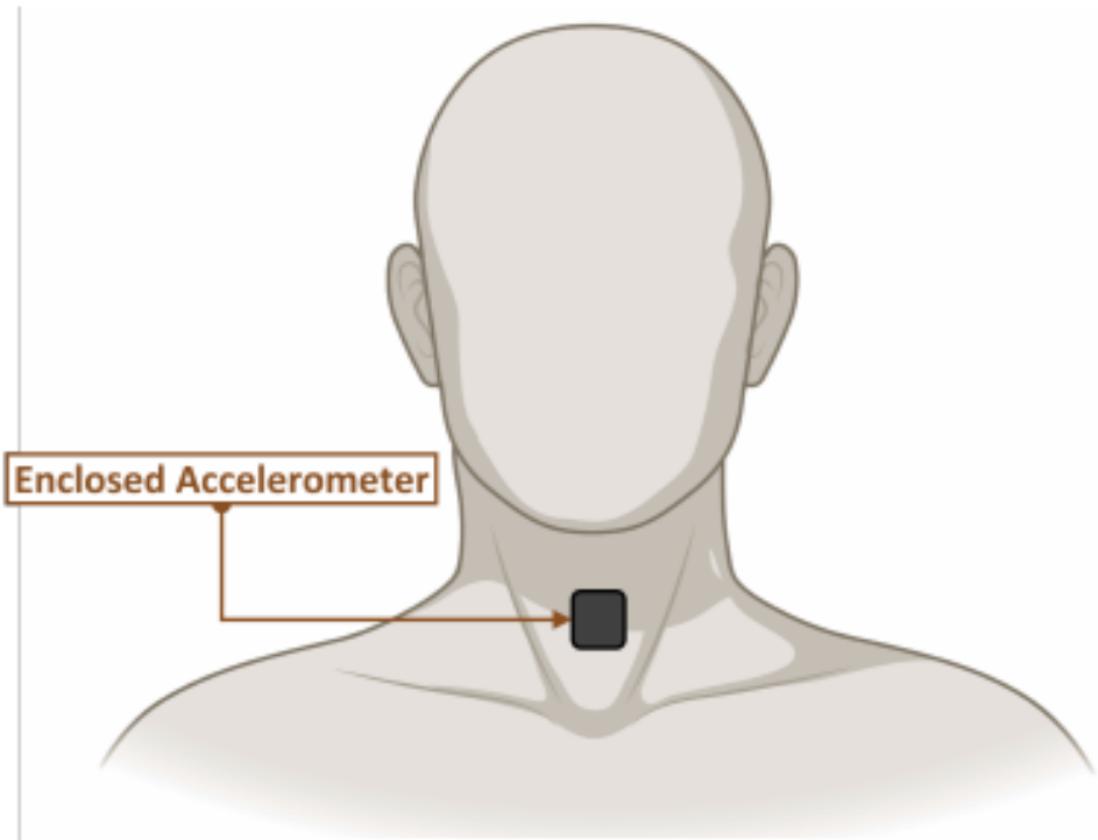
trong môi trường có kiểm soát. Ví dụ, Khan và cộng sự [64] ứng dụng kết hợp camera RGB và camera nhiệt, huấn luyện mô hình VGG16 và ResNet50, đạt độ chính xác 99%. Mahvash và cộng sự [65] tiếp tục mở rộng hướng nghiên cứu này bằng cách sử dụng camera hồng ngoại 2D, kết hợp học sâu với mô hình ResNet-152, 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 Giáo sư Lê Tiên Thường, trường đại học Bách Khoa TP Hồ Chí Minh 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, hiện nay với sự phát triển vượt bậc của điện thoại di động, 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 [21] 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.

Tuy nhiên, các hệ thống trên đều có hạn chế về tính phức tạp phần cứng, yêu cầu nhiều cảm biến, khó áp dụng trong điều kiện thực tế tại nhà. Vì vậy, xu hướng nghiên cứu gần đây 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 [15] phát triển hệ thống iSleePost 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 0.82-0.84. 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 tuyến 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.

Nghiên cứu [20], 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), vị trí thuận lợi cho việc kết hợp theo dõi đa thông số sinh lý trong các ứng dụng y tế như OSA. Ba mô hình được thử nghiệm gồm Decision Tree (DT), Extra-Trees (ET) và LSTM Neural Network (LSTM-NN), nhằm đánh giá hiệu năng trong điều kiện giới hạn về năng lượng và bộ nhớ của thiết bị đeo. Kết quả cho thấy chỉ với tần số lấy mẫu 5 Hz và cửa sổ 1 s, các mô hình vẫn đạt độ chính xác trung bình  $> 98\%$  và f1-score lần lượt 0.945, 0.975, 0.965, trong đó 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 hình trên biển. Đặc biệt, tác giả Vũ Hoàng Diệu và cộng sự [19] 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%. 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 thấp và khả năng triển khai thực tế.



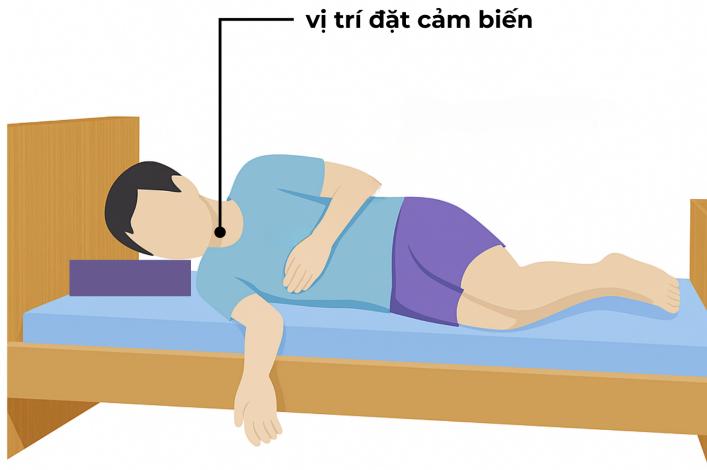
Hình 1.7: Ví trí đặt cảm biến được đề xuất trong nghiên cứu [20]

Bám sát định hướng này, luận văn đề xuất một hệ thống sử dụng duy nhất một cảm biến gia tốc gắn dưới 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 giá 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 khả năng ước lượng chỉ số AHI.

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

Từ góc độ học thuật, các công trình nghiên cứu gần đây đã khẳng định tính hiệu quả và độ tin cậy của cảm biến gia tốc trong việc phân loại tư thế ngủ ở người.

Ưu điểm nổi bật của cảm biến gia tốc nằm ở khả năng vận hành độc lập với mức



Hình 1.8: Vị trí tối ưu để gắn cảm biến gia tốc

tiêu thụ năng lượng thấp, đồng thời có thể tích hợp dễ dàng vào các hệ thống vi điều khiển. 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, giới hạn kỹ thuật chủ yếu của cảm biến gia tốc xuất phát từ vị trí gắn cảm biến. Việc đặt sai vị trí hoặc xoay lệch trục có thể dẫn đến sai lệch trong kết quả phân loại tư thế. Hơn nữa, cảm biến này không cung cấp thông tin về các thông số sinh lý quan trọng như độ bão hòa oxy trong máu ( $SpO_2$ ), nhịp thở hoặc nhịp tim. Do đó, để chẩn đoán toàn diện hội chứng ngưng thở khi ngủ tắc nghẽn (OSA), việc tích hợp cảm biến đa kênh hoặc ứng dụng các mô hình suy luận dựa trên học máy là cần thiết nhằm bù đắp cho những thiếu hụt về dữ liệu sinh lý.

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 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**

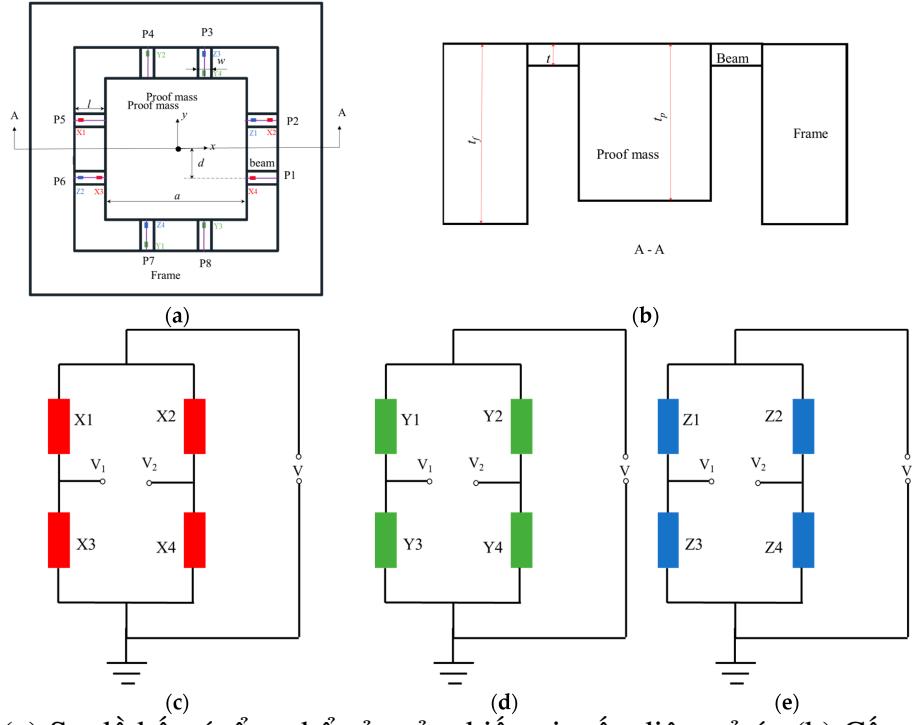
Cảm biến gia tốc là một thiết bị đo lường có khả năng phát hiện và ghi nhận gia tốc theo thời gian. Với ưu điểm nhỏ gọn, tiêu thụ năng lượng thấp và chi phí hợp lý, cảm biến gia tốc được ứng dụng rộng rãi trong nhiều lĩnh vực như điện tử tiêu dùng, ô tô, công nghiệp, trong các thiết bị theo dõi hoạt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

Hiện nay, có 3 dạng chính của cảm biến gia tốc vi cơ điện tử (Micro-Electro-Mechanical Systems - MEMS) là kiểu điện trở áp, kiểu điện dung và kiểu áp điện [74].

### **Cảm biến gia tốc MEMS điện trở áp**

Đầu tiên, cảm biến MEMS điện trở áp ra đời sớm nhất, có cấu tạo gồm một khối dầm bằng vật liệu có tính điện trở áp, kèm theo một khối trọng (mass) và khung đỡ [75]. Bên mặt trên của khối trọng, các dầm và khung nằm trên cùng một mặt phẳng, đảm bảo tính đối xứng cấu trúc và khả năng tương thích cao với quy trình chế tạo vi cơ điện tử. Khi cảm biến chịu tác động của gia tốc ngoài, ứng suất cơ học xuất hiện tại vùng đầu 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. Dựa trên đặc tính hình học và sự đối xứng của cấu trúc, mô hình toán học của cảm biến có thể được chia thành hai trường hợp chính: (1) Gia tốc tác dụng theo phương vuông góc với mặt phẳng (trục Z); (2) Gia tốc tác dụng trong mặt phẳng (trục X hoặc Y).



Hình 1.9: (a) Sơ đồ bố trí tổng thể của cảm biến gia tốc điện trở áp (b) Cấu trúc mặt cắt qua khối trọng và 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.

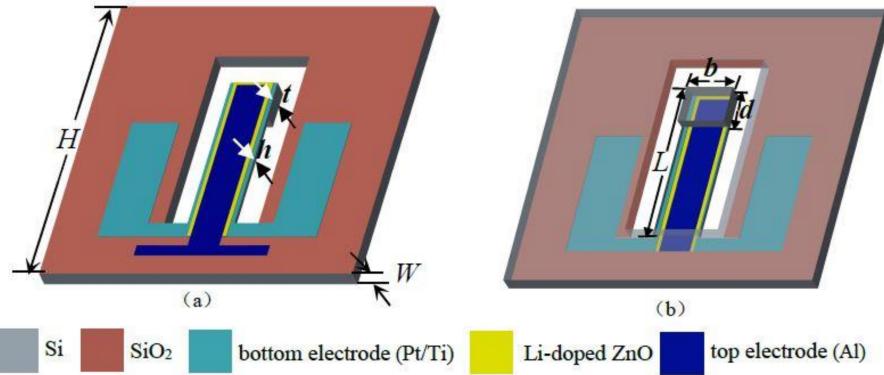
### Cảm biến gia tốc MEMS áp điện

Cảm biến gia tốc được cấu tạo gồm hai phần chính [76]: (1) Thanh dầm là một dầm silicon có một đầu cố định liên kết với khối trọng. Khi có gia tốc tác động, khối lượng này tạo ra lực quán tính khiến 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 ( $t$ ). 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.

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



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

Lí lầm 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:

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

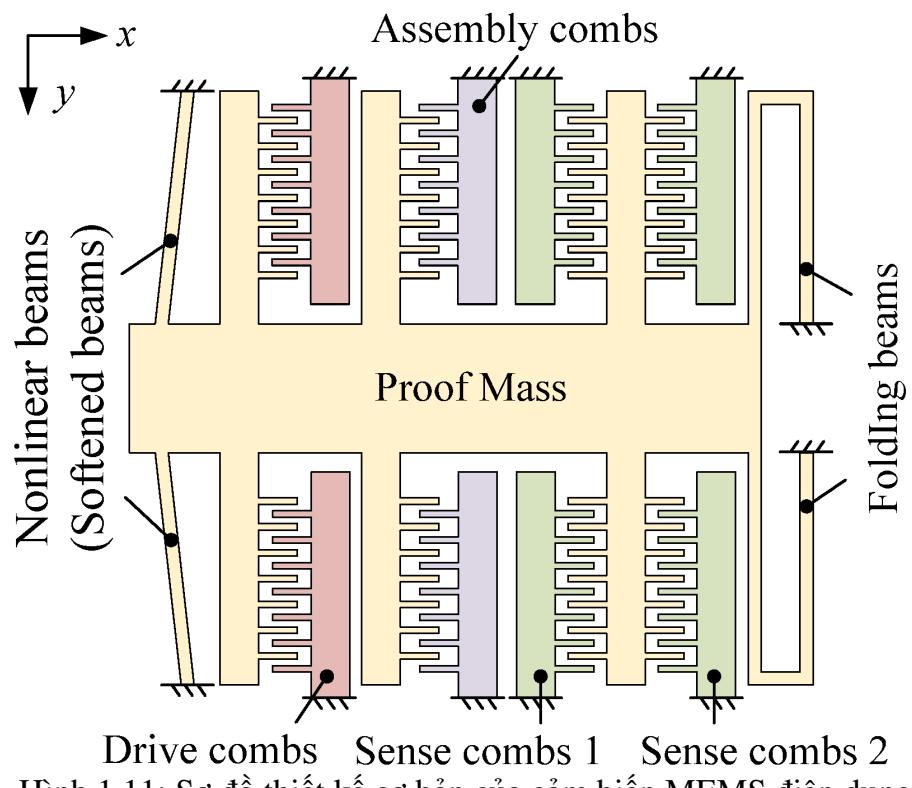
Kết hợp với định nghĩa độ nhạy của cảm biến, ta có biểu thức:

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

### Cảm biến gia tốc MEMS điện dung

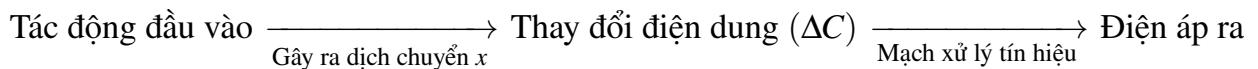
Cảm biến gia tốc điện dung được gồm ba khối chức năng chính: khối trọng, hệ dầm đàn hồi và cụm lược điện cực [78]. Khối trọng là phần trung tâm của cảm biến, dao động

khi có gia tốc tác động. Các dầm có nhiệm vụ đỡ và liên kết khối lượng này, đồng thời cho phép chuyển động đàn hồi trong giới hạn thiết kế. Cụm lược ghép (assembly combs) được sử dụng để căn chỉnh trong quá trình chế tạo, trong khi cụm lược kích thích (drive combs) tạo dao động cưỡng bức cho khối trọng trong các thí nghiệm đo đáp ứng. Cụm lược đo (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 lược đo (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.



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 tu

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  $V_{\text{out}}$ .

### **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 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 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 (4), có thể thấy rằng điện áp vi sai đầu ra  $V_{out}$  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:

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

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 (&gt;10 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 tuý, 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 thấp. 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 Cortex-M4 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 giắc 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**

Hiện nay, trí tuệ nhân tạo (AI) đã và đang len lỏi vào hầu hết các lĩnh vực của đời sống, đặc biệt là trong các thiết bị y tế tại nhà. Việc ứng dụng các kỹ thuật học máy (machine learning) không chỉ giúp nâng cao hiệu suất xử lý, giảm chi phí vận hành, mà còn đảm bảo tính ổn định và độ lặp lại cao, đồng thời giảm thiểu sự can thiệp trực tiếp của con người.

Với lượng dữ liệu phong phú và liên tục được thu thập từ các cảm biến sinh học, 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

<b>STT</b>	<b>Giai đoạn</b>	<b>Mô tả tổng quát</b>
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 [15, 21, 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, 19, 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 [19, 20].

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 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 thống 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* [20], 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 [19, 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 điện tử. 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

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
[20]	18 đối tượng	4 tư thế ngủ	12 đặc trưng miền thời gian	DT, Extra-Trees, LSTM-NN	>98%	BLE -> laptop	Có
[15]	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 trục	Nguồn	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	Nguồn	>99.2%	cảm biến -> PC	Không
[59]	18 đối tượng	4 tư thế ngủ	12 đặc trưng miền thời gian	Nguồn , Extra-Trees	>95%	cảm biến -> PC	Không
[18]	13 đối tượng	4 tư thế ngủ và không phải	trung bình giá trị gia tốc 3 trục	Nguồn	>96%	cảm biến -> PC	Không
[70]	7 đối tượng	4 tư thế ngủ	trung bình giá trị gia tốc 3 trục	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, XG-Boost, AdaBoost	95.65%	cảm biến -> PC	Không

NM = Không đề cập; DT = Decision Tree; LSTM = Long Short-Term Memory; RF = Random Forest; SVM = Support Vector Machine;

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 [20] 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.

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 của Vu et al. [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(Decision Tree - DT), rừng ngẫu nhiên (Random Forest - RF) 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**

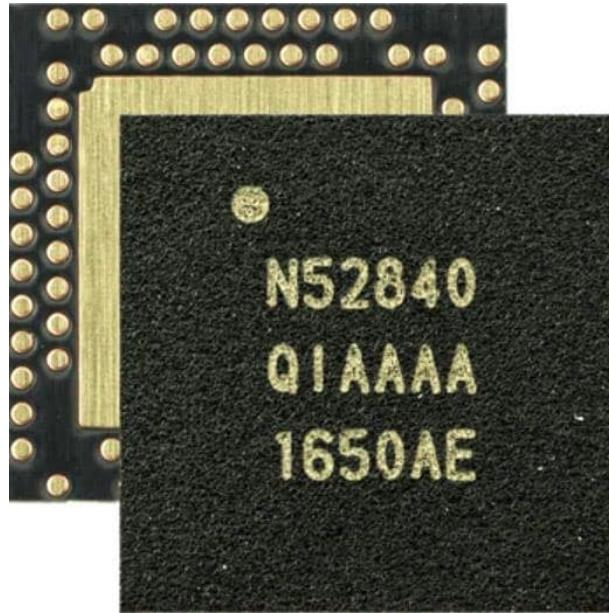
# **HỆ THỐNG THU THẬP VÀ XỬ LÝ TÍN HIỆU CẢM BIẾN**

### **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 nguyên lý, 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 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



Hình 2.1: Nordic NRF52840

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, 20]. 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 gấp 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 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

# 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><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></li><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>

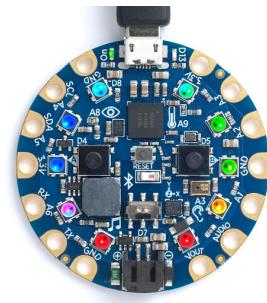
Hình 2.2: Các tính năng Nordic nRF52840

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



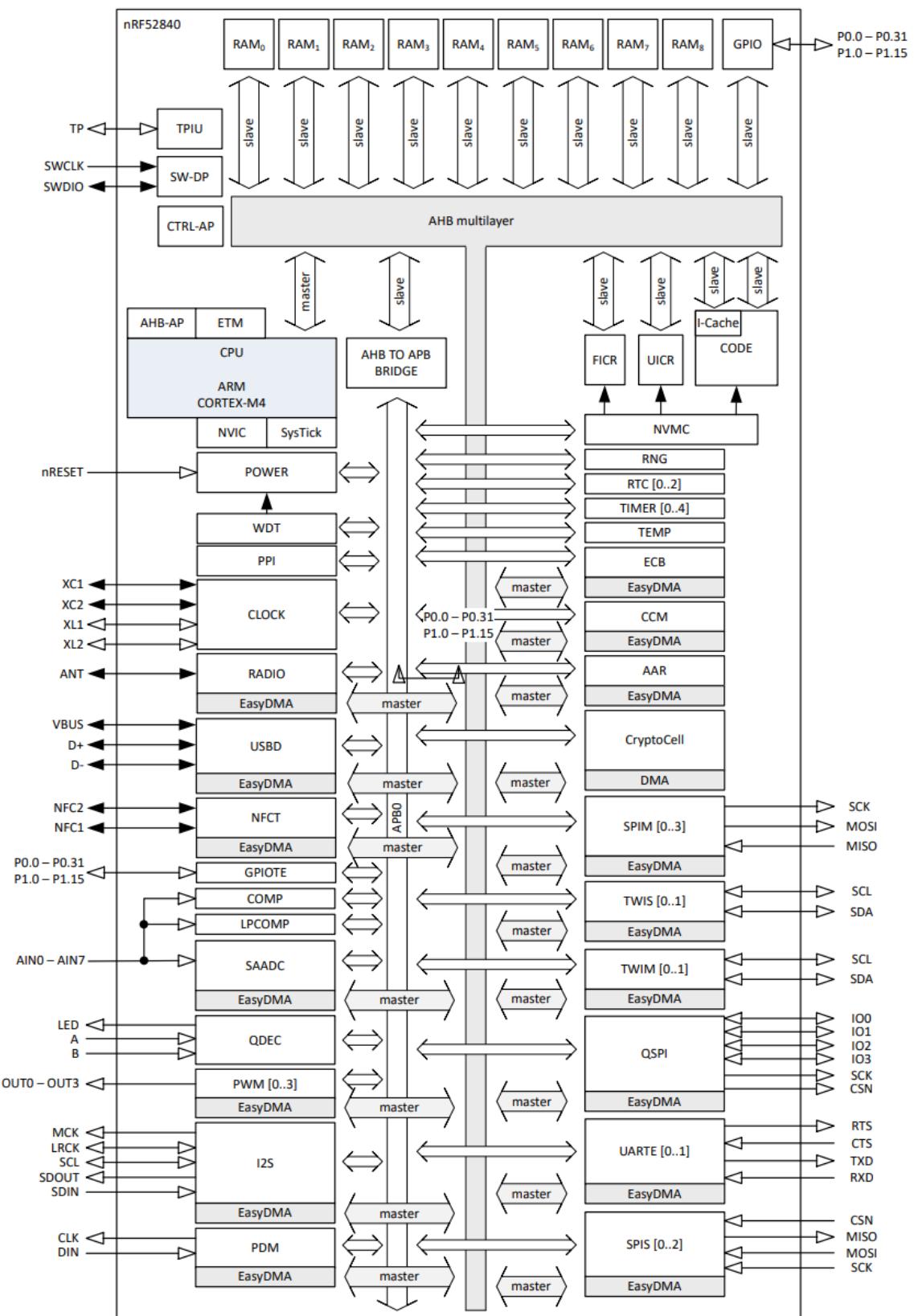
(b) Adafruit Circuit Playground Bluefruit

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 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



Hình 2.3: Sơ đồ khối Nordic NRF52840

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 biên độ nhỏ trong khi ngủ.

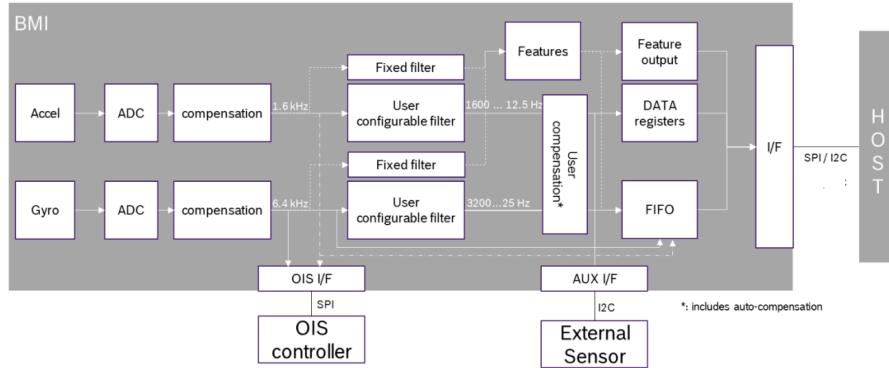
Thứ hai, mức tiêu thụ năng lượng thấp và khả năng hoạt động ổn định: cảm biến tiêu thụ trung bình chỉ khoảng  $685 \mu A$  khi hoạt động bình thường, đồng thời được trang bị bộ quản lý năng lượng tích hợp hỗ trợ nhiều chế độ tiết kiệm điện khác nhau. Điều này cho phép thiết bị hoạt động lâu dài trên 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 0.7Hz đến 1.6 kHz cho gia tốc kế, phù hợp bài toán phân loại giấc ngủ yêu cầu tần số lấy mẫu thấp 10Hz.

