

**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:** Ngưng thở tắc nghẽn khi ngủ (Obstructive Sleep Apnea – OSA) là một rối loạn hô hấp khi ngủ do sự hẹp hoặc tắc nghẽn một phần hay toàn bộ đường hô hấp trên, bao gồm vùng mũi họng, hầu họng hoặc cả hai. Tình trạng này được đặc trưng bởi các cơn ngưng thở hoặc giảm thở ngắn, lặp đi lặp lại trong khi ngủ, gây gián đoạn giấc ngủ do thức giấc thường xuyên và dẫn đến giảm oxy máu từng đợt[1]. Nghiên cứu[2] đã phân tích dữ liệu từ 15 nghiên cứu với tổng cộng 42.924 người. Kết quả cho thấy tỷ lệ mắc chứng buồn ngủ ban ngày quá mức (EDS) ở bệnh nhân ngưng thở tắc nghẽn khi ngủ (OSA) trên toàn cầu là 39,9% (khoảng tin cậy 95%: 34,4%–45,7%).

Tư thế ngủ đóng vai trò quan trọng trong việc khởi phát và làm trầm trọng thêm các triệu chứng của OSA. Đặc biệt, tư thế nằm ngửa thường làm tăng mức độ tắc nghẽn đường hô hấp so với các tư thế khác, dẫn đến dạng OSA phụ thuộc tư thế (positional OSA). Trong bối cảnh đó, việc theo dõi chính xác tư thế ngủ trong thời gian thực có thể cung cấp thông tin quan trọng phục vụ chẩn đoán sớm, đánh giá nguy cơ và cải thiện tình trạng OSA bằng việc thay đổi tư thế ngủ.

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 đã trở thành một yếu tố then chốt trong quá trình tích hợp chúng vào các thiết bị điện tử có kích thước nhỏ gọn. Không chỉ góp phần nâng cao độ chính xác trong việc đo lường các thông số sinh lý quan trọng, nó còn giúp giảm kích thước thiết bị, tăng tính di động và mở rộng khả năng ứng dụng trong nhiều lĩnh vực, đặc biệt nổi bật trong y học cá thể hóa.

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. Trong bài toán chẩn đoán hội chứng ngưng thở khi ngủ, sự phối hợp giữa dữ liệu từ cảm biến và thuật toán học máy không chỉ đảm bảo độ chính xác cao trong thu thập dữ liệu mà còn mở ra khả năng phân tích chuyên sâu về các yếu tố sinh lý, phục vụ quá trình đánh giá toàn diện. Đặc biệt, sự nổi lên của lĩnh vực học máy triển khai trực tiếp trên các vi điều khiển hoặc thiết bị biên có tài nguyên hạn chế (Tiny Machine Learning - TinyML) đã

đánh dấu bước tiến quan trọng trong việc hiện thực hóa các hệ thống giám sát sức khỏe thuận tiện, tiết kiệm năng lượng, chi phí thấp và có khả năng hoạt động độc lập không phụ thuộc vào kết nối mạng hoặc nền tảng đám mây.

Trên cơ sở đó, luân văn này tập trung nghiên cứu, phát triển hệ thống thu thập, mô hình học máy biên nhầm phân loại tư thế ngủ, sử dụng đơn dữ liệu cảm biến gia tốc. Luận văn được thực hiện qua hai giai đoạn chính. Giai đoạn thứ nhất tập trung vào việc nghiên cứu và phát triển hệ thống thu thập, xử lý, lưu trữ dữ liệu cảm biến; đồng thời tiến hành phân tích, trích xuất đặc trưng và huấn luyện, đánh giá một số mô hình học máy phù hợp cho bài toán phân loại tư thế ngủ. Giai đoạn thứ hai tập trung vào nghiên cứu và phát triển phần cứng riêng, cũng như triển khai mô hình học máy đã lựa chọn lên nền tảng phần cứng đó.

**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>	viii
<b>Danh sách bảng</b>	ix
<b>Đặt vấn đề</b>	1
<b>Chương 1 Hội chứng ngưng thở khi ngủ do tắc nghẽn (OSA): Ảnh hưởng của tư thế ngủ và các tiến bộ công nghệ trong giám sát tư thế ngủ tự động</b>	7
1.1. Hội chứng ngưng thở khi ngủ . . . . .	8
1.2. Ứng dụng cảm biến gia tốc trong đánh giá tư thế ngủ của người mắc OSA tại nhà . . . . .	15
1.2.1. Quy trình tổng quát trong hệ thống ứng dụng AI cho đánh giá OSA và phân loại tư thế ngủ . . . . .	27
<b>Chương 2 XÂY DỰNG HỆ THỐNG THU THẬP DỮ LIỆU, HUẤN LUYỆN MÔ HÌNH VÀ CHUẨN BỊ TRIỂN KHAI TRÊN THIẾT BỊ BIÊN.</b>	40
2.1. Phần cứng thực nghiệm . . . . .	40
2.1.1. Cảm biến . . . . .	41
2.1.2. Vị xử lý . . . . .	46
2.1.3. Bluetooth năng lượng thấp . . . . .	51

2.1.4. Thiết bị thực nghiệm . . . . .	54
2.2. Hệ thống thu thập, xử lý, lưu trữ dữ liệu . . . . .	56
2.2.1. Lập trình vi xử lý . . . . .	56
2.2.2. Hiệu chuẩn cảm biến . . . . .	62
2.2.3. Hiệu chuẩn cảm biến . . . . .	63
2.2.4. Xây dựng phần mềm ứng dụng . . . . .	66
2.2.5. Thiết kế và xây dựng hệ thống lưu trữ . . . . .	73
2.2.6. Học máy trong phân loại tư thế ngủ . . . . .	79
<b>Chương 3 KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ</b>	<b>88</b>
3.1. Hệ thống thực nghiệm . . . . .	88
3.2. Thu thập và gắn nhãn dữ liệu . . . . .	93
3.3. Phân loại tư thế ngủ bằng học máy . . . . .	95
3.3.1. Phân tích dữ liệu . . . . .	95
3.3.2. Xử lý và trích xuất đặc trưng . . . . .	97
3.3.3. Kịch bản kiểm thử và lựa chọn tính năng . . . . .	98
3.3.4. Huấn luyện mô hình . . . . .	103
3.3.5. Đánh giá kết quả . . . . .	105
3.4. Triển khai trên vi điều khiển nhúng . . . . .	109
3.4.1. Quy trình triển khai mô hình . . . . .	110
3.4.2. Đánh giá hiệu suất và tài nguyên . . . . .	113
<b>Chương 4 Kết luận</b>	<b>116</b>

# Danh sách hình vẽ

1.1	Các tư thế ngủ cơ bản của con người . . . . .	16
1.2	Vị trí tối ưu để đặt cảm biến gia tốc . . . . .	18
1.3	Nguyên lý cơ bản của cảm biến gia tốc . . . . .	22
1.4	Cấu trúc cảm biến gia tốc điện dung . . . . .	23
1.5	Cấu trúc cảm biến áp điện trở . . . . .	24
1.6	Cấu trúc cảm biến áp điện . . . . .	25
2.1	Cảm biến gia tốc LIS3DH và sơ đồ chân kết nối . . . . .	46
2.2	Thành phần chính của vi điều khiển Cortex-M4 . . . . .	49
2.3	Nordic Semiconductor NRF52840 . . . . .	50
2.4	Các kiểu kết nối không dây trong họ chip nRF52 . . . . .	52
2.5	Cấu trúc của GATT . . . . .	54
2.6	Cấu trúc các thành phần trên Circuit Playground . . . . .	56
2.7	Lưu đồ luồng gửi thông tin BLE . . . . .	61
2.8	Minh họa kết quả phân tích đường cong Allan . . . . .	62
2.9	Biểu đồ phương sai Allan của trục X . . . . .	63
2.10	Minh họa kết quả phân tích đường cong Allan . . . . .	64
2.11	Biểu đồ phương sai Allan của trục X . . . . .	65
2.12	Kết quả bộ lọc Kalman cho dữ liệu trục X của cảm biến gia tốc . . . . .	67
2.13	Cấu trúc kiến trúc BLoC trong ứng dụng Flutter . . . . .	70

2.14	Mô hình tích hợp giữa mạng cảm biến và cấu trúc dữ liệu đám mây . . . . .	74
2.15	Lưu đồ thuật toán lưu trữ dữ liệu cảm biến . . . . .	78
2.16	Phân bố thời gian thực hiện đối với dự án học máy . . . . .	79
2.17	Minh họa mặt siêu phẳng phân tách hai lớp trong SVM . . . . .	82
2.18	Chiến lược phân loại đa lớp bằng phương pháp One-vs-One . . . . .	83
2.19	Chiến lược phân loại đa lớp bằng phương pháp One-vs-Rest . . . . .	83
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. . . . .	89
3.2	Giao diện chức năng đăng ký và đăng nhập . . . . .	90
3.3	Giao diện trang chủ . . . . .	90
3.4	Giao diện màn hình danh sách BLE và chi tiết các dịch vụ kết nối với phần cứng . . . . .	91
3.5	Giao diện chức năng chatbot và bộ câu hỏi tầm soát . . . . .	92
3.6	Giao diện hiển thị dữ liệu gia tốc ba trục . . . . .	93
3.7	Mô phỏng thực nghiệm thực tế . . . . .	94
3.8	Phân bố dữ liệu cảm biến theo ba trục $x$ , $y$ , $z$ ứng với các tư thế ngủ khác nhau. . . . .	96
3.9	Phân bố số lượng mẫu trong tập huấn luyện theo từng tư thế. . . . .	97
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. . . . .	101
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. . . . .	103
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. . . . .	107

# Danh sách bảng

1.1	Phân loại mức độ OSA dựa trên chỉ số AHI . . . . .	10
1.2	Phân loại thiết bị đeo trong phát hiện OSA và tài liệu tham khảo liên quan	13
1.3	So sánh ưu và nhược điểm của các vị trí gắn cảm biến gia tốc trong theo dõi tư thế ngủ . . . . .	19
1.4	Các bước chính trong hệ thống ứng dụng AI cho phân tích tư thế ngủ và hỗ trợ chẩn đoán OSA . . . . .	28
1.5	Hiệu quả mô hình học máy và khả năng triển khai biên trong nhận diện tư thế ngủ . . . . .	37
2.1	Các nhóm chức năng chính của ứng dụng . . . . .	69
3.1	Các đặc trưng thống kê và tín hiệu được sử dụng trong phân loại tư thế ngủ	99
3.2	Các kịch bản lựa chọn và sử dụng đặc trưng trong nghiên cứu . . . . .	102
3.3	Các mô hình học máy và siêu tham số sử dụng trong nghiên cứu . . . . .	104
3.4	Độ chính xác phân loại của các mô hình trong 8 kịch bản . . . . .	105
3.5	Kích thước mô hình (KB) trong 8 kịch bản . . . . .	107
3.6	So sánh hiệu năng Logistic Regression và Neural Network trên nRF52840	114

# Đặt vấn đề

OSA là một rối loạn hô hấp khi ngủ do sự hẹp hoặc tắc nghẽn một phần hay toàn bộ đường hô hấp trên, bao gồm vùng mũi họng, hầu họng hoặc cả hai. Tình trạng này được đặc trưng bởi các cơn ngưng thở hoặc giảm thở ngắn, lặp đi lặp lại trong khi ngủ, gây gián đoạn giấc ngủ do thức giấc thường xuyên và dẫn đến giảm oxy máu từng đợt. Tỷ lệ hiện mắc OSA tại Việt Nam ước tính khoảng 8,5% [3]. OSA hiện được công nhận là một yếu tố có nguy cơ độc lập đối với nhiều bệnh lý liên quan đến tim mạch, đặc biệt là tăng huyết áp. Ngoài ra, hội chứng này còn có mối liên hệ đáng kể với các nguy cơ như tai nạn giao thông, tai nạn lao động dẫn đến làm gia tăng gánh nặng kinh tế xã hội[4][5]. Đáng chú ý, tình trạng ngưng thở khi ngủ kéo dài và không được phát hiện, điều trị có thể ảnh hưởng nghiêm trọng đến sức khỏe gây ra rối loạn nhịp tim và một trong những nguyên nhân gây đột tử [6]. Theo PSG.TS Nguyễn Thy Khuê, Hội Y học Giấc ngủ Việt Nam, ngưng thở khi ngủ còn là một trong những dấu hiệu rõ ràng của bệnh đái tháo đường, bệnh thận. OSA được phát hiện ở hơn 20% người bệnh đái tháo đường và làm trầm trọng thêm các rối loạn chuyển hóa, đặc biệt là đái tháo đường type 2. Trong một nghiên cứu tiêu biểu tại Việt Nam, GS.TS.BS. Dương Quý Sỹ và cộng sự đã khảo sát 524 trẻ em mắc rối loạn tăng động giảm chú ý (Attention Deficit Hyperactivity Disorder - ADHD) tại Bệnh viện Nhi Trung ương Việt Nam [7]. 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.

Tư thế ngủ được xem là một trong những yếu tố quan trọng có thể làm trầm trọng thêm tình trạng ngưng thở khi ngủ, đặc biệt là ở tư thế nằm ngửa. Dạng đặc biệt của hội chứng này được gọi là ngưng thở khi ngủ do tư thế (Positional Obstructive Sleep Apnea – pOSA). Theo tiêu chuẩn chẩn đoán, bệnh nhân được xác định mắc pOSA khi chỉ số

Apnea–Hypopnea Index (AHI) lớn hơn 5, đồng thời giá trị AHI ở tư thế nằm ngửa cao gấp ít nhất hai lần so với khi nằm ở các tư thế khác [8]. Các nghiên cứu gần đây cho thấy tỷ lệ hiện mắc pOSA chiếm tới khoảng 50% trong tổng số bệnh nhân OSA [9]. Phát hiện này nhấn mạnh rằng tư thế ngủ có ảnh hưởng sinh lý rõ rệt đến mức độ xẹp của đường hô hấp trên, đặc biệt ở tư thế nằm ngửa, làm gia tăng nguy cơ tắc nghẽn đường thở.

Việc chẩn đoán (OSA) hiện nay chủ yếu được thực hiện thông qua hai phương pháp: đa ký giấc ngủ (Polysomnography – PSG) và thiết bị kiểm tra giấc ngủ tại nhà (Home Sleep Test – HST). Trong đó, PSG được xem là tiêu chuẩn vàng trong việc đánh giá OSA. Nó thường được thực hiện tại bệnh viện hoặc trung tâm giấc ngủ. Thông thường, bệnh nhân sẽ được yêu cầu đến vào buổi tối để trung tâm có thể ghi lại và theo dõi giấc ngủ ban đêm của bệnh nhân. Đôi khi, PSG được thực hiện vào ban ngày để phù hợp với những người làm việc theo ca, những người đã quen với việc ngủ ban ngày. Quá trình đo PSG phải được thực hiện dưới sự giám sát trực tiếp của bác sĩ chuyên ngành giấc ngủ hoặc kỹ thuật viên có chuyên môn[10][11]. Trong khi bạn ngủ, kỹ thuật viên sẽ theo dõi sóng não, chuyển động mắt, nhịp tim, nhịp thở, nồng độ oxy trong máu, tư thế và chuyển động cơ thể, tiếng ngáy và các tiếng ồn khác mà bạn có thể phát ra khi ngủ. Tất cả các phép đo này sẽ được ghi lại trên biểu đồ. Trong quá trình nghiên cứu, kỹ thuật viên có thể yêu cầu bạn sử dụng máy thở áp lực dương (PAP) nếu bạn có dấu hiệu ngưng thở khi ngủ. PAP là một thiết bị mặt nạ cung cấp không khí với mức oxy phù hợp thông qua miệng hoặc ống mũi để hỗ trợ trong trường hợp khó thở.