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



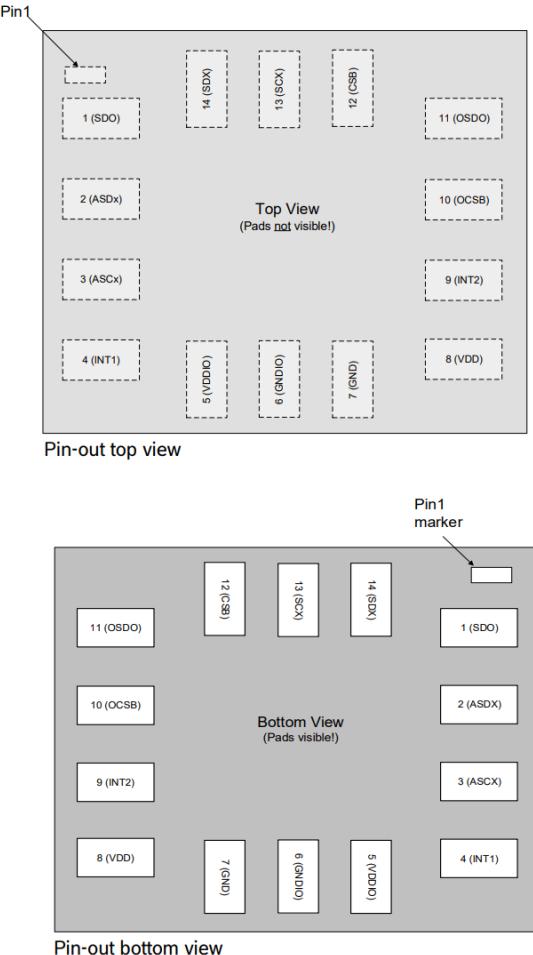
Hình 2.6: Sơ đồ khái niệm Bosch BMI270

### 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.

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.



Hình 2.7: Sơ đồ chân Bosch BMI270

### Các thành phần, chức năng

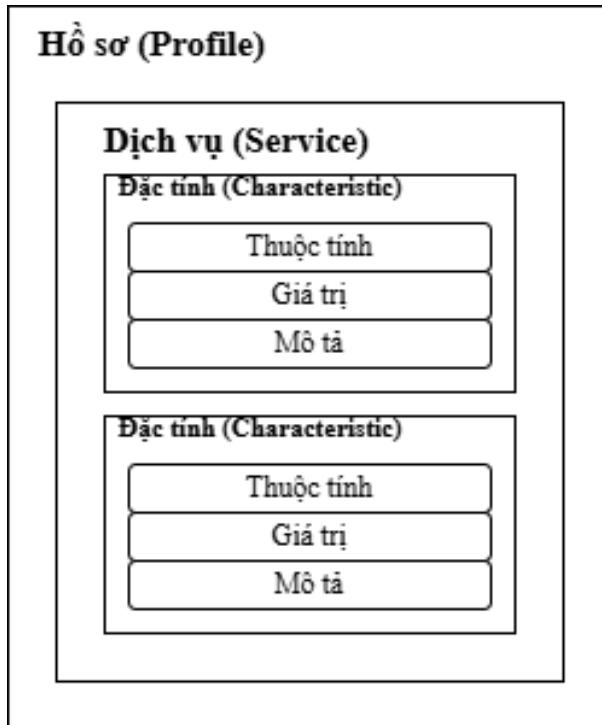
BLE là giao thức kết nối không dây được thiết kế chuyên biệt cho các ứng dụng năng lượng thấp, hoạt động ở băng tần ISM 2.4 GHz, hỗ trợ thông lượng ứng dụng lên đến 1.4 Mbps [104]. BLE hiện được hỗ trợ phổ biến trên hầu hết các hệ điều hành như iOS, Android, macOS, Windows và Linux, cũng như trong các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

	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

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 - 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 mạng 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 nguyên lý**

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?

Qua các câu hỏi đó, tác giả từng bước nghiên cứu và hoàn thiện thiết kế sơ đồ nguyên lý.

### **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 4cm. 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 khối 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 kháng, 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 khối điều khiển [106].



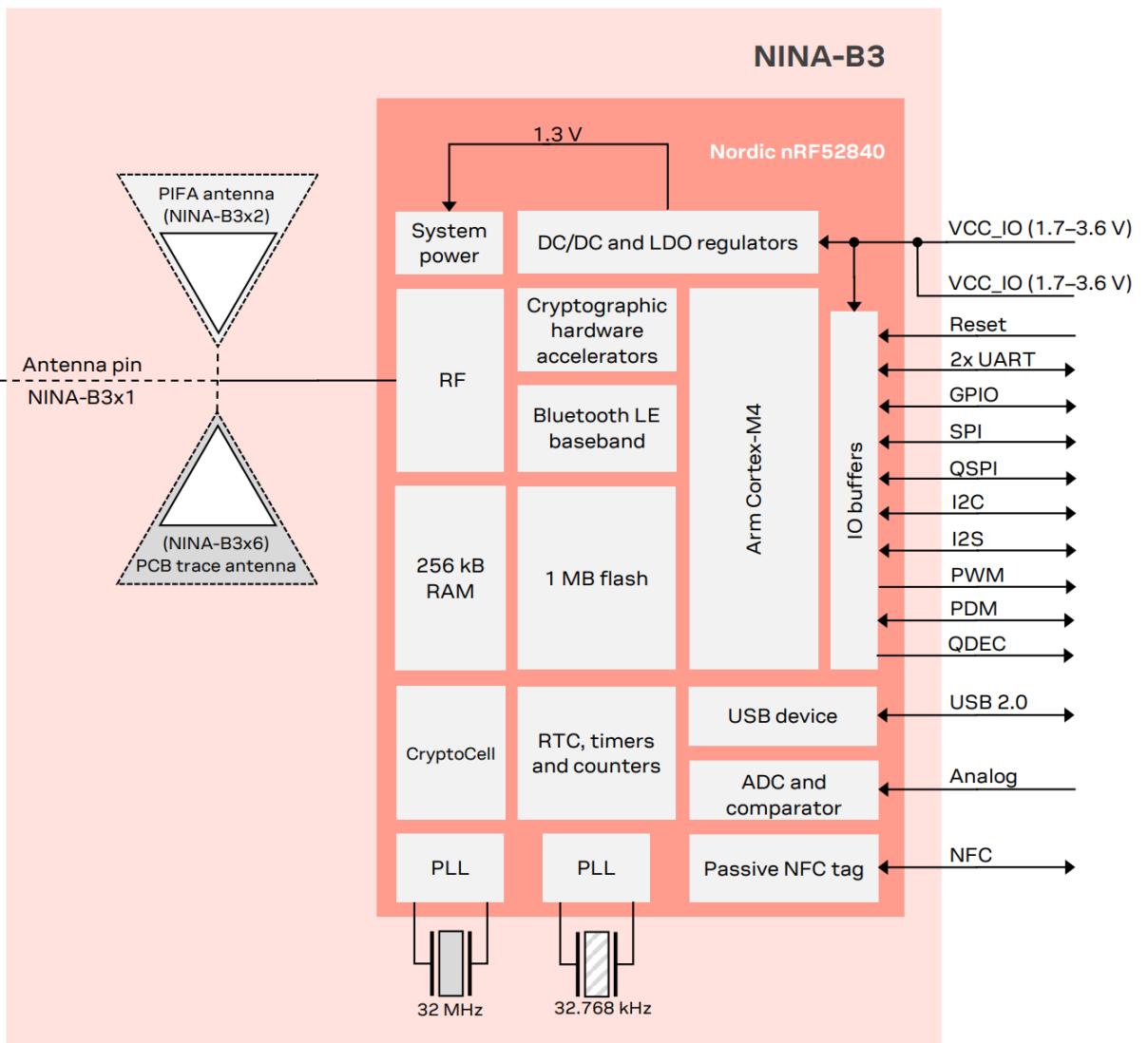
Hình 2.10: U-blox NINA-B306

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. 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.

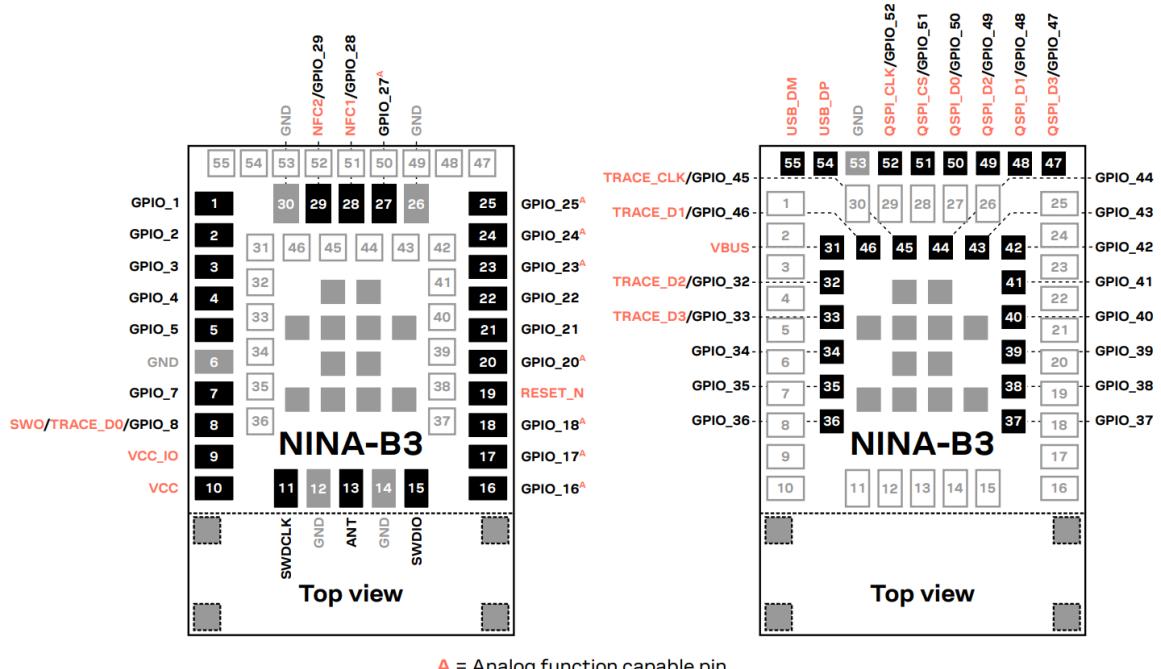
### **Khối cảm biến**

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.

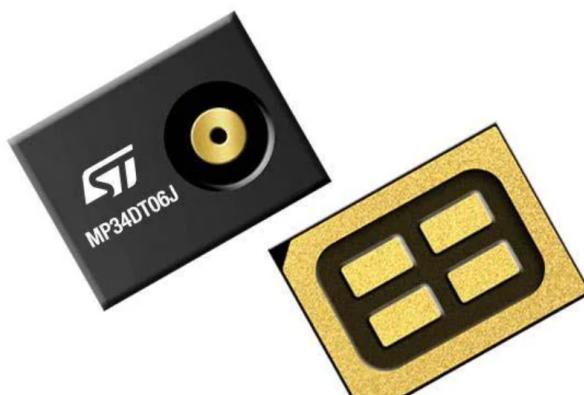


Hình 2.11: Sơ đồ khối U-blox NINA-B306



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

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.



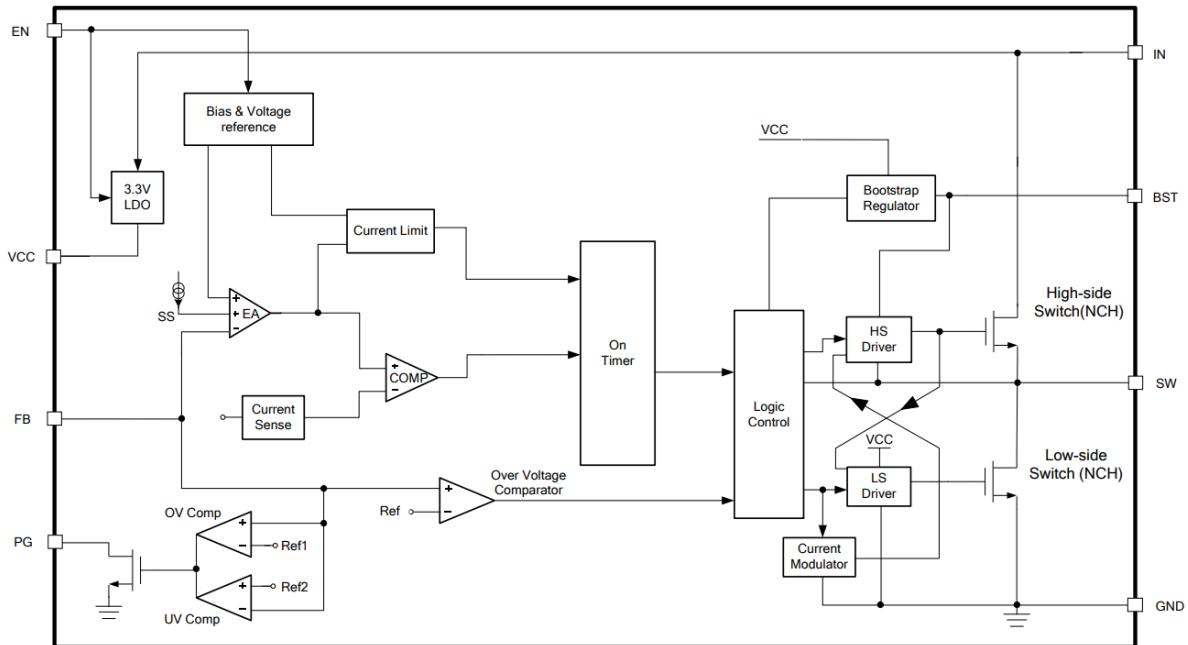
Hình 2.13: STM MP34DT06J

### **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 khối 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 2.14: Sơ đồ khối IC MP2322GQH, dải đầu vào 3V–22V, dòng tải tối đa 1A, hiệu suất cao, tần số chuyển mạch 1.25MHz (Monolithic Power Systems)

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 khôi 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 khôi nạp và gõ 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.

## **Mạch nguyên lý**

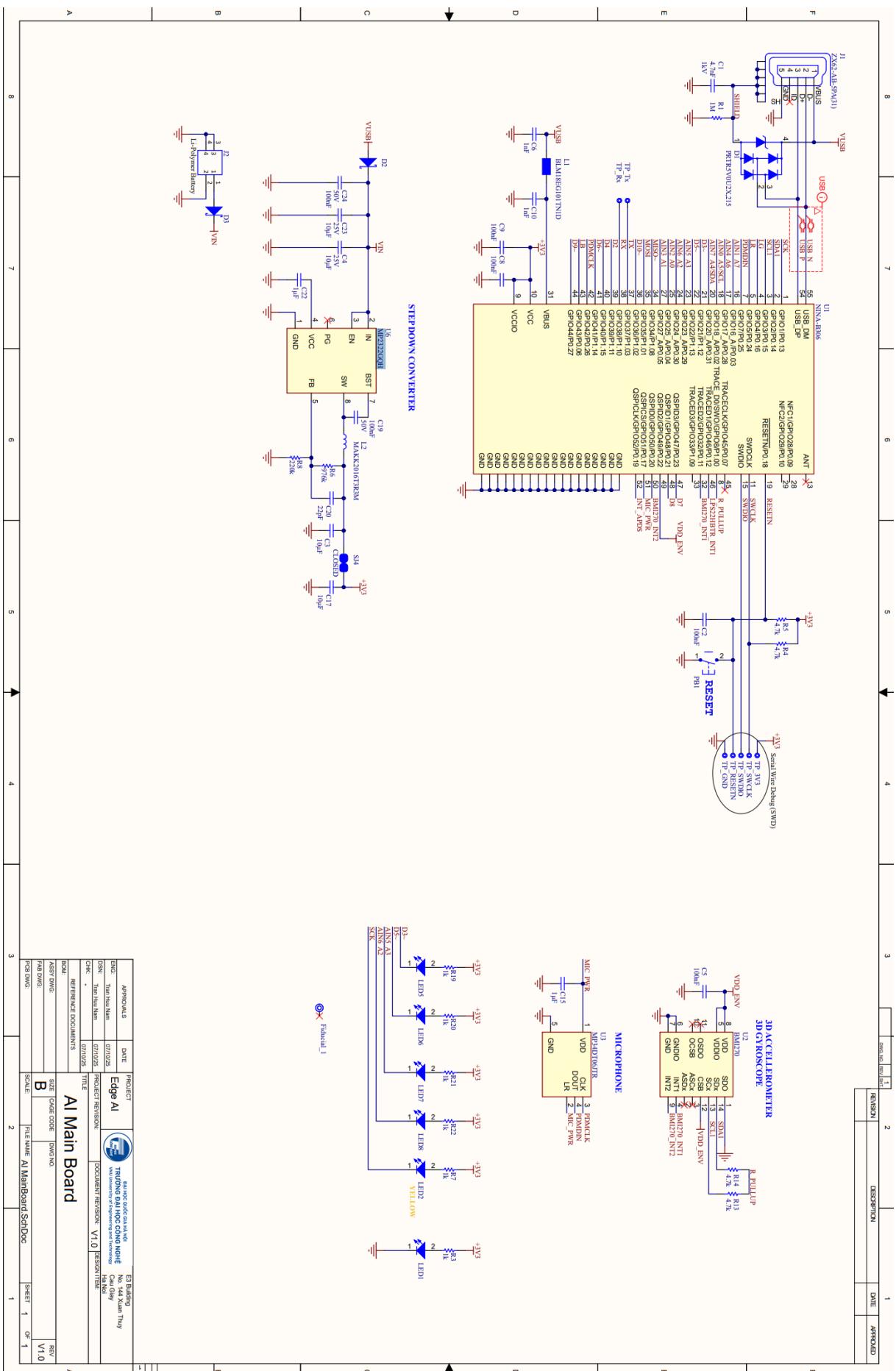
Sau khi xác định các linh kiện, các yêu cầu về điện áp, kiểu dữ liệu đầu vào, đầu ra của các thành phần kể trên, tác giả tiến hành vẽ nguyên lý mạch thông qua phần mềm Altium Designer.

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

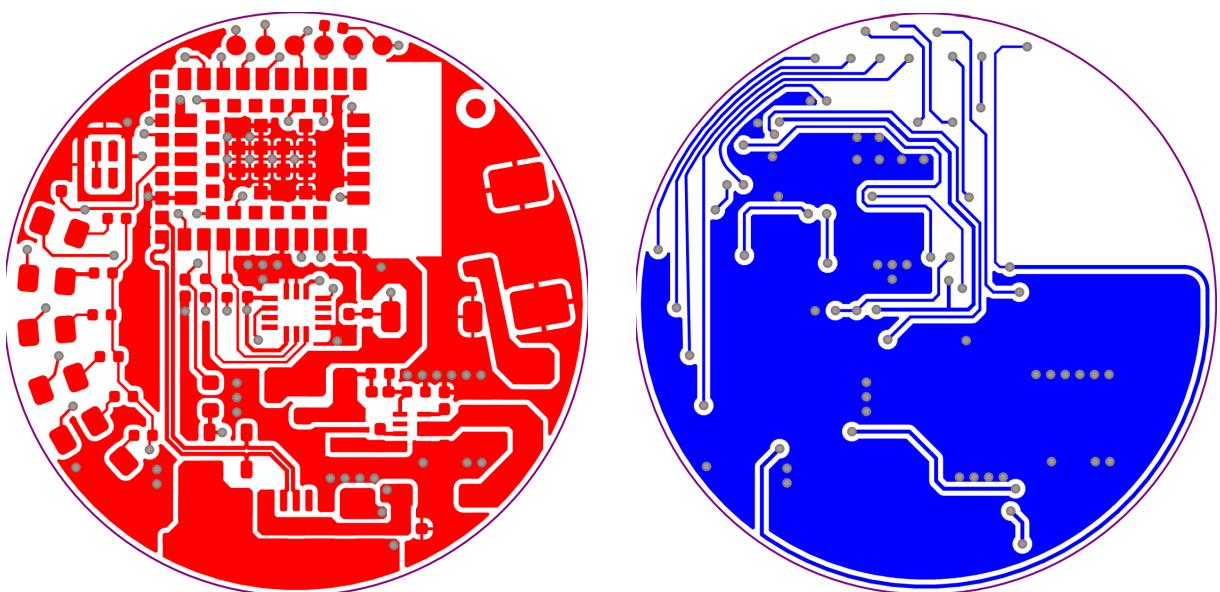
Kết quả được thể hiện tại Hình 2.15

Sau khi hoàn thiện thiết kế, tác giả thực hiện đặt in mạch, hàn lắp các linh kiện theo sơ đồ nguyên lý, 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.

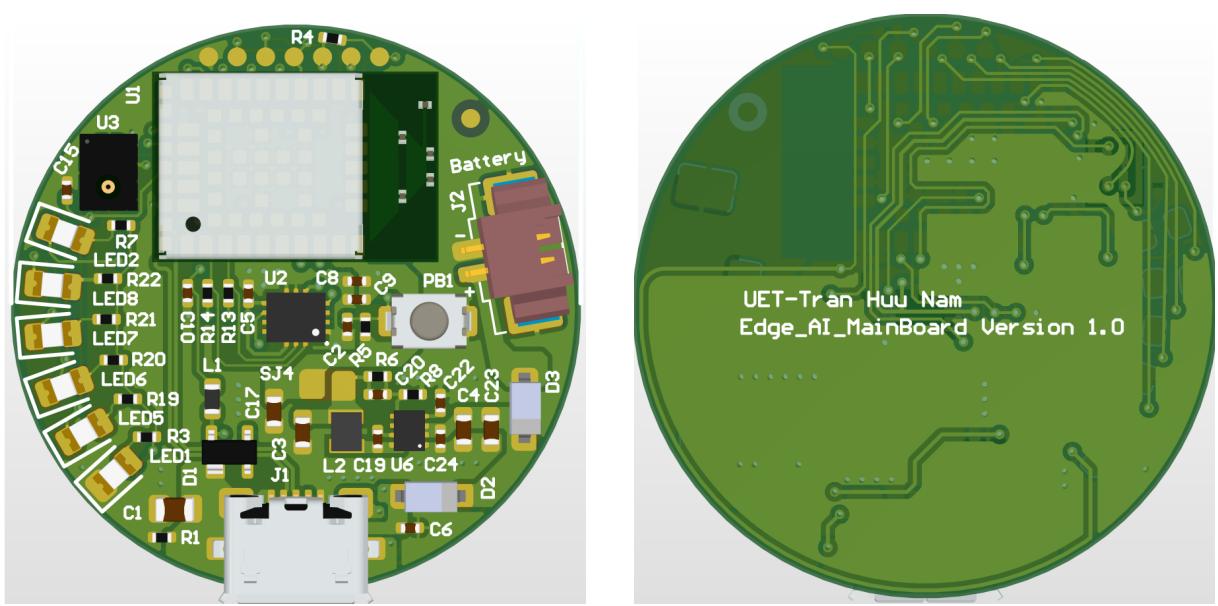
Hình 2.18 là kết quả mạch đã hoàn thiện.



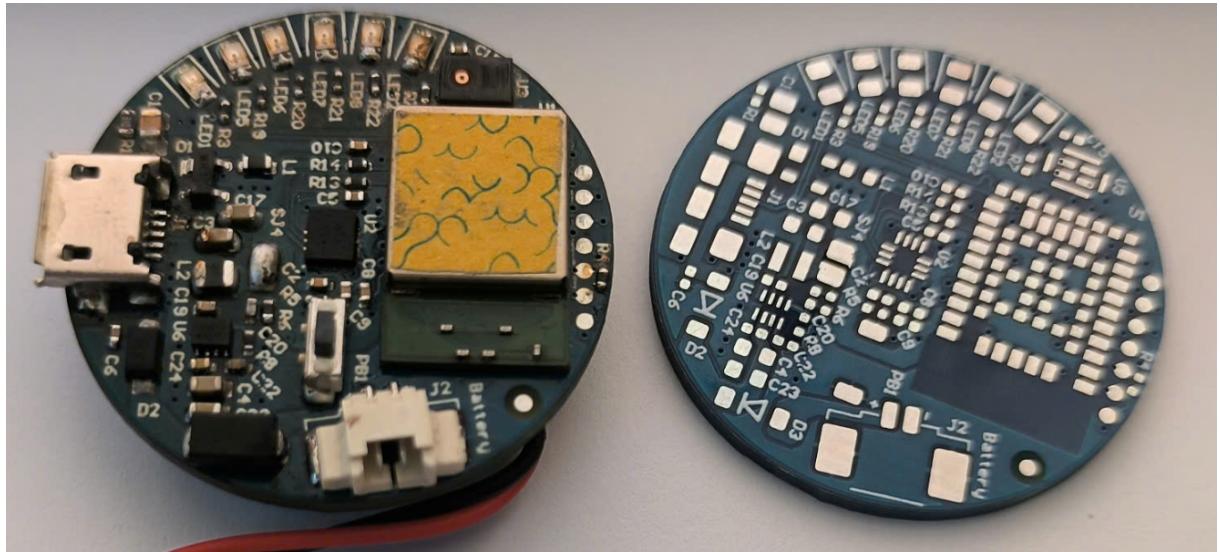
Hình 2.15: Sơ đồ nguyên lý <sup>49</sup>



Hình 2.16: Bố trí mạch in hai mặt của bo mạch: (a) lớp trên; (b) lớp dưới



Hình 2.17: Mô hình mạch 3D: (a) lớp trên; (b) lớp dưới



Hình 2.18: Mạch hoàn chỉnh

## 2.2. Hệ thống phần mềm

Phần này trình bày tổng quan kiến trúc hệ thống bao gồm: lập trình firmware trên vi điều khiển để thu thập dữ liệu cảm biến, thiết kế ứng dụng di động làm cầu nối giữa phần cứng và hệ thống đám mây, cùng với backend và cơ sở dữ liệu lưu trữ phục vụ huấn luyện mô hình. 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 vi xử lý

Mục tiêu của phần lập trình không chỉ dừng lại ở việc xây dựng các đoạn mã điều khiển cơ bản, mà còn hướng đến việc tận dụng tối đa hệ sinh thái thư viện và framework đã được cộng đồng kiểm chứng. Cách tiếp cận này giúp rút ngắn đáng kể thời gian phát triển, giảm thiểu rủi ro sai sót khi lập trình ở mức thanh ghi thấp, đồng thời bảo đảm tính ổn định và khả năng mở rộng của hệ thống trong các giai đoạn triển khai tiếp theo.

Quá trình phát triển phần mềm được thực hiện trên nền tảng *Arduino IDE*, kết hợp hai thư viện chính: Adafruit Circuit Playground (cho bo Adafruit Playground) và Arduino\_LSM9DS1/ArduinoBLE (cho bo Arduino Nano 33 BLE Sense). Việc sử dụng các thư viện này không chỉ đơn giản hóa thao tác giao tiếp với cảm biến và quản lý tín

hiệu, mà còn tạo điều kiện tương thích trực tiếp với *TensorFlow Lite for Microcontrollers*, giúp rút ngắn đáng kể giai đoạn triển khai TinyML trên thiết bị.

## Cấu hình vi xử lý và cảm biến

Trước khi tích hợp kết nối không dây, hệ thống được thiết lập các thông số hoạt động của vi xử lý và cảm biến dựa trên định hướng tối ưu năng lượng và đảm bảo chất lượng dữ liệu đã phân tích ở các chương trước. Mục tiêu của giai đoạn này là duy trì thời lượng pin dài trên thiết bị đeo, đồng thời vẫn thu được dữ liệu gia tốc đủ chính xác để phục vụ huấn luyện mô hình nhận diện tư thế ngủ.

Cụ thể, vi xử lý nRF52840 được vận hành ở chế độ tiết kiệm năng lượng trong trạng thái chờ, và chỉ chuyển sang xử lý khi nhận tín hiệu ngắn từ cảm biến. Cảm biến gia tốc MEMS LIS3DH (trên Adafruit Playground) và IMU tích hợp (trên Arduino Nano 33 BLE Sense) được cấu hình với tần số lấy mẫu 10 Hz và dải đo  $\pm 2g$ . Thiết lập này được lựa chọn vì:

Tần số 10 Hz đủ để ghi nhận chuyển động chậm và biến đổi tư thế trong khi ngủ, đồng thời giúp giảm thiểu khối lượng dữ liệu cần xử lý và lưu trữ.

Dải đo  $\pm 2g$  cung cấp độ nhạy cao, phù hợp với các chuyển động mức thấp của cơ thể khi thay đổi tư thế, đồng thời hạn chế tiêu thụ năng lượng hơn so với các dải đo rộng hơn.

Ngoài ra, cảm biến được cấu hình sử dụng ngắn dữ liệu (INT pin), tức là chỉ khi có mẫu mới trong thanh ghi thì tín hiệu ngắn mới được gửi tới vi xử lý. Cách tiếp cận này cho phép vi xử lý duy trì trạng thái ngủ phần lớn thời gian, chỉ thức dậy để đọc dữ liệu cần thiết, từ đó giảm đáng kể mức tiêu thụ năng lượng.

## Cấu hình BLE

Trong quá trình khởi tạo (`setup()`), hệ thống thực hiện cấu hình các thành phần cốt lõi của giao thức Bluetooth Low Energy (BLE), bao gồm: tạo bản tin quảng bá (advertising), thiết lập cơ chế kết nối/ngắn kết nối, cũng như định nghĩa cấu trúc dịch vụ và đặc tính theo mô hình GATT. Điểm mạnh của BLE là cho phép thiết bị đeo duy trì mức tiêu thụ năng lượng cực thấp trong khi vẫn bảo đảm tốc độ truyền dữ liệu cảm biến đủ nhanh để

phục vụ các ứng dụng thời gian thực. Đây là một yêu cầu quan trọng trong các thiết bị y tế đeo người, vốn cần hoạt động liên tục trong nhiều giờ hoặc nhiều ngày chỉ với một pin nhỏ. Quy trình này được minh họa trong Hình ??.

Trong đoạn mã 2.1, hàm `startAdv()` đảm nhiệm cấu hình quảng bá BLE cho thiết bị. Quá trình này bao gồm: thiết lập cờ kết nối tổng quát, chèn thông tin công suất truyền (Tx Power), thêm UUID của dịch vụ tư thế (`positionService`) và tên thiết bị vào gói quảng bá. Đặc biệt, các thông số quảng bá được cấu hình theo khuyến nghị của Apple (20 ms ở chế độ nhanh, 152.5 ms ở chế độ chậm, chuyển đổi sau 30 giây), nhằm đảm bảo mức độ tương thích tối đa với hệ sinh thái iOS. Cách cấu hình này giúp thiết bị duy trì trạng thái “luôn sẵn sàng” để ghép nối, nhưng không tiêu tốn năng lượng quá mức.

Trong cấu trúc dịch vụ, hệ thống định nghĩa một dịch vụ chính với UUID 0x1821, kèm theo hai đặc tính cảm biến: gia tốc (UUID 0x2713, đơn vị  $\text{m/s}^2$ ) và gia tốc góc (UUID 0x2744, đơn vị  $\text{rad/s}^2$ ). Mặc dù phần cứng hỗ trợ thu thập song song nhiều loại dữ liệu cảm biến, khoá luận này tập trung vào tín hiệu gia tốc tuyến tính - loại dữ liệu được coi là trực quan nhất để phân loại tư thế ngủ, đồng thời giảm độ phức tạp của mô hình TinyML trong giai đoạn thử nghiệm.

Cách tiếp cận này chứng minh rằng việc lựa chọn đúng nền tảng và thư viện không chỉ giải quyết nhu cầu trước mắt (truyền dữ liệu ổn định, tiết kiệm pin) mà còn mở ra khả năng tích hợp lâu dài với các hệ thống phân tích dữ liệu trên di động hoặc đám mây, phù hợp với hướng phát triển thiết bị chăm sóc sức khoẻ thông minh theo chuẩn công nghiệp.

Ngoài các thao tác khởi tạo dịch vụ, thư viện BLE của Adafruit còn cung cấp các phương thức cấu hình đặc tính (*characteristics*) nhằm kiểm soát hành vi và bảo mật của kết nối BLE.

Cụ thể, phương thức `setProperties` cho phép cấu hình quyền truy cập của đặc tính, với các lựa chọn phổ biến như:

`CHR_PROPS_BROADCAST` phát sóng đặc tính (bit 0)

`CHR_PROPS_READ` cho phép thiết bị đọc (bit 1)

`CHR_PROPS_WRITE_WO_RESP` ghi không cần phản hồi (bit 2)

`CHR_PROPS_WRITE` ghi với phản hồi (bit 3)

---

```
1 void startAdv(void)
2 {
3     // Advertising packet
4     Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE);
5     Bluefruit.Advertising.addTxPower();
6
7     // Include HRM Service UUID
8     Bluefruit.Advertising.addService(positionService);
9
10    // Include Name
11    Bluefruit.Advertising.addName();
12
13    /* Start Advertising
14     * - Enable auto advertising if disconnected
15     * - Interval: fast mode = 20 ms, slow mode = 152.5 ms
16     * - Timeout for fast mode is 30 seconds
17     * - Start(timeout) with timeout = 0 will advertise forever (until
18       connected)
19
20     * For recommended advertising interval
21     * https://developer.apple.com/library/content/qa/qa1931/\_index.html
22     */
23
24     Bluefruit.Advertising.restartOnDisconnect(true);
25     Bluefruit.Advertising.setInterval(32, 244); // in unit of 0.625 ms
26     Bluefruit.Advertising.setFastTimeout(30); // number of seconds in
         fast mode
27     Bluefruit.Advertising.start(0);           // 0 = Don't stop
         advertising after n seconds
28 }
```

---

Mã nguồn 2.1: Tập lệnh khởi tạo và kết nối Bluetooth từ thư viện của AdaFruit

---

```
1 void setupPosition(void)
2 {
3
4     positionService.begin();
5
6     accelerometerCharacter.setProperties(CHR_PROPS_NOTIFY+CHR_PROPS_READ+CHR_PROPS_WRITE);
7     accelerometerCharacter.setPermission(SECMODE_OPEN, SECMODE_NO_ACCESS);
8     accelerometerCharacter.setFixedLen(9);
9     accelerometerCharacter.setCccdWriteCallback(cccd_callback); // Optionally capture CCCD updates
10    accelerometerCharacter.begin();
11    uint8_t accelerometerData[9] = { 0b00000000, 0b00000000,
12                                    0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000 };
13    // Set the characteristic to use 8-bit values, with the sensor
14    // connected and detected
15    accelerometerCharacter.write(accelerometerData, 9);
16
17    gyroscopeCharacter.setProperties(CHR_PROPS_READ);
18    gyroscopeCharacter.setPermission(SECMODE_OPEN, SECMODE_NO_ACCESS);
19    gyroscopeCharacter.setFixedLen(1);
20    gyroscopeCharacter.begin();
21    gyroscopeCharacter.write8(2); // Set the characteristic to 'Wrist'
22                                (2)
23 }
```

---

Mã nguồn 2.2: Gửi dữ liệu từ BLE

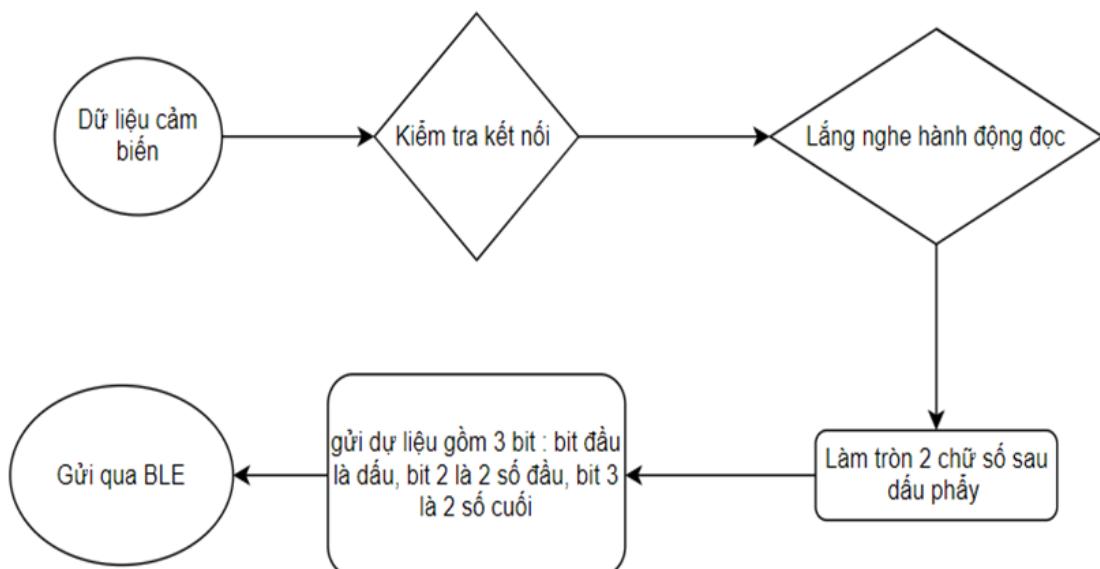
CHR\_PROPS\_NOTIFY gửi thông báo không xác nhận (bit 4)

CHR\_PROPS\_INDICATE gửi thông báo có xác nhận (bit 5)

Ngoài ra, một số phương thức bổ trợ khác bao gồm:

setPermission thiết lập quyền truy cập và mức độ bảo mật (ví dụ: không cần xác thực, cần mã hoá, v.v.)

setFixedLen xác định độ dài cố định của dữ liệu truyền



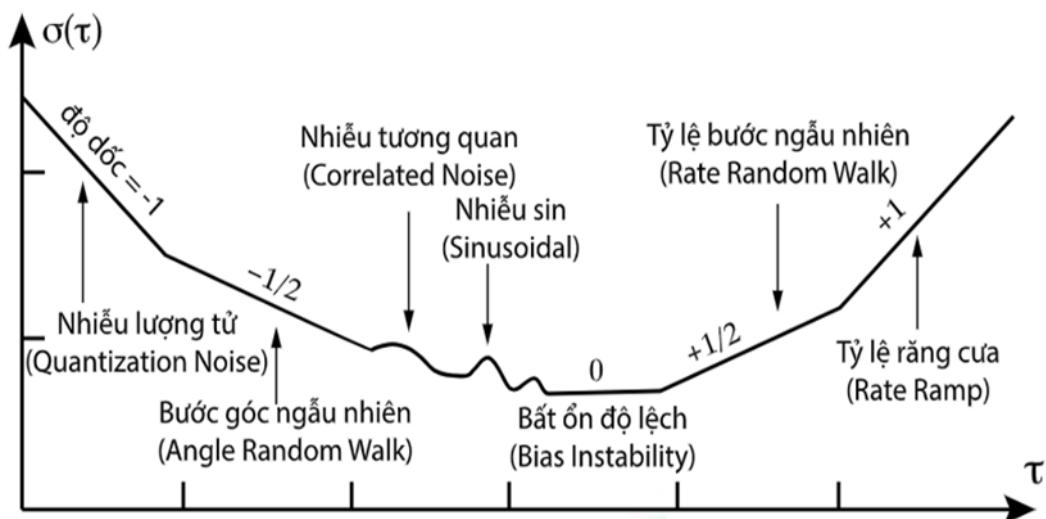
Hình 2.19: Lưu đồ luồng gửi thông tin BLE

Luồng xử lý dữ liệu BLE được minh họa tại Hình 2.19. Sau khi thu nhận dữ liệu cảm biến, thiết bị kiểm tra trạng thái kết nối BLE. Nếu kết nối hợp lệ, nó sẽ tiếp tục lắng nghe hành động đọc từ phía thiết bị trung tâm. Dữ liệu sau đó được làm tròn đến hai chữ số thập phân và mã hóa thành ba byte: byte đầu tiên lưu dấu, byte thứ hai chứa hai chữ số đầu, và byte cuối là hai chữ số cuối của giá trị giá tốc. Chuỗi dữ liệu này được gửi qua BLE theo đặc tính đã định nghĩa trước đó.

## 2.2.2. Hiệu chuẩn cảm biến

Việc thu nhận và tiền xử lý dữ liệu là bước quan trọng trong các hệ đo lường. Mặc dù cảm biến thường được hiệu chuẩn từ nhà sản xuất, nhưng vẫn cần được hiệu chuẩn lại trong môi trường đo thực tế để cải thiện hiệu năng và giảm thiểu sai số. Các sai số này được chia thành hai loại chính: (i) sai số hệ thống (mặc định) và (ii) sai số ngẫu nhiên.

**Hiệu chuẩn sai số hệ thống.** Tác giả sử dụng gia tốc trọng trường để hiệu chuẩn cảm biến theo hướng tĩnh. Khi xoay cảm biến sao cho một trục hướng lên vuông góc với mặt phẳng nằm ngang, giá trị đo được là  $-1g$ ; khi hướng xuống dưới, giá trị là  $+1g$ . Bằng cách xoay cảm biến lần lượt qua sáu vị trí tĩnh tương ứng với các hướng trục chính, có thể xác định được các điểm chuẩn, từ đó nội suy để xác định giá trị  $0g$  một cách chính xác và đáng tin cậy.

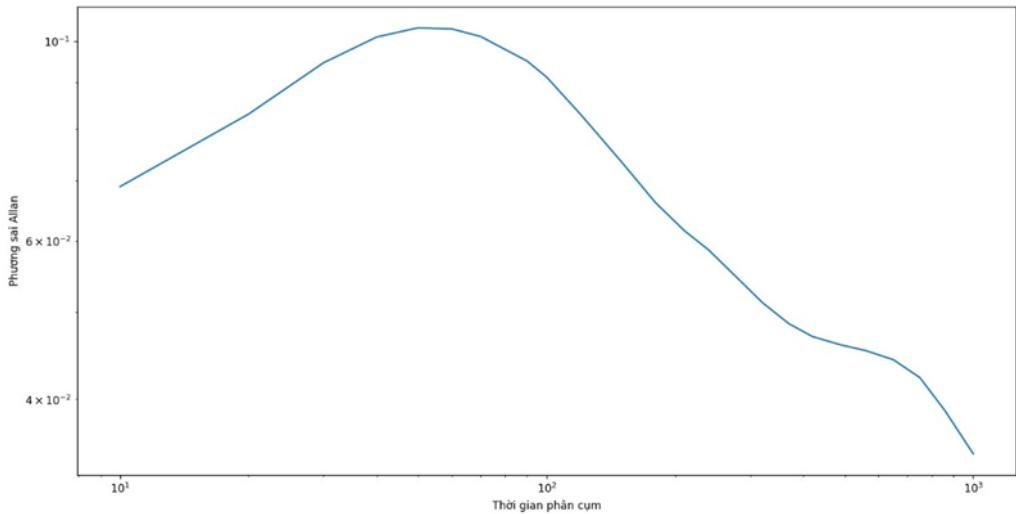


Hình 2.20: Minh họa kết quả phân tích đường cong Allan

**Phân tích sai số ngẫu nhiên.** Tác giả sử dụng phương sai Allan để phân tích các thành phần nhiễu trong dữ liệu cảm biến [107]. Đây là phương pháp phân tích miền thời gian phổ biến nhằm đánh giá độ ổn định tần số và định lượng các loại nhiễu khác nhau như nhiễu trắng, trôi ngẫu nhiên, và nhiễu lượng tử. Biểu đồ Allan log-log cho phép nhận diện các thành phần nhiễu thông qua độ dốc của từng đoạn đường cong.

Trong thử nghiệm, cảm biến được đặt cố định trong phòng ở điều kiện nhiệt độ ổn định, với tần số lấy mẫu 10 Hz, thu được tổng cộng 1.211.210 mẫu. Kết quả biểu diễn trong Hình 2.23 cho thấy nhiễu chiếm ưu thế là nhiễu lượng tử (quantization noise), đặc

trưng bởi hệ số góc tương ứng trong đồ thị.



Hình 2.21: Biểu đồ phương sai Allan của trục X

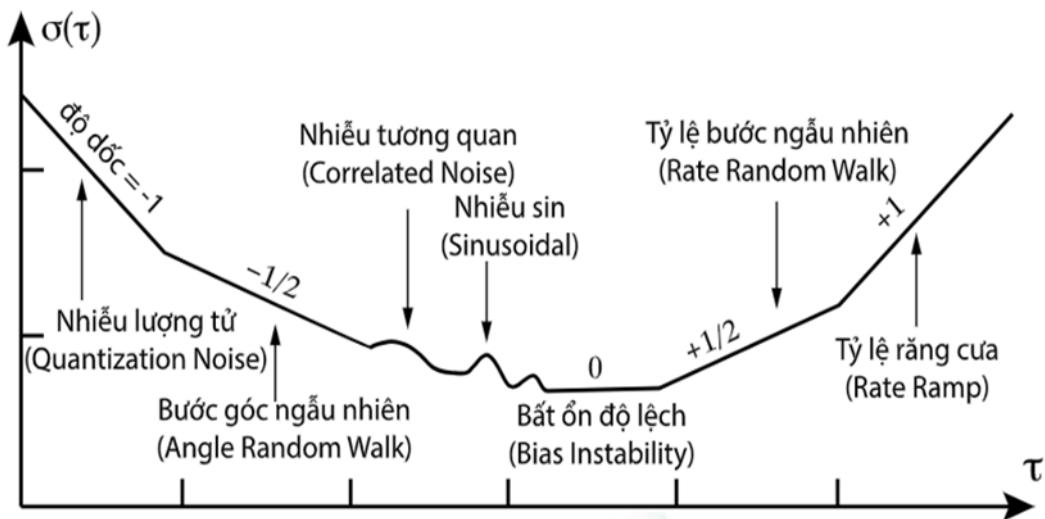
### 2.2.3. Hiệu chuẩn cảm biến

Trong bất kỳ hệ thống đo lường nào, chất lượng dữ liệu đầu vào quyết định trực tiếp đến hiệu năng xử lý và độ tin cậy của kết quả đầu ra. Các mô hình học máy, đặc biệt trong bối cảnh nhúng với tài nguyên hạn chế, rất nhạy cảm với sai số và nhiễu. Do đó, hiệu chuẩn cảm biến không chỉ là bước kỹ thuật đơn thuần mà còn mang ý nghĩa khoa học quan trọng: nó đảm bảo rằng dữ liệu thu thập được phản ánh đúng hiện tượng vật lý, hạn chế tối đa sai lệch hệ thống và định lượng rõ ràng các yếu tố ngẫu nhiên, qua đó nâng cao khả năng tổng quát hóa của mô hình.

Mặc dù các cảm biến thương mại thường được nhà sản xuất hiệu chuẩn ở mức cơ bản, nhưng sự khác biệt về môi trường, cách gắn lắp, và độ trôi theo thời gian vẫn tạo ra những sai lệch đáng kể. Trong nghiên cứu này, cảm biến được hiệu chuẩn theo hai hướng: (i) loại bỏ sai số hệ thống cố hữu, và (ii) phân tích, định lượng các thành phần nhiễu ngẫu nhiên.

**Hiệu chuẩn sai số hệ thống.** Sai số hệ thống bao gồm độ lệch offset, sai số độ nhạy và sai số giao thoa trực (cross-axis). Để hiệu chỉnh, tác giả sử dụng giá tốc trọng trường như một tín hiệu chuẩn có giá trị tuyệt đối ổn định. Khi đặt cảm biến ở trạng thái

tĩnh và xoay lần lượt để các trục X, Y, Z hướng lên hoặc xuống vuông góc với mặt phẳng ngang, giá trị lý tưởng phải là  $\pm 1g$ . Bằng cách đo tại sáu vị trí này, có thể xác định được đường cong đặc trưng của từng trục, từ đó hiệu chỉnh offset (xác định 0g chính xác) và scale factor (tuyến tính hóa độ nhạy). Cách tiếp cận này đảm bảo rằng dữ liệu đầu ra từ mỗi trục gia tốc bám sát hiện tượng vật lý, hạn chế tối đa sự sai lệch tích lũy khi tích hợp dữ liệu trong các thuật toán phân loại.

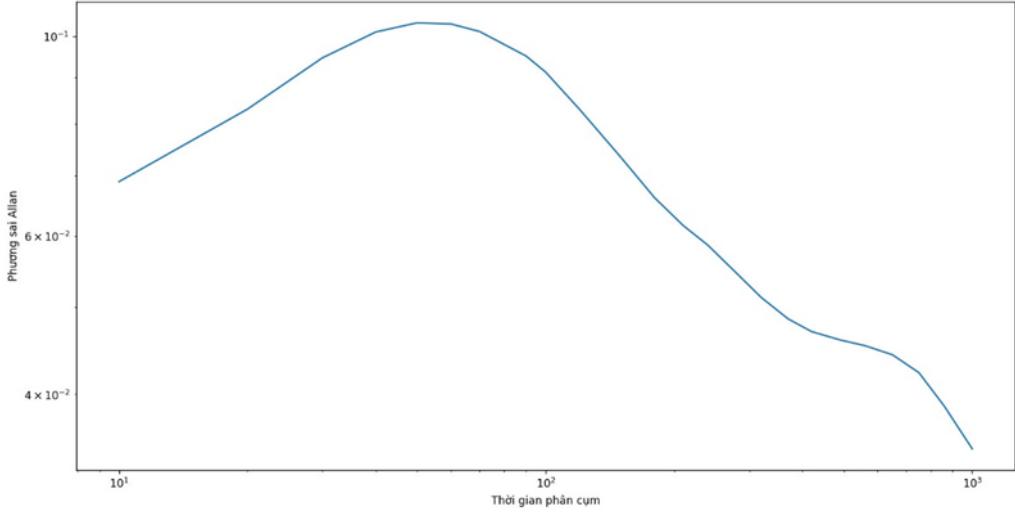


Hình 2.22: Minh họa kết quả phân tích đường cong Allan

**Phân tích sai số ngẫu nhiên.** Bên cạnh sai số hệ thống, cảm biến còn chịu ảnh hưởng của nhiễu ngẫu nhiên mà không thể loại bỏ bằng hiệu chuẩn tĩnh. Để đánh giá các thành phần này, tác giả sử dụng phương pháp phương sai Allan [107], một công cụ kinh điển trong phân tích ổn định tín hiệu và được ứng dụng rộng rãi trong lĩnh vực quan tính học. Phương pháp này cho phép phân tách và định lượng các loại nhiễu như nhiễu trắng (white noise), trôi ngẫu nhiên (random walk), và nhiễu lượng tử (quantization noise), dựa trên độ dốc đặc trưng ở từng miền thời gian trong biểu đồ log-log.

Trong thí nghiệm, cảm biến được cố định trong môi trường nhiệt độ ổn định, hoạt động ở tần số lấy mẫu 10 Hz và dải đo  $\pm 2g$ , thu được tổng cộng 1.211.210 mẫu liên tục. Kết quả biểu diễn trong Hình 2.23 cho thấy thành phần nhiễu trôi là nhiễu lượng tử, thể hiện bằng độ dốc  $-1$  trong miền thời gian ngắn của đồ thị Allan. Phát hiện này có ý nghĩa thực tiễn: do dữ liệu chủ yếu chịu ảnh hưởng bởi nhiễu lượng tử, các phương pháp lọc tuyến tính đơn giản (như trung bình trượt hoặc bộ lọc thông thấp bậc thấp) là đủ để cải thiện chất lượng tín hiệu, thay vì phải áp dụng các kỹ thuật lọc phức tạp vốn tiêu tốn

tài nguyên xử lý.



Hình 2.23: Biểu đồ phương sai Allan của trục X

Việc hiệu chuẩn cảm biến theo hai bước nêu trên giúp hệ thống đạt được hai mục tiêu quan trọng: (i) loại bỏ sai số hệ thống để dữ liệu phản ánh trung thực hiện tượng vật lý, và (ii) định lượng đặc tính nhiễu nhằm lựa chọn phương pháp lọc tối ưu, cân bằng giữa chất lượng tín hiệu và tiêu thụ năng lượng. Quan trọng hơn, quá trình này nâng cao độ tin cậy của toàn bộ nghiên cứu: mô hình học máy được huấn luyện trên dữ liệu chuẩn hóa, ít sai lệch, từ đó tăng khả năng tổng quát khi áp dụng trên các thiết bị đeo thực tế khác nhau. Nhờ vậy, nghiên cứu không chỉ dừng lại ở việc xây dựng một nguyên mẫu, mà còn khẳng định được giá trị khoa học ở mức cao hơn: dữ liệu và mô hình có khả năng tái sử dụng, mở rộng và thích ứng trong nhiều bối cảnh ứng dụng khác nhau.