Mặc dù PSG vẫn giữ vai trò là phương pháp tham chiếu trong chẩn đoán và theo dõi chất lượng giấc ngủ cũng như các rối loạn liên quan, nhưng việc triển khai kỹ thuật này thường đòi hỏi chi phí cao, trang thiết bị chuyên dụng và điều kiện thực hiện tại các cơ sở y tế chuyên khoa. Một thách thức khác của PSG là người bệnh thường cảm thấy bất tiện và cảm giác khó chịu do mang nhiều cảm biến gắn trên cơ thể trong suốt đêm, dẫn đến nguy cơ gián đoạn hoặc sai lệnh dữ liệu trong quá trình ghi nhận. Vì vậy, các hệ thống HST đang ngày càng thu hút sự quan tâm từ cộng đồng khoa học toàn cầu[12][13][14]. Các thiết bị HST hiện đại dùng các cảm biến không xâm lấn nhằm ghi nhận và phân tích một số tín hiệu sinh lý cơ bản như luồng khí hô hấp, tư thế ngủ, áp suất mũi, độ bão hòa oxy và nhịp tim. Việc cải thiện chất lượng, kéo dài thời lượng sử dụng và tăng độ chính xác và cải thiện mức độ thoái mái vẫn là những thách thức lớn đối với giới nghiên cứu và các nhà sản xuất thiết bị y tế. Tuy nhiên, với những tiến bộ công nghệ đang diễn ra

nhanh chóng, HST có tiềm năng trở thành một công cụ chẩn đoán quan trọng và được ứng dụng rộng rãi trong lâm sàng. Điều này không chỉ mang lại sự thuận tiện và chấp nhận cao hơn từ phía người bệnh, mà còn góp phần làm giảm gánh nặng chi phí và áp lực cho hệ thống chăm sóc sức khỏe.

Trong những năm gần đây, nhiều nhóm nghiên cứu đang chú trọng nghiên cứu, phát triển hệ thống HST nhằm mục đích thay thế hoặc hỗ trợ cho đa ký giấc ngủ. Năm 2011, Collop và cộng sự đã phát triển một hệ thống phân loại SCOPER (Sleep, Cardiovascular, Oximetry, Position, Effort, and Respiration) để đánh giá các tín hiệu sinh lý thu nhận trong việc chẩn đoán OSA[15]. Tác giả Morillo D và cộng sự đề xuất một phương pháp sàng lọc ngưng thở tắc nghẽn khi ngủ dựa trên cảm biến gia tốc gắn tại vị trí hõm ức, cho phép trích xuất các tín hiệu hô hấp, tim mạch và tiếng ngáy bằng kỹ thuật xử lý tín hiệu số, từ đó chứng minh tính khả thi của thiết bị di động đơn giản và chi phí thấp trong hỗ trợ chẩn đoán hội chứng ngưng thở – giảm thở khi ngủ[16]. Trong nhóm các thiết bị đeo, A.H. Yüzer và cộng sự đã phát triển thiết bị đeo tay sử dụng cảm biến gia tốc ADXL345 để phát hiện và phát tín hiệu rung khi cảnh báo. [17]. Tương tự, 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) [18]. Tại Việt Nam, nhóm nghiên cứu của Giáo sư Lê Tiến 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 [19].

Trong các phương pháp tiếp cận hiện nay, phần lớn đều ứng dụng các mô hình học máy nhằm khai thác hiệu quả tín hiệu sinh lý (như ECG, PPG và accelerometer) để phát hiện các sự kiện hô hấp bất thường cũng như phân loại mức độ nghiêm trọng của OSA một cách tự động, nhanh chóng và chính xác [20]. Trong số các thuật toán học máy truyền thống, Random Forest (RF) đã được chứng minh là một trong những phương pháp hiệu quả nhất nhờ khả năng kháng quá khớp và độ chính xác cao, với độ chính xác trên 93% trong nhiều nghiên cứu [21, 22]. SVM, KNN và LDA cũng được sử dụng phổ biến nhờ khả năng phân loại mạnh mẽ và khả năng thích ứng với nhiều loại dữ liệu sinh lý [23, 24, 25]. Bên cạnh đó, các thuật toán tiên tiến như XGBoost đã được triển khai để tăng tốc huấn luyện và cải thiện hiệu suất trong các hệ thống học máy gọn nhẹ (TinyML)

phục vụ chẩn đoán lâm sàng [26].

Bên cạnh các thuật toán học máy truyền thống, các mô hình học sâu (deep learning) như mạng nơ-ron nhân tạo (ANN), mạng nơ-ron tích chập (CNN), và mạng nơ-ron sâu (DNN) đã cho thấy hiệu quả vượt trội trong việc xử lý tín hiệu sinh lý phức tạp và nhận diện sự kiện hô hấp liên quan đến OSA. Nhiều nghiên cứu đã ứng dụng các kiến trúc CNN một chiều (1D-CNN), mạng kết hợp CNN–RNN, và mạng tích chập đa tầng để phát hiện nhịp thở bất thường và ước lượng chỉ số AHI với độ chính xác có thể đạt trên 90% [27, 28]. Các mô hình này tận dụng khả năng tự động trích chọn đặc trưng từ dữ liệu cảm biến thô, hỗ trợ phát hiện OSA theo thời gian thực và tối ưu hóa hiệu năng. Những kết quả này khẳng định tiềm năng lớn của học sâu trong việc xây dựng các hệ thống giám sát giấc ngủ thế hệ mới, đặc biệt là trên các nền tảng nhúng và thiết bị đeo thông minh.

Còn đối với bài toán phân loại tư thế ngủ ở người, tác giả Xi Xu và cộng sự đã đề xuất một mô hình học máy stacking cải tiến, kết hợp ba thuật toán mạnh (XGBoost, SVM, DNDT) và tối ưu bằng Bayesian optimization cùng phương pháp entropy weighting để nhận dạng tư thế ngủ dựa trên dữ liệu áp suất từ nệm khí [29]. Kết quả thực nghiệm cho thấy mô hình đạt độ chính xác 94.48%, vượt trội hơn các mô hình đơn lẻ và có tiềm năng ứng dụng trong các hệ thống giám sát giấc ngủ thông minh. Trong một nghiên cứu khác, tác giả giới thiệu một thiết bị đeo di động sử dụng cảm biến gia tốc đơn để nhận diện mười hai tư thế ngủ, hỗ trợ bệnh nhân GERD cải thiện thói quen và chất lượng giấc ngủ tại nhà [30]. Mô hình AnpoNet kết hợp 1D-CNN và LSTM, đạt độ chính xác 94,67% và F1-score 92,94%, cho thấy tiềm năng ứng dụng thực tiễn trong giám sát tư thế ngủ không xâm lấn.

Với mục tiêu triển khai hệ thống trên thiết bị đeo có tài nguyên tính toán hạn chế, việc áp dụng các thuật toán học máy nhẹ (TinyML) đóng vai trò then chốt. Khác với các mô hình học sâu truyền thống yêu cầu phần cứng mạnh và tiêu tốn năng lượng, TinyML cho phép thực thi mô hình trực tiếp trên vi điều khiển có dung lượng bộ nhớ và tốc độ xử lý thấp, đồng thời đảm bảo độ trễ tối thiểu và khả năng hoạt động ngoại tuyến. Đặc điểm này đặc biệt phù hợp cho các ứng dụng y sinh tại nhà, nơi thiết bị cần hoạt động liên tục, chi phí thấp và không phụ thuộc vào kết nối mạng.

Tiny machine learning tập trung vào việc giảm thiểu số lượng tham số mô hình và độ phức tạp tính toán, nhờ đó trở thành một hướng tiếp cận đầy hứa hẹn trong các ứng

dụng chăm sóc sức khỏe thông minh [31, 32]. Tuy nhiên, vẫn tồn tại sự đánh đổi giữa việc đơn giản hóa mô hình và duy trì độ chính xác trong phân loại. Bài toán đặt ra là làm sao tối ưu hóa đồng thời kích thước mô hình, tốc độ xử lý và hiệu suất trong việc phân loại tình trạng OSA, nhận diện tư thế, đảm bảo hệ thống vừa đủ nhẹ để chạy trên phần cứng nhúng, vừa đủ chính xác để có giá trị trong giám sát y tế.

Các nghiên cứu trước đây đã chứng minh hiệu quả của học máy trong phân loại tư thế ngủ và phát hiện OSA, tuy nhiên phần lớn các hệ thống vẫn phụ thuộc vào dữ liệu đa kênh phức tạp, mô hình tính toán nặng, hoặc cần xử lý tập trung trên máy chủ. Điều này gây hạn chế khi triển khai thực tế tại nhà hoặc trên thiết bị đeo. Hơn nữa, vẫn còn thiếu các giải pháp tích hợp đầy đủ từ thu thập, xử lý, đến phân loại tín hiệu ngay trên vi điều khiển với độ trễ thấp và khả năng hoạt động độc lập không cần kết nối mạng. Qua đó, luận văn này đề xuất một hướng tiếp cận sử dụng TinyML tại biên, nhằm xây dựng một hệ thống theo dõi tư thế ngủ, có tiềm năng hỗ trợ sàng lọc pOSA theo thời gian thực. Thiết bị được thiết kế theo hướng đeo được, nhỏ gọn, chi phí thấp, cho phép ghi nhận và phân loại chính xác các tư thế ngủ: nằm ngửa, nằm sấp, nằm nghiêng trái và phải. Hệ thống tích hợp khả năng thu thập – xử lý – phân loại ngay trên vi điều khiển, hướng tới độ chính xác cao, độ trễ thấp và khả năng triển khai tại nhà với chi phí hợp lý. Tuy chưa thực hiện chức năng sàng lọc OSA, hệ thống được xây dựng như một nền tảng kỹ thuật tiềm năng, phục vụ cho các nghiên cứu ứng dụng trong tương lai, đặc biệt là hỗ trợ đánh giá nguy cơ pOSA dựa trên tư thế ngủ.

Mục tiêu cụ thể của khóa luận 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 ba chương chính như sau:

**Chương 1:** Hội chứng ngưng thở khi ngủ do tắc nghẽn (OSA): Ảnh hưởng của tư thế ngủ và các tiến bộ công nghệ trong giám sát tư thế ngủ tự động

**Chương 2:** Xây dựng hệ thống thu thập dữ liệu, huấn luyện, đánh giá mô hình học máy.

**Chương 3:** Xây dựng phần cứng, triển khai tại biên.

## **Chương 1**

# **Hội chứng ngưng thở khi ngủ do tắc nghẽn (OSA): Ảnh hưởng của tư thế ngủ và các tiến bộ công nghệ trong giám sát tư thế ngủ tự động**

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 (pOSA). Tiếp đó, chương tập trung làm rõ cơ sở khoa học cho việc phát triển các thiết bị theo dõi giấc ngủ và thiết bị HST (Home Sleep Test), nhận dạng tư thế ngủ tự động. Cuối cùng, tác giả đưa ra xu hướng ứng dụng trí tuệ nhân tạo (AI) và điện toán biên (Edge Computing) trong phân tích dữ liệu sinh lý và nhận dạng tư thế ngủ tự động, 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, hỗ trợ phân tầng nguy cơ và lựa chọn phương pháp điều trị phù hợp. Theo tiêu chuẩn chấm điểm của AASM [33], 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).

Ngưng thở (Apnea) được Hiệp hội Y học Giấc ngủ Hoa Kỳ (AASM) định nghĩa là sự ngưng luồng khí hô hấp qua mũi và miệng trong thời gian tối thiểu 10 giây. 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 [34]: ngưng thở tắc nghẽn, ngưng thở trung ương, Ngưng thở hỗn hợp. Trong đó: 01) Ngưng thở khi ngủ do tắc nghẽn (Obstructive Sleep Apnea – OSA) là dạng phổ biến nhất, xảy ra khi các cơ vùng họng giãn ra và làm tắc đường thở, cản trở không khí đi vào phổi [35]; 02) Ngưng thở khi ngủ do trung ương (Central Sleep Apnea – CSA) là tình trạng não không gửi tín hiệu đúng đến các cơ kiểm soát hô hấp [36]. 03) Ngưng thở hỗn hợp (Mixed Apnea) là sự kết hợp của cả hai yếu tố: giai đoạn đầu của sự kiện không có nỗ lực hô hấp (giống CSA), sau đó xuất hiện nỗ lực hô hấp (giống OSA). Dạng này thường xuất hiện ở những bệnh nhân OSA nặng.

Giảm thở (Hypopnea) với hai mức tiêu chuẩn đánh giá: 01) Tiêu chuẩn khuyến nghị: một sự kiện được xác định là hypopnea nếu thỏa mãn đồng thời ba điều kiện: (i) biên độ tín hiệu luồng khí giảm  $\geq 30\%$  so với nền trước sự kiện, đo bằng cảm biến áp lực mũi hoặc thiết bị CPAP; (ii) thời gian giảm tín hiệu kéo dài  $\geq 10$  giây; và (iii) kèm theo giảm độ bão hòa oxy  $\geq 3\%$  và/hoặc gây kích hoạt điện não (arousal); 02) Tiêu chuẩn chấp nhận được (Acceptable): tương tự như trên, tuy nhiên yêu cầu giảm độ bão hòa oxy phải đạt từ 4% trở lên.

RERA 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 apnea hoặc hypopnea. 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 (Obstructive Sleep Apnea – OSA) là dạng phổ biến nhất và có tác động sâu rộng đến sức khỏe cộng đồng. Đáng chú ý, một phân nhóm quan trọng của OSA là ngưng thở khi ngủ có liên quan đến tư thế (positional OSA – pOSA), trong đó tần suất các sự kiện ngưng thở tăng rõ rệt khi người bệnh nằm ngửa so với các tư thế khác. Do vậy, luận văn này tập trung phân tích chuyên sâu về OSA và đặc biệt là pOSA, làm rõ cơ chế bệnh sinh, tiêu chí chẩn đoán và các chỉ số phân tầng mức độ nặng. Trên cơ sở đó, nghiên cứu đề xuất và phát triển một hệ thống giám sát tư thế ngủ tích hợp cảm biến gia tốc và mô hình học máy gọn nhẹ (TinyML), hướng tới khả năng sàng lọc OSA tại nhà bằng thiết bị đeo thông minh hoạt động độc lập trên nền tảng vi điều khiển.

Các mức độ của hội chứng ngưng thở khi ngủ do tắc nghẽn (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 hép thở cho tổng số giờ đã ngủ, với mỗi sự kiện phải kéo dài ít nhất 10 giây Bảng 1.1 [35]. Hội chứng ngưng thở khi ngủ tắc nghẽn tư thế (positional Obstructive Sleep Apnea – pOSA) là một dạng đặc biệt của OSA, trong đó mức độ nghiêm trọng của hội chứng ngưng thở chịu ảnh hưởng đáng kể từ tư thế nằm của bệnh nhân. Cụ thể, pOSA được xác định khi chỉ số AHI (Apnea–Hypopnea Index) ở tư thế nằm ngửa cao hơn đáng kể so với các tư thế khác, và thường gặp nhất ở bệnh nhân OSA mức độ nhẹ đến trung bình [8, 37]. Người mắc pOSA thường có các đặc điểm như trẻ tuổi hơn, chỉ số khối cơ thể (BMI) thấp hơn, và mức độ OSA tổng thể nhẹ hơn so với nhóm bệnh nhân không thuộc dạng tư thế (non-positional OSA – NpOSA).

Trong nhiều năm qua, các nhà nghiên cứu đã đề xuất nhiều tiêu chí khác nhau nhằm chẩn đoán pOSA, từ đơn giản đến phức tạp. Định nghĩa cổ điển nhất được giới thiệu bởi Cartwright, theo đó bệnh nhân được coi là pOSA nếu AHI ở tư thế nằm ngửa lớn hơn ít nhất hai lần so với AHI ở tư thế không nằm ngửa [38]. Mador sau đó kế thừa định nghĩa này và bổ sung tiêu chí rằng AHI ở tư thế không nằm ngửa phải nhỏ hơn 5 lần/giờ, nhằm tăng tính đặc hiệu trong chẩn đoán [39]. Song song đó, Levendowski 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 AHI ở tư thế không nằm ngửa [40]. Các tiêu chí này có ưu điểm là đơn giản và dễ áp dụng trong lâm sàng cũng như trên thiết bị theo dõi tại nhà, nhưng có thể bỏ sót những trường hợp ranh giới hoặc đa yếu tố.