**Lọc nhiễu bằng bộ lọc Kalman.** Để xử lý nhiễu trong dữ liệu cảm biến, đặc biệt là nhiễu trắng và nhiễu lượng tử vốn chiếm ưu thế theo phân tích Allan, tác giả lựa chọn bộ lọc Kalman [85]. Khác với các bộ lọc số cổ điển như trung bình trượt hoặc bộ lọc thông thấp FIR/IIR chỉ đơn thuần làm trơn tín hiệu, Kalman là một bộ lọc đệ quy dựa trên lý thuyết ước lượng Bayes. Nó có khả năng ước lượng trạng thái tối ưu của hệ thống từ chuỗi quan sát bị nhiễu, đồng thời duy trì khả năng thích ứng theo thời gian mà không cần lưu trữ toàn bộ dữ liệu lịch sử.

**Lý do lựa chọn Kalman.** So với moving average (để gây trễ tín hiệu) hay Butterworth low-pass (cần bậc lọc cao, tốn tài nguyên), Kalman phù hợp hơn cho hệ thống

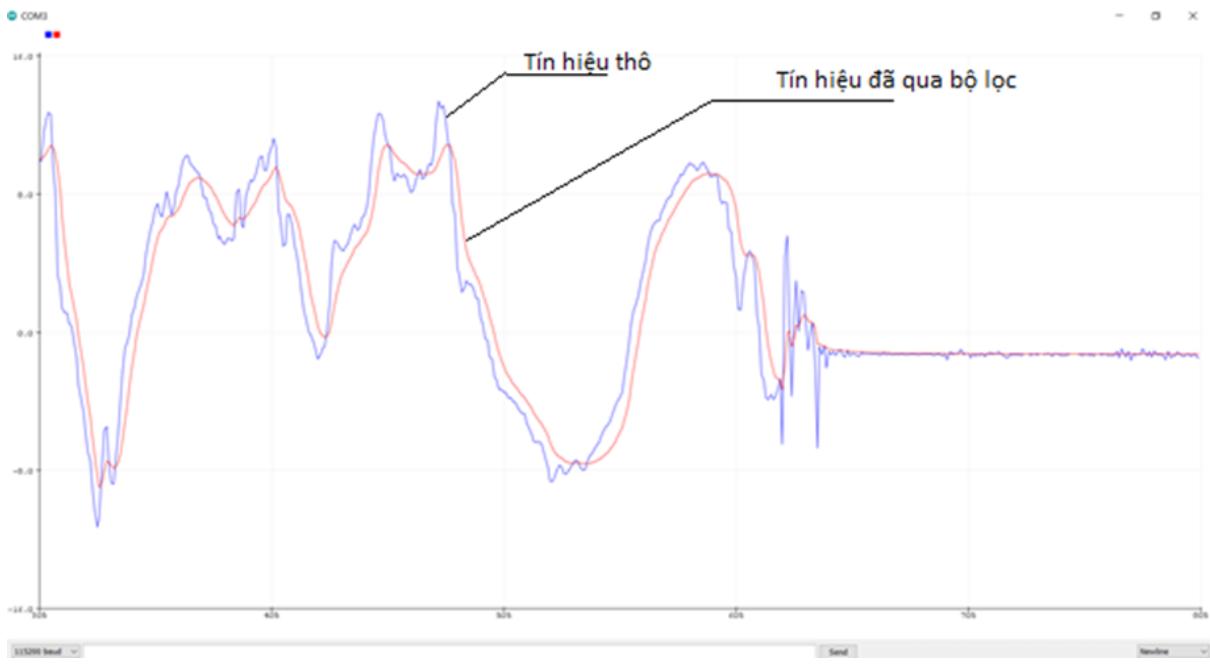
nhúng vì ba lý do: (i) hoạt động theo cơ chế cập nhật đê quy, tiết kiệm bộ nhớ và chi phí tính toán; (ii) cho phép cân bằng độ tin cậy của mô hình và tín hiệu đo; (iii) hiệu quả đặc biệt với dữ liệu nhiễu trắng và nhiễu lượng tử, vốn đã được xác định chiếm ưu thế. Điều này giúp tín hiệu sau lọc vừa mượt, vừa bảo toàn các đặc trưng quan trọng để phân loại tư thế ngủ.

**Lựa chọn tham số.** Trong bộ lọc Kalman, ma trận hiệp phương sai quá trình ( $Q$ ) và đo lường ( $R$ ) được xác định từ thực nghiệm. Với dữ liệu cảm biến ở tần số lấy mẫu 10 Hz,  $R$  được ước lượng dựa trên phương sai nhiễu trắng từ phân tích Allan, trong khi  $Q$  được chọn nhỏ hơn nhằm giả định trạng thái hệ thống thay đổi chậm theo thời gian (phù hợp với chuyển động cơ thể khi ngủ). Ma trận  $P$  ban đầu được đặt gần bằng  $R$  để phản ánh độ tin cậy trung bình ở giai đoạn khởi tạo, sau đó được cập nhật động theo từng bước lặp.

**Kết quả thực nghiệm.** Sau khi áp dụng bộ lọc Kalman trực tiếp tại vi điều khiển, tín hiệu đầu ra trở nên mượt và ổn định hơn, loại bỏ phần lớn dao động tức thời do nhiễu mà không gây trễ đáng kể. Hình 2.24 minh họa dữ liệu trực X trước và sau lọc, cho thấy biên độ dao động được làm giảm trong khi các biến thiên chính của tư thế vẫn được giữ nguyên. Kết quả này chứng tỏ Kalman không chỉ cải thiện chất lượng dữ liệu mà còn giúp duy trì đặc tính động học quan trọng, đảm bảo độ chính xác cho giai đoạn phân loại mô hình TinyML.

#### 2.2.4. Xây dựng phần mềm ứng dụng

Phần mềm ứng dụng 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 pipeline 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 framework học máy phổ biến. So với các hệ thống Home Sleep Testing (HST) thương mại thường đóng kín và khó tùy biến, ứng dụng của chúng tôi đặt trọng tâm vào tính mở, khả năng cấu hình, và khả năng mở rộng để đáp ứng yêu cầu của cả nghiên cứu học thuật lẫn triển khai lâm sàng. Trong giai đoạn đầu, kiến trúc phân tầng được xây dựng nhằm tách biệt rõ chức năng của từng thành phần:



Hình 2.24: Kết quả bộ lọc Kalman cho dữ liệu trục X của cảm biến gia tốc

**Thiết bị nhúng (device):** cảm biến gia tốc trên kit thu nhận tín hiệu thô, được vi xử lý xử lý sơ bộ và truyền qua giao thức *Bluetooth Low Energy (BLE)*.

**Kết nối BLE - Ứng dụng di động:** ứng dụng trên điện thoại thông minh thiết lập liên kết BLE để nhận dữ liệu theo thời gian thực. Bên cạnh việc hiển thị trực quan tín hiệu trên màn hình, ứng dụng còn cho phép người dùng cấu hình tham số thiết bị (tần số lấy mẫu, thời lượng ghi, định dạng dữ liệu).

**Ứng dụng - Máy chủ (server):** dữ liệu cảm biến được đồng bộ về server qua kết nối Internet. Server lưu trữ dữ liệu trong cơ sở dữ liệu quan hệ và cung cấp giao diện quản lý thí nghiệm, gắn nhãn dữ liệu và phân quyền truy cập.

**Xuất dữ liệu - Định dạng CSV:** để phục vụ cho giai đoạn phân tích và huấn luyện, hệ thống cho phép xuất dữ liệu dưới dạng chuẩn CSV. Dữ liệu này bao gồm chuỗi tín hiệu thô, nhãn tư thế, và siêu dữ liệu (thời gian, ID thiết bị, thông số cấu hình).

**Huấn luyện mô hình học máy:** dữ liệu CSV sau khi được làm sạch và tiền xử lý được đưa vào pipeline huấn luyện mô hình (Python/Scikit-learn/TensorFlow). Các mô hình được đánh giá ngoại tuyến (offline) trước khi lựa chọn phương án tối ưu để triển khai trên thiết bị nhúng.

Cách tiếp cận phân tầng này đảm bảo rằng dữ liệu được thu thập từ môi trường

thực tế được lưu trữ một cách nhất quán, có thể tái sử dụng và dễ dàng tích hợp vào các framework học máy phổ biến. Đồng thời, nó tạo điều kiện cho việc mở rộng hệ thống: sau giai đoạn huấn luyện, mô hình tối ưu có thể được đưa ngược trở lại ứng dụng để kiểm chứng trong thời gian thực trên thiết bị.

### Các công nghệ và thành phần sử dụng được tóm tắt như sau:

Ứng dụng được xây dựng bằng ngôn ngữ lập trình Dart và phát triển trên nền tảng Flutter, cho phép triển khai hiệu quả trên hệ điều hành Android. Về mặt giao tiếp phần cứng, ứng dụng sử dụng chuẩn *Bluetooth Low Energy (BLE)* nhằm tối ưu hóa tiêu thụ năng lượng và bảo đảm kết nối ổn định giữa thiết bị và điện thoại di động. Các chức năng chính của ứng dụng bao gồm: thiết lập kết nối với thiết bị đeo, thu nhận dữ liệu cảm biến, hiển thị tín hiệu theo thời gian thực, lưu trữ dữ liệu có cấu trúc, đồng thời hỗ trợ cá nhân hóa trải nghiệm người dùng thông qua các tùy chọn cấu hình và quản lý hồ sơ cá nhân.

Ứng dụng được thiết kế xoay quanh ba nhóm chức năng chính như minh họa trong Bảng 2.1. Trước hết, nhóm **bảo mật** tập trung vào các thao tác đăng nhập, xác thực và khôi phục tài khoản, nhằm đảm bảo an toàn dữ liệu và bảo vệ quyền riêng tư của người dùng. Tiếp theo, nhóm **chức năng chung** bao gồm các tính năng cốt lõi như kết nối với thiết bị phần cứng, thu thập tín hiệu từ cảm biến và hiển thị dữ liệu theo thời gian thực, qua đó tạo nên cầu nối trực tiếp giữa người dùng và hệ thống nhúng. Cuối cùng, nhóm **cá nhân hoá** cho phép theo dõi các chỉ số sức khỏe, khai báo bảng hỏi STOP-BANG, cũng như lưu trữ hồ sơ người dùng, góp phần xây dựng nền tảng dữ liệu cá nhân phục vụ phân tích lâm sàng và huấn luyện mô hình học máy trong tương lai.

## Kiến trúc phần mềm

Ứng dụng sử dụng mô hình **BLoC (Business Logic Component)** để tách biệt giao diện người dùng và logic xử lý. BLoC hoạt động dựa trên nguyên tắc nhận sự kiện đầu vào và trả về trạng thái phù hợp, giúp quản lý luồng dữ liệu hiệu quả. Cấu trúc tổng thể của kiến trúc BLoC gồm ba lớp chính được mô tả trong Hình 2.25.

Sau khi kết nối BLE được thiết lập thành công, ứng dụng truy xuất đối tượng đặc tính cảm biến (characteristic instance) và liên tục gửi yêu cầu đọc (read) đến vi điều khiển. Thiết bị phản hồi bằng cách trả về dữ liệu cảm biến dưới dạng mảng UInt8. Các

Bảng 2.1: Các nhóm chức năng chính của ứng dụng

Nhóm	Chức năng	Mô tả
Bảo mật	Đăng ký	Người dùng đăng ký thông tin và được gửi xác thực qua email
	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
Chức năng chung	Kết nối Bluetooth	Kết nối ứng dụng với phần cứng thông qua BLE
	Chọn cảm biến	Lựa chọn cảm biến như (giá tốc, âm thanh, ...)
	Câu hỏi STOPBANG	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	Dữ liệu thời gian thực thông qua màn hình giao diện và có đánh giá về tư thế ngủ
	Chatbot y tế	Chatbot xây dựng trên bộ câu hỏi 2000 câu về y học giấc ngủ và công nghệ RAG
Cá nhân hóa	Đổi thông tin cá nhân, mật khẩu	Sửa đổi thông tin cá nhân trên ứng dụng
	Thêm người dùng phụ	Thêm người dùng phụ ví dụ trong cùng 1 gia đình
	Đăng xuất	Thoát khỏi ứng dụng

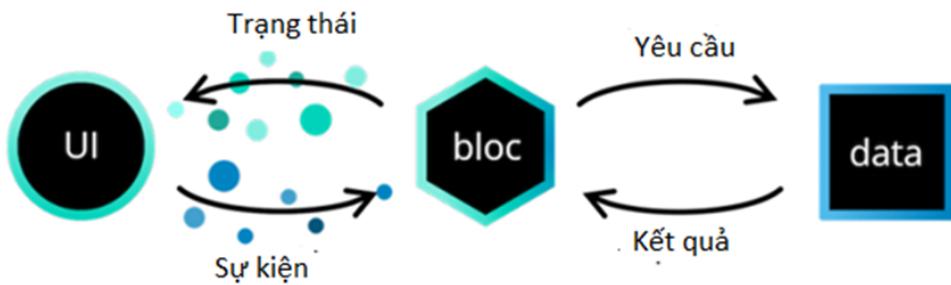
---

```
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 2.3: Tập lệnh để tìm kiếm dịch vụ cảm biến



Hình 2.25: Cấu trúc kiến trúc BLoC trong ứng dụng Flutter

---

```

1  {
2      "value": "0.88%0.66%0.99@2022-01-01/0.88%0.66%0.99@2022-01-01/0.88%0.66%0.99@2022-01-01"
3      "customer": "62a5f5672ad9c724ef117d76"
4  }

```

---

Mã nguồn 2.4: "Cấu trúc dữ liệu của phần nội dung đẩy lên máy chủ"

giá trị này được ứng dụng giải mã, chuyển đổi sang dạng số thực tương ứng với giá tốc trên ba trục (X, Y, Z), và gắn nhãn thời gian thực.

Quá trình xử lý này được thực hiện trong một vòng lặp có kiểm soát độ trễ ngắn nhằm đảm bảo khả năng cập nhật liên tục nhưng vẫn tối ưu hiệu suất hệ thống.

Mã 2.3 minh họa toàn bộ quy trình xử lý: từ kết nối BLE, truy xuất đặc tính giá tốc, đọc giá trị nhị phân từ thiết bị, đến việc chuẩn hóa và gửi dữ liệu lên backend. Trong đoạn mã này, dữ liệu dạng Uint8List nhận từ cảm biến được tách và chuyển đổi thành ba thành phần tương ứng với ba trục giá tốc. Dữ liệu sau khi được xử lý sẽ được đóng gói theo định dạng JSON và gửi đến máy chủ thông qua phương thức POST, sử dụng thư viện http trong Flutter.

Định dạng dữ liệu BLE được chuẩn hóa như trong Mã 2.4, với trường "value" là chuỗi liên tục các giá trị cảm biến (phân tách bằng ký tự đặc biệt) và trường "customer" để định danh người dùng.

Việc tối ưu hóa cả quá trình đọc BLE và đẩy dữ liệu HTTP theo lô như vậy giúp giảm độ trễ, tránh tình trạng nghẽn băng thông, đồng thời vẫn đảm bảo độ chính xác và toàn vẹn của dữ liệu cảm biến.

Ngoài các chức năng thu thập và truyền dữ liệu cảm biến, ứng dụng còn tích hợp

các công cụ hỗ trợ đánh giá y học lâm sàng ban đầu nhằm phục vụ cho việc sàng lọc và phân loại nguy cơ mắc hội chứng ngưng thở khi ngủ (OSA). Trong đó, ba thành phần quan trọng được triển khai bao gồm:

**01) Bộ câu hỏi STOP-BANG:** Đây là một bảng sàng lọc lâm sàng được sử dụng phổ biến trong y học giấc ngủ để đánh giá nguy cơ mắc OSA. Dữ liệu từ bảng này được lưu trữ cùng với dữ liệu cảm biến và đóng vai trò như đầu vào bổ sung cho các mô hình học máy dự đoán chỉ số AHI (Apnea-Hypopnea Index).

**02) Thang điểm Epworth Sleepiness Scale (ESS):** Tác giả triển khai thêm bảng câu hỏi ESS nhằm đánh giá mức độ buồn ngủ ban ngày của người dùng. Thang điểm này giúp phát hiện tình trạng buồn ngủ quá mức và có thể hỗ trợ phân tầng nguy cơ trong mô hình phân loại rối loạn giấc ngủ.

**03) Đánh giá BMI (Body Mass Index):** BMI được tự động tính toán dựa trên chiều cao và cân nặng người dùng nhập vào. Chỉ số này đóng vai trò là một trong các yếu tố nguy cơ chính trong chẩn đoán OSA, đặc biệt khi kết hợp cùng STOP-BANG.

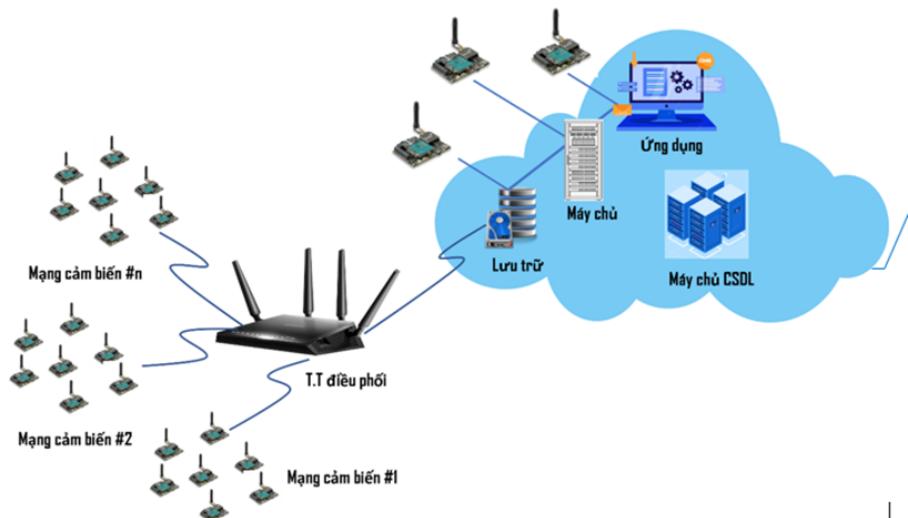
Ngoài ra, nhằm cải thiện trải nghiệm người dùng và hỗ trợ trả lời câu hỏi liên quan đến giấc ngủ, tác giả phát triển thêm tính năng **chatbot y học giấc ngủ** dựa trên kỹ thuật **Retrieval-Augmented Generation (RAG)**. Chatbot này được xây dựng từ cơ sở dữ liệu gồm hơn 2000 câu hỏi và câu trả lời chuyên 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.

Hệ thống RAG kết hợp khả năng truy vấn ngữ nghĩa từ tập văn bản lớn (document retrieval) và khả năng sinh văn bản linh hoạt từ mô hình ngôn ngữ lớn (LLM), từ đó cung cấp các câu trả lời chính xác, có căn cứ và dễ hiểu cho người dùng không chuyên.

**Tính năng quản lý người dùng** cũng được mở rộng. Người dùng có thể tạo tài khoản một lần và sử dụng lại trong các lần đăng nhập sau. Cơ chế này giúp rút ngắn thao tác, đồng thời vẫn đảm bảo tính bảo mật và khả năng khôi phục dữ liệu khi quên tài khoản hoặc mất khẩu. Dữ liệu người dùng (câu hỏi, chỉ số BMI, lịch sử cảm biến) được liên kết thống nhất qua một ID định danh duy nhất, hỗ trợ tốt cho việc phân tích, theo dõi tiến triển và huấn luyện mô hình học máy cá nhân hóa trong tương lai.

Có thể thấy rằng, việc đáp ứng đầy đủ các **yêu cầu chức năng** mới chỉ đảm bảo hệ thống thực hiện được các nhiệm vụ cốt lõi đã đề ra, chẳng hạn như thu thập dữ liệu, hiển thị tín hiệu hay quản lý hồ sơ người dùng. Tuy nhiên, trong bối cảnh triển khai thực tiễn, điều này là chưa đủ. Một hệ thống có thể hoạt động đúng chức năng nhưng sẽ nhanh chóng trở nên khó sử dụng hoặc thậm chí thất bại nếu thiếu đi các **yêu cầu phi chức năng** quan trọng. Những yêu cầu này bao gồm độ tin cậy khi vận hành liên tục trong môi trường thực tế, tính bảo mật nhằm bảo vệ dữ liệu y tế nhạy cảm, khả năng mở rộng để thích ứng với số lượng người dùng tăng dần, cũng như hiệu năng xử lý để đảm bảo dữ liệu được truyền tải theo thời gian thực. Do đó, chỉ khi cả hai nhóm yêu cầu được quan tâm song song và được tích hợp hài hòa ngay từ giai đoạn thiết kế, hệ thống mới vừa đáp ứng đúng mục tiêu nghiên cứu, vừa có tính khả thi và bền vững khi ứng dụng trong y tế lâm sàng và cộng đồng.

### 2.2.5. Thiết kế và xây dựng hệ thống lưu trữ



Hình 2.26: Mô hình tích hợp giữa mạng cảm biến và cấu trúc dữ liệu đám mây

Trong hệ thống đề xuất, dữ liệu cảm biến giữ vai trò trung tâm trong việc huấn luyện và triển khai các mô hình trí tuệ nhân tạo (AI). Do giới hạn bộ nhớ của vi điều khiển và thiết bị đầu cuối, giải pháp lưu trữ dữ liệu trên nền tảng đám mây trở thành lựa chọn tất yếu. Không chỉ giúp vượt qua hạn chế về dung lượng, cloud còn loại bỏ rào cản về địa lý, cho phép chia sẻ và phân tích dữ liệu từ bất kỳ đâu có kết nối Internet. Để bảo đảm hiệu quả, kiến trúc lưu trữ được phân chia thành ba khối chức năng chính: **máy chủ**

**xử lý, tầng bộ nhớ đệm (cache), và cơ sở dữ liệu lưu trữ.**

**(i) Máy chủ xử lý.** Máy chủ được triển khai trên nền tảng Node.js và vận hành trong hạ tầng Amazon Web Services (AWS), thể hiện một sự lựa chọn chiến lược trong thiết kế kiến trúc hệ thống. Trong bối cảnh các ứng dụng IoT y tế tạo ra luồng dữ liệu liên tục và khối lượng lớn, việc sử dụng Node.js với mô hình bất đồng bộ (asynchronous, event-driven) cho phép máy chủ xử lý đồng thời nhiều kết nối mà không cần tiêu tốn tài nguyên theo kiểu đa luồng truyền thống. Điều này đặc biệt quan trọng khi hệ thống cần thu nhận dữ liệu từ nhiều thiết bị đeo cùng lúc trong các nghiên cứu lâm sàng quy mô lớn, vốn tiềm ẩn nguy cơ quá tải nếu không có cơ chế tối ưu.

Một nguyên tắc cốt lõi của thiết kế máy chủ là *batch processing*. Thay vì gửi từng gói dữ liệu riêng lẻ, hệ thống tích luỹ các gói tin thành lô 1000 mẫu trước khi truyền tới backend. Cách tiếp cận này không chỉ giảm số lượng yêu cầu HTTP, mà còn tối ưu băng thông, giảm độ trễ mạng và hạ thấp chi phí vận hành trên cloud. Từ góc độ khoa học dữ liệu, xử lý theo lô cũng giúp bảo toàn tính toàn vẹn của chuỗi thời gian, cho phép phân tích tín hiệu gia tốc ( $x$ ,  $y$ ,  $z$ ) đi kèm mốc thời gian (timestamp) một cách nhất quán. Đây là yếu tố then chốt để đảm bảo tính tái lập (reproducibility) của các thí nghiệm và độ tin cậy của mô hình học máy huấn luyện từ dữ liệu đó.

Máy chủ còn được trang bị cơ chế *retry logic* để tăng tính bền bỉ (resilience) của hệ thống trước những sự cố mạng thường gặp trong môi trường thực tế. Thay vì chấp nhận mất gói dữ liệu, hệ thống sẽ thử gửi lại nhiều lần; chỉ khi thất bại liên tục quá 10 lần mới thông báo lỗi và tạm dừng tiến trình lưu trữ. Cơ chế này phản ánh nguyên tắc thiết kế quan trọng trong các hệ thống y tế: “*fail-safe and fault-tolerant*”, tức là ngay cả khi gặp lỗi, hệ thống vẫn đảm bảo không gây ra sai lệch dữ liệu, đồng thời cung cấp tín hiệu rõ ràng cho người vận hành.

Tóm lại, tầng máy chủ không chỉ đơn thuần là nơi tiếp nhận dữ liệu từ thiết bị, mà còn là lớp điều phối trung tâm, kết hợp giữa hiệu quả tính toán (nhờ kiến trúc bất đồng bộ của Node.js), khả năng mở rộng linh hoạt (nhờ hạ tầng AWS), và các cơ chế bảo đảm tính toàn vẹn dữ liệu (batch processing và retry logic). Thiết kế này cho thấy một hướng tiếp cận hệ thống phân tán trong lĩnh vực IoT y tế, nơi yêu cầu vừa đáp ứng tải dữ liệu cao, vừa bảo đảm chất lượng khoa học và tính khả thi triển khai trong môi trường lâm sàng.

**(ii) Bộ nhớ đệm (cache).** Trong kiến trúc hệ thống IoT y tế, tầng cache giữ vai trò

chiến lược như một lớp trung gian giữa ứng dụng và cơ sở dữ liệu, không chỉ để lưu tạm dữ liệu khi mất kết nối mạng, mà còn để tối ưu hiệu năng và giảm áp lực cho backend. Nếu ứng dụng gửi trực tiếp từng gói dữ liệu lên cơ sở dữ liệu, hệ thống sẽ phải xử lý một lượng lớn yêu cầu nhỏ lẻ, gây nghẽn băng thông và tăng độ trễ phản hồi. Đặc biệt với dữ liệu cảm biến liên tục theo thời gian thực, chiến lược này dễ dẫn đến tình trạng “request storm” khiến máy chủ và cơ sở dữ liệu bị quá tải. Việc bổ sung một lớp cache giúp gom dữ liệu thành lô, tối ưu băng thông truyền tải và đồng thời cung cấp cơ chế *write-back* tới cơ sở dữ liệu khi mạng ổn định.

Ý nghĩa của tầng cache không chỉ dừng ở việc lưu trữ tạm thời. Nó còn là thành phần bảo đảm **tính liên tục dịch vụ (service continuity)**: người dùng vẫn có thể quan sát dữ liệu cảm biến theo thời gian thực ngay cả khi mất mạng, và hệ thống sẽ tự động đồng bộ dữ liệu sau khi kết nối được khôi phục. Điều này đặc biệt quan trọng trong nghiên cứu y học giấc ngủ tại nhà, nơi mà gián đoạn kết nối Internet là tình huống phổ biến. Từ góc nhìn lâm sàng, sự hiện diện của cache giúp giảm thiểu nguy cơ mất dữ liệu, một yếu tố có thể ảnh hưởng trực tiếp tới độ tin cậy của mô hình học máy huấn luyện trên dữ liệu đó.

Trong giai đoạn mở rộng, tầng cache có thể triển khai bằng các hệ thống bộ nhớ đệm phân tán như Redis hoặc Memcached, hỗ trợ hàng nghìn truy vấn đồng thời và cho phép xử lý dữ liệu theo cơ chế *publish/subscribe*. Ở mức độ cao hơn, tầng cache có thể kết hợp với hàng đợi thông điệp (*message queue*) như RabbitMQ hoặc Kafka, nhằm điều phối luồng dữ liệu giữa ứng dụng và backend. Giải pháp này đặc biệt hữu ích khi hệ thống mở rộng tới hàng trăm hoặc hàng nghìn thiết bị đeo cùng hoạt động, giúp cân bằng tải, tránh tình trạng *data race*, và tăng tính ổn định tổng thể.

Tóm lại, tầng cache không chỉ là một cơ chế tối ưu hiệu năng, mà còn là **cấu phần bảo đảm tính tin cậy và khả năng mở rộng** của toàn bộ hệ thống. Nếu coi máy chủ là trung tâm điều phối và cơ sở dữ liệu là nền tảng lưu trữ, thì tầng cache chính là “vùng đệm an toàn” giúp hệ thống hấp thụ sự biến động của môi trường mạng và duy trì hoạt động ổn định trong mọi kịch bản triển khai.

**(iii) Cơ sở dữ liệu lưu trữ.** Trong các hệ thống IoT y tế, đặc biệt là khi dữ liệu cảm biến và dữ liệu lâm sàng cùng tồn tại, không một mô hình cơ sở dữ liệu đơn lẻ nào có thể đáp ứng tối ưu tất cả yêu cầu. Do đó, kiến trúc lưu trữ được thiết kế theo hướng **lai (hybrid storage architecture)**, kết hợp cả NoSQL (MongoDB Atlas) và SQL (MySQL),

nhằm tận dụng thế mạnh riêng của từng loại hình.

Với dữ liệu cảm biến thời gian thực, **MongoDB Atlas** thể hiện ưu thế vượt trội nhờ khả năng lưu trữ phi cấu trúc, phân tán trên nhiều cụm máy chủ và hỗ trợ mở rộng theo chiều ngang (*horizontal scaling*). Đặc thù của dữ liệu gia tốc là liên tục, khối lượng lớn, và thay đổi nhanh, do đó cơ chế đánh chỉ mục theo timestamp cùng với tính năng TTL (Time-To-Live index) của MongoDB giúp tối ưu hóa truy vấn, giảm dung lượng lưu trữ. Điều này mang lại ý nghĩa khoa học quan trọng: cho phép xử lý tín hiệu theo chuỗi thời gian mà không bị nhiễu bởi dữ liệu cũ, đồng thời bảo đảm tính nhất quán khi huấn luyện mô hình học máy.

Trong khi đó, **MySQL** được sử dụng để quản lý các dữ liệu có cấu trúc ổn định và đòi hỏi tính toàn vẹn quan hệ, chẳng hạn như thông tin định danh người dùng, mật khẩu mã hoá, kết quả bảng điểm STOP-BANG, thang điểm Epworth, chỉ số BMI, và tiền sử bệnh nền. Việc sử dụng hệ quản trị quan hệ cho phép đảm bảo **tính ràng buộc (constraints)** giữa các bảng và duy trì độ chính xác dữ liệu trong các thao tác xác thực, phân quyền và theo dõi tiến triển y khoa. Từ góc nhìn triển khai lâm sàng, đây là điều kiện tiên quyết để hệ thống đáp ứng được các quy chuẩn bảo mật dữ liệu y tế (ví dụ HIPAA hoặc các tiêu chuẩn tương tự).

Do đó, kiến trúc lai chính là sự cân bằng giữa **tính linh hoạt** và **tính toàn vẹn**, vừa bảo đảm hiệu năng, vừa bảo đảm tính tin cậy khi ứng dụng trong môi trường y tế.

**Mở rộng về sau**, khi khối lượng dữ liệu ngày càng tăng và đa dạng hoá về nguồn gốc (cảm biến đeo, hồ sơ bệnh án điện tử, dữ liệu hình ảnh y tế), hệ thống có thể được mở rộng thành một **datalake** trên nền tảng cloud. Khác với cơ sở dữ liệu truyền thống chỉ lưu dữ liệu đã chuẩn hóa, datalake cho phép lưu trữ song song dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc, mở ra khả năng phân tích hợp nhất trên toàn bộ phổ dữ liệu. Điều này mang lại giá trị chiến lược cho các nghiên cứu y học giấc ngủ, vì cho phép liên kết dữ liệu tư thế ngủ với dữ liệu sinh học khác (EEG, SpO<sub>2</sub>, nhịp tim) trong các nghiên cứu đa trung tâm.

Song song, việc tích hợp các công cụ phân tích và trực quan hóa như **Microsoft Power BI** giúp chuyển đổi dữ liệu thô thành báo cáo và biểu đồ động, hỗ trợ các bác sĩ và nhà nghiên cứu ra quyết định kịp thời. Power BI không chỉ đơn thuần là công cụ hiển thị, mà còn là một lớp phân tích trung gian, cho phép truy vấn dữ liệu từ cả MySQL, MongoDB và datalake, sau đó biểu diễn các chỉ số quan trọng theo thời gian thực. Đây

chính là cầu nối giữa **dữ liệu lớn (big data)** và **tri thức y khoa**, tạo ra tác động trực tiếp đến chất lượng chẩn đoán và theo dõi bệnh nhân.

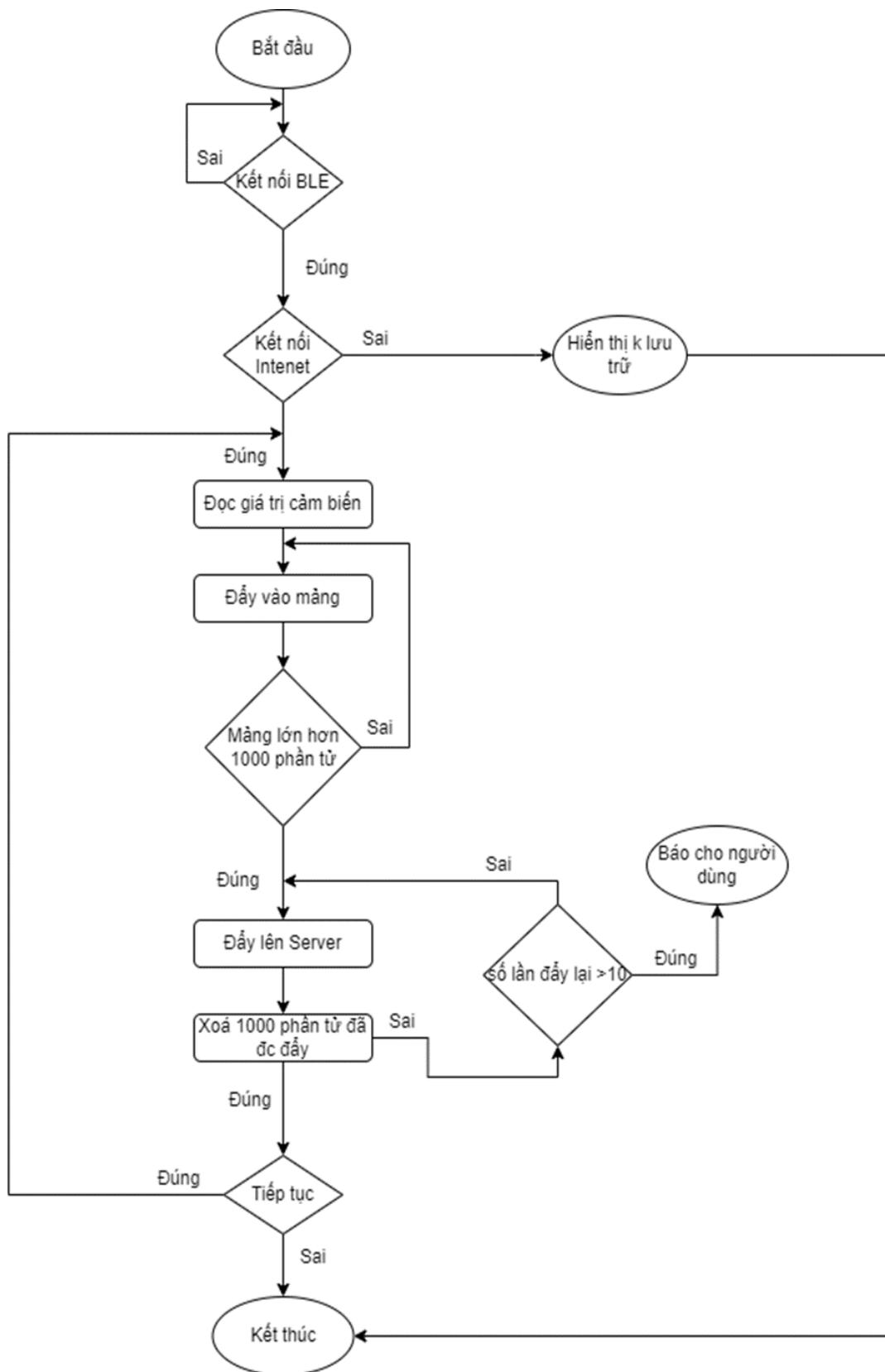
Ở mức độ khái quát hơn, việc lựa chọn kiến trúc lưu trữ lai kết hợp mở rộng sang datalake và trực quan hóa bằng Power BI phản ánh xu hướng thiết kế hệ thống y tế hiện đại: “*data-centric and insight-driven architecture*”. Không chỉ dừng ở việc thu thập và lưu trữ dữ liệu, hệ thống còn tiến tới khai thác tri thức từ dữ liệu để hỗ trợ chẩn đoán, dự báo nguy cơ và nâng cao chất lượng chăm sóc sức khỏe cộng đồng.

Lưu đồ thuật toán lưu trữ dữ liệu được thể hiện trong Hình 2.27, bao gồm hai nhánh xử lý nhằm đảm bảo tính liên tục dịch vụ và độ tin cậy dữ liệu. (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/ứng dụng 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 batch để gửi lên máy chủ. Cơ chế này vừa giảm overhead giao thức, vừa bảo toàn tính liên tục của chuỗi thời gian để phục vụ phân tích tín hiệu. Trong trường hợp thao tác gửi thất bại liên tiếp quá 10 lần, hệ thống chủ động phát cảnh báo và tạm ngừng quá trình lưu trữ nhằm bảo đảm tính nhất quán (consistency) và tránh sai lệch dữ liệu.

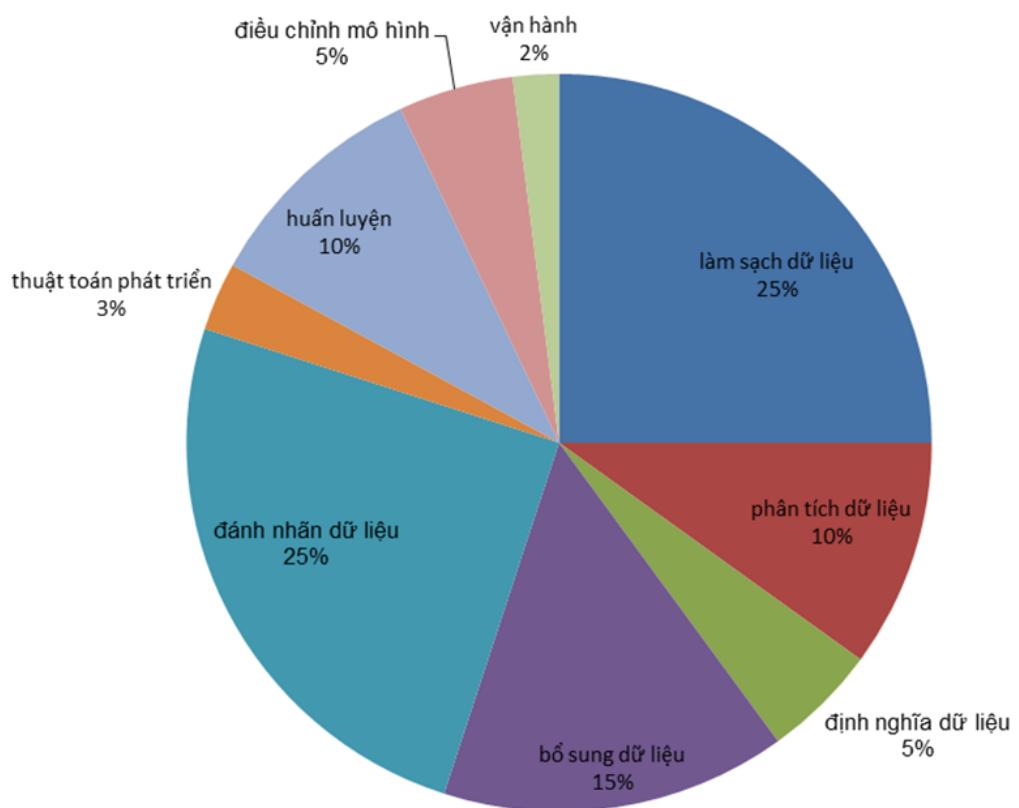
## 2.2.6. Học máy trong phân loại tư thế ngủ

Học máy ngày càng được xem là công cụ hiệu quả trong việc phân loại tư thế ngủ dựa trên dữ liệu cảm biến. Các thuật toán học máy cho phép phát hiện và nhận diện tư thế với độ chính xác cao hơn so với các phương pháp truyền thống. Nhờ đó, chúng tạo nền tảng quan trọng cho việc phát triển các hệ thống giám sát giấc ngủ thông minh, hỗ trợ sàng lọc và chẩn đoán hội chứng ngưng thở khi ngủ.

Hình 2.28 trình bày phân bố thời gian tương đối giữa các công đoạn trong quá trình triển khai một dự án học máy thực tế. Dữ liệu trong biểu đồ cho thấy rằng phần lớn thời gian không nằm ở bước huấn luyện mô hình, mà được dành cho các công việc tiền xử lý dữ liệu - chiếm đến hơn 60% tổng thời gian. Cụ thể, hai hoạt động tốn thời gian nhất là **làm sạch dữ liệu** và **gán nhãn dữ liệu**, mỗi hoạt động chiếm 25% tổng thời lượng thực hiện. Tiếp theo là **bổ sung dữ liệu** (15%) và **phân tích dữ liệu** (10%). Bốn công đoạn này là nền tảng quyết định chất lượng đầu vào, ảnh hưởng trực tiếp đến độ chính xác và



Hình 2.27: Lưu đồ thuật toán lưu trữ dữ liệu cảm biến



Hình 2.28: Phân bổ thời gian thực hiện đối với dự án học máy

khả năng tổng quát hóa của mô hình sau khi huấn luyện.