Một hệ thống phân loại toàn diện hơn là Amsterdam Positional Obstructive Sleep Apnea Classification (APOC), được thiết kế để phản ánh chính xác hơn ảnh hưởng của tư thế đến mức độ nghiêm trọng của OSA [41]. Tiêu chí APOC xác định pOSA khi bệnh nhân có AHI toàn bộ lớn hơn 5 lần/giờ, đồng thời tổng thời gian ngủ (Total Sleep Time – TST) ở tư thế tốt nhất (Best Sleeping Position – BSP) và tư thế gây ra chỉ số AHI cao nhất (Worst Sleeping Position – WSP) đều chiếm tối thiểu 10% TST. Ngoài ra, bệnh nhân cần thỏa mãn ít nhất một trong ba điều kiện sau: 01) AHI ở BSP nhỏ hơn 5; 02) AHI ở BSP thấp hơn AHI toàn bộ; 03) AHI ở BSP thấp hơn tối thiểu 25% so với AHI toàn bộ trong trường hợp AHI toàn bộ vượt quá 40. Hơn nữa, APOC còn cho phép phân nhóm bệnh nhân thành ba mức độ đáp ứng điều trị: nhóm APOC-I bao gồm bệnh nhân có thể khỏi hoàn toàn nhờ PT; nhóm APOC-II và APOC-III bao gồm các trường hợp có cải thiện một phần như giảm phân loại OSA hoặc giảm chỉ số AHI sau can thiệp tư thế.

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

Ngưng thở tắc nghẽn khi ngủ thường xảy ra ở người lớn tuổi và những người thừa cân béo phì. Yếu tố gây ra có thể liên quan đến cấu trúc hoặc phi cấu trúc, bao gồm cả yếu tố di truyền. Tỷ lệ ngưng thở tắc nghẽn là từ 2% đến 9% ở người lớn. Ngưng thở tắc nghẽn khi ngủ có thể tăng gấp 4 lần ở nam giới và gấp 7 lần hơn ở những người béo phì (ví dụ chỉ số khối cơ thể (Body mass Index - BMI) > 30). OSA nặng (AHI > 30/h) làm tăng nguy cơ tử vong ở nam giới trung niên. Nguyên nhân chủ yếu là do diện tích vòng họng hoặc khoang mũi bị thu hẹp như viêm xoang, các khối u, bệnh phì đại tuyến lưỡi, amydal, phì đại tuyến mõ (đặc biệt ở trẻ em), béo phì hoặc đến từ các bệnh lý: tiểu đường, huyết áp cao, các bệnh tim mạch v.v [42]. Ngoài ra, có thể đến từ thói quen không lành mạnh của con người như là sử dụng các chất kích thích, hút thuốc, ngáy khi ngủ [43][44]. Bên cạnh đó, các yếu tố không giải phẫu như hoạt động kém của cơ gián họng, nồng độ thức giấc thấp và sự điều hòa hô hấp không ổn định cũng góp phần quan

trọng vào cơ chế bệnh sinh. Sự tương tác giữa các yếu tố này tạo nên tính đa dạng trong biểu hiện và mức độ nặng của OSA.

Phần lớn bệnh nhân mắc hội chứng ngưng thở khi ngủ tắc nghẽn (OSA) không tự nhận thức được các rối loạn hô hấp xảy ra trong lúc ngủ. Điều này đặc biệt đúng với những người sống hoặc ngủ một mình, do thiếu sự quan sát từ bên ngoài. Đáng lưu ý, hơn 80% các trường hợp OSA được phát hiện ở những bệnh nhân mắc các bệnh lý liên quan đến béo phì như tiểu đường, bệnh thận, rối loạn lipid máu, v.v. [42]. Hội chứng OSA ảnh hưởng nghiêm trọng đến chất lượng cuộc sống. Những hệ lụy thường gặp bao gồm: suy giảm trí nhớ, giảm tinh táo, dễ cáu gắt, trầm cảm, đau đầu và giảm khả năng tập trung [45]. Các tác động này làm giảm hiệu suất làm việc, gây rối loạn trong các mối quan hệ xã hội và làm tăng nguy cơ tai nạn giao thông. Một nghiên cứu bởi Mooe và cộng sự [46] thực hiện trên nam giới bị bệnh mạch vành (Coronary Artery Disease – CAD) cho thấy có tới 37% bệnh nhân có chỉ số AHI vượt quá 10. Trong khi đó, nghiên cứu của Young và cộng sự (1997) đã phát hiện rằng chỉ số AHI tăng tỉ lệ thuận với huyết áp tâm thu và tâm trương, với mức ý nghĩa lần lượt là  $p = 0.003$  và  $p = 0.01$  [47].

Trong điều kiện hiện tại, đa số bệnh nhân nghi ngờ mắc hội chứng ngưng thở tắc nghẽn khi ngủ được khám bởi bs chuyên khoa Tai Mũi Họng và bác sĩ chuyên gia về ngủ ngáy. Khám tổng quát kết hợp khai thác bệnh sử liên quan, sử dụng các thang điểm đánh giá buồn ngủ và nguy cơ ngưng thở khi ngủ, như Epworth Sleepiness Scale, STOP-BANG (đã được dịch sang tiếng Việt) được chấp thuận tại Việt Nam như một phương án sàng lọc bệnh nhân OSA. hoặc có thể khám nội soi Tai Mũi Họng để tìm nguyên nhân. Vì đa số các trường hợp ngáy, ngưng thở khi ngủ là do tắc nghẽn ngoại biên, nguyên nhân từ Mũi – Họng – Màn hẫu , VA và amidan, và những bất thường về hàm mặt khác. Việc đánh giá ngưng thở khi ngủ bắt đầu thường bắt đầu bằng một khảo sát giấc ngủ toàn diện, bao gồm khai thác bệnh sử liên quan đến các triệu chứng lâm sàng đặc trưng, sau đó tiến hành đánh giá khách quan thông qua đa ký giấc ngủ (PSG) [48][49]. Phương pháp do dùng đa ký giấc ngủ (Polysomnography) với sự giám sát của các bác sĩ chuyên môn được coi là tiêu chuẩn vàng trong chẩn đoán chứng ngưng thở khi ngủ.

Polysomnography là một phương pháp ghi đa kênh liên tục trong suốt một đêm, bao gồm nhiều thông số sinh lý nhằm đánh giá toàn diện hoạt động hô hấp và thần kinh khi ngủ. Các thành phần chính trong một đánh giá polysomnography bao gồm: điện não đồ (EEG) để ghi lại hoạt động điện của não; điện cơ ký (EMG) nhằm đo trương lực cơ,

đặc biệt là ở cằm và chân; điện động mắt (EOG) để theo dõi chuyển động của nhãn cầu, giúp xác định các giai đoạn của giấc ngủ; và điện tâm đồ (ECG) để theo dõi hoạt động điện của tim. Bên cạnh đó, quá trình đo cũng bao gồm theo dõi độ bão hòa oxy trong máu ( $\text{SpO}_2$ ), đo lưu lượng khí thở qua mũi và miệng, đánh giá nỗ lực hô hấp thông qua chuyển động của ngực và bụng, đo áp lực khí thở qua mũi, và ghi nhận cường độ tiếng ngáy. Tư thế ngủ là một tín hiệu quan trọng trong polysomnography (PSG), đặc biệt có giá trị trong chẩn đoán và phân loại hội chứng ngưng thở khi ngủ tắc nghẽn phụ thuộc tư thế (positional OSA – pOSA). Trong quá trình ghi đa ký giấc ngủ, việc theo dõi liên tục tư thế cơ thể giúp xác định mối liên hệ giữa tư thế nằm (như nằm ngửa, nằm nghiêng hoặc nằm sấp) với tần suất và mức độ nghiêm trọng của các rối loạn hô hấp. Tập hợp các thông số này cho phép bác sĩ chẩn đoán chính hội chứng ngưng thở khi ngủ tắc nghẽn (OSA).

Một trong những hạn chế của phương pháp đánh giá sử dụng (PSG) là sự bất tiện, chi phí cao và khả năng phổ biến thấp, nhất là đối với phần lớn người bệnh có thu nhập thấp. Việc yêu cầu bệnh nhân phải lưu trú qua đêm tại cơ sở y tế, cùng với việc gắn nhiều thiết bị theo dõi sinh lý lên cơ thể, không chỉ gây cảm giác khó chịu mà còn tiềm ẩn nguy cơ ảnh hưởng đến chất lượng và tính chính xác của dữ liệu thu thập được. Chính những bất cập này đã thúc đẩy sự phát triển của các thiết bị theo dõi giấc ngủ ngoài trung tâm (Out-of-Center devices) hay còn gọi là thiết bị kiểm tra giấc ngủ tại nhà (Home Sleep Test – HST). Những thiết bị này thường được thiết kế với số lượng cảm biến tối giản hơn so với PSG truyền thống, đồng thời tích hợp các thuật toán phân tích tự động – được xử lý trực tiếp trên thiết bị hoặc thông qua phần mềm chuyên dụng – nhằm hỗ trợ chẩn đoán ngưng thở khi ngủ do tắc nghẽn (OSA) một cách thuận tiện và tiết kiệm hơn. Những thông số SCOPERA được coi là cơ sở để xây dựng thiết bị HST trong đó giấc ngủ (Sleep - S), tim mạch (Cardiovascular - C), oxi trong máu (Oximetry - O), cố gắng thở (Effort - E), luồng không khí lưu thông (Respiratory - R), âm thở (Audio - A).

Thiết bị đeo hỗ trợ theo dõi giấc ngủ (wearable Health Sleep Technology – HST) đang trở thành một xu hướng chủ đạo trong nghiên cứu và ứng dụng lâm sàng nhờ khả năng thu thập liên tục dữ liệu sinh lý một cách không xâm lấn, thuận tiện và có thể triển khai tại nhà. Dựa trên đặc điểm hình thái và vị trí gắn trên cơ thể, các thiết bị này có thể được phân thành các nhóm: vòng tay (bracelet), đai ngực (chest band), miếng dán (adhesive patch), tai nghe (headset), nhẫn thông minh (ring), v.v. Các thiết bị này có thể

là sản phẩm thương mại sẵn có hoặc được thiết kế riêng cho mục đích nghiên cứu.

Bảng 1.2: Phân loại thiết bị đeo trong phát hiện OSA và tài liệu tham khảo liên quan

<b>Loại thiết bị đeo</b>	<b>Tài liệu tham khảo</b>
Vòng tay (Bracelet)	[50], [51] [12] [20]
Đai ngực (Chest band)	[52], [53] [12] [20]
Miếng dán (Adhesive patch)	[30], [54], [55], [56] [20]
Dạng khác	[28], [17] [20] [13] [14] [17]

Jeon và cộng sự [50] đã đề xuất một hệ thống chẩn đoán và dự đoán hội chứng ngưng thở khi ngủ dựa trên thiết bị đeo. Nghiên cứu giới thiệu hệ thống chỉ sử dụng một thiết bị đeo duy nhất – Sleep Care Kit (SCK) – có khả năng đo các thông số như nhịp thở, độ bão hòa oxy, nhịp tim và gia tốc ba trục. Thiết bị được đeo ở cổ tay, đảm bảo tiếp xúc trực tiếp với da để thu thập tín hiệu sinh lý một cách chính xác. Sau một giai đoạn học ban đầu, hệ thống tiến hành phân tích mối tương quan giữa nhịp tim và gia tốc ba trục nhằm dự đoán nguy cơ ngưng thở khi ngủ. Trong nghiên cứu [53], Chen và cộng sự sử dụng thiết bị đeo dạng vòng tay thu tín hiệu PPG để phát hiện ngưng thở khi ngủ ở 100 tình nguyện viên. Thiết bị được thiết kế tối ưu về bộ nhớ, độ trễ và năng lượng, phù hợp triển khai trên các hệ thống tính toán giới hạn. Dữ liệu thu được được đồng bộ với PSG nhằm đảm bảo độ chính xác, đồng thời hướng tới ứng dụng giám sát dài hạn tại nhà. Trong các nghiên cứu [54, 55], Yeo và cộng sự sử dụng thiết bị dán T-REX TR100A để ghi điện tâm đồ (ECG) một kênh từ vùng bụng trên. Thiết bị được dán trực tiếp lên da, đảm bảo tiếp xúc ổn định và tín hiệu thu nhận chính xác, giúp theo dõi liên tục và hạn chế nguy cơ bị bong tróc trong quá trình sử dụng. Một thiết bị HST dạng miếng dán cổ đã được phát triển, tích hợp cảm biến gia tốc ba trục LIS2DH12 và vi điều khiển nRF5232, cho phép truyền dữ liệu không dây qua Bluetooth Low Energy (BLE) với tần số lấy mẫu 100 Hz [28]. Thiết bị có kích thước nhỏ gọn ( $\sim 3207 \text{ mm}^3$ ), được dán trực tiếp lên vùng cổ bằng băng dính hai mặt, giúp đảm bảo tiếp xúc ổn định với da. Thiết kế này cho phép ghi nhận chính xác các tư thế ngủ phổ biến như nằm ngửa, nằm nghiêng và nằm sấp, phù hợp với mục tiêu theo dõi tại nhà và hỗ trợ đánh giá nguy cơ mắc pOSA. Nghiên cứu [52] chứng minh rằng tín hiệu chuyển động ngực (Thoracic movement signal - THO) và bụng (Abdominal movement signal - ABD), thu từ các dải piezoelectric đeo được, có thể được sử dụng hiệu quả để phân loại các dạng rối loạn thở khi ngủ thông qua mô hình

thuật toán SVM. Kết quả cho thấy khi kết hợp cả hai tín hiệu, độ chính xác phân loại đạt trung bình 81.8%, khẳng định tiềm năng ứng dụng của phương pháp này trong sàng lọc và theo dõi OSA tại nhà hoặc trong lâm sàng. Gần đây, Domingues và cộng sự (2024) xây dựng một mô hình mạng nơ-ron nhân tạo dựa trên dữ liệu từ máy đo ( $\text{SpO}_2$ ), cảm biến gia tốc và ghi âm tiếng ngáy của hệ thống Biologix, nhằm dự đoán chính xác trạng thái ngủ. Kết quả cho thấy mô hình này có khả năng cao độ chính xác trong chẩn đoán ngưng thở khi ngủ tại nhà, tiệm cận với tiêu chuẩn của đa ký giấc ngủ truyền thống [57]. Một hướng nghiên cứu khác là cảm biến đặt dưới nệm giường. Tác giả Andrei Boiko và cộng sự đánh giá hệ thống phát sử dụng cảm biến gia tốc đặt dưới đệm giường để ghi dao động do cử động ngực khi thở. Kết quả cho thấy thuật toán phát hiện ngưng thở đạt độ chính xác, độ đặc hiệu và độ nhạy lần lượt là 94.6%, 95.3% và 93.7% [58].

Tuy nhiên, một rào cản lớn cho việc triển khai HST trên diện rộng vẫn nằm ở chi phí. Theo báo cáo tổng quan [59], giá trung bình của một thiết bị HST khoảng 2300 USD – vượt quá khả năng chi trả của đại đa số bệnh nhân, đặc biệt ở các nước có thu nhập trung bình – thấp. Do đó, việc tối ưu hóa chi phí, giảm số lượng cảm biến và đơn giản hóa quy trình sử dụng vẫn là các mục tiêu cấp thiết.