Trong khi đó, các bước thường được quan tâm trong các tài liệu học thuật như huấn luyện mô hình (10%), phát triển thuật toán (3%) và tinh chỉnh mô hình (5%) lại chiếm tỷ trọng thấp hơn. Giai đoạn vận hành thực tế (deployment) cũng chỉ chiếm khoảng 2%, tuy nhiên vẫn đóng vai trò quan trọng trong việc chuyển giao ứng dụng ra ngoài môi trường thử nghiệm.

Sự phân bố này phản ánh đặc điểm phổ biến trong các dự án học máy với dữ liệu thực tế từ cảm biến: chất lượng mô hình phụ thuộc chủ yếu vào dữ liệu và quy trình xử lý trước huấn luyện. Do đó, việc đầu tư thời gian vào xử lý dữ liệu là hoàn toàn cần thiết và hợp lý.

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. Trong mục này, tác giả đi sâu vào phân tích các thuật toán học máy đã được lựa chọn, lý do lựa chọn, đặc điểm cấu trúc của từng mô hình, cũng như hiệu quả của chúng trong bối cảnh bài toán sử dụng dữ liệu cảm biến gia tốc ba trục.

**Hồi quy Logistic (Logistic Regression - LR)** là một trong những thuật toán cơ bản và phổ biến nhất trong học máy, đặc biệt phù hợp với các bài toán phân loại nhị phân. Về mặt cấu trúc, LR tương tự như hồi quy tuyến tính ở chỗ sử dụng tổ hợp tuyến tính giữa các đặc trưng đầu vào và trọng số, tuy nhiên kết quả đầu ra được đưa qua một hàm kích hoạt phi tuyến gọi là **hàm logistic (sigmoid)** để ánh xạ về miền giá trị [0, 1] [108]:

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

Trong đó,  $\mathbf{w}$  là vector trọng số,  $\mathbf{x}$  là vector đặc trưng đầu vào, và  $b$  là hệ số điều chỉnh (bias). Giá trị  $\sigma(z)$  thể hiện xác suất điểm dữ liệu  $\mathbf{x}$  thuộc lớp 1. Nếu xác suất này lớn hơn ngưỡng (thường là 0.5), mô hình phân loại  $\mathbf{x}$  thuộc lớp dương.

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 (multiclass classification), có thể sử dụng biến thể **Softmax Regression**, trong đó mô hình ước lượng xác suất đầu ra theo phân phối softmax:

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

Trong đó,  $K$  là tổng số lớp,  $\mathbf{w}_j$  là vector trọng số tương ứng với lớp  $j$ .

Trong khuôn khổ đề tài này, Logistic Regression được lựa chọn nhờ ưu điểm về đơn giản, hiệu quả tính toán và kích thước mô hình nhỏ gọn (< 5 KB), cho phép triển khai trực tiếp trên các vi điều khiển như nRF52840. Mặc dù độ chính xác có thể thấp hơn một số mô hình phức tạp hơn như Random Forest hoặc Gradient Boosting, LR vẫn đảm bảo hiệu năng chấp nhận được trong bối cảnh hệ thống nhúng giới hạn tài nguyên.

**Máy vector hỗ trợ (Support Vector Machine - SVM)** 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 (margin) từ nó đến các điểm dữ liệu gần nhất của mỗi lớp - gọi là **support vectors** - là tối đa. Bài toán tối ưu hóa trong SVM có thể biểu diễn như sau:

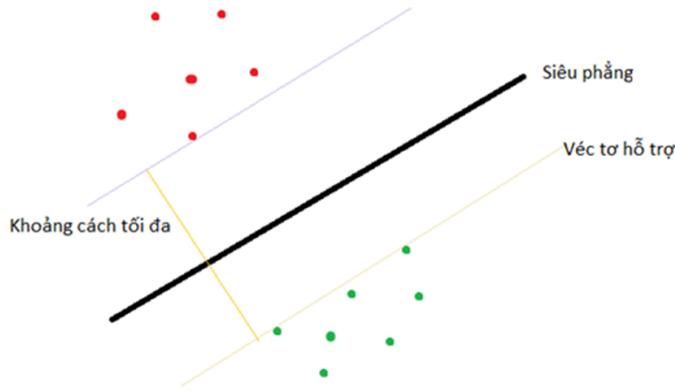
$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \forall i \quad (2.3)$$

Trong đó,  $\mathbf{w}$  là vector trọng số,  $b$  là hệ số bias, và  $(\mathbf{x}_i, y_i)$  là tập dữ liệu huấn luyện.

Hình 2.29 minh họa khái niệm mặt siêu phẳng và các support vectors trong không gian hai chiều.

Ưu điểm nổi bật của SVM là khả năng xử lý hiệu quả trong không gian đặc trưng cao.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.

Tuy nhiên, SVM cũng tồn tại một số hạn chế. Khi dữ liệu không thể phân tách



Hình 2.29: Minh họa mặt siêu phẳng phân tách hai lớp trong SVM

tuyến tính rõ ràng hoặc chứa nhiều nhiễu, hiệu quả phân loại có thể suy giảm đáng kể. Ngoài ra, chi phí tính toán trong giai đoạn huấn luyện tăng nhanh theo kích thước tập dữ liệu, điều này làm cho SVM trở nên khó triển khai trong các hệ thống có tài nguyên hạn chế hoặc yêu cầu thời gian thực, như thiết bị nhúng hoặc điện.

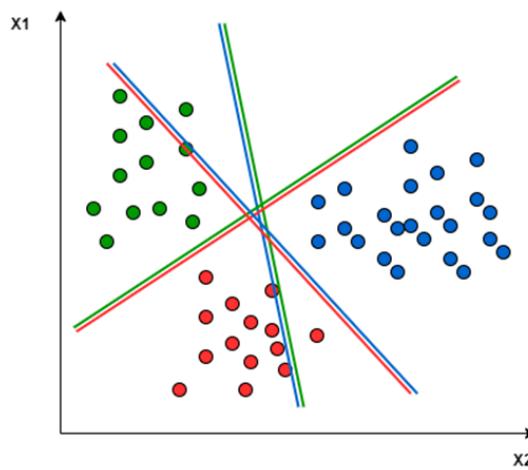
Một khái niệm quan trọng trong SVM là **biên (margin)** - khoảng cách giữa mặt siêu phẳng và các điểm dữ liệu gần nhất thuộc hai lớp. Mặt siêu phẳng tối ưu là mặt phẳng có biên lớn nhất, và chỉ những điểm nằm gần sát biên mới ảnh hưởng đến việc xác định mặt siêu phẳng, được gọi là **các vectơ hỗ trợ (support vectors)**. Các điểm này hỗ trợ việc xác định biên phân cách và trực tiếp ảnh hưởng đến hàm quyết định (decision function) của mô hình. Bài toán tối ưu hoá của SVM tìm ra các trọng số và bias sao cho biên được cực đại.

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 2.30 và Hình 2.31.

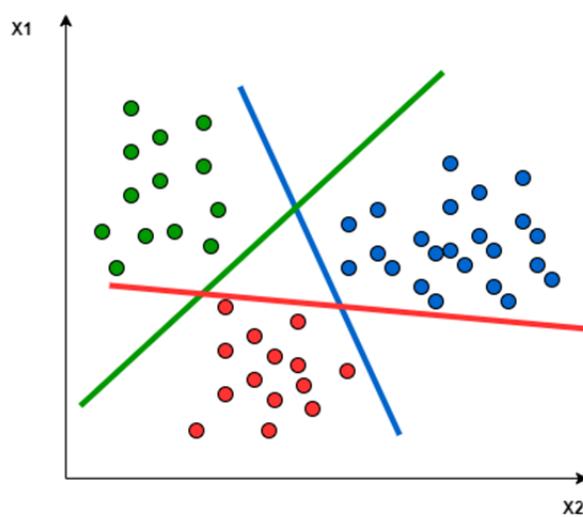
**One-vs-One (OvO):** Trong phương pháp này, một mô hình SVM được huấn luyện cho mỗi cặp lớp. Với  $K$  lớp, tổng cộng  $\frac{K(K-1)}{2}$  mô hình con được huấn luyện. Mỗi mô hình học cách phân biệt giữa hai lớp cụ thể và bỏ qua các lớp còn lại. Trong quá trình dự đoán, một cơ chế bỏ phiếu (voting)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



Hình 2.30: Chiến lược phân loại đa lớp bằng phương pháp One-vs-One



Hình 2.31: Chiến lược phân loại đa lớp bằng phương pháp One-vs-Rest

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 (Random Forest - RF)** 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)** [110]. Khác với việc sử dụng một cây duy nhất như trong Decision Tree truyền thống, Random Forest 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 (voting) hoặc trung bình hoá (trong bài toán hồi quy).

Ý tưởng chính của Random Forest nhằm giảm thiểu hiện tượng **quá khớp (overfitting)** thường gặp trong Decision Tree 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.

Do các cây trong Random Forest được huấn luyện trên những tập dữ liệu và tập thuộc tính khác nhau, mỗi cây đơn lẻ có thể có sai số lớn (high bias hoặc underfitting). Tuy nhiên, việc tổng hợp kết quả của nhiều cây giúp giảm phương sai (variance), cải thiện khả năng tổng quát hoá. Nhờ đó, Random Forest đạt được sự cân bằng giữa bias và variance - một trong những đặc điểm lý tưởng của mô hình học máy tốt.

Một đặc điểm quan trọng khác là **tính ổn định** của RF đối với nhiễu và dữ liệu không cân bằng, cùng với khả năng **đánh giá mức độ quan trọng của đặc trưng (feature importance)** thông qua chỉ số Gini hoặc entropy trung bình trên toàn bộ cây.

Tuy nhiên, mô hình Random Forest có kích thước lớn do lưu trữ nhiều cây quyết định, mỗi cây có thể có độ sâu đáng kể. Điều này khiến RF khó triển khai trực tiếp trong

môi trường hạn chế tài nguyên như các vi điều khiển hoặc thiết bị đeo (wearables). Trong nghiên cứu này, RF được sử dụng như một mô hình tham chiếu mạnh về độ chính xác, nhưng chưa phải là lựa chọn phù hợp cho triển khai biên (TinyML).

**Học tăng cường (Gradient Boosting - GB)** là một phương pháp học có giám sát thuộc nhóm thuật toán tổ hợp (ensemble learning), trong đó nhiều mô hình yếu (weak learners) - thường là các cây quyết định có độ sâu nông - được kết hợp theo cách tuần tự để tạo thành một mô hình mạnh hơn [111].

Khác với Random Forest - nơi các cây được xây dựng song song và độc lập - Gradient Boosting xây dựng mô hình theo từng bước lặp (iteration), 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:

$$F_{t+1}(x) = F_t(x) + \gamma h_t(x) \quad (2.4)$$

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 (MSE) 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 tuần 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 nhiễu sẽ dễ dàng bị mô hình "học theo", dẫn đến hiện tượng quá khớp (overfitting) nếu không áp dụng kỹ thuật regularization hoặc early stopping.

Trong nghiên cứu này, tác giả sử dụng **GB** để phân loại tư thế ngủ đạt độ chính xác cao nhất, điều này đúng trên toàn bộ kịch bản thử nghiệm. Tuy nhiên, do số lượng cây lớn và trọng số tổng thể cao (trên 500 KB), Gradient Boosting chưa phù hợp để triển khai trực tiếp trên các thiết bị vi điều khiển hạn chế tài nguyên. Thay vào đó, mô hình này được sử dụng để thiết lập ngưỡng hiệu năng tham chiếu (baseline) trong môi trường

huấn luyện trên máy chủ hoặc máy tính cá nhân.

**Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN)** là một 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) [112]. Trong nghiên cứu này, tác giả sử dụng kiến trúc **Multilayer Perceptron (MLP)** - một loại mạng nơ-ron đơn giản gồm ít nhất ba lớp: lớp đầu vào (input layer), một hoặc nhiều 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):

$$f(x) = \max(0, x) \quad (2.5)$$

Đầu ra của mạng được tính thông qua lan truyền tiến (forward propagation), và mô hình được huấn luyện bằng cách tối thiểu hóa một hàm mất mát (loss function), chẳng hạn như hàm binary cross-entropy trong phân loại nhị phân, thông qua thuật toán lan truyền ngược (backpropagation) và phương pháp tối ưu như Adam hoặc SGD.

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 tín hiệu cảm biến có nhiều, biến đổi theo thời gian hoặc không gian.

Tuy nhiên, ANN cũng tồn tại nhiều thách thức trong thực tế triển khai: 01) Lựa chọn tham số 02) Yêu cầu tính toán cao 03) Khó giải thích: ANN hoạt động như một hộp đen, khó hiểu về mặt trực quan so với cây quyết định hoặc hồi quy logistic.

Trong đề tài này, tác giả sử dụng một kiến trúc MLP đơn giản gồm hai lớp ẩn với số lượng nơ-ron tương đối nhỏ (8, 4) và hàm kích hoạt ReLU, được huấn luyện bằng thuật toán tối ưu Adam với tốc độ học ban đầu là 0.01.

**Convolutional Neural Network (CNN)** là một mô hình học sâu được thiết kế chuyên biệt để xử lý các loại dữ liệu có cấu trúc lưới, chẳng hạn như ảnh hai chiều hoặc tín hiệu chuỗi thời gian một chiều. Không giống như mạng neural truyền thống, CNN sử dụng phép tích chập để trích xuất tự động các đặc trưng cục bộ trong dữ liệu, từ đó giảm thiểu đáng kể nhu cầu tiền xử lý và cải thiện hiệu quả học biểu diễn [113].

Một kiến trúc CNN điển hình bao gồm các tầng tích chập (convolutional layers), tiếp theo là các hàm kích hoạt phi tuyến như ReLU và các tầng giảm mẫu (pooling layers). Sau các tầng này, các đặc trưng được đưa vào một hoặc nhiều tầng kết nối đầy đủ (fully connected layers) để thực hiện phân loại. Quá trình này cho phép CNN học được cả đặc trưng cục bộ lẫn toàn cục trong tín hiệu đầu vào.

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 (features) 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. Để thực hiện điều đó, tác giả xây dựng tám kịch bản khác nhau tương ứng với các tổ hợp đặc trưng và kích thước cửa sổ khác nhau. Việc huấn luyện và đánh giá trên nhiều kịch bản đòi hỏi một mô hình đủ linh hoạt, dễ kiểm soát về kích thước và thời gian huấn luyện.

## **Chương 3**

# **KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ**

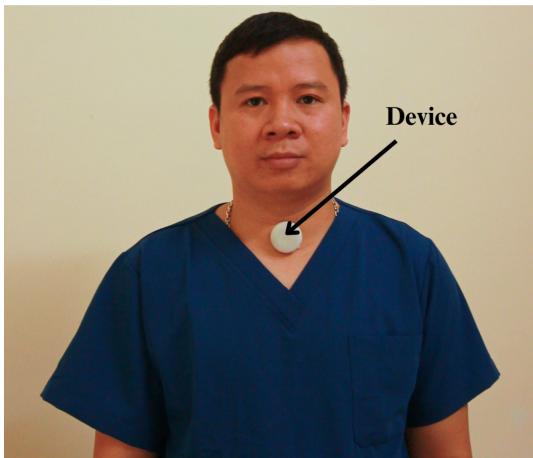
Trong phần này, tác giả trình bày các kết quả đạt được từ hai giai đoạn chính: (i) thu thập và xử lý dữ liệu cảm biến từ các tư thế ngủ khác nhau; (ii) huấn luyện mô hình học máy và triển khai mô hình tối ưu lên vi điều khiển nhằm đánh giá tính khả thi của giải pháp trên thiết bị biến.

### **3.1. Hệ thống thực nghiệm**

Tác giả đã hoàn thiện việc lập trình firmware cho bộ vi mạch cảm biến tích hợp vi điều khiển NRF52840 và cảm biến gia tốc ba trục LIS3DH. Firmware được xây dựng sử dụng ngôn ngữ C/C++ trên nền tảng Arduino Core, tối ưu hóa để vận hành ổn định trong môi trường năng lượng thấp và hỗ trợ giao tiếp không dây chuẩn Bluetooth Low Energy (BLE).

Để đảm bảo khả năng hoạt động liên tục trong suốt một đêm ngủ (tối thiểu 8 giờ), hệ thống được thiết kế sử dụng pin cúc áo CR2032, với dòng tiêu thụ trung bình được đo đạt dưới 8 mA trong chế độ ghi nhận liên tục và truyền dữ liệu định kỳ.

Mã nguồn firmware bao gồm các khối chức năng chính: khởi tạo cảm biến, hiệu chỉnh dải đo và tần số lấy mẫu (10 Hz), lọc nhiễu đầu vào (bằng kỹ thuật trung bình trượt), đóng gói dữ liệu, và truyền dữ liệu qua BLE đến ứng dụng Android. Ngoài ra, tác



(a)



(b)

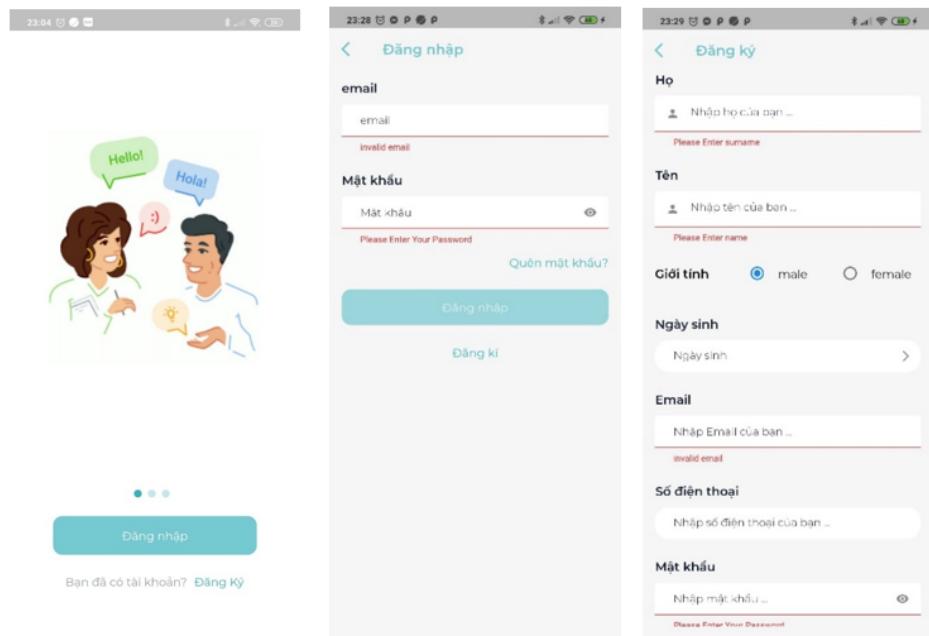
Hình 3.1: Hệ thống thử nghiệm: (a) vị trí đặt thiết bị cảm biến; (b) minh họa thực nghiệm thực tế trong tư thế nằm.

giả cung tích hợp cơ chế báo hiệu bằng LED để xác nhận trạng thái hoạt động của thiết bị (kết nối, truyền dữ liệu, và lỗi). Kết quả thử nghiệm thực tế cho thấy hệ thống hoạt động ổn định trong suốt thời gian ghi nhận dữ liệu qua đêm, không xảy ra hiện tượng rớt kết nối hay tràn bộ đệm dữ liệu. Toàn bộ dữ liệu thu được được đồng bộ theo thời gian thực tới ứng dụng di động, phục vụ cho các giai đoạn xử lý tín hiệu và huấn luyện mô hình học máy ở các chương sau.

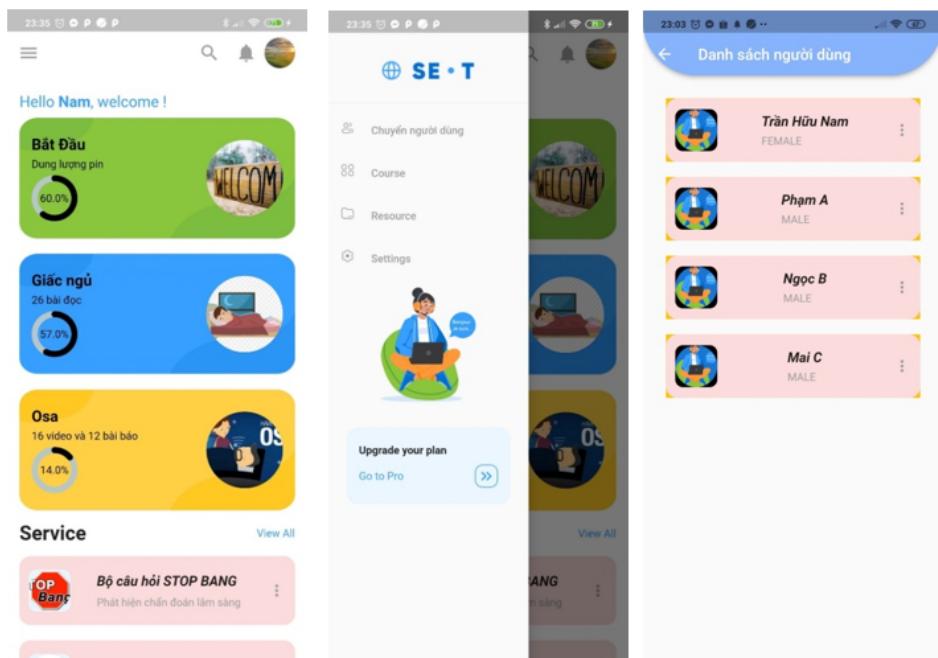
Một trong các nhiệm vụ chính mà tác giả thực hiện là phát triển ứng dụng di động phục vụ cho quá trình thu thập, hiển thị và xử lý dữ liệu. Dựa trên phản hồi từ nhóm nghiên cứu, tư vấn khoa học của Thầy PGS.TS. Mai Anh Tuấn, tư vấn y khoa của Thầy GS.TS. Dương Quý Sỹ, ứng dụng được thiết kế với tiêu chí giao diện thân thiện, thao tác đơn giản và tính năng tập trung vào mục tiêu thử nghiệm.

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 3.2. 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 3.3 là giao diện khi người dùng đăng nhập thành công, bao gồm các tính năng: Kết nối BLE và đọc dữ liệu, chuyển người dùng, xem thông tin người dùng v.v. Hình 3.4 thể hiện danh sách BLE có thể kết nối và dịch vụ kết nối với phần cứng đã được nhắc tới bên trên.

Hình 3.5 minh họa giao diện của các chức năng hỗ trợ người dùng trong quá trình

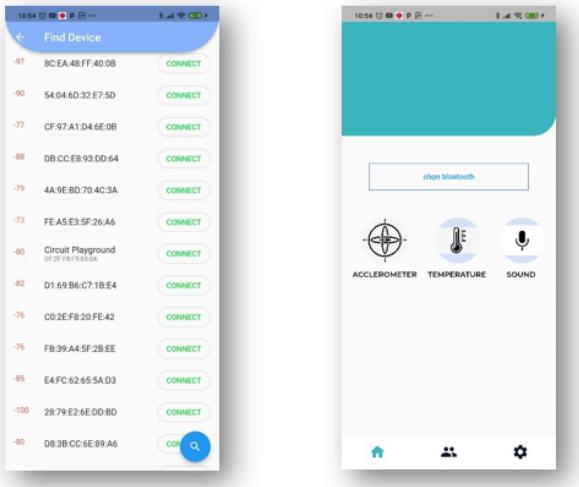


Hình 3.2: Giao diện chức năng đăng ký và đăng nhập



Hình 3.3: Giao diện trang chủ

sàng lọc nguy cơ ngưng thở khi ngủ và cung cấp thông tin về chất lượng giấc ngủ. Giao diện đầu tiên (tù trái sang) hiển thị mục "Hỏi – Đáp về Giác Ngủ", nơi người dùng có thể tra cứu các thông tin được tổng hợp từ chuyên gia trong lĩnh vực y học giấc ngủ. Giao diện thứ hai trình bày tập hợp các công cụ đo lường phổ biến như thang điểm Epworth,

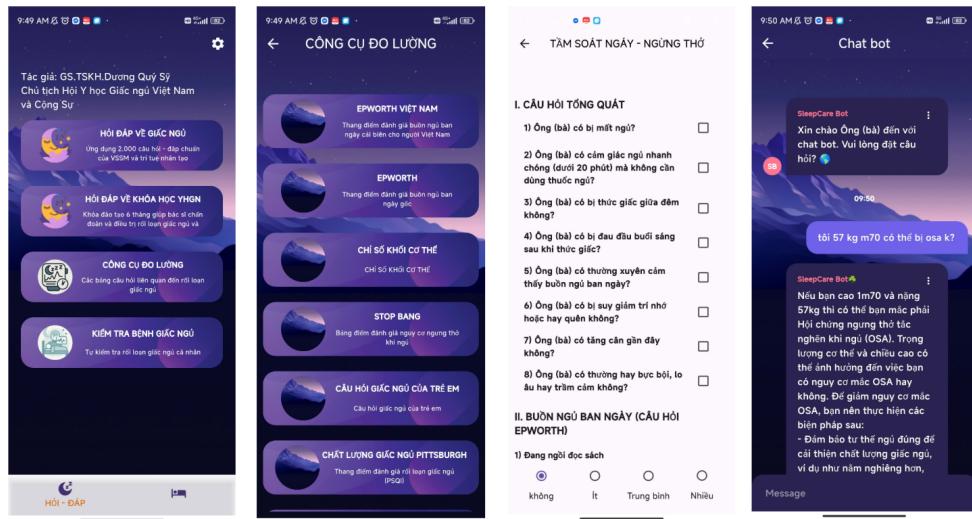


Hình 3.4: Giao diện màn hình danh sách BLE và chi tiết các dịch vụ kết nối với phần cứng

STOP-BANG, chỉ số khó có thể duy trì sự tĩnh táo (ESS), và các bộ câu hỏi dành riêng cho trẻ em hoặc đánh giá chất lượng giấc ngủ theo thang điểm Pittsburgh (PSQI). Giao diện thứ ba mô tả chi tiết một bộ câu hỏi sàng lọc nguy cơ ngưng thở khi ngủ (tên mục: “Tầm soát ngày – Ngưng thở”) bao gồm các câu hỏi tổng quát và chuyên biệt nhằm đánh giá các yếu tố liên quan đến OSA (Obstructive Sleep Apnea), như tần suất ngáy, triệu chứng ngủ gật ban ngày, gián đoạn giấc ngủ, hoặc các đặc điểm nhân trắc học có liên quan.

Giao diện thứ tư là chức năng chatbot – nơi người dùng có thể trao đổi trực tiếp với hệ thống trí tuệ nhân tạo được lập trình sẵn để phản hồi các câu hỏi về OSA. Chatbot có khả năng nhận diện từ khóa và cung cấp phản hồi ngắn gọn dựa trên cơ sở dữ liệu đã huấn luyện. Trong ví dụ minh họa, chatbot phản hồi một truy vấn liên quan đến chỉ số BMI và nguy cơ mắc OSA, thể hiện vai trò hỗ trợ tư vấn bước đầu cho người dùng nghi ngờ có hội chứng ngưng thở khi ngủ.

Hình 3.6 minh họa giao diện hiển thị giá trị cảm biến theo thời gian thực. Phần đầu hiển thị biểu đồ ba trục x, y, z. Phần thứ hai là tổng thời gian theo từng tư thế ngủ được tính toán dựa trên tín hiệu nhận dạng. Phần cuối cùng cho biết tư thế hiện tại mà hệ thống đang xác định được. Tuy nhiên, phương pháp xác định tư thế dựa trên ngưỡng chưa có tính tổng quát cao. Do đó, trong các phần tiếp theo, các mô hình học máy sẽ được áp dụng để cải thiện độ chính xác và ổn định của hệ thống nhận diện tư thế. Việc cập nhật tư thế được thực hiện định kỳ mỗi 10 giây để tăng khả năng phản hồi theo thời



Hình 3.5: Giao diện chức năng chatbot và bộ câu hỏi tâm soát

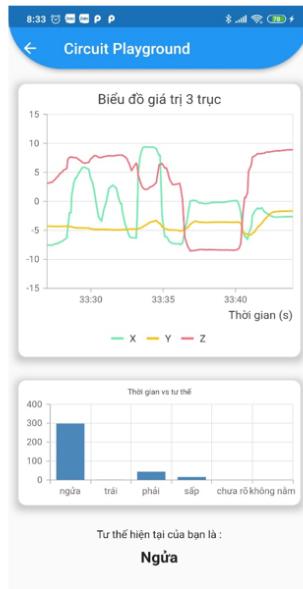
```

1 static Function getPositionSleep = (double x, double y, double z) {
2     if ((-6.5 < y && y < 6.5)) {
3         if (-7.07 < x && x < 7.07) {
4             if (z > 0) {
5                 return 1; // ngua
6             }
7             if (z < 0) {
8                 return 4; //sap
9             }
10        }
11        if (x > 3) return 2; //trai
12        if (x < -3) return 3; //phai
13    }
14    return 6; // khong phai nam
15 };

```

Mã nguồn 3.1: Tập lệnh đánh giá tư thế ngủ bằng ngưỡng

gian thực.



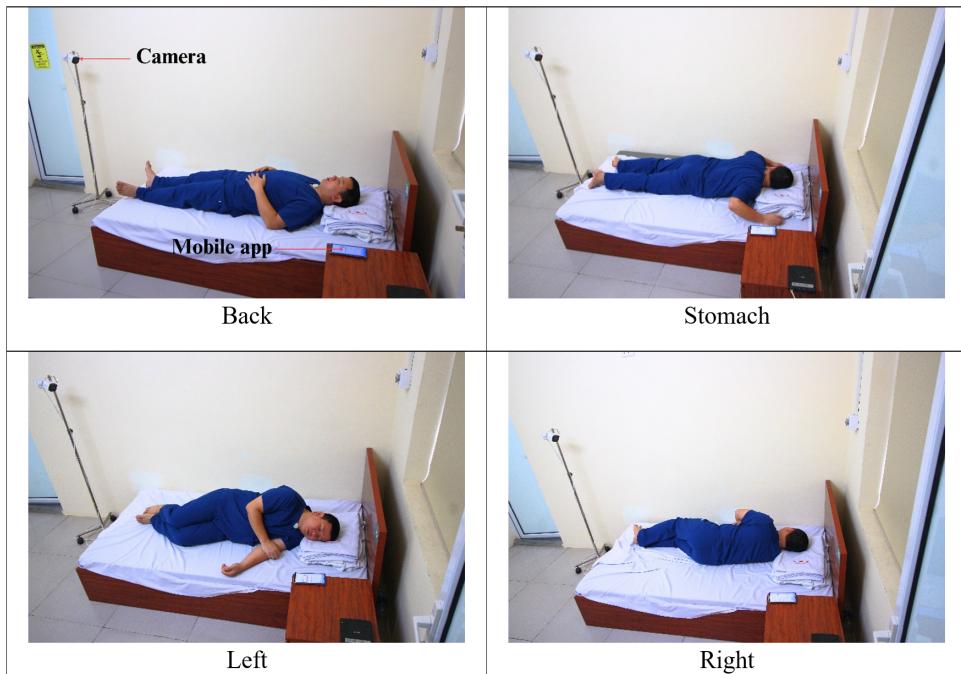
Hình 3.6: Giao diện hiển thị dữ liệu gia tốc ba trục

Về mặt kiến trúc lưu trữ dữ liệu, hệ thống được thiết kế phân tách giữa dữ liệu định tính và dữ liệu định lượng. Cụ thể, thông tin người dùng (tài khoản, cấu hình cá nhân), các bộ câu hỏi tầm soát (ví dụ: STOP-BANG, ESS), cùng với nội dung trả lời và phản hồi của chatbot được lưu trữ trong cơ sở dữ liệu quan hệ **PostgreSQL**. Cơ sở dữ liệu này hỗ trợ tính nhạy cảm cao và dễ dàng cho việc mở rộng truy vấn phức tạp trong các bài toán phân tích sau này. Trong khi đó, dữ liệu cảm biến gia tốc được lưu trữ song song tại cơ sở dữ liệu phi quan hệ **MongoDB**, với định dạng BSON linh hoạt, phù hợp cho việc ghi nhận chuỗi thời gian lớn và truy xuất nhanh theo timestamp. Ngoài ra, hệ thống được mở rộng với các API cho phép trích xuất dữ liệu dưới dạng Excel, nhằm hỗ trợ phân tích và chia sẻ thông tin một cách linh hoạt.

### 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



Hình 3.7: Mô phỏng thực nghiệm thực tế

nguyễn 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 xương ức (ngay dưới homb 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 5 phút cho mỗi tư thế. Thứ tự thay đổi tư thế được thực hiện ngẫu nhiên nhằm tránh thiên lệch theo trình tự. 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 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 thực. 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 Hình 3.7.

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 sẽ là dữ liệu trong suốt một đêm ngủ tự nhiên của tác giả. Việc gán nhãn dữ liệu được thực hiện thủ công bằng cách đồng bộ thời gian giữa tín hiệu cảm biến và dữ liệu video, sau đó loại bỏ toàn bộ các đoạn có chuyển động hoặc tư thế không rõ ràng. Kết quả là bộ dữ liệu kiểm thử gồm **64.258** mẫu đảm bảo độ chính xác cao về mặt nhãn.

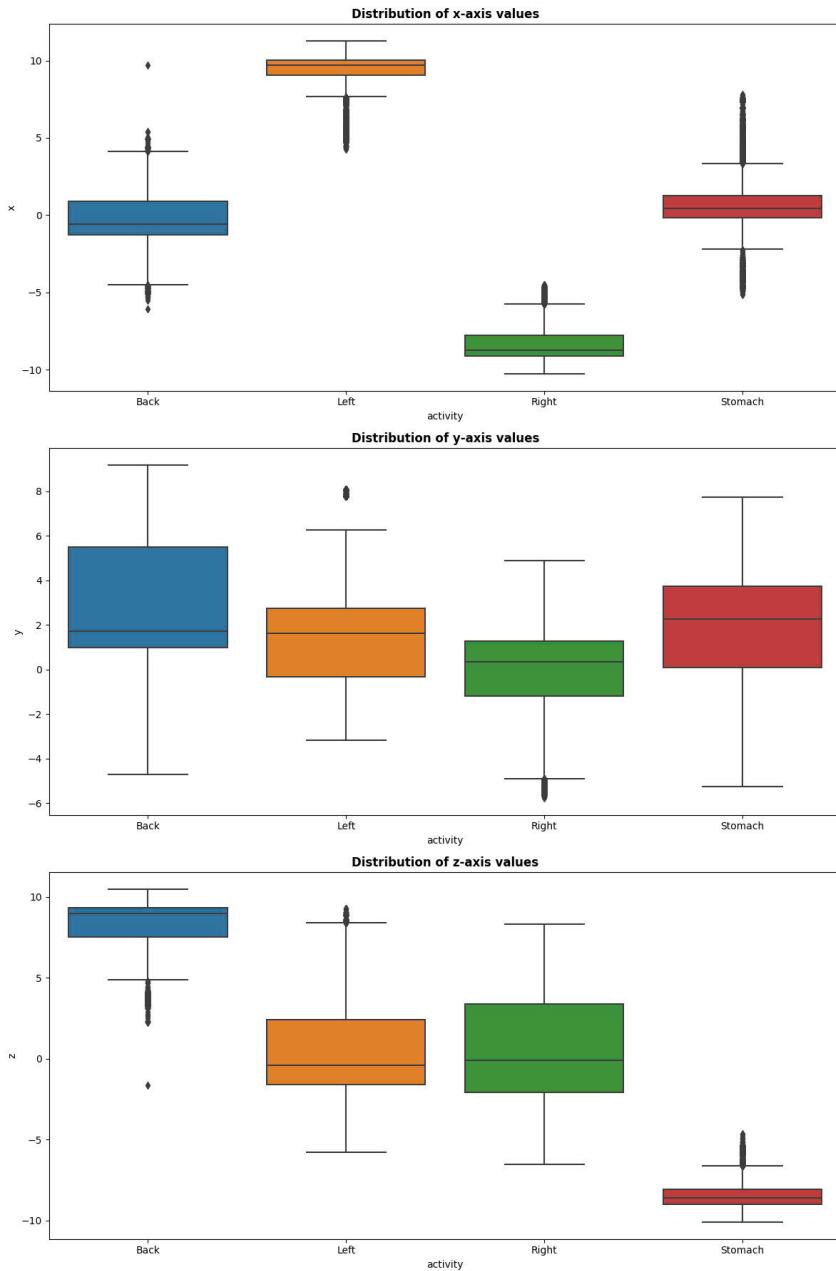
Tất cả dữ liệu thu thập từ các tình nguyện viên và tác giả đều được xuất ra định dạng CSV, bao gồm thông tin thời gian (timestamp), giá trị cảm biến trên ba trục  $x$ ,  $y$ ,  $z$ , và nhãn tư thế tương ứng (nếu có). Dữ liệu này được sử dụng làm đầu vào cho quá trình trích xuất đặc trưng và huấn luyện mô hình học máy.

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

#### 3.3.1. Phân tích dữ liệu

Tư thế ngủ ban đầu có thể được ước lượng bằng phương pháp dựa trên ngưỡng (threshold-based), áp dụng trực tiếp lên dữ liệu cảm biến gia tốc ba trục. Trong phương pháp này, các ngưỡng được thiết lập trước cho từng trục ( $x$ ,  $y$ ,  $z$ ), và sự thay đổi tư thế được suy đoán khi giá trị gia tốc đo được vượt quá ngưỡng tương ứng. Kỹ thuật này có ưu điểm là đơn giản, chi phí tính toán thấp, và đặc biệt phù hợp với các hệ thống nhúng tiêu thụ năng lượng thấp. Mặc dù kỹ thuật dựa trên ngưỡng có ưu điểm đơn giản và phù hợp với các hệ thống nhúng có tài nguyên hạn chế, nó tồn tại một số hạn chế nhất định. Cụ thể, phương pháp này khó phát hiện các chuyển động nhẹ hoặc tư thế trung gian giữa các

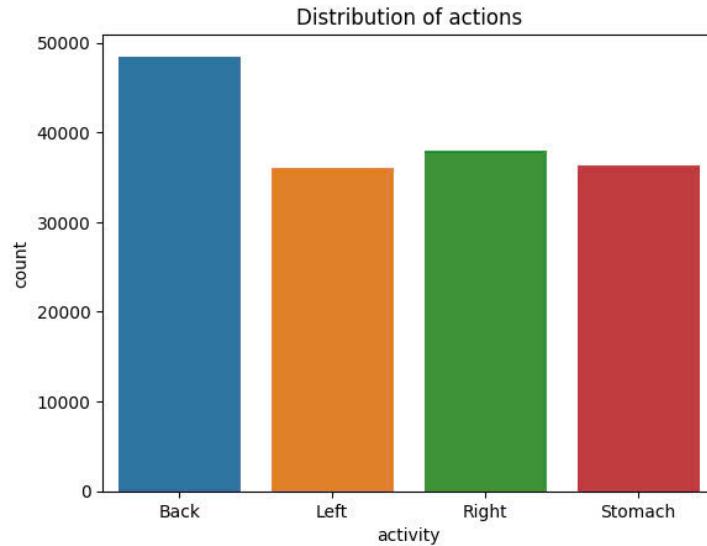
trạng thái rõ ràng. Ngoài ra, các ngưỡng thường cần hiệu chỉnh theo từng cá nhân do sự khác biệt về hình thể, kiểu vận động và vị trí gắn cảm biến.



Hình 3.8: Phân bố dữ liệu cảm biến theo ba trục  $x$ ,  $y$ ,  $z$  ứng với các tư thế ngủ khác nhau.

Hình 3.8 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. Ở trục  $x$ , các phân bố tương đối biệt lập, đặc biệt 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. Ngược lại, trục  $y$  thể hiện mức độ chồng lấn lớn giữa các tư thế, dẫn đến khả năng tách biệt thấp và ít giá trị trong việc xác định tư thế

ngủ. Đối 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.



Hình 3.9: Phân bố số lượng mẫu trong tập huấn luyện theo từng tư thế.

Hình 3.9 minh họa sự phân bố số lượng mẫu trong tập huấn luyện theo từng tư thế. Tư thế nằm ngửa (Back) chiếm tỷ trọng cao nhất với khoảng 50.000 mẫu, trong khi ba tư thế còn lại (nghiêng trái, nghiêng phải và nằm sấp) có số lượng tương đối cân bằng, dao động từ 30.000 đến 35.000 mẫu. Phân bố này phản ánh xu hướng phổ biến của tư thế nằm ngửa trong giấc ngủ tự nhiên, đồng thời cho thấy tầm quan trọng lâm sàng của tư thế này, đặc biệt trong bối cảnh hội chứng ngưng thở khi ngủ (OSA), khi tư thế nằm ngửa có thể làm trầm trọng tình trạng bệnh.

### 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 (differential technique), 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 liên tiếp nhằm tăng độ mịn của chuỗi dữ liệu đầu vào. 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 vốn là 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.

#### **3.3.3. Kịch bản kiểm thử và lựa chọn tính năng**

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 (overfitting). Nguyên lý chung là các đặc trưng hiệu quả phải có mối tương quan cao với biến mục tiêu (tư thế ngủ), đồng thời có mức tương quan

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)	$\frac{1}{n} \sum_{i=1}^n ( x_i  +  y_i  +  z_i )$

thấp với nhau nhằm tránh dư thừa thông tin.

**Thứ nhất**, phân tích ma trận tương quan Pearson (Hình 3.10) 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.

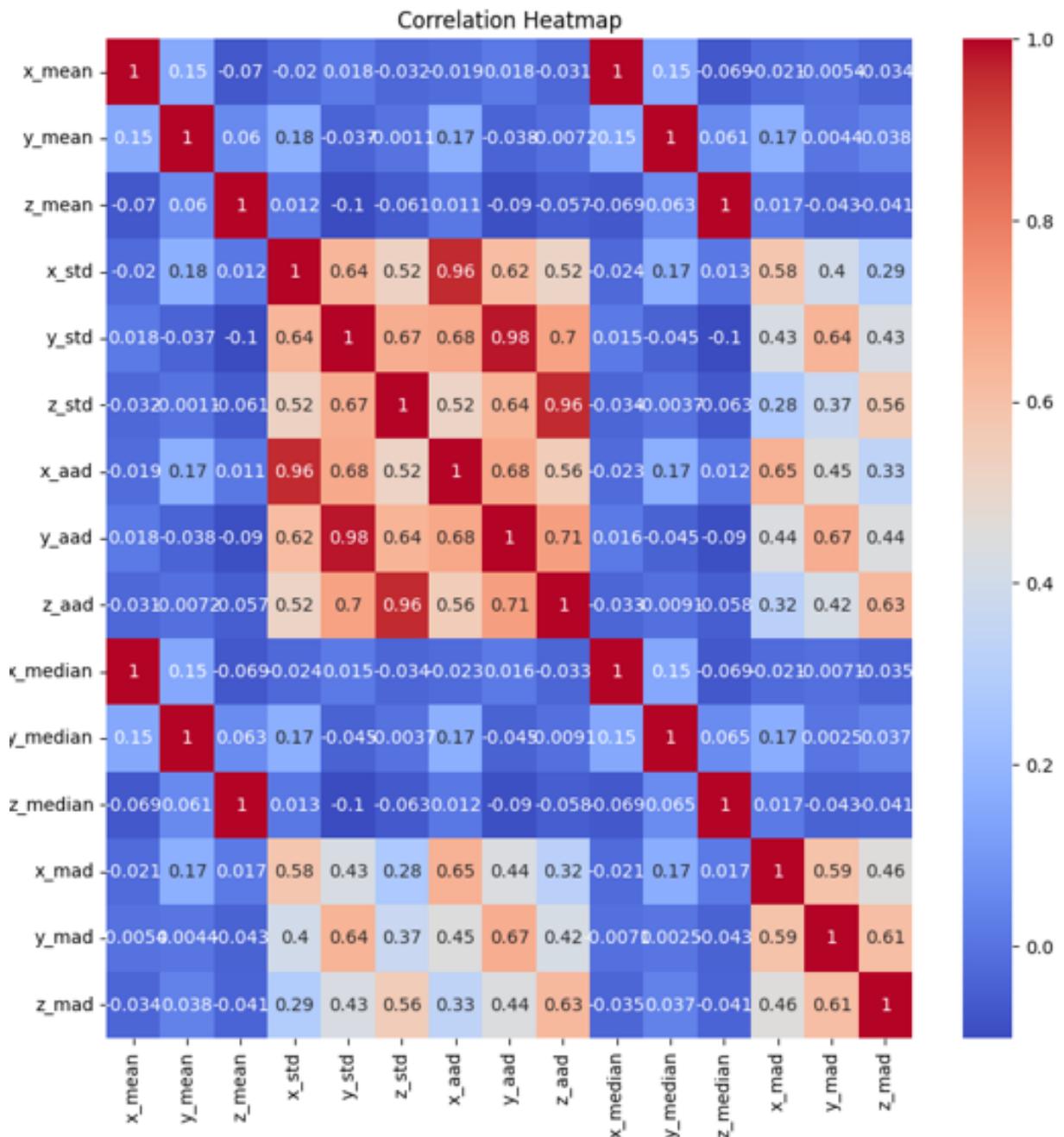
**Thứ hai**, kết quả phân tích SHAP (SHapley Additive exPlanations) ở Hình 3.11 chỉ ra rằng một số đặc trưng – đặc biệt là các đặc trưng miền thời gian trên trục  $z$  như trung bình, năng lượng, trung vị – có ảnh hưởng vượt trội đến dự đoán của mô hình. Do đó, việc ưu tiên các đặc trưng này trong kịch bản triển khai nhẹ (TinyML) là hoàn toàn hợp lý về mặt kỹ thuật.

**Thứ ba**, để hệ thống hóa việc đánh giá vai trò của từng nhóm đặc trưng và cấu hình mô hình, tác giả đã xây dựng tám kịch bản thực nghiệm được trình bày trong Bảng 3.2. Các kịch bản được thiết kế nhằm phản ánh đầy đủ các yếu tố cần đánh giá như loại đặc trưng (miền thời gian, tần số), mức độ tương quan, độ dài cửa sổ tín hiệu và bộ đặc trưng tối ưu hoá bằng SHAP.

Đối với các kịch bản 5 đến 8, việc lựa chọn 11 đặc trưng được thực hiện bằng cách huấn luyện mô hình Random Forest trên toàn bộ tập đặc trưng, sau đó tính giá trị SHAP trung bình cho từng đặc trưng và chọn ra nhóm có ảnh hưởng cao nhất. Việc rút gọn đặc trưng này giúp mô hình nhẹ hơn, nhanh hơn, phù hợp với môi trường nhúng có giới hạn về bộ nhớ và tính toán.

Các kịch bản 6 đến 8 thay đổi độ dài cửa sổ trượt (1 giây, 2 giây, 3 giây) và mức chồng lấn nhằm khảo sát tác động của kích thước đoạn tín hiệu đến hiệu quả mô hình, đồng thời phản ánh điều kiện sử dụng thực tế trong các thiết bị đeo (wearables) hoặc hệ thống biên (edge-AI).

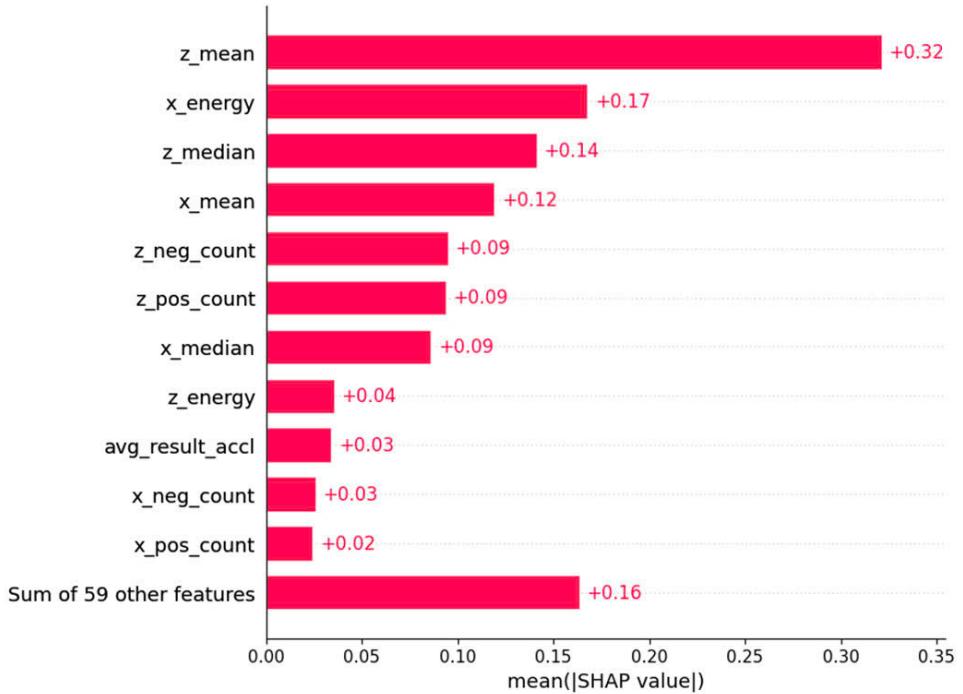
Với thiết kế kịch bản như trên, luận văn không chỉ đánh giá hiệu quả mô hình theo nhiều hướng khác nhau, mà còn hướng tới việc xác định cấu hình tối ưu giữa độ chính xác, chi phí tính toán và khả năng triển khai thực tiễn.



Hình 3.10: Ma trận tương quan giữa các đặc trưng trích xuất. Cường độ màu thể hiện hệ số tương quan Pearson. Màu đỏ là tương quan dương mạnh, xanh là tương quan âm mạnh, xám là không tương quan.

Bảng 3.2: Các kịch bản lựa chọn và sử dụng đặc trưng trong nghiên cứu

Kịch bản	Mô tả
1	Sử dụng toàn bộ đặc trưng để đánh giá ảnh hưởng tổng thể đến mô hình.
2	Áp dụng toàn bộ đặc trưng trong miền thời gian.
3	Áp dụng toàn bộ đặc trưng trong miền tần số.
4	Sử dụng đặc trưng miền thời gian, loại bỏ các đặc trưng có tương quan $> 95\%$ .
5	Chọn ra 11 đặc trưng quan trọng nhất theo giá trị SHAP.
6	Dùng cửa sổ 3 giây (50% overlap), với 11 đặc trưng SHAP.
7	Dùng cửa sổ 1 giây (50% overlap), với 11 đặc trưng SHAP.
8	Dùng cửa sổ 2 giây (25% overlap), với 11 đặc trưng SHAP.



Hình 3.11: Phân tích giá trị SHAP nhằm xác định tầm quan trọng của các đặc trưng trong mô hình phân loại tư thế ngủ. Các đặc trưng từ trục  $z$  chiếm ưu thế về mức ảnh hưởng đến đầu ra mô hình.

### 3.3.4. Huấn luyện mô hình

Dựa trên các phương pháp phân loại được đề cập ở các phần trước — bao gồm phương pháp ngẫu ngamage, học máy và học sâu — tác giả đã lựa chọn một tập hợp đại diện các mô hình để tiến hành đánh giá hiệu quả trong bài toán nhận diện tư thế ngủ từ dữ liệu cảm biến gia tốc. Cụ thể, bốn mô hình học máy truyền thống được lựa chọn từ thư viện scikit-learn gồm: **Random Forest (RF)**, **Logistic Regression (LR)**, **Support Vector Machine (SVM)**, và **Gradient Boosting (GB)**. Đây đều là các mô hình đã được chứng minh hiệu quả trong việc xử lý dữ liệu cảm biến có cấu trúc, đặc biệt trong các bài toán phân loại đa lớp.

Để đảm bảo tính công bằng trong so sánh và khả năng triển khai thực tế trên vi điều khiển, các siêu tham số (hyperparameters) của từng mô hình được lựa chọn dựa trên kinh nghiệm thực tiễn trong các công trình trước và quá trình tinh chỉnh sơ bộ nhằm đạt được sự cân bằng giữa độ chính xác và độ phức tạp tính toán. Chi tiết tham số của từng mô hình được trình bày trong Bảng 3.3.

Bảng 3.3: Các mô hình học máy và siêu tham số sử dụng trong nghiên cứu

Mô hình	Tham số cấu hình
<b>Random Forest (RF)</b>	Số cây quyết định: 50; Độ sâu tối đa: 5; Số đặc trưng được xét tại mỗi nút: log2
<b>Logistic Regression (LR)</b>	Chiến lược đa lớp: one-vs-rest; Số vòng lặp tối đa: 50; Hàm tối ưu: lbfgs
<b>Support Vector Machine (SVM)</b>	Hàm kernel: sigmoid; Tham số điều chuẩn $C = 2$ ; Chiến lược đa lớp: one-vs-rest
<b>Gradient Boosting (GB)</b>	Tốc độ học: 0.01; Số lượng cây tăng cường: 50; Độ sâu tối đa: 3; Số đặc trưng được chọn: log2
<b>Mạng nơ-ron (Neural Network, Keras)</b>	Cấu trúc: [8, 4, <i>num_classes</i> ]; Hàm kích hoạt: ReLU, ReLU, Softmax; Thuật toán tối ưu: Adam (learning rate = 0.01); Hàm mất mát sparse_categorical_crossentropy; Độ đo đánh giá: accuracy

Ngoài các mô hình học máy, một mạng nơ-ron nhân tạo tuyến tính đơn giản (feed-forward neural network) được xây dựng bằng thư viện TensorFlow/Keras để đại diện cho phương pháp học sâu. Mạng bao gồm hai lớp ẩn với số lượng nơ-ron lần lượt là 8 và 4, sau là một lớp đầu ra sử dụng hàm kích hoạt Softmax cho bài toán phân loại đa lớp. Các lớp ẩn sử dụng hàm kích hoạt ReLU nhằm mô hình hóa các quan hệ phi tuyến hiệu quả hơn. Mô hình này được tối ưu bằng thuật toán Adam với tốc độ học (learning rate) là 0.01 và được huấn luyện bằng hàm mất mát sparse\_categorical\_crossentropy, sử dụng độ đo đánh giá accuracy để phản ánh hiệu suất phân loại.

Việc lựa chọn kết hợp các mô hình với mức độ phức tạp khác nhau cho phép đánh giá toàn diện về hiệu quả phân loại trong các điều kiện thực tế. Từ các mô hình cây đơn giản và dễ diễn giải, đến các mô hình mạnh hơn như Gradient Boosting hoặc mạng nơ-ron – nghiên cứu nhằm tìm ra giải pháp cân bằng tối ưu giữa độ chính xác, kích thước mô hình, tốc độ suy luận (inference latency) và mức sử dụng bộ nhớ, phục vụ cho các ứng dụng thực tiễn như hệ thống AI biên (Edge-AI) hoặc thiết bị đeo thông minh.

### 3.3.5. Đánh giá kết quả

Để đánh giá hiệu quả của các mô hình học máy và tác động của lựa chọn đặc trưng đầu vào, tám kịch bản thực nghiệm đã được thiết kế như trình bày ở các phần trước. Các kịch bản này không chỉ cho phép phân tích ảnh hưởng của đặc trưng, cửa sổ tín hiệu và trực cảm biến, mà còn hướng đến tối ưu hóa trọng số mô hình phục vụ triển khai trên thiết bị nhúng.

Bảng 3.4: Độ chính xác phân loại của các mô hình trong 8 kịch bản

Mô hình	S1	S2	S3	S4	S5	S6	S7	S8
LR	0.970	0.970	0.368	0.970	0.987	0.990	0.990	0.987
RF	0.995	0.996	0.426	0.994	0.993	0.993	0.993	0.991
SVM	0.995	0.985	0.280	0.991	0.989	0.982	0.982	0.870
GB	0.996	0.996	0.439	0.995	0.995	0.996	0.996	0.996
NN	0.920	–	–	–	–	–	–	–

Kết quả trong Bảng 3.4 làm sáng tỏ sự khác biệt căn bản giữa các nhóm đặc trưng và cách tiếp cận mô hình. Trước hết, **Gradient Boosting (GB)** nổi bật với hiệu năng ổn định nhất, liên tục đạt giá trị chính xác tối đa (0.996) trong năm kịch bản (S1, S2, S6, S7, S8). Đặc điểm này cho thấy ưu thế vượt trội của các phương pháp tăng cường mô

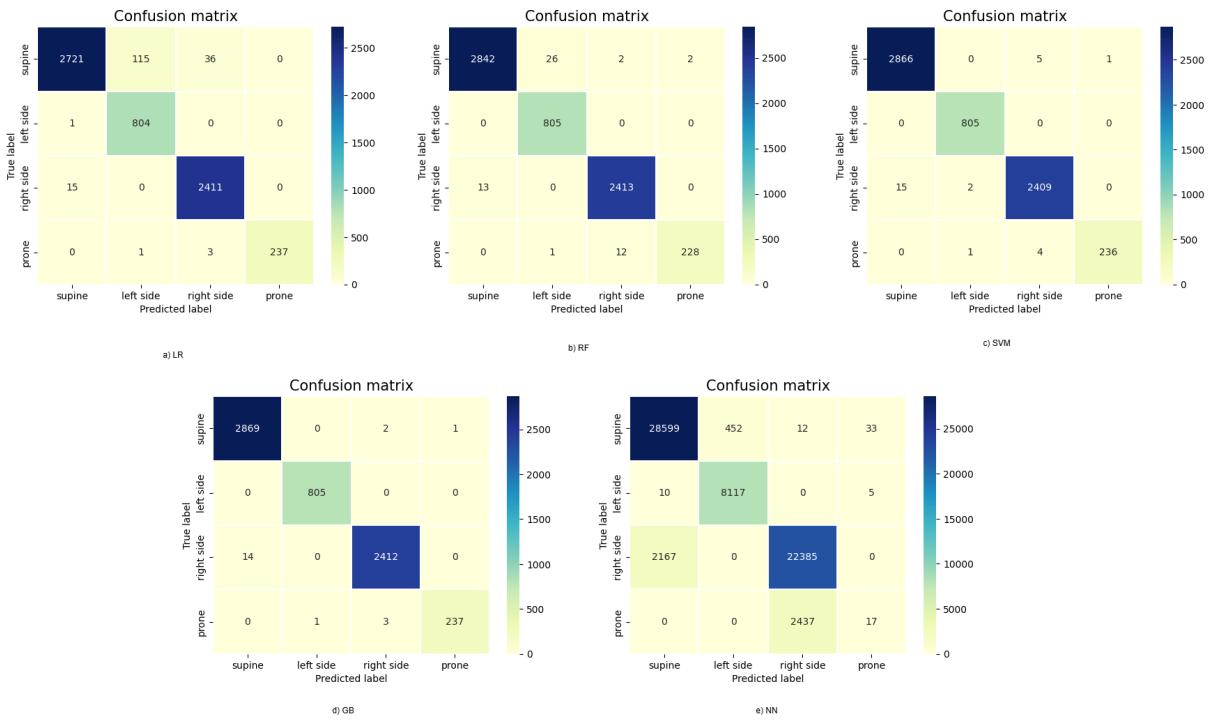
hình dựa trên cây quyết định, khả năng khai thác tốt cả quan hệ phi tuyến và sự tương tác giữa các đặc trưng. **Random Forest (RF)** mặc dù có đôi chút biến thiên, nhưng vẫn duy trì độ chính xác vượt ngưỡng 0.99 trong hầu hết kịch bản, cung cấp vai trò của các mô hình ensemble dựa trên bootstrap aggregation trong việc giảm phương sai và cải thiện khả năng tổng quát hoá.

Trái lại, **Logistic Regression (LR)** và **Support Vector Machine (SVM)** chỉ đạt được hiệu năng tiệm cận 0.99 trong các kịch bản sử dụng đặc trưng miền thời gian hoặc đặc trưng được lựa chọn bằng SHAP. Đặc biệt, ở kịch bản S3 – vốn chỉ khai thác đặc trưng miền tần số – cả hai mô hình này sụt giảm nghiêm trọng về độ chính xác (LR còn 0.368, SVM chỉ 0.280). Kết quả nhấn mạnh rằng các tín hiệu động học quan trọng để phân loại chủ yếu được mã hoá trong miền thời gian. Điều này hoàn toàn tương thích với phân tích SHAP (Hình 3.11), khi các đặc trưng đóng góp nhiều nhất đều tập trung ở miền thời gian, đặc biệt trên trục  $z$ , nơi thể hiện rõ sự khác biệt về động học tư thế.

Một điểm đáng chú ý khác là khi áp dụng cơ chế chọn lọc đặc trưng dựa trên **SHAP** (các kịch bản S5–S8), độ chính xác của các mô hình hầu như không suy giảm, thậm chí trong một số trường hợp còn được cải thiện nhẹ (ví dụ LR đạt 0.990 ở S6 và S7 so với 0.970 ở S1). Điều này chứng tỏ việc loại bỏ các đặc trưng dư thừa và tập trung vào những đặc trưng quan trọng nhất không chỉ giúp giảm chiều dữ liệu mà còn hạn chế hiện tượng nhiễu, từ đó gia tăng độ khái quát hoá. Kết quả này có ý nghĩa thực tiễn quan trọng, vì nó cho phép duy trì hiệu năng phân loại cao trong khi giảm thiểu chi phí tính toán và bộ nhớ – yếu tố then chốt khi triển khai trên các thiết bị nhúng hoặc hệ thống IoT có tài nguyên hạn chế.

Tóm lại, Bảng 3.4 không chỉ xác nhận ưu thế vượt trội của các mô hình ensemble như GB và RF, mà còn chứng minh tính hiệu quả của phương pháp chọn lọc đặc trưng dựa trên SHAP. Sự thất bại rõ rệt của kịch bản S3 nhấn mạnh rằng miền tần số không mang lại giá trị phân loại đáng kể trong trường hợp này, trong khi các đặc trưng miền thời gian mới là nguồn thông tin then chốt để mô hình học được ranh giới phân lớp một cách chính xác và bền vững.

Kết quả này nhấn mạnh vai trò then chốt của lựa chọn đặc trưng đầu vào đối với hiệu quả mô hình. Các kịch bản sử dụng đặc trưng đã rút gọn theo SHAP (như S5–S8) vừa đạt độ chính xác cao, vừa giảm số chiều dữ liệu đầu vào, qua đó hỗ trợ triển khai mô hình nhẹ trong môi trường nhúng. Ngược lại, các kịch bản thiếu chọn lọc như S3 dẫn



Hình 3.12: Ma trận nhầm lẫn (confusion matrix) của năm mô hình phân loại trong kịch bản S1. GB và RF cho kết quả chính xác cao nhất. Mô hình NN được huấn luyện trực tiếp trên dữ liệu thô.

đến hiệu suất kém.

Bảng 3.5: Kích thước mô hình (KB) trong 8 kịch bản

Mô hình	S1	S2	S3	S4	S5	S6	S7	S8
LR	4	2	2	2	2	2	2	2
RF	187	151	291	176	89	89	141	103
SVM	315	232	3051	294	150	92	183	274
GB	605	602	615	603	587	587	587	589
NN	55	—	—	—	—	—	—	—

Bảng 3.5 cho thấy sự khác biệt đáng kể về kích thước giữa các mô hình. **Gradient Boosting (GB)** luôn duy trì dung lượng trên 580 KB bất kể kịch bản, cho thấy tính ổn định nhưng đồng thời cũng phản ánh hạn chế khi triển khai trên thiết bị nhúng có bộ nhớ giới hạn. **Random Forest (RF)** có kích thước biến thiên rõ rệt (89–291 KB) phụ thuộc vào số lượng và loại đặc trưng đầu vào, song vẫn duy trì ở mức chấp nhận được đối với các nền tảng nhúng có tài nguyên trung bình.

Ngược lại, **Logistic Regression (LR)** chỉ chiếm 2–4 KB, một dung lượng cực kỳ

nhỏ, khiến mô hình này trở thành ứng viên lý tưởng trong các hệ thống vi điều khiển hoặc IoT cần tối ưu bộ nhớ, mặc dù độ chính xác có phần thấp hơn so với các mô hình ensemble. **Support Vector Machine (SVM)** thường dao động trong khoảng 92–315 KB, tuy nhiên ở kịch bản S3 kích thước tăng vọt lên 3051 KB, nguyên nhân xuất phát từ số chiều đầu vào lớn và cấu trúc bộ nhớ của kernel. Điều này cho thấy SVM kém ổn định về mặt tài nguyên và khó kiểm soát khi triển khai thực tế.

Đáng chú ý, **Neural Network (NN)** ở kịch bản S1 đạt độ chính xác 0.92 với dung lượng chỉ 55 KB. Mặc dù chưa đạt hiệu năng tối ưu, kết quả này mở ra hướng tiếp cận tiềm năng cho các ứng dụng Edge AI, nơi sự cân bằng giữa hiệu năng và tính gọn nhẹ được đặt lên hàng đầu.

Tổng thể, Bảng 3.5 nhấn mạnh bài toán đánh đổi (trade-off) giữa *hiệu quả dự đoán* và *tài nguyên triển khai*. Trong khi RF và GB cho kết quả phân loại vượt trội về độ chính xác, thì LR và NN lại nổi bật nhờ kích thước gọn nhẹ, đặc biệt phù hợp với môi trường hạn chế tài nguyên. Sự đánh đổi này cho thấy rằng lựa chọn mô hình tối ưu không chỉ dựa vào độ chính xác thuần tuý, mà còn phụ thuộc vào yêu cầu hệ thống và ngữ cảnh triển khai cụ thể. Từ hai bảng kết quả có thể thấy rõ sự đánh đổi giữa *hiệu năng phân loại* và *tài nguyên triển khai*. Trong bối cảnh nghiên cứu hướng đến ứng dụng trên thiết bị nhúng và hệ thống IoT, các mô hình có kích thước gọn nhẹ nhưng vẫn duy trì độ chính xác ở mức cao cần được ưu tiên. Do đó, mặc dù **RF** và **GB** thể hiện hiệu năng vượt trội về mặt độ chính xác, luận án lựa chọn thử nghiệm chuyên sâu với **Logistic Regression (LR)** và **Neural Network (NN)**. Hai mô hình này có ưu điểm quan trọng: dung lượng bộ nhớ rất nhỏ (2–4 KB đối với LR, 55 KB đối với NN) và khả năng triển khai thuận lợi trên vi điều khiển, vốn thường chỉ có vài trăm kilobyte bộ nhớ khả dụng.

Một điểm cần lưu ý là các kịch bản S1–S5 được xây dựng với cửa sổ trượt 2 giây, trong khi các kịch bản S7 sử dụng cửa sổ 1 giây với mức chồng lấn 50%. Kết quả cho thấy việc giảm độ dài cửa sổ từ 2 giây xuống 1 giây không làm suy giảm đáng kể độ chính xác (thậm chí trong một số trường hợp, LR và GB còn cải thiện nhẹ, ví dụ LR đạt 0.990 ở S7 so với 0.987 ở S5). Việc lựa chọn cửa sổ 1 giây thay vì 2 giây có ý nghĩa thực tiễn quan trọng: nó cho phép hệ thống phản hồi nhanh hơn, nắm bắt kịp thời sự thay đổi tư thế, đồng thời giảm độ trễ trong nhận dạng – một yêu cầu thiết yếu khi triển khai trong môi trường y tế thực tế.

Tóm lại, mặc dù các mô hình ensemble như RF và GB đạt hiệu năng cao nhất về

mặt số liệu, luận án định hướng ưu tiên triển khai thử nghiệm với LR và NN kết hợp cửa sổ 1 giây. Đây là lựa chọn cân bằng hợp lý giữa *độ chính xác, tính gọn nhẹ của mô hình, và khả năng đáp ứng thời gian thực* khi triển khai trong các hệ thống nhúng hỗ trợ giám sát sức khỏe.

### 3.4. Triển khai trên vi điều khiển nhúng

Sau khi hoàn tất quá trình huấn luyện và đánh giá trên máy tính, bước tiếp theo của nghiên cứu là kiểm chứng khả năng triển khai mô hình trong môi trường thực tế sử dụng vi điều khiển nhúng. Mục tiêu khoa học ở giai đoạn này không chỉ dừng ở việc “chạy được” mô hình trên phần cứng hạn chế, mà còn nhằm làm sáng tỏ mối quan hệ đánh đổi giữa hiệu năng thuật toán và giới hạn tài nguyên của hệ thống nhúng.