Hiện nay, đã có nhiều nghiên cứu và sản phẩm thương mại được phát triển nhằm mục đích theo dõi tư thế ngủ. Trong đó, iSleePost là một hệ thống thiết bị đeo được phát triển bởi Po-Yuan Jeng và cộng sự [60], gồm hai thiết bị đeo độc lập sử dụng cảm biến gia tốc để theo dõi và phân loại tư thế ngủ. Trong đó, một thiết bị được đeo ở cổ tay nhằm ghi nhận chuyển động, còn thiết bị thứ hai được gắn ở ngực để thu thập nhãn tư thế một cách tự động phục vụ huấn luyện mô hình. Dữ liệu từ cổ tay được xử lý bằng kỹ thuật cửa sổ trượt để trích xuất đặc trưng, sau đó ánh xạ với tư thế cơ thể dựa trên dữ liệu từ thiết bị ngực. Một hướng tiếp cận khác, Zhang và cộng sự sử dụng duy nhất một thiết bị đeo gắn ở ngực, có tích hợp cảm biến gia tốc ba trục, để theo dõi cả tư thế ngủ và các chỉ số sinh lý như nhịp tim và nhịp thở [61]. Trong nghiên cứu được thực hiện trên 7 tình nguyện viên khỏe mạnh trong điều kiện phòng lab, hệ thống sử dụng bộ phân loại tuyến tính (LDA) để nhận diện bốn tư thế ngủ phổ biến trong trạng thái tĩnh, kết hợp với thuật toán phát hiện chuyển động hiệu quả nhằm phân biệt giữa tư thế tĩnh và vận động. Một hệ thống thiết bị đeo theo dõi tư thế ngủ (Wearable Sleep System – WSS), được Kwasnicki và cộng sự phát triển với thiết kế riêng biệt, gồm các cảm biến đeo trên mỗi cánh tay và vùng ngực, kết nối không dây với bộ xử lý trung tâm (trong nghiên cứu là máy tính xách

tay) thông qua bộ thu phát sóng vô tuyến [62]. Nền tảng phần cứng của hệ thống sử dụng bộ vi xử lý tiêu thụ điện năng cực thấp TI MSP430, kết hợp với mô-đun truyền thông không dây Chipcon CC2420, cùng pin Li-ion polymer nhẹ. Mỗi nút cảm biến được tích hợp cảm biến gia tốc ba trục ADXL330, con quay hồi chuyển số ITG-3200, và cảm biến từ trường ba trục HMC5843. Toàn bộ mô-đun có kích thước  $20 \times 30 \times 17$  mm, trọng lượng 10 g. Trong nghiên cứu của Asma Channa và cộng sự [63], một hệ thống theo dõi tư thế ngủ dựa trên công nghệ Internet vạn vật (IoT) đã được đề xuất, sử dụng hai thảm cảm biến áp lực thương mại để thu thập dữ liệu từ 13 người tham gia ở các tư thế ngủ khác nhau. Dữ liệu được xử lý bằng nhiều thuật toán học máy giám sát, trong đó các mô hình như Weighted KNN và Linear SVM đạt độ chính xác nhận dạng tư thế lên tới 98,7%.

Các nghiên cứu hiện nay đã phát triển nhiều hệ thống theo dõi tư thế ngủ với độ chính xác cao, sử dụng đa dạng nền tảng như thiết bị đeo tích hợp cảm biến gia tốc, PPG, ECG hoặc thảm cảm biến áp lực. Nhiều hệ thống đã chứng minh hiệu quả trong việc phân loại tư thế ngủ và ghi nhận các chỉ số sinh lý liên quan đến hội chứng ngưng thở khi ngủ tắc nghẽn (OSA).

Tuy nhiên, để đáp ứng yêu cầu triển khai thực tế tại nhà, các giải pháp cần được tiếp tục hoàn thiện theo hướng tối giản phần cứng, giảm số lượng cảm biến, đồng thời tối ưu mô hình học máy cho nền tảng tính toán giới hạn (TinyML). Việc xây dựng một quy trình thu thập và xử lý dữ liệu rõ ràng, có thể lặp lại, cũng đóng vai trò quan trọng trong việc đảm bảo tính ổn định và khả năng áp dụng rộng rãi.

Bám sát định hướng này, luận văn đề xuất một hệ thống sử dụng duy nhất một cảm biến gia tốc dán dưới hõm ức để phát hiện các tư thế ngủ liên quan đến OSA. Trên cơ sở dữ liệu thu được, mô hình học máy nhẹ sẽ được xây dựng và tối ưu, hướng tới việc phát triển thiết bị IoT nhỏ gọn, chi phí thấp, có khả năng ước lượng chỉ số AHI – phục vụ cho sàng lọc và theo dõi OSA tại nhà một cách hiệu quả.

## 1.2. Ứng dụng cảm biến gia tốc trong đánh giá tư thế ngủ của người mắc OSA tại nhà

Tư thế ngủ của con người thường được phân loại thành bốn nhóm chính: nằm ngửa, nghiêng trái, nghiêng phải và nằm sấp (Hình 1.1) [64]. Việc phân biệt rõ ràng các tư thế



Hình 1.1: Các tư thế ngủ cơ bản của con người

này giúp nâng cao độ chính xác trong việc phân tích ảnh hưởng của tư thế đến các chỉ số sinh lý trong giấc ngủ.

Việc theo dõi tư thế cơ thể đặc biệt hữu ích trong phát hiện và điều trị hội chứng ngừng thở khi ngủ phụ thuộc tư thế (positional OSA). Hiểu được mối quan hệ giữa tư thế ngủ và rối loạn hô hấp sẽ mở ra hướng điều trị cá thể hóa, chẳng hạn như liệu pháp định hướng tư thế.

Trong bối cảnh đó, cảm biến gia tốc ba trục (triaxial accelerometers) nổi lên như một công cụ đầy hứa hẹn nhờ vào đặc tính đo lường trực tiếp chuyển động và hướng trọng lực của cơ thể. So với các phương pháp truyền thống như ghi hình hồng ngoại, cảm biến áp suất hay hệ thống camera đa góc, gia tốc kế mang lại ưu thế về kích thước, mức độ xâm nhập, khả năng tiêu thụ năng lượng thấp và đặc biệt là chi phí triển khai hợp lý – các yếu tố quyết định khi xét đến ứng dụng tại nhà trên quy mô lớn.

Các hệ thống ghi hình sử dụng camera hồng ngoại có khả năng ghi nhận toàn bộ hoạt động của cơ thể trong điều kiện ánh sáng yếu, và có thể kết hợp với học sâu để phân loại tư thế ngủ một cách chính xác. Tuy nhiên, phương pháp này đòi hỏi không

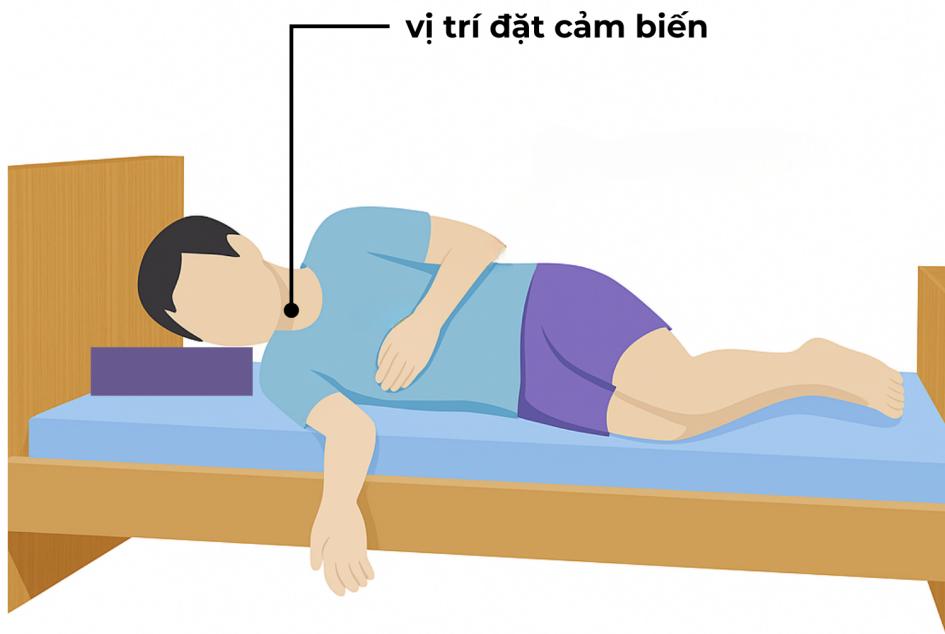
gian phòng ngủ cố định, điều kiện lắp đặt tối ưu, và đặt ra các thách thức về quyền riêng tư [65]. Việc tái tạo các góc khuôn mặt hoặc cơ thể khi bị chắn/mềm che khuất cũng làm giảm hiệu quả hệ thống. Trong khi đó, cảm biến áp suất bô trí dưới đệm có thể ghi nhận phân bố trọng lực và áp lực tiếp xúc, từ đó gián tiếp suy đoán tư thế. Channa và cộng sự [63] sử dụng tới 2048 điểm cảm biến để tăng cường độ phân giải, cho phép phân biệt chính xác các tư thế cơ bản. Tuy nhiên, nhược điểm của phương pháp này nằm ở tính cố định (không đeo được), chi phí lắp đặt cao và không phù hợp cho người hay di chuyển nơi ngủ hoặc sinh sống trong điều kiện không gian hạn chế.

Trái lại, cảm biến gia tốc mang tính linh hoạt cao. Thiết bị có thể đeo trực tiếp lên cơ thể (dạng vòng tay, miếng dán, đai ngực) hoặc tích hợp vào các vật dụng quen thuộc như điện thoại di động [66, 67]. Thiết kế này vừa đảm bảo tính di động, vừa cho phép giám sát liên tục với mức độ can thiệp tối thiểu.

Từ góc độ học thuật, các nghiên cứu gần đây đã xác nhận hiệu quả của cảm biến gia tốc trong phân loại tư thế ngủ. Jeng và cộng sự [60] sử dụng thiết bị đeo tay kết hợp cảm biến ngực để huấn luyện mô hình học máy đạt độ chính xác trên 85%. Zhang và cộng sự [61] chỉ sử dụng một cảm biến tại vị trí xương ức đã có thể phân biệt chính xác các tư thế phổ biến. Đặc biệt, việc chọn vị trí gắn cảm biến là yếu tố then chốt. Các vị trí như cổ tay hay trán thường chịu nhiều chuyển động ngoại ý và lệch trực so với thân người, làm giảm tính đại diện của tín hiệu. Vùng xương ức, cụ thể là ngay dưới hõm cổ (suprasternal notch), được xem là vị trí lý tưởng do sự ổn định hình học và gần đường trung trực cơ thể, từ đó phản ánh đúng hướng trọng lực – nền tảng để suy luận tư thế một cách chính xác.

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 [66]. Nhóm tác giả trong [68] đã báo cáo nghiên cứu đánh giá tư thế ngủ của bệnh nhân sử dụng thiết bị di động đeo ở xương ức 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 tiêu biểu, 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 (Total Sleep Time – TST), độ trễ vào giấc (Sleep Onset Latency – SOL) và hiệu quả giấc ngủ (Sleep Efficiency – SE). 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[67]. 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.



Hình 1.2: Vị trí tối ưu để đặt cảm biến gia tốc

Ưu điểm nổi bật nhất của cảm biến gia tốc là khả năng hoạt động độc lập với mức tiêu thụ điện năng thấp, dễ dàng tích hợp vào hệ thống vi điều khiển nhúng như nRF52, ESP32, hoặc STM32 để xây dựng thiết bị đeo có khả năng xử lý cục bộ (on-device) – phù hợp với định hướng TinyML. Ngoài ra, cảm biến gia tốc không yêu cầu điều kiện môi trường đặc biệt như ánh sáng hay bề mặt phẳng, đồng thời không ảnh hưởng đến quyền riêng tư như hệ thống camera.

Tuy nhiên, giới hạn của cảm biến gia tốc là việc nhận diện tư thế phụ thuộc mạnh vào vị trí gắn cảm biến. Một cảm biến đặt sai vị trí hoặc xoay lệch trực có thể làm sai lệch hoàn toàn kết quả phân loại. Hơn nữa, gia tốc kê không cung cấp thông tin về các chỉ số sinh lý như SpO<sub>2</sub>, nhịp thở hay nhịp tim, vì vậy để chẩn đoán toàn diện OSA, cần tích hợp thêm các cảm biến khác hoặc kết hợp với các mô hình suy luận (inference models) để bù đắp cho thiếu sót này.

Xuất phát từ các phân tích nêu trên, nghiên cứu này lựa chọn thiết kế một thiết bị

Bảng 1.3: So sánh ưu và nhược điểm của các vị trí gắn cảm biến gia tốc trong theo dõi tư thế ngủ

Vị trí	Ưu điểm	Nhược điểm
<b>Cổ tay</b>	- Dễ đeo, quen thuộc (giống đồng hồ, vòng tay). - Thuận tiện cho người dùng tự gắn.	- Chịu nhiều chuyển động ngoài ý muốn của tay. - Tín hiệu kém đại diện cho tư thế toàn thân.
<b>Trán</b>	- Có thể cố định bằng băng quấn đầu. - Ít bị ảnh hưởng bởi quần áo hay chăn phủ.	- Không phản ánh chính xác trực cơ thể. - Gây khó chịu, ảnh hưởng thẩm mỹ và giấc ngủ.
<b>Ngực (lateral)</b>	- Gần trung tâm thân người, phản ánh tương đối hướng trọng lực. - Thuận lợi để kết hợp với dây đeo ngực (chest strap).	- Có nguy cơ xô lệch khi nằm nghiêng. - Có thể gây khó chịu khi nằm sấp.
<b>Xương ức (suprasternal notch)</b>	- Ổn định hình học, gần trục trung tâm cơ thể. - Phản ánh chính xác hướng trọng lực và tư thế toàn thân. - Ít chịu ảnh hưởng từ chuyển động tay/chân. - Thuận lợi để tích hợp thêm cảm biến (âm thanh, nhiệt độ, SpO <sub>2</sub> ).	- Yêu cầu thiết kế miếng dán/thiết bị đeo chuyên dụng để cố định đúng vị trí.

đeo tiếp xúc sử dụng cảm biến gia tốc ba trục, được gắn tại vùng xương ức (suprasternal notch). Việc lựa chọn vị trí và cấu trúc thiết bị được cân nhắc dựa trên các yếu tố sau (Hình 1.2). Bảng 1.3 dưới đây so sánh ưu và nhược điểm của các vị trí gắn cảm biến gia tốc thường gặp, qua đó cho thấy vùng xương ức (suprasternal notch) là lựa chọn hợp lý nhất:

1. Vị trí xương ức có độ ổn định cao trong suốt quá trình ngủ, hạn chế dịch chuyển ngoài ý muốn, từ đó đảm bảo tính nhất quán của dữ liệu thu nhận.
2. Vị trí này phản ánh chính xác hướng trọng lực của cơ thể, giúp tăng độ chính xác trong phân loại tư thế ngủ dựa trên gia tốc ( $x, y, z$ ).
3. Cấu trúc gắn tại xương ức thuận lợi cho việc tích hợp thêm các cảm biến khác như cảm biến âm thanh (microphone) hoặc cảm biến nhiệt độ, phục vụ cho các mục tiêu mở rộng trong các nghiên cứu tiếp theo.

Sau khi xác định được vị trí gắn tối ưu, bước tiếp theo là làm rõ cơ sở nguyên lý đo lường của cảm biến, nhằm lý giải vì sao thiết bị này có thể phản ánh chính xác tư thế cơ thể trong khi ngủ.

## Cảm biến gia tốc và nguyên lý ứng dụng trong theo dõi tư thế ngủ

Cảm biến gia tốc là một thiết bị đo lường có khả năng phát hiện và ghi nhận gia tốc – tức là sự thay đổi vận tốc theo thời gian – của một vật thể trong không gian ba chiều. 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, và đặc biệt là y học, trong các thiết bị theo dõi hoạt động và giấc ngủ.

Tuy nhiên, giới hạn của cảm biến gia tốc là việc nhận diện tư thế phụ thuộc mạnh vào vị trí gắn cảm biến. Một cảm biến đặt sai vị trí hoặc xoay lệch trực có thể làm sai lệch hoàn toàn kết quả phân loại. Hơn nữa, gia tốc kê không cung cấp thông tin về các chỉ số sinh lý như SpO<sub>2</sub>, nhịp thở hay nhịp tim, vì vậy để chẩn đoán toàn diện OSA, cần tích hợp thêm các cảm biến khác hoặc kết hợp với các mô hình suy luận (inference models) để bù đắp cho thiếu sót này.

Nguyên lý hoạt động của cảm biến gia tốc dựa trên **Định luật II Newton**:

$$F = ma \quad (1.1)$$

Trong đó,  $F$  là lực tác động lên một khối lượng  $m$ , tạo ra gia tốc  $a$ . Trong cấu trúc vi cơ điện tử (MEMS) của cảm biến gia tốc, một khối lượng nhỏ được treo bằng các thanh đòn hồi. Khi cảm biến chịu tác động gia tốc, khối lượng này dịch chuyển, gây ra sự thay đổi về đặc tính điện, chẳng hạn như: 01) thay đổi điện dung trong cảm biến kiểu điện dung (capacitive type); 02) thay đổi điện tích do hiệu ứng áp điện trong cảm biến kiểu áp điện (piezoelectric type); và 03) thay đổi điện áp trong các cảm biến điện trở áp (piezoresistive type).

Cấu trúc của cảm biến gia tốc MEMS bao gồm một khối lượng vi mô được gắn với hệ thống lò xo vi cơ. Khi có gia tốc tác động, khối lượng này dịch chuyển so với vị trí cân bằng ban đầu, gây ra sự biến dạng trong hệ lò xo. Theo định luật Hooke, lực đòn hồi sinh ra bởi sự biến dạng này được xác định bởi:

$$F = k \cdot \Delta l \quad (1.2)$$

kết hợp với phương trình Newton ta thu được:

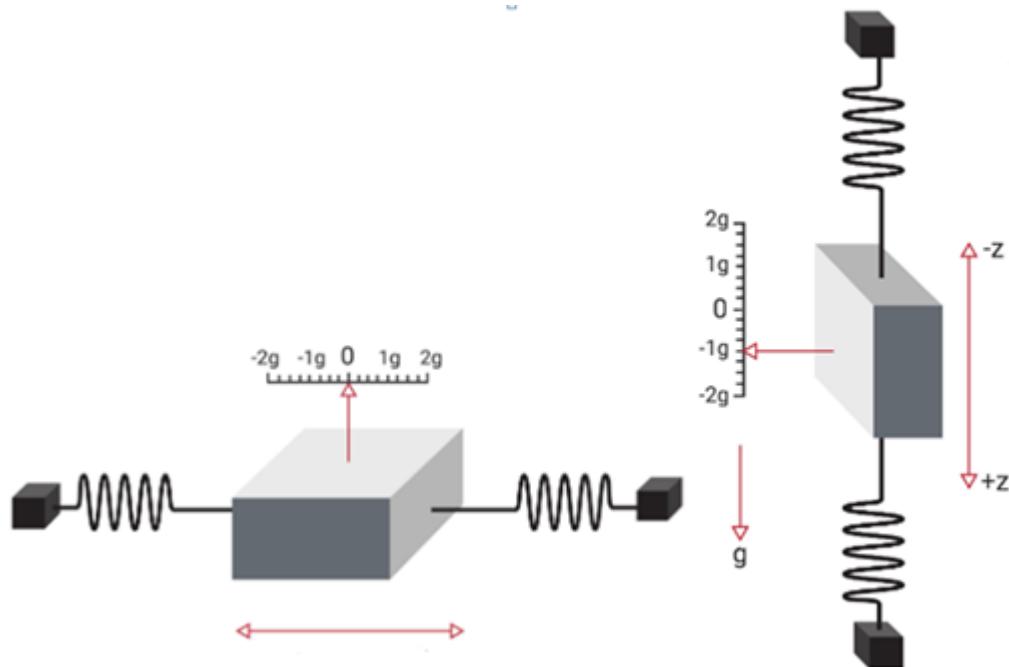
$$a = \frac{k \cdot \Delta l}{m} \quad (1.3)$$

Điều này cho thấy: gia tốc có thể được xác định gián tiếp thông qua độ biến dạng  $\Delta l$  của phần tử đòn hồi – yếu tố có thể đo được bằng các cảm biến vật lý (điện dung, áp điện, áp điện trở). Đây là cơ sở vật lý cho toàn bộ hoạt động của các cảm biến gia tốc hiện đại.

Tín hiệu điện sinh ra từ quá trình này được khuếch đại và số hóa để xử lý trong các ứng dụng khác nhau. Chính khả năng chuyển đổi giữa năng lượng cơ học và điện học giúp cảm biến gia tốc hoạt động hiệu quả trong việc ghi nhận các trạng thái động học của vật thể, bao gồm: 01) dịch chuyển tuyến tính (linear movement), 02) góc nghiêng (tilt), 03) rung động (vibration), và 04) va chạm (shock) hoặc rơi tự do (free fall). Trong lĩnh vực y sinh, đặc biệt là trong nghiên cứu giấc ngủ và hội chứng ngưng thở khi ngủ (OSA), cảm biến gia tốc không chỉ giúp xác định tư thế ngủ (supine, prone, lateral) mà còn ghi nhận các dao động nhỏ trong lồng ngực do hô hấp – yếu tố gián tiếp giúp phát

hiện bất thường hô hấp như hypopnea hoặc apnea.

Một điểm đặc biệt trong ứng dụng là **hệ trục tọa độ trong cảm biến**. Khi cảm biến được gắn trên cơ thể, trục z thường hướng vuông góc với mặt phẳng ngang và sẽ ghi nhận giá tốc trọng trường. Ở trạng thái tĩnh, trục z đo được giá trị gần bằng  $g = 9.81, \text{m/s}^2$ . Điều này không chỉ giúp hiệu chuẩn cảm biến mà còn cung cấp thông tin định hướng không gian – yếu tố thiết yếu để phân biệt các tư thế ngủ dựa trên định hướng của trục trọng lực. Nhờ khả năng tích hợp dễ dàng vào các thiết bị đeo (vòng tay, miếng dán, nhẫn), cảm biến giá tốc trở thành phần cốt lõi trong các hệ thống theo dõi không xâm lấn, hỗ trợ hiệu quả cho việc sàng lọc và đánh giá OSA tại nhà hoặc trong môi trường lâm sàng.



Hình 1.3: Nguyên lý cơ bản của cảm biến giá tốc

Như minh họa trong Hình 1.3, khi cảm biến giá tốc chịu tác động từ một chuyển động, khối giá trị (proof mass) sẽ dịch chuyển, làm lò xo kết nối bị biến dạng.

Trong khuôn khổ luận văn này, tác giả tập trung tìm hiểu và ứng dụng cảm biến giá tốc được chế tạo dựa trên công nghệ vi cơ điện tử (Micro-Electro-Mechanical Systems – MEMS). Đây là một công nghệ tiên tiến cho phép tích hợp các thành phần phần cứng siêu nhỏ và linh kiện điện tử ngay trên cùng một chip bán dẫn, với kích thước cấu trúc có thể dưới 10 micromet. Một trong những ưu điểm nổi bật của cảm biến giá tốc MEMS là

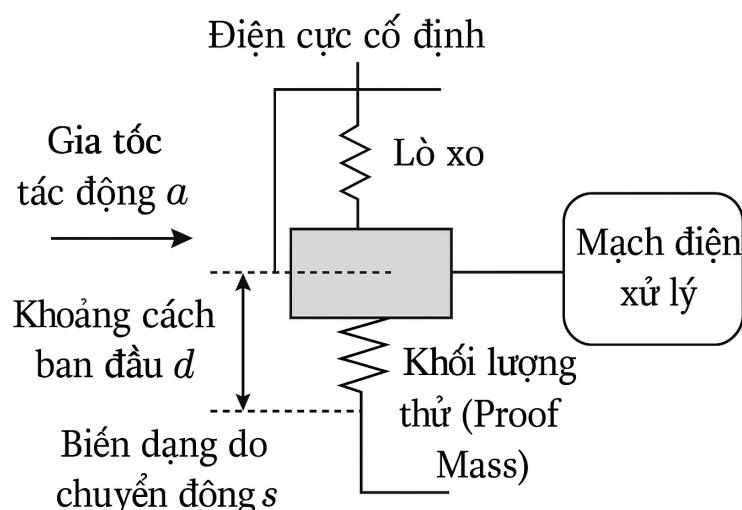
khả năng tích hợp trực tiếp lên bo mạch in (Printed Circuit Board – PCB), qua đó giảm thiểu thể tích chiếm dụng, tiết kiệm chi phí sản xuất và đơn giản hóa thiết kế hệ thống nhúng. Nhờ đó, công nghệ này đặc biệt phù hợp cho các ứng dụng trong thiết bị đeo cá nhân, điện thoại di động, và các hệ thống theo dõi sức khỏe thế hệ mới.

Dựa trên nguyên lý chuyển đổi dao động cơ học thành tín hiệu điện, cảm biến gia tốc MEMS được chia thành ba loại chính, mỗi loại ứng dụng một cơ chế vật lý riêng biệt [69, 70]:

### 1. Cảm biến gia tốc kiểu điện dung (Capacitive Accelerometers):

Đây là loại cảm biến phổ biến nhất trong các thiết bị điện tử tiêu dùng như điện thoại thông minh, vòng đeo tay sức khỏe và các hệ thống theo dõi chuyển động đeo được. Nguyên lý hoạt động của cảm biến dựa trên sự thay đổi điện dung giữa các bản cực trong cấu trúc tụ điện khi khối gia trọng (*proof mass*) dịch chuyển dưới tác động của gia tốc.

Khi có gia tốc tác động, khối lượng vi mô được treo bằng hệ lò xo vi cơ (MEMS spring system) sẽ dịch chuyển, làm thay đổi khoảng cách giữa các bản cực. Biến thiên điện dung này được ghi nhận thông qua mạch đo điện dung nhạy, sau đó được khuếch đại và chuyển đổi thành tín hiệu điện tỷ lệ với gia tốc. Ưu điểm chính bao gồm: độ nhạy cao ở vùng tần số thấp, kích thước nhỏ, tiêu thụ năng lượng thấp và khả năng tích hợp tốt vào các hệ thống nhúng. Loại cảm biến này đặc biệt phù hợp với các ứng dụng theo dõi tư thế và chuyển động chậm.

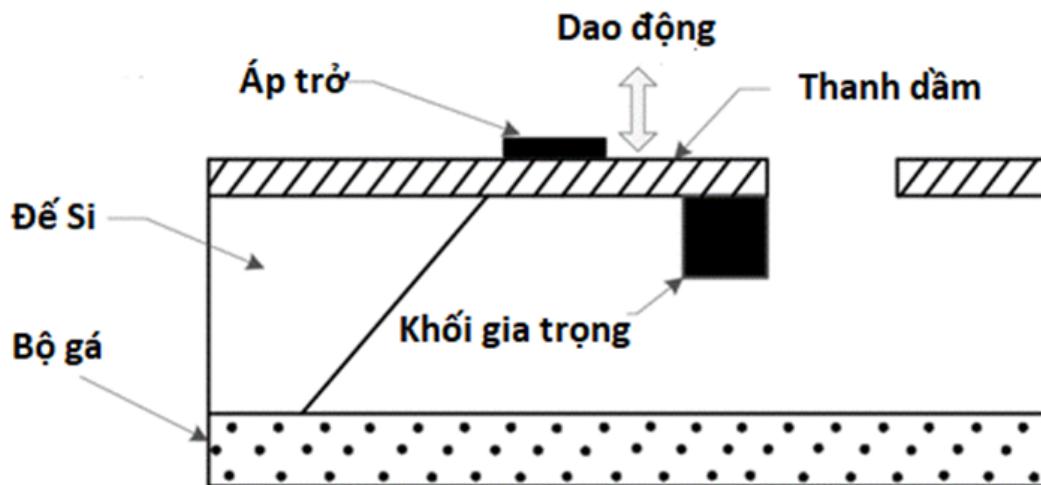


Hình 1.4: Cấu trúc cảm biến gia tốc điện dung

## 2. Cảm biến gia tốc kiểu áp điện trở (Piezoresistive Accelerometers):

Loại cảm biến này hoạt động dựa trên hiện tượng thay đổi điện trở của vật liệu bán dẫn khi chịu ứng suất cơ học – còn gọi là hiệu ứng áp điện trở. Các phần tử cảm biến thường được gắn trực tiếp lên thanh dầm (cantilever) nối với khối gia trọng. Khi có gia tốc, sự biến dạng cơ học của thanh dầm làm thay đổi điện trở của phần tử cảm biến.

Để tăng độ nhạy và khả năng khuếch đại tín hiệu, cấu trúc cảm biến thường được tích hợp trong mạch cầu Wheatstone, giúp tăng cường tỷ số tín hiệu trên nhiễu (Signal-to-Noise Ratio – SNR).

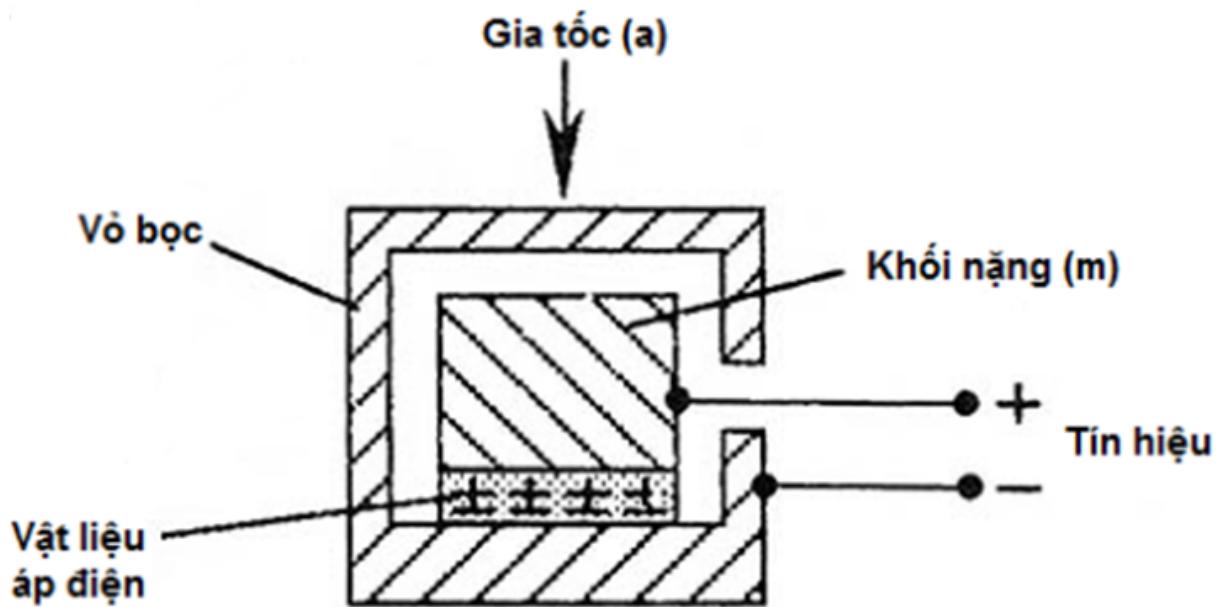


Hình 1.5: Cấu trúc cảm biến áp điện trở

Ưu điểm chính của cảm biến áp điện trở là khả năng hoạt động ổn định trong môi trường nhiệt độ cao hoặc điều kiện công nghiệp khắc nghiệt. Tuy nhiên, chúng tiêu thụ nhiều năng lượng hơn và kém nhạy hơn với chuyển động biên độ nhỏ, do đó ít phù hợp với các ứng dụng theo dõi sinh lý học dài hạn như tư thế ngủ.

## 3. Cảm biến gia tốc kiểu áp điện (Piezoelectric Accelerometers):

Dựa trên hiệu ứng áp điện, cảm biến kiểu này sử dụng các vật liệu như thạch anh hoặc gốm sứ có khả năng tạo ra điện tích khi bị biến dạng (nén hoặc kéo). Lượng điện tích sinh ra tỉ lệ thuận với lực cơ học tác động lên cảm biến.



Hình 1.6: Cấu trúc cảm biến áp điện

Cảm biến kiểu áp điện có ưu điểm nổi bật là khối lượng nhẹ, tốc độ phản hồi nhanh, và dải tần hoạt động rất rộng (có thể lên tới hàng MHz). Chúng thường được ứng dụng trong đo rung động chính xác, giám sát kết cấu, và phân tích chấn động ngắn hạn trong môi trường khắc nghiệt.

Tuy cảm biến áp điện có độ nhạy cao với rung động nhanh và tần số cao, nhưng chúng không phù hợp để đo gia tốc tĩnh hoặc chuyển động chậm kéo dài – vốn là đặc trưng của bài toán theo dõi tư thế ngủ. Nguyên nhân là vì cảm biến này không phản hồi với lực không đổi như trọng lực, do điện tích tạo ra bị rò rỉ dần theo thời gian. Bên cạnh đó, tín hiệu đầu ra có trở kháng rất cao, dễ suy hao, đòi hỏi bộ khuếch đại điện tích chuyên dụng, làm tăng độ phức tạp, kích thước và tiêu thụ năng lượng của hệ thống. Những yếu tố này khiến cảm biến áp điện không phù hợp để tích hợp vào thiết bị đeo cá nhân dùng trong theo dõi OSA tại nhà.