Cụ thể, nghiên cứu tiến hành triển khai song song hai mô hình: mạng nơ-ron nông (Neural Network – NN) với khả năng biểu diễn phi tuyến mạnh mẽ, và hồi quy logistic (Logistic Regression – LR) với cấu trúc tuyến tính cực kỳ gọn nhẹ. NN được kỳ vọng duy trì độ chính xác cao trong phân loại tư thế ngủ, trong khi LR đóng vai trò như một đối chứng quan trọng, minh chứng cho khả năng đạt được sự cân bằng tối ưu giữa độ chính xác vừa đủ và mức tiêu thụ tài nguyên tối thiểu.

Điều này cho thấy trong bối cảnh phần cứng hạn chế, giá trị khoa học không nằm ở việc đạt độ chính xác tuyệt đối trong điều kiện lý tưởng, mà ở khả năng thiết kế một mô hình “đủ tốt” nhưng có thể vận hành bền vững trên chip. Chính sự đánh đổi này khẳng định nguyên lý cốt lõi của TinyML: hy sinh một phần nhỏ về độ chính xác để đổi lấy tính khả thi, hiệu quả năng lượng và độ tin cậy trong môi trường thực.

Kết quả cũng cho thấy sự song hành giữa hai mô hình được lựa chọn. Mạng nơ-ron (NN) duy trì độ chính xác cao nhưng tiêu tốn nhiều tài nguyên, trong khi hồi quy logistic (LR) có dung lượng siêu nhỏ, tốc độ suy luận nhanh, và vẫn giữ mức chính xác tiệm cận. Việc triển khai song song cả NN và LR trên chip vì vậy không chỉ mang ý nghĩa kiểm chứng kỹ thuật, mà còn cung cấp bằng chứng khoa học cho thấy ranh giới cân bằng giữa “độ chính xác tối đa” và “khả năng ứng dụng thực tế” trong hệ thống nhúng y sinh.

### 3.4.1. Quy trình triển khai mô hình

Dựa trên các kết quả phân tích và đánh giá ở giai đoạn 1, tác giả quyết định bước sang giai đoạn 2 với mục tiêu triển khai thực tế trên phần cứng nhúng. Như đã trình bày, hai mô hình được lựa chọn cho thử nghiệm là **Neural Network (NN)** và **Logistic Regression (LR)**, bởi chúng đáp ứng tốt yêu cầu về tính gọn nhẹ và khả năng triển khai trên vi điều khiển.

Quy trình triển khai bao gồm các bước sau:

1. **Xác định vi điều khiển mục tiêu:** lựa chọn nền tảng phần cứng phù hợp với giới hạn bộ nhớ và khả năng tính toán.
2. **Thu thập lại dữ liệu trực tiếp trên vi điều khiển:** đảm bảo dữ liệu phản ánh đúng điều kiện hoạt động của phần cứng thực tế, tránh sai lệch do khác biệt môi trường so với giai đoạn mô phỏng.
3. **Huấn luyện lại mô hình:** sử dụng bộ dữ liệu thu thập mới để tinh chỉnh và tái huấn luyện, nhằm tối ưu hóa mô hình cho nền tảng phần cứng được chọn.
4. **Chuyển đổi mô hình sang mã C/C++:** áp dụng các công cụ biên dịch và chuyển đổi chuyên dụng để xuất mô hình dưới dạng mã nguồn có thể nhúng trực tiếp.
5. **Triển khai trên chip:** nạp mã nguồn vào vi điều khiển, bộ lọc, kiểm tra khả năng suy luận và hiệu năng thời gian thực.

Vi điều khiển được lựa chọn là **Arduino Nano 33 BLE Sense**, sử dụng chip nRF52840 (ARM Cortex-M4F, 64 MHz), 1 MB flash và 256 KB RAM. Bo mạch này cũng tích hợp sẵn cảm biến tốc độ, rất phù hợp để xây dựng hệ thống nhận diện tư thế ngủ hoàn chỉnh và hoạt động độc lập.

Qua nhiều lần triển khai thực nghiệm trực tiếp trên vi điều khiển, tác giả đã rút ra một kết luận quan trọng: việc **giảm số lượng mẫu huấn luyện** kết hợp với **rút gọn tập đặc trưng** mang lại hiệu quả rõ rệt trong việc tối ưu mô hình cho môi trường nhúng. Cụ thể, các mô hình sau khi được tinh giản có *kích thước tệp nhỏ hơn, mức sử dụng bộ nhớ giảm đáng kể*, và *thời gian suy luận nhanh hơn*, nhưng độ chính xác chỉ suy giảm ở mức rất nhỏ và hoàn toàn nằm trong ngưỡng chấp nhận được đối với ứng dụng thực tế.

Điều này chứng tỏ rằng, trong bối cảnh triển khai Edge AI, sự đánh đổi giữa số lượng đặc trưng và hiệu năng mô hình có thể được cân bằng một cách hợp lý, từ đó vừa đảm bảo tính khả thi trên phần cứng hạn chế, vừa duy trì độ tin cậy trong dự đoán.

Đặc biệt, ở **bước 2 của quy trình**, tác giả đã tiến hành *thu thập lại 10.430 mẫu dữ liệu* trực tiếp từ thiết bị **Arduino Nano 33 BLE Sense**. Khác với bộ dữ liệu mô phỏng trên máy tính, tập dữ liệu này phản ánh sát thực tế điều kiện hoạt động của phần cứng, bao gồm cả đặc điểm nhiễu, độ trễ và sai số phép đo. Nhờ vậy, việc huấn luyện lại mô hình trên tập dữ liệu thực nghiệm giúp tăng tính tương thích giữa mô hình và nền tảng nhúng, giảm thiểu nguy cơ sai lệch do khoảng cách giữa môi trường mô phỏng và môi trường thực thi.

Trong quá trình triển khai thực tế, tác giả tiến hành hai hướng tiếp cận riêng biệt tương ứng với hai mô hình đã được lựa chọn từ giai đoạn 1: **Logistic Regression (LR)** và **Neural Network (NN)**.

- **Đối với LR:** sau khi huấn luyện lại mô hình trên tập dữ liệu thu thập từ Arduino Nano 33, toàn bộ tham số bao gồm trọng số, hệ số bias và giá trị chuẩn hoá (min, max, scale) được xuất trực tiếp sang mã C/C++. Cách tiếp cận này cho phép mô hình LR được biểu diễn dưới dạng các mảng hằng số `const float[]` trong chương trình Arduino, từ đó vi điều khiển có thể tính toán đầu ra bằng phép nhân ma trận và cộng bias đơn giản. Việc xuất mô hình theo phương thức này đảm bảo kích thước file rất nhỏ (chỉ vài kilobyte) và suy luận có thể thực hiện nhanh chóng mà không phụ thuộc vào thư viện học máy phức tạp.
- **Đối với NN:** mô hình được thiết kế với hai lớp ẩn (8 và 4 nơ-ron), sử dụng hàm kích hoạt ReLU và lớp đầu ra Softmax. Sau khi huấn luyện, mô hình được chuyển đổi sang định dạng **TensorFlow Lite (TFLite)** bằng công cụ TFLiteConverter. File nhị phân `.tflite` sau đó được ánh xạ sang mã C thông qua tiện ích `xxd -i`, tạo thành một mảng byte `const unsigned char[]` để nạp trực tiếp vào bộ nhớ của vi điều khiển. Cách tiếp cận này cho phép duy trì toàn bộ cấu trúc của mạng nơ-ron, đồng thời tận dụng khả năng tối ưu hoá suy luận của TensorFlow Lite trên nền tảng nhúng.

Sự khác biệt giữa hai phương pháp này phản ánh bản chất của từng mô hình: LR dựa trên phương trình tuyến tính với đặc trưng đã trích xuất, nên việc chuyển đổi tham

số sang C/C++ là tối ưu và gọn nhẹ nhất. Ngược lại, NN có cấu trúc phi tuyến và nhiều lớp, do đó cần sử dụng định dạng TFLite để đóng gói toàn bộ mô hình dưới dạng nhị phân, vừa đảm bảo tính toàn vẹn, vừa khai thác được khả năng tối ưu hoá suy luận trên chip.

Qua đó, luận án đã thiết lập được hai quy trình triển khai hoàn chỉnh: (i) LR với mô hình tuyến tính tối giản, phù hợp với các hệ thống nhúng cực kỳ hạn chế tài nguyên; (ii) NN với mô hình phi tuyến phức tạp hơn, tận dụng TFLite để cân bằng giữa độ chính xác và tốc độ suy luận.

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)**. Sau khi huấn luyện, mô hình Keras được chuyển đổi sang tệp nhị phân `.tflite` và sau đó nhúng trực tiếp vào chương trình Arduino dưới dạng mảng byte (`const unsigned char model[]`). Việc này cho phép vi điều khiển thực thi suy luận với sự hỗ trợ của thư viện TFLM, vốn đã được tối ưu hóa cho các hệ thống nhúng có bộ nhớ giới hạn.

Trong chương trình triển khai, bộ gia tốc của Arduino Nano 33 BLE Sense cung cấp dữ liệu ba trục ( $x, y, z$ ) liên tục. Tiếp đó dữ liệu được chuẩn hóa để bảo đảm sự tương thích với dải giá trị đầu vào mà mô hình đã được huấn luyện.

Khối MicroInterpreter trong TFLM chịu trách nhiệm phân bổ bộ nhớ, thực thi các toán tử (Dense, ReLU, Softmax), và trả về xác suất dự đoán cho từng lớp tư thế (*ngửa, nghiêng trái, nghiêng phải, sấp*). 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 thông qua khả năng biểu diễn phi tuyến, phù hợp cho các ứng dụng đòi hỏi độ chính xác cao và tính khái quát.

### 3.4.2. 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 (CPU):** Intel(R) Core(TM) Ultra 5 226V, tốc độ xung nhịp 2.10 GHz.
- **Bộ nhớ trong (RAM):** 16.0 GB DDR5, tốc độ 8533 MT/s.
- **Ổ lưu trữ (Storage):** SSD dung lượng 954 GB (trong đó đã sử dụng 188 GB).
- **Card đồ họa (GPU):** 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 3.6 trình bày sự so sánh chi tiết giữa Logistic Regression (LR) và Neural Network (NN) 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 *real-time* nơi độ trễ (latency) quyết định khả năng đáp ứng của hệ thống.

Bảng 3.6: So sánh hiệu năng Logistic Regression và Neural Network 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)
Độ phức tạp mô hình	Thấp (1 lớp tuyến tính)	Trung bình – cao (nhiều lớp fully connected)
Sức mạnh biểu diễn	Hạn chế (chỉ quan hệ tuyến tính)	Cao (biểu diễn quan hệ phi tuyến)
Ổn định trên MCU	Ổn định	Ổn định
Thời gian suy luận	$501 \mu s$	$8000 \mu s$

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 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).

## Hướng phát triển tương lai

Trong tương lai, hệ thống có thể được mở rộng theo hướng **tích hợp đa cảm biến** (multi-sensor fusion), thay vì chỉ dựa trên tốc kế đơn lẻ. Cụ thể, việc bổ sung thêm các cảm biến như *cảm biến nhịp tim (PPG/ECG)*, *cảm biến SpO<sub>2</sub>*, *cảm biến hô hấp (respiratory effort)*, hoặc *microphone áp suất âm thanh* sẽ cung cấp nhiều kênh dữ liệu sinh lý hơn, từ đó hỗ trợ phân tích toàn diện về chất lượng giấc ngủ.

Mục tiêu xa hơn là hướng tới **bài toán phát hiện sớm và theo dõi ngưng thở tắc nghẽn khi ngủ (Obstructive Sleep Apnea – OSA)**. Đặc trưng của OSA không chỉ thể hiện qua tư thế ngủ, mà còn gắn liền với các tín hiệu sinh lý như giảm oxy máu, sự thay đổi nhịp tim, và các chu kỳ hô hấp bất thường. Việc kết hợp dữ liệu tư thế ngủ với các chỉ số sinh lý học quan trọng có thể cho phép mô hình nhận diện các giai đoạn ngưng thở hoặc thở nông, vốn là dấu hiệu lâm sàng đặc trưng của OSA.

Hướng nghiên cứu này cũng mở ra những thách thức về **tối ưu hóa mô hình** để đảm bảo hệ thống vẫn vận hành được trên phần cứng nhúng giới hạn tài nguyên. Điều này đòi hỏi chiến lược như *quantization*, *pruning*, *knowledge distillation*, hoặc thậm chí triển khai các kiến trúc chuyên biệt như *TinyCNN* hay *RNN nhẹ* nhằm xử lý dữ liệu chuỗi thời gian đa kênh. Do đó, nghiên cứu trong giai đoạn tiếp theo không chỉ tập trung vào độ chính xác mô hình, mà còn cần chú trọng đến khả năng triển khai thực tiễn trong môi trường chăm sóc sức khỏe từ xa (telehealth) và thiết bị đeo thông minh (wearable devices).

## Chương 4

# Kết luận

### Hướng phát triển trong thời gian tới

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 (lên tới 99.6% với Gradient Boosting, 98.7% với Logistic Regression). Đồng thời, một mô hình mạng nơ-ron nồng (NN) cũng đã được triển khai thành công trên vi điều khiển Arduino Nano 33 BLE Sense, chứng minh tính khả thi của việc chạy mô hình trực tiếp trên phần cứng nhúng.

Trong giai đoạn tiếp theo, nhóm nghiên cứu sẽ tập trung vào việc:

- **Tối ưu mô hình:** Tiếp tục khai thác Logistic Regression do ưu thế về kích thước nhỏ gọn và tốc độ suy luận, đồng thời thử nghiệm các kiến trúc học sâu nhẹ (lightweight deep learning) như CNN, MobileNet hoặc TinyML framework nếu điều kiện phần cứng cho phép;
- **Phát triển phần cứng:** Tự thiết kế mạch nguyên lý và PCB trên Altium, tích hợp cảm biến, vi điều khiển, module truyền không dây và khối xử lý tín hiệu nhằm xây dựng thiết bị chuyên biệt thay vì phụ thuộc vào bo mạch thương mại;
- **Mở rộng tín hiệu:** Bổ sung thêm các cảm biến sinh lý (như microphone phát hiện ngáy, cảm biến nhịp thở, nhịp tim, SpO<sub>2</sub>) để từng bước hướng tới đánh giá chỉ số AHI (Apnea–Hypopnea Index) – thước đo quan trọng trong chẩn đoán hội chứng ngưng thở khi ngủ (OSA);

- **Cải tiến phần mềm:** Bổ sung kết nối Wi-Fi/BLE Mesh để dữ liệu được gửi trực tiếp lên server, hạn chế việc phụ thuộc vào ứng dụng di động phải chạy liên tục, từ đó nâng cao trải nghiệm người dùng và tiết kiệm năng lượng.

Ngoài ra, một trong những ưu tiên quan trọng là **xây dựng bộ dữ liệu huấn luyện có độ tin cậy cao**. Nhóm nghiên cứu dự kiến triển khai các phương pháp gán nhãn tự động và bán tự động, bao gồm:

- Ghi hình kết hợp đồng bộ thời gian với dữ liệu cảm biến;
- Gán nhãn theo khoảng thời gian định trước;
- Gán nhãn thủ công trực tiếp trên ứng dụng di động thông qua nút bấm hỗ trợ.

Ứng dụng di động trong tương lai sẽ được bổ sung tính năng gán nhãn và quản lý dữ liệu tập trung, hỗ trợ quá trình thu thập và huấn luyện mô hình.

## Kết luận định hướng

Mục tiêu dài hạn của nghiên cứu không chỉ dừng lại ở việc nhận diện tư thế ngủ, mà còn hướng đến phát triển một hệ thống **Home Sleep Testing (HST)** đơn giản, chi phí thấp, có khả năng **sàng lọc nguy cơ mắc hội chứng ngưng thở khi ngủ tắc nghẽn (OSA)** ngay tại nhà.

Các định hướng cụ thể gồm:

- Tích hợp đa cảm biến để giám sát đồng thời tư thế, nhịp thở, âm thanh và SpO<sub>2</sub>;
- Phát triển thuật toán ước lượng chỉ số AHI dựa trên dữ liệu tổng hợp;
- Thiết kế giao diện giám sát từ xa dành cho bác sĩ, kết hợp đề xuất can thiệp lâm sàng;
- Tối ưu và lượng tử hóa mô hình học máy để triển khai trực tiếp trên vi điều khiển hoặc thiết bị đeo tay.

Với định hướng này, sản phẩm kỳ vọng trở thành một giải pháp HST gọn nhẹ, dễ tiếp cận và có tính ứng dụng cao, góp phần hỗ trợ bệnh nhân và các cơ sở y tế trong công

tác **sàng lọc sớm, theo dõi và quản lý OSA** tại cộng đồng, đặc biệt ở các khu vực còn hạn chế về thiết bị PSG chuẩn.

# 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. *inJAMA*: 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. *inISRN Otolaryngology*: (2013), **page** 670381. DOI: 10.1155/2013/670381.
  - [8] Rosalind Dymond Cartwright. Effect of Sleep Position on Sleep Apnea Severity. *inSleep*: 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. *inCurrent Sleep Medicine Reports*: 6.4 (2020), **pages** 309–321. DOI: 10.1007/s40675-020-00197-5.
  - [10] 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. *inIEEE Journal of Biomedical and Health Informatics*: 17.6 (2013), **pages** 985–993. DOI: 10.1109/JBHI.2013.2252911.
  - [11] X. Nam, Y. Kim **and** J. Lee. Sleep monitoring based on a tri-axial accelerometer and a pressure sensor. *inSensors*: 16.5 (2016), **page** 750.
  - [12] M. Enayati, M. Skubic, J. M. Keller, M. Popescu **and** N. Z. Farahani. ?Sleep posture classification using bed sensor data and neural networks? *inProceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*: july 2018, **pages** 461–465.
  - [13] 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? *inProceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*: july 2019, **pages** 3115–3118.
  - [14] 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.

- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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? *inProceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*: august 2015, **pages** 4942–4945.
- [19] 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>.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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 Respi-

- ratory 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>.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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>.
- [29] 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.
- [30] 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.

- [31] 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>.
- [32] 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>.
- [33] 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.
- [34] 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.
- [35] D. Cary, A. Jacques **and** K. Briffa. Examining relationships between sleep posture, waking spinal symptoms and quality of sleep: A cross sectional study. *inPLoS 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.
- [36] 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>.
- [37] 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>.

- [38] R. D. Cartwright. Effect of Sleep Position on Sleep Apnea Severity. *inSleep*: 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>.
- [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? *in2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*: Austin, TX, USA, 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 Yuze. 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 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*: 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. **in***Scientific 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. **in***Procedia 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. **in***IEEE 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. **in***Entropy (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. **in***CMES - 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. **in***IEEE 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.
- [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. 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.

- [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. **in***Sensors*: 18.11 (2018), **page** 3641. DOI: 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. **in***Micromachines*: 9.4 (2018), **page** 178. DOI: 10 . 3390/mi9040178.
- [77] J. Y. Hao **and** W. Bai. Mechanics analysis of cantilever beam with uniform load. **in***Advanced 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. **in***Sensors*: 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. **in***Electronics 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. **in***Water*: 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. **in***IEEE 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. **in***IEEE 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. **in***Sensors 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. **in***arxiv*: (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.compbiomed.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.
- [95] O. Elnaggar, R. Arelihi, 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.
- [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.
- [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.
- [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] N. Djermanova M. B. Marinov B. Ganev **and** T. D. Tashev. Analysis of Sensors Noise Performance Using Allan Deviation. *inIEEE XXVIII International Scientific Conference Electronics (ET)*: 7 (2019). DOI: 10.1109/ET.2019.8878552..
- [108] J. S. Cramer. *The Origins of Logistic Regression*. Tinbergen Institute Discussion Paper TI 2001-119/4. Available at SSRN: <https://doi.org/10.2139/ssrn.360300>. Tinbergen Institute, **january** 2002. DOI: 10.2139/ssrn.360300. URL: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=360300](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=360300).
- [109] Corinna Cortes **and** Vladimir Vapnik. Support-Vector Networks. *inMachine Learning*: 20.3 (1995), **pages** 273–297. ISSN: 0885-6125. DOI: 10.1007/BF00994018. URL: <https://doi.org/10.1007/BF00994018>.
- [110] Leo Breiman. Random Forests. *inMachine Learning*: 45.1 (2001), **pages** 5–32. DOI: 10.1023/A:1010933404324. URL: <https://doi.org/10.1023/A:1010933404324>.
- [111] Tianqi Chen **and** Carlos Guestrin. ?XGBoost: A Scalable Tree Boosting System? *inProceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*: ACM, 2016, **pages** 785–794. DOI: 10.1145/2939672.2939785. URL: <https://doi.org/10.1145/2939672.2939785>.
- [112] Anil K. Jain, Jianchang Mao **and** K. M. Mohiuddin. Artificial neural networks: a tutorial. *inComputer*: 29.3 (**march** 1996), **pages** 31–44. DOI: 10.1109/2.485891.
- [113] Yann LeCun, Yoshua Bengio **and** Geoffrey Hinton. Deep learning. *inNature*: 521.7553 (**may** 2015), **pages** 436–444. DOI: 10.1038/nature14539. URL: <https://www.nature.com/articles/nature14539>.