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.

Thứ nhất, từ góc độ *đặc trưng tín hiệu*, bài toán nhận diện tư thế ngủ yêu cầu hệ thống có khả năng ghi nhận chính xác các chuyển động biên độ nhỏ, xảy ra chậm rãi trong thời gian dài (ví dụ như quá trình xoay người khi ngủ hoặc chuyển từ nằm ngửa sang nghiêng). Đây là điều kiện lý tưởng cho cảm biến điện dung MEMS – vốn có độ nhạy cao trong vùng tần số thấp (DC đến vài chục Hz), cho phép nhận diện rõ ràng sự thay đổi hướng trọng lực tác động lên trực cảm biến. Không giống như cảm biến áp điện, vốn chỉ phản hồi với dao động nhanh và không ghi nhận được trạng thái tĩnh, cảm biến điện dung cho phép theo dõi liên tục tư thế cơ thể với độ ổn định tín hiệu cao, phù hợp với đặc điểm sinh lý của giấc ngủ.

Thứ hai, về *yêu cầu hệ thống*, thiết bị theo dõi tư thế ngủ tại nhà cần đảm bảo ba yếu tố: (i) tiêu thụ năng lượng thấp để hoạt động xuyên đêm bằng pin; (ii) kích thước nhỏ gọn để đeo thoải mái mà không ảnh hưởng đến giấc ngủ; và (iii) khả năng tích hợp lên vi điều khiển nhúng (như dòng ARM Cortex-M) nhằm xử lý tín hiệu tại chỗ mà không cần truyền tải liên tục về máy chủ. Cảm biến điện dung MEMS đáp ứng hoàn toàn các yêu cầu này: tiêu thụ năng lượng ở mức vài  $\mu\text{A}$ , kích thước thường chỉ vài  $\text{mm}^2$ , và tín hiệu đầu ra dạng điện áp hoặc giao tiếp I<sup>2</sup>C/SPI – thuận tiện cho việc kết nối với hệ thống nhúng tiêu chuẩn.

Thứ ba, xét về *tính khả thi triển khai*, cảm biến điện dung MEMS có chi phí sản xuất và tích hợp thấp hơn rõ rệt so với các loại cảm biến dùng hiệu ứng áp điện hoặc áp điện trở, vốn yêu cầu mạch xử lý tín hiệu chuyên biệt (như khuếch đại điện tích hoặc cầu đo điện trở chính xác). Hơn nữa, loại cảm biến này có độ phổ biến cao trong chuỗi cung ứng linh kiện, dễ dàng tìm thấy các model tối ưu (như LIS3DH, ADXL345, MMA8451Q...) với đầy đủ tài liệu kỹ thuật, thư viện lập trình và driver mã nguồn mở – giúp rút ngắn thời gian phát triển, tăng tính linh hoạt và khả năng tái sử dụng của hệ thống.

Cuối cùng, việc sử dụng cảm biến điện dung không chỉ đảm bảo độ chính xác đo lường trong môi trường thử nghiệm mà còn mang lại độ tin cậy và tính thực tiễn cao khi triển khai ở quy mô cộng đồng. Thiết kế hệ thống đơn giản, tiêu thụ điện năng thấp, dễ chế tạo và mở rộng – đây là những yếu tố then chốt trong các giải pháp y học số (digital health) hướng tới phòng ngừa sớm và giám sát cá nhân hoá tại nhà.

**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 chẩn đoán hội chứng ngưng thở khi ngủ – đặc biệt là thể ngưng thở tắc nghẽn phụ thuộc tư thế (positional OSA – pOSA) – đòi hỏi một hệ thống giám sát liên tục, có khả năng thu thập dữ liệu sinh lý trong thời gian dài, nhận diện chính xác tư thế ngủ, và đưa ra cảnh báo kịp thời 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 giấc ngủ truyền thống và trí tuệ nhân tạo ứng dụng – một lĩnh vực đang phát triển mạnh mẽ trong thập niên gần đây [20].

### **1.2.1. Quy trình tổng quát trong hệ thống ứng dụng AI cho đánh giá OSA và phân loại tư thế ngủ**

Quy trình này có thể được điều chỉnh tùy theo loại tín hiệu đầu vào và mục tiêu phân tích cụ thể (nhận diện tư thế, phát hiện ngưng thở, theo dõi nhịp thở,...). Tuy nhiên, nguyên tắc cơ bản là đảm bảo tín hiệu đầu vào chất lượng cao và mô hình đủ nhẹ để triển khai thực tế (xem Bảng 1.4).

**Thu thập tín hiệu** là bước đầu tiên và đóng vai trò nền tảng trong toàn bộ quy trình phân tích tư thế ngủ, chẩn đoán hội chứng ngưng thở khi ngủ (OSA) bằng trí tuệ nhân tạo. Trong các hệ thống theo dõi tại nhà (HST), quá trình này được thực hiện thông qua các thiết bị đeo hoặc cảm biến gắn ngoài, với mục tiêu ghi nhận liên tục các tín hiệu sinh lý phản ánh hoạt động hô hấp, tuần hoàn và chuyển động cơ thể trong suốt giấc ngủ. Tín hiệu được ghi nhận có thể bao gồm: gia tốc ba trục nhằm phát hiện chuyển động và tư

Bảng 1.4: Các bước chính trong hệ thống ứng dụng AI cho phân tích tư thê ngủ và hỗ trợ chẩn đoán OSA

<b>STT</b>	<b>Giai đoạn</b>	<b>Mô tả tổng quát</b>
1	<b>Thu thập tín hiệu</b> (Data Acquisition)	Ghi nhận tín hiệu từ cảm biến như gia tốc ba trục, PPG, ECG hoặc áp lực. Thiết bị đeo nhỏ gọn, truyền dữ liệu qua BLE, tốc độ lấy mẫu 10–100 Hz.
2	<b>Tiền xử lý tín hiệu</b> (Preprocessing)	Lọc nhiễu (notch, bandpass), loại bỏ trôi nền, phân đoạn theo cửa sổ thời gian. Mục tiêu là làm sạch và ổn định dữ liệu đầu vào.
3	<b>Trích xuất đặc trưng</b> (Feature Extraction)	Tính toán các đặc trưng thời gian (mean, std, energy...) và tần số (FFT, wavelet), đại diện cho nội dung sinh lý trong từng đoạn tín hiệu.
4	<b>Lựa chọn và huấn luyện mô hình</b> (Model Selection & Training)	Lựa chọn thuật toán học máy (SVM, RF, LR, NN nhẹ) phù hợp với bài toán phân loại tư thê và/hoặc đánh giá OSA.
5	<b>Đánh giá hiệu năng</b> (Evaluation)	Sử dụng các chỉ số đánh giá mô hình như Accuracy, Precision, Recall, F1-Score, AUC, confusion matrix.
6	<b>Tối ưu mô hình</b> (Model Optimization)	Ứng dụng kỹ thuật pruning, quantization (8-bit) để giảm kích thước và độ phức tạp mô hình nhằm phục vụ triển khai biên.
7	<b>Triển khai thực tế</b> (Deployment)	Triển khai mô hình trên vi điều khiển (MCU) hỗ trợ TinyML, tích hợp trong thiết bị đeo nhằm theo dõi tư thê ngủ và ước lượng AHI tại nhà.

thế nằm; tín hiệu PPG để đo nhịp tim và độ bão hòa oxy; tín hiệu ECG để đánh giá hoạt động điện tim; hoặc tín hiệu từ cảm biến hô hấp, cảm biến áp lực gắn trên giường để xác định tư thế ngủ qua phân bố trọng lực. Tuỳ vào mục tiêu nghiên cứu và cấu hình thiết bị, các tín hiệu này được lấy mẫu với tần số phù hợp, thường dao động từ 10 Hz đến 500 Hz [28, 60, 61, 68, 71, 20]. Một trong những yêu cầu quan trọng của bước thu thập là đảm bảo tính ổn định và chất lượng của dữ liệu. Các cảm biến cần được gắn chắc chắn tại những vị trí tối ưu, chẳng hạn như vùng ngực, cổ tay hoặc hõm ức, để giảm thiểu nhiễu do 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 hoặc truyền dữ liệu hiệu quả thông qua các giao thức không dây như Bluetooth Low Energy (BLE) hoặc Wi-Fi [19, 17]. Trong một số nghiên cứu gần đây, các hệ thống HST hiện đại đã sử dụng vi điều khiển công suất thấp tích hợp cảm biến gia tốc, cho phép truyền dữ liệu liên tục với mức tiêu thụ năng lượng tối ưu. Ví dụ, trong nghiên cứu [28], thiết bị dán cổ tích hợp cảm biến gia tốc LIS2DH12 và vi điều khiển nRF5232 đã được sử dụng để thu tín hiệu ở tần số 100 Hz và truyền không dây qua BLE, đảm bảo dữ liệu ổn định và chính xác trong điều kiện thực tế tại nhà.

Chất lượng của dữ liệu thu được trong bước này ảnh hưởng trực tiếp đến hiệu quả của toàn bộ hệ thống. Do đó, quá trình thu thập tín hiệu cần được thiết kế sao cho vừa đáp ứng được yêu cầu về độ chính xác sinh lý, vừa đảm bảo tính khả thi khi triển khai lâu dài trong môi trường ngoài lâm sàng. Tuy nhiên, trong nhiều nghiên cứu hiện nay, cấu trúc phần mềm cho quá trình thu thập và lưu trữ dữ liệu vẫn chưa được trình bày một cách đầy đủ và thống nhất. Các yếu tố như kiến trúc hệ thống (ứng dụng di động, web, hay máy chủ trung tâm), giao thức truyền thông (MQTT, HTTP), phương thức lưu trữ (cục bộ, cloud, hoặc cơ sở dữ liệu liên kết), cũng như các vấn đề liên quan đến mã hóa – bảo mật dữ liệu sức khỏe – và tuân thủ quy định của cơ quan chuyên môn (chuẩn HIPAA, GDPR, hoặc quy định y tế quốc gia) thường chưa được đề cập rõ ràng. Những yếu tố này có thể ảnh hưởng đến khả năng mở rộng, tích hợp, và ứng dụng thực tiễn của hệ thống trong môi trường chăm sóc sức khỏe tại nhà.

## Tiền xử lý dữ liệu

**Tiền xử lý dữ liệu** là một giai đoạn quan trọng nhằm cải thiện chất lượng tín hiệu, chuẩn hóa định dạng đầu vào và đảm bảo tính đồng nhất của dữ liệu trước khi đưa vào mô

hình học máy. Các kỹ thuật tiền xử lý phổ biến bao gồm lọc tín hiệu (filtering), nội suy (interpolation), chuẩn hóa dữ liệu (normalization), phân đoạn tín hiệu (Segmentation – Epoch Creation).

**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, SpO<sub>2</sub> và gia tốc. Các loại nhiễu thường gặp bao gồm: nhiễu điện lưới (50–60 Hz), chuyển động cơ thể (motion artifacts), nhiễu tần số cao và trôi đường cơ sở (baseline drift) [sheta2021osa , 72]. Trong nghiên cứu của Sheta và cộng sự [73], tác giả đã sử dụng bộ lọc notch IIR bậc hai nhằm loại bỏ nhiễu điện lưới 60 Hz khỏi tín hiệu ECG — vốn là loại nhiễu phổ biến gây khó khăn cho việc phân tích và trích xuất đặc trưng hình thái. 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. Các kỹ thuật lọc phổ biến bao gồm notch filter (đặc biệt là IIR notch filter bậc hai) để loại bỏ nhiễu điện lưới, bandpass filter để giữ lại dải tần sinh lý, và các bộ lọc mượt như Butterworth bậc ba, median filter hoặc moving average filter. 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 được sử dụng rộng rãi để làm mượt chuỗi tín hiệu thời gian và giảm nhiễu ngẫu nhiên, còn total variation filter được đánh giá cao trong việc bảo toàn biên chuyển động rõ nét [74, 66]. 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 bước cần thiết để chuẩn hóa chuỗi thời gian. Các phương pháp như cubic spline, piecewise cubic Hermite (PCHIP) hay linear interpolation thường được áp dụng để điều chỉnh các chuỗi dữ liệu về cùng tần số chuẩn, đảm bảo đầu vào đồng bộ cho mô hình học máy hoặc mạng nơ-ron. Ngoài ra, đối với các chuỗi tín hiệu rời rạc như nhịp RR từ ECG hoặc các chuỗi SpO<sub>2</sub> có thể bị gián đoạn hoặc mất mẫu, kỹ thuật nội suy còn đóng vai trò quan trọng trong việc làm đầy các giá trị bị thiếu (missing/null values) và tái cấu trúc tín hiệu thành chuỗi liên tục, giúp mô hình học được đặc trưng đầy đủ hơn từ dữ liệu [75]. Trong nghiên cứu [76], cả hai tín hiệu gia tốc (ACC) và xung quang học (PPG) đều được nội suy để tạo thành chuỗi thời gian đồng nhất với tần số lấy mẫu 32 Hz. Đối với các tín hiệu có tần số cao hơn 32 Hz, một bộ lọc thông thấp (lowpass filter) được áp dụng trước khi giảm tần số (down-sampling) nhằm hạn chế hiện tượng aliasing. Ngược lại, các tín hiệu có tần số lấy mẫu thấp hơn 32 Hz được nội suy bằng phương pháp đa thức Hermite từng đoạn (PCHIP – Piecewise Cubic Hermite Interpolation Polynomial)

để khớp với tần số mong muốn.

**Chuẩn hóa dữ liệu** giúp đưa các đặc trưng đầu vào về cùng một miền giá trị, tránh hiện tượng các đặc trưng có biên độ lớn chi phối quá trình học. Một số phương pháp phổ biến bao gồm: Z-score normalization (chuẩn hóa về trung bình 0 và độ lệch chuẩn 1), min-max scaling (đưa về khoảng [0,1]), và robust scaling (dựa trên trung vị và IQR, phù hợp với dữ liệu có nhiễu hoặc ngoại lệ). Trong các nghiên cứu về OSA, chuẩn hóa đã được áp dụng cho nhiều loại tín hiệu như SpO<sub>2</sub>, PPG và gia tốc, giúp tăng độ ổn định mô hình và cải thiện tốc độ hội tụ khi huấn luyện mạng nơ-ron [28, 30, 72].

**Phân đoạn tín hiệu** là một bước thiết yếu trong xử lý tín hiệu sinh lý, đặc biệt trong các hệ thống theo dõi vị trí giấc ngủ và phát hiện ngưng thở khi ngủ (OSA). Mục tiêu của bước này là chia chuỗi dữ liệu liên tục thành các đoạn thời gian ngắn cố định (epoch), giúp mô hình học máy nhận diện hiệu quả các đặc trưng biến đổi theo thời gian. Thời lượng epoch 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. Trong một số trường hợp, việc phân đoạn còn dựa vào các đặc điểm sinh lý như đỉnh sóng R trong ECG. Sau khi phân đoạn, các bước tiền xử lý như lọc nhiễu, chuẩn hóa hoặc làm mượt được áp dụng trong từng đoạn để nâng cao chất lượng dữ liệu đầu vào cho mô hình học máy [28, 30, 27, 75, 20].

Tóm lại, tiền xử lý tín hiệu 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 biến đổi dữ liệu cảm biến thành tập hợp các đặc trưng có ý nghĩa, phản ánh các mẫu hình sinh lý liên quan đến mục đích của nghiên cứu. Quá trình này giúp làm nổi bật những thông tin cốt lõi từ tín hiệu sinh học như ECG, PPG, SpO<sub>2</sub>, gia tốc hay tín hiệu hô hấp, phục vụ cho các mô hình học máy trong việc phân loại tư thế ngủ hoặc phát hiện hội chứng ngưng thở khi ngủ (OSA) [13, 27, 63, 77, 71]. Các đặc trưng thường được sử dụng bao gồm biến thiên nhịp tim (HRV), khoảng RR, 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, thông số

Hjorth từ EEG. Ngoài ra, các đặc trưng từ tín hiệu PPG và gia tốc như entropy, phương sai hoặc số lượng chuyển động cũng đóng vai trò quan trọng trong việc phản ánh mức độ rối loạn hô hấp trong giấc ngủ. Ngoài các đặc trưng truyền thống trong miền thời gian và tần số, các đặc trưng hỗn hợp như biến đổi wavelet đã chứng minh vai trò đặc biệt quan trọng trong việc mô tả tín hiệu sinh lý một cách toàn diện hơn. Cụ thể, biến đổi wavelet cho phép phân tích tín hiệu ở nhiều thang thời gian khác nhau, đồng thời bảo toàn được cả thông tin về thời điểm và cấu trúc tần số của tín hiệu. Điều này đặc biệt hữu ích trong bối cảnh tín hiệu sinh lý thường không ổn định (non-stationary), như tín hiệu ECG, PPG hoặc tín hiệu gia tốc trong quá trình theo dõi tư thế ngủ. Các hệ số wavelet thu được từ tín hiệu gốc có thể phản ánh rõ ràng sự thay đổi chu kỳ hô hấp, chuyển động cơ thể, cũng như các đặc điểm bất thường liên quan đến cơn ngưng thở khi ngủ. Nhờ đó, việc trích xuất đặc trưng wavelet đóng góp quan trọng vào hiệu suất của các mô hình phân loại tư thế ngủ và phát hiện OSA [28].

**Gán nhãn (labeling)** 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”. Một số nghiên cứu còn mở rộng sang các nhãn liên quan đến chất lượng tín hiệu (“Good” vs. “Poor”), kiểu hô hấp (“Bình thường”, “Bất thường”), các nhãn liên quan đến tư thế ngủ. Việc gán nhãn đúng đắn giúp huấn luyện mô hình AI chính xác và đánh giá mức độ nghiêm trọng của OSA một cách hiệu quả. 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.

**Phát hiện đỉnh (peak detection)** là kỹ thuật thường được áp dụng để trích xuất thông tin từ tín hiệu dạng sóng, đặc biệt là ECG và PPG. Trong các nghiên cứu về OSA, đỉnh sóng R từ ECG thường được xác định để tính toán HRV, sử dụng các thuật toán như Pan–Tompkins, Hamilton hoặc ngưỡng thích nghi (adaptive threshold) [20]. Với tín hiệu PPG, đỉnh tâm thu được phát hiện thông qua các thuật toán lặp hoặc wavelet nhị phân để xác định các khoảng thời gian giữa hai đỉnh (peak-to-peak intervals). Kỹ thuật này giúp nâng cao độ chính xác trong việc phát hiện các thay đổi sinh lý, góp phần cải thiện năng lực chẩn đoán tự động OSA trong các hệ thống ứng dụng trí tuệ nhân tạo.

## Lựa chọn và huấn luyện mô hình

Các kỹ thuật học máy và kiến trúc mạng nơ-ron nhân tạo đã được ứng dụng hiệu quả trong các bài toán phân loại tư thế ngủ và chẩn đoán hội chứng ngưng thở khi ngủ (OSA). Học máy (Machine Learning – ML) cung cấp một tập hợp đa dạng các thuật toán cho phép khai thác mẫu hình từ dữ liệu cảm biến, hỗ trợ phát hiện những thay đổi trong tư thế cơ thể và các rối loạn hô hấp liên quan đến giấc ngủ. Trong khi đó, mạng nơ-ron nhân tạo (Artificial Neural Networks – ANN), đặc biệt là các kiến trúc sâu như CNN, LSTM hoặc hybrid CNN-LSTM, thể hiện ưu thế trong việc xử lý tín hiệu sinh lý phi tuyến, phức tạp, có độ biến thiên cao theo thời gian như ECG, SpO<sub>2</sub> hay tín hiệu gia tốc ba trục.

Trong số các **thuật toán học máy truyền thống** được sử dụng phổ biến, Random Forest (RF) [21] nổi bật nhờ khả năng kháng chống lại quá khớp và độ chính xác cao. Trong nghiên cứu [22], Wang và cộng sự đã ứng dụng RF để phân loại các trường hợp ngưng thở khi ngủ, đạt độ chính xác 93.88%, độ nhạy 89.93% và độ đặc hiệu 91.8%. Một nghiên cứu khác [55], Yeo và cộng sự sử dụng RF cho nhiệm vụ phân loại sự kiện hô hấp, thu được độ chính xác 83%, độ nhạy 99% và F1-score 81%. Mặc dù có sự khác biệt về nguồn dữ liệu và phương pháp trích chọn đặc trưng, RF vẫn cho thấy hiệu quả vượt trội khi so sánh với các thuật toán khác như SVM, LDA hay QDA [22], [55], [78].

Bên cạnh đó, thuật toán SVM [23] cũng đã được áp dụng nhằm xác định siêu phẳng tối ưu để phân loại các nhóm trong không gian đặc trưng. Trong nghiên cứu [22], Wang cũng sử dụng thêm mô hình SVM và đạt độ chính xác 88,28%, độ đặc hiệu 91,69% và độ nhạy 83,94%, cho thấy hiệu quả cao trong phát hiện ngưng thở khi ngủ, dù mô hình Random Forest thường có kết quả cao hơn. Ở nghiên cứu [55], SVM đạt độ chính xác 83% và hệ số Cohen's kappa 0,53 trong phân loại sự kiện hô hấp theo từng phút. Trong [78], SVM được huấn luyện trên tín hiệu ECG một kênh, đạt độ chính xác 69,13%, góp phần cải thiện hiệu suất của hệ thống phân loại khi tích hợp trong mô hình tổ hợp. Những kết quả này cho thấy SVM vẫn là một phương pháp có giá trị trong ứng dụng học máy cho chẩn đoán ngưng thở khi ngủ.

K-Nearest Neighbors (KNN) [24] là một thuật toán khác cũng thường xuyên được áp dụng trong các nghiên cứu về phát hiện ngưng thở khi ngủ [22], [50]. Dựa trên nguyên lý đo độ tương đồng trong không gian đặc trưng, KNN phân loại một điểm dữ liệu mới

dựa trên nhãn của các điểm lân cận gần nhất. Wang và cộng sự đã dùng mô hình KNN xử lý tín hiệu quang học PPG và đạt độ chính xác 85.06%, với độ đặc hiệu 86.11% và độ nhạy 83.72% [22]. Trong khi đó, nghiên cứu [50] báo cáo hiệu quả vượt trội hơn với accuracy lên đến 95%, đồng thời vẫn đảm bảo thời gian thực thi đáp ứng yêu cầu hệ thống. Thành công này được cho là đến từ khả năng đo lường chính xác độ tương đồng giữa dữ liệu quan sát và dữ liệu đã học, giúp mô hình KNN đưa ra dự đoán phù hợp với mức độ nghiêm trọng của OSA.

Bên cạnh các thuật toán truyền thống, mô hình XGBoost [26] cũng được đưa vào thử nghiệm trong nghiên cứu [22] nhằm đánh giá khả năng phân loại các mức độ ngưng thở khi ngủ. Là một biến thể của thuật toán boosting, XGBoost được thiết kế tối ưu cho hiệu suất tính toán và có khả năng xử lý hiệu quả cả bài toán hồi quy và phân loại. Kết quả cho thấy XGBoost đạt độ chính xác 82.05%, độ đặc hiệu 84.91% và độ nhạy 78.42%, cho thấy tiềm năng lớn của mô hình này trong ứng dụng lâm sàng, đặc biệt trong các hệ thống đòi hỏi cân bằng giữa độ chính xác và tốc độ huấn luyện. Trong nghiên cứu [55], thuật toán Linear Discriminant Analysis (LDA) [25] được đánh giá là một phương pháp quan trọng. LDA sử dụng trung bình và ma trận hiệp phương sai của từng lớp để xác định ranh giới quyết định tối ưu, nhằm tối đa hóa sự phân biệt giữa các lớp và giảm thiểu phương sai nội bộ. Trong bối cảnh nghiên cứu, LDA cho thấy hiệu quả vượt trội trong phát hiện sự kiện hô hấp với độ chính xác 81%, độ nhạy 88%, độ đặc hiệu 79% và điểm F1 đạt 81%.

**Học sâu (Deep learning - DL)** là một lĩnh vực thuộc học máy tập trung vào việc khai thác các mạng nơ-ron nhân tạo nhiều lớp để trích xuất đặc trưng và mô hình hóa dữ liệu phức tạp. Các mô hình học sâu đã chứng minh hiệu quả trong việc phát hiện tư thế ngủ, đánh giá tình trạng OSA. Học sâu cho phép phân tích các tín hiệu sinh lý không tuyến tính, có nhiều và mang tính thời gian, chẳng hạn như tín hiệu PPG, ECG, SpO<sub>2</sub> và gia tốc kế.

Khác với các mô hình học máy truyền thống vốn đòi hỏi kỹ thuật trích xuất đặc trưng thủ công, các mạng học sâu có khả năng tự động học các đặc trưng biểu diễn từ dữ liệu đầu vào thông qua quá trình huấn luyện. Điều này đặc biệt hữu ích trong phát hiện hội chứng ngưng thở khi ngủ (OSA – *Obstructive Sleep Apnea*), nơi tín hiệu thu nhận từ thiết bị đeo thường biến thiên theo tư thế, giai đoạn ngủ và tình trạng sinh lý của từng bệnh nhân.

Trong các ứng dụng chẩn đoán OSA, nhiều kiến trúc học sâu đã được đề xuất và triển khai, bao gồm mạng nơ-ron tích chập (Convolutional Neural Networks – CNN), mạng nơ-ron hồi tiếp (Recurrent Neural Networks – RNN) với các biến thể như LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Unit), mạng nơ-ron truyền thẳng (Feedforward Neural Networks – FNN), và mạng perceptron đa tầng (Multilayer Perceptron – MLP). Ngoài ra, các mô hình học chuyển tiếp (Transfer Learning) sử dụng các kiến trúc sâu như ResNet và ResNeXt cũng được chứng minh mang lại hiệu quả cao trong bối cảnh dữ liệu hạn chế [20, 22, 72, 76]. Gần đây, các mô hình kết hợp giữa học sâu và phương pháp phân cụm phân cấp (Hierarchical Deep Clustering – HDC) cũng được áp dụng nhằm giảm chi phí tính toán và mở rộng khả năng triển khai trên các thiết bị vi xử lý nhúng.

Trong [72], Rossi và cộng sự đã phát triển một mô hình mạng nơ-ron sâu (DNN) được phát triển để xử lý tín hiệu siêu âm thu từ cảm biến đeo ngực, theo hướng kết hợp đa cảm biến. Mô hình được thiết kế cho bài toán phân loại ba chiều: chất lượng tín hiệu, các mẫu hô hấp và các mẫu giấc ngủ. Mô hình đạt độ chính xác 96% trong phân biệt tín hiệu nhiễu và bình thường, với độ chính xác cao hơn trong dự đoán mẫu hô hấp (0.93) so với mẫu giấc ngủ (0.76). Trong [76], một mạng nơ-ron tích chập sâu (CNN) được triển khai nhằm phân loại các giai đoạn ngủ và phát hiện ngưng thở khi ngủ từ thiết bị CST (consumer sleep technologies) đeo tay, sử dụng tín hiệu từ cảm biến gia tốc và PPG. Mô hình gồm hai luồng xử lý riêng biệt, và kết quả cho thấy hiệu suất được cải thiện khi sử dụng dữ liệu thô, đặc biệt trong phát hiện sự kiện apnea trong giấc ngủ REM. Ji et al và cộng sự triển khai mô hình Long Short-Term Memory (LSTM) – một dạng mạng nơ-ron hồi tiếp (RNN) – nhằm phân tích chuỗi tín hiệu sinh lý phục vụ phát hiện hội chứng ngưng thở – giảm thở khi ngủ (SAHS) [79]. Mô hình gồm ba lớp lặp LSTM kết hợp với bốn lớp kết nối đầy đủ, có số nút lần lượt là 128, 64, 32 và 1. Kết quả cho thấy mô hình đạt độ chính xác trung bình từ 84–85%, khẳng định tính hiệu quả của kiến trúc LSTM trong việc nhận dạng các đặc trưng liên quan từ dữ liệu tuần tự. Tổng thể, các mô hình học sâu như CNN, LSTM và các biến thể kết hợp đã chứng minh hiệu quả vượt trội trong việc phát hiện OSA từ các tín hiệu sinh lý đa dạng. Tuy nhiên, thách thức về chi phí tính toán, dung lượng bộ nhớ và khả năng triển khai thực tế trên thiết bị đeo yêu cầu các hướng nghiên cứu tối ưu hóa mô hình và giảm độ phức tạp.

Tập trung vào mục tiêu phát hiện các hoạt động thường ngày và tư thế ngủ của con

người, rất nhiều nghiên cứu gần đây đã ứng dụng các phương pháp học máy và trí tuệ nhân tạo để xây dựng hệ thống nhận dạng từ dữ liệu cảm biến đeo được. *Bảng 1.5* dưới đây tổng hợp một số nghiên cứu tiêu biểu trong lĩnh vực này, bao gồm cả các mô hình học máy truyền thống (như cây quyết định, KNN, SVM) và các phương pháp học sâu (CNN, LSTM). Các hệ thống được triển khai với mục đích phân loại tư thế ngủ hoặc hoạt động thể chất, sử dụng nhiều loại dữ liệu cảm biến khác nhau như gia tốc ba trục, tín hiệu quang học (PPG) hoặc đặc trưng miền thời gian. Nhiều nghiên cứu cũng đã chú trọng đến việc triển khai mô hình trên thiết bị nhúng (on-chip), nhằm hướng đến khả năng theo dõi tại nhà trong thời gian thực với chi phí và mức tiêu thụ năng lượng thấp.

Sự phát triển nhanh chóng của trí tuệ nhân tạo (AI) trong lĩnh vực y học đã mở ra nhiều hướng tiếp cận mới trong việc đánh giá tư thế ngủ và chẩn đoán OSA. Các hệ thống AI đang dần chứng minh hiệu quả vượt trội trong việc xử lý dữ liệu cảm biến lớn và phức tạp, từ đó cung cấp các phân tích chính xác về hành vi giấc ngủ của bệnh nhân. Các mô hình học máy được huấn luyện trên tập dữ liệu cảm biến từ accelerometer, gyroscope hoặc thiết bị đeo thông minh có thể tự động phân loại các tư thế ngủ theo thời gian thực, với độ chính xác lên đến trên 90% trong nhiều nghiên cứu gần đây [28][30][27].

**Tiny Machine Learning (TinyML)** là lĩnh vực nghiên cứu và ứng dụng các mô hình học máy có kích thước nhỏ gọn, tiêu thụ năng lượng thấp, được triển khai trực tiếp trên các thiết bị nhúng như vi điều khiển (microcontrollers – MCU) hoặc thiết bị đeo (wearables) [31]. Không giống như các mô hình học máy truyền thống thường yêu cầu nền tảng phần cứng mạnh và tính toán trên cloud, TinyML cho phép xử lý dữ liệu tại chỗ (on-device inference) mà không cần kết nối internet, nhờ đó nâng cao tính riêng tư, độ phản hồi thời gian thực và khả năng hoạt động ngoại tuyến.

Trong lĩnh vực chăm sóc sức khỏe, công nghệ **TinyML** (Tiny Machine Learning) ngày càng được ứng dụng rộng rãi để phân tích các tín hiệu sinh lý như nhịp tim, nhịp thở, chuyển động và tư thế ngủ. Với ưu điểm tiêu thụ năng lượng thấp, chi phí triển khai rẻ và khả năng vận hành lâu dài trên phần cứng hạn chế, các mô hình TinyML đặc biệt phù hợp cho các hệ thống theo dõi y tế tại nhà, chẳng hạn như phát hiện hội chứng ngưng thở khi ngủ (OSA) bằng thiết bị đeo không xâm lấn. Các thuật toán thường được sử dụng trong TinyML bao gồm các biến thể nhẹ và được đơn giản hóa từ các mô hình học sâu như *Convolutional Neural Networks (CNN)*, *Long Short-Term Memory (LSTM)*, cùng với các mô hình cây quyết định gọn nhẹ như *Decision Tree*, *Random Forest* hoặc

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
[80]	18 đối tượng	Tư thế ngủ	12 đặc trưng miền thời gian	DT, Extra-Trees, LSTM-NN	>98%	NM	Có
[60]	Không nêu	Tư thế ngủ	NM	RF, SVM	85%	Có	Không
[81]	561.859 mẫu	Tư thế ngủ	NM	LSTM_sq, CNN, MLP, LSTM_con	>99%	Không	Không
[62]	16 đối tượng	Tư thế ngủ	Trung bình, 3 trục	Ngưỡng cố định	99.5%	Không	Không
[82]	5 đối tượng	Tư thế ngủ	NM	Ngưỡng cố định	>99.2%	Không	Không
[83]	8 đối tượng	Hoạt động người dùng	Trung bình, Độ lệch chuẩn (6)	KNN, RF, SVM	97%	Không	Không
[84]	Public + 20 private	Hoạt động người dùng	31 đặc trưng miền thời gian	RF	97%	Có	Không
[85]	10 đối tượng (public)	Hoạt động người dùng	11 đặc trưng miền thời gian	KNN, NB, SVM, CTree, RF	98%	Không	Không

NM = Không đề cập; DT = Decision Tree; CNN = Convolutional Neural Network; MLP = Multilayer Perceptron;

LSTM = Long Short-Term Memory; RF = Random Forest; SVM = Support Vector Machine; NB = Naive Bayes; CTree = Conditional Inference Tree.

*Gradient Boosting*, vốn đã được tối ưu hoá cho việc hoạt động trên nền tảng vi điều khiển có giới hạn tài nguyên (MCU).

Dữ liệu trong Bảng 1.5 cho thấy sự đa dạng trong cách tiếp cận bài toán nhận diện tư thế ngủ và hoạt động người dùng. Hầu hết các nghiên cứu đạt độ chính xác cao (trên 95%), ngay cả khi sử dụng các thuật toán đơn giản như ngưỡng cố định hoặc mô hình học máy cổ điển như KNN, SVM và RF. Tuy nhiên, một điểm đáng lưu ý là chỉ duy nhất nghiên cứu của Abdulsadig và cộng sự [80] thực sự được triển khai trên phần cứng nhúng (on-chip), 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 hiệu năng cao, chưa đáp ứng được yêu cầu triển khai thực tế trong các thiết bị đeo.

Đi sâu hơn, nhóm mô hình học sâu như CNN, LSTM hoặc kết hợp nhiều lớp (LSTM\_sq, LSTM\_con, MLP) tuy đạt độ chính xác vượt trội (trên 99%) như trong nghiên cứu của Vu et al. [81], 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 đơn giản hơn như Decision Tree, Random Forest 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 (mean, std, energy, entropy) — vốn dễ tính toán và không cần FFT — giúp giảm đáng kể độ phức tạp tính toán. Điều này rất phù hợp với triết lý thiết kế của TinyML: đơn giản, hiệu quả và khả thi khi triển khai thực tế trên các nền tảng hạn chế như MCU.

Một điểm sáng là khả năng cải thiện hiệu năng của mô hình bằng cách lựa chọn đặc trưng phù hợp với tín hiệu đầu vào. Ví dụ, các đặc trưng trung bình và phương sai theo trực trong các nghiên cứu [83], [85] đủ mạnh để phân biệt các tư thế ngủ hoặc hoạt động cơ bản, mà không cần tới biểu diễn phức tạp trong miền tần số hay wavelet.

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 chỉ yêu cầu vài kilobyte bộ nhớ đã trở nên khả thi. Điều này mở ra hướng tiếp cận mới trong chẩn đoán y tế cá nhân hóa, thông qua thiết bị đeo thông minh hoạt động liên tục tại nhà.

Một điểm mạnh nổi bật của các mô hình AI triển khai trên nền TinyML là khả năng học và tự hiệu chỉnh theo dữ liệu thực tế. Nhờ vậy, hệ thống có thể phân biệt hiệu quả giữa các tư thế ngủ cơ bản — nghiêng trái, nghiêng phải, nằm ngửa, và nằm sấp — ngay cả khi có những sai lệch nhỏ về góc đặt thiết bị hoặc chuyển động cơ thể không mong muốn. Hơn nữa, AI còn cho phép tích hợp dữ liệu tư thế với các thông số sinh lý khác như nhịp tim, nhịp thở, độ bão hòa oxy máu ( $\text{SpO}_2$ ), và âm thanh để xây dựng mô hình chẩn đoán OSA đa chiều. Việc tổng hợp đa nguồn tín hiệu này giúp hệ thống:

- Phát hiện chính xác các giai đoạn ngưng thở và giảm thở (apnea/hypopnea),
- Phân tích tác động của từng tư thế ngủ đến mức độ hẹp đường thở,
- Ước lượng mức độ nghiêm trọng của OSA theo từng cá nhân.

Đây là một bước tiến quan trọng, hướng đến mục tiêu cá nhân hóa chẩn đoán và điều trị OSA – điều mà các phương pháp truyền thống như đa ký giấc ngủ (*polysomnography – PSG*) còn nhiều hạn chế do chi phí cao, quy trình phức tạp và tính không đại diện cho điều kiện giấc ngủ tự nhiên. Khi được triển khai trên các thiết bị đeo thông minh, hệ thống AI này cho phép theo dõi liên tục trong thời gian dài mà không gây xâm lấn hay

ánh hưởng đến chất lượng giấc ngủ của người dùng. Dữ liệu thu thập trong điều kiện sinh hoạt tự nhiên tại nhà có độ chính xác cao hơn, giúp nâng cao giá trị chẩn đoán và hỗ trợ quyết định điều trị chính xác hơn cho bác sĩ.

Cuối cùng, sự kết hợp giữa **AI**, **thiết bị đeo** và công nghệ **điện toán biên (edge computing)** mang lại tiềm năng rất lớn cho các hệ thống sàng lọc, chẩn đoán sớm và quản lý cá thể hóa OSA theo thời gian thực — mở ra hướng phát triển bền vững cho y tế thông minh trong kỷ nguyên số.

Đây là một bước tiến quan trọng, hướng đến mục tiêu cá nhân hóa chẩn đoán và điều trị OSA – điều mà đa ký giấc ngủ (*polysomnography – PSG*) truyền thống còn hạn chế do chi phí cao, quy trình phức tạp và tính không đại diện cho điều kiện ngủ tự nhiên. Khi được triển khai trên thiết bị đeo thông minh, hệ thống AI cho phép theo dõi liên tục, không xâm lấn trong bối cảnh sinh hoạt tại nhà, nhờ đó nâng cao độ tin cậy của dữ liệu và hỗ trợ quyết định điều trị chính xác hơn cho bác sĩ. Sự kết hợp giữa **AI**, **thiết bị đeo** và **điện toán biên (edge computing)** vì thế mở ra tiềm năng lớn cho sàng lọc, chẩn đoán sớm và quản lý OSA theo thời gian thực.

**Vì vậy, luận văn được tổ chức theo hai giai đoạn mạch lạc:** (i) *Dánh 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 nhẹ (ngưỡng-hình học, LR, SVM, RF, NN nồng). Tiêu chí lựa chọn không chỉ dựa trên độ chính xác/F1 vi mô-vĩ mô và ma trận nhầm lẫn cho từng tư thế, mà còn cân nhắc tính bền vững trước sai lệch gắn cảm biến, chi phí tính toán và khả năng co giãn khi giảm tần số lấy mẫu. (ii) *Triển khai on-chip (TinyML) trên vi điều khiển mục tiêu* – ánh xạ pipeline suy luận lên nền tảng Cortex-M4 với các tối ưu đặc thù phần cứng (quantization số nguyên, rút gọn đặc trưng, sử dụng FIFO/INT để giảm đánh thức CPU), bảo đảm ràng buộc bộ nhớ–độ trễ–năng lượng, đồng thời kiểm chứng sai lệch hiệu năng so với giai đoạn (i) ở mức chấp nhận được cho ứng dụng thực tế.

Cấu trúc hai giai đoạn này phản ánh lộ trình từ bằng chứng thuật toán trên dữ liệu đến khả năng vận hành ngoài đời thực trên thiết bị đeo chi phí thấp, qua đó hiện thực hóa mục tiêu cá thể hóa chẩn đoán và theo dõi OSA trong môi trường tại nhà.

## Chương 2

# XÂY DỰNG HỆ THỐNG THU THẬP DỮ LIỆU, HUẤN LUYỆN MÔ HÌNH VÀ CHUẨN BỊ TRIỂN KHAI TRÊN THIẾT BỊ BIÊN.

### 2.1. Phần cứng thực nghiệm

Trong thiết kế hệ thống giám sát tư thế ngủ, quá trình lựa chọn phần cứng được xây dựng theo một lộ trình rõ ràng:

- Xác định mục tiêu:** Hệ thống cần đạt độ chính xác cao trong phân loại tư thế ngủ, tiêu thụ năng lượng thấp để hoạt động suốt đêm, đồng thời có khả năng triển khai mô hình TinyML trực tiếp trên thiết bị đeo.
- Lựa chọn cảm biến:** Dựa trên đặc tính chuyển động chậm – biên độ nhỏ trong giấc ngủ, cảm biến gia tốc số với độ nhạy cao, dải đo hẹp và khả năng tích hợp tốt được ưu tiên.

3. **Lựa chọn vi xử lý:** Trên cơ sở yêu cầu về xử lý tín hiệu thời gian thực và triển khai mô hình học máy nhẹ, dòng ARM Cortex-M4 được chọn nhờ sự cân bằng giữa hiệu năng, năng lượng và hỗ trợ thư viện DSP/TinyML.
4. **Lựa chọn kit phát triển:** Sau khi xác định được cảm biến và vi xử lý, bước cuối cùng là chọn nền tảng phần cứng cụ thể. Luận văn sử dụng hai kit: (i) bo Adafruit Playground với nRF52840 + LIS3DH để thu thập dữ liệu và huấn luyện mô hình; (ii) Arduino Nano 33 BLE Sense để triển khai mô hình tối ưu trên thiết bị đeo thực tế.
5. **Nghiên cứu và thiết kế phần cứng riêng:** Bên cạnh việc sử dụng kit thương mại nhằm rút ngắn quá trình thử nghiệm, luận văn cũng hướng tới việc chế tạo bo mạch chuyên dụng tích hợp cảm biến và vi xử lý đã chọn. Mục tiêu là tạo ra nguyên mẫu thiết bị đeo có kích thước nhỏ gọn, tối ưu năng lượng, đáp ứng chuẩn y sinh và sẵn sàng cho các nghiên cứu mở rộng trong tương lai.

Cách tiếp cận này đảm bảo hệ thống phần cứng vừa có tính thực tiễn trong giai đoạn thử nghiệm, vừa có tính định hướng dài hạn trong phát triển sản phẩm hoàn chỉnh, từ đó tạo cơ sở cho việc chuyển giao công nghệ và ứng dụng lâm sàng

### 2.1.1. Cảm biến

Trong quá trình ngủ, các chuyển động thân thể chủ yếu là chuyển động chậm, với biên độ nhỏ và không mang tính đột ngột. Các chuyển động thân thể chủ yếu mang tính chậm và thường xảy ra trong giai đoạn ngủ không chuyển động mắt nhanh (NREM), khi cơ thể có khả năng tự do thay đổi tư thế. Ngược lại, trong giai đoạn ngủ REM, hiện tượng ức chế trương lực cơ khiến 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, khả năng phân giải tốt và ổn định với nhiều nền thấp. 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 kích thước nhỏ gọn, tiêu thụ năng lượng thấp, tần số lấy mẫu phù hợp và đặc biệt là độ nhạy cao với chuyển động cường độ thấp.

Trong nghiên cứu của Vu và cộng sự (2023), dữ liệu tư thế ngủ được thu thập thông qua một thiết bị đeo đặt tại vùng bụng của người tham gia. Thiết bị này tích hợp cảm

biến gia tốc ba trục ADXL345, bộ điều khiển ESP8266 và pin Lithium, tất cả được đóng gói trong một hộp nhựa nhỏ gọn [81]. Trong nghiên cứu của Boiko và cộng sự, cảm biến gia tốc ba trục ADXL355z được sử dụng để thu nhận tín hiệu hô hấp từ vùng ngực và bụng, với tần số lấy mẫu 62 Hz. Đây là cảm biến có độ nhiễu thấp, độ trôi nhiệt nhỏ và phù hợp với các ứng dụng y sinh. Trước đó, dòng cảm biến này cũng đã được ứng dụng thành công trong các phép đo tim-phổi [58]. Dữ liệu từ ADXL355z được đổi chiều với tín hiệu chuẩn thu từ dây đeo hô hấp của hệ thống SOMNO HD eco PSG, nhằm đảm bảo độ chính xác trong đánh giá tín hiệu sinh lý trong khi ngủ. Trong nghiên cứu của Abdulsadig và cộng sự, dữ liệu gia tốc được thu thập bằng bo mạch điện tử thiết kế riêng tích hợp cảm biến gia tốc ba trục LIS2DH12 (STMicroelectronics) và vi điều khiển nRF5232 (Nordic Semiconductor) [28].

Việc lựa chọn cảm biến gia tốc phù hợp đòi hỏi phải cân nhắc đồng thời nhiều yếu tố kỹ thuật nhằm đảm bảo độ chính xác đo lường, khả năng tích hợp phần cứng và tính khả thi khi triển khai thực tế trong thiết bị đeo.

### **Dải đo động (Full-Scale Range):**

Một trong những tham số quan trọng hàng đầu trong thiết kế hệ thống đo lường là *dải đo động* (full-scale range) của cảm biến gia tốc. Đối với bài toán giám sát tư thế ngủ, các chuyển động của cơ thể thường diễn ra chậm và có biên độ thấp – chủ yếu dưới 1g, tương ứng với các thao tác xoay người hoặc thay đổi tư thế nhẹ nhàng trong khi ngủ. Vì vậy, lựa chọn dải đo nhỏ như  $\pm 2g$  hoặc  $\pm 4g$  là hợp lý nhằm tối ưu hóa độ phân giải của cảm biến.

Về mặt kỹ thuật, độ phân giải của cảm biến phụ thuộc trực tiếp vào dải đo và số bit của bộ chuyển đổi tín hiệu nội bộ. Chẳng hạn, với cảm biến có độ phân giải 16-bit và dải đo  $\pm 2g$ , độ phân giải lý thuyết đạt được sẽ vào khoảng:

$$\frac{2 \times 2g}{2^{16}} \approx 0.000061g/LSB$$

Ngược lại, nếu mở rộng dải đo lên đến  $\pm 16g$ , độ phân giải bị giảm xuống còn:

$$\frac{2 \times 16g}{2^{16}} \approx 0.000488g/LSB$$

Sự suy giảm độ phân giải này có thể khiến cảm biến bỏ sót các dao động vi mô đặc trưng, làm ảnh hưởng đến hiệu quả phân loại tư thế hoặc phát hiện các chuyển động nhẹ trong trạng thái ngủ.

Từ đó, có thể thấy rằng việc lựa chọn dải đo động phù hợp là yếu tố then chốt trong tối ưu hiệu năng của hệ thống. Việc đặt dải đo quá lớn để “bao phủ toàn bộ biên độ có thể xảy ra” sẽ dẫn đến lãng phí độ phân giải và giảm độ nhạy trong các tình huống thực tế vốn không yêu cầu dải đo rộng. Do đó, việc lựa chọn dải đo  $\pm 2g$  là lựa chọn ưu tiên trong hệ thống theo dõi tư thế ngủ với yêu cầu chính xác cao và chuyển động cường độ thấp.

### **Tín hiệu đầu ra (Analog vs. Digital):**

Một trong những tiêu chí quan trọng trong thiết kế hệ thống cảm biến nhúng là lựa chọn loại tín hiệu đầu ra – giữa cảm biến đầu ra tương tự (*analog output*) và cảm biến đầu ra số (*digital output*). Trong khuôn khổ hệ thống theo dõi tư thế ngủ, cảm biến kỹ thuật số được ưu tiên vì nhiều lợi thế vượt trội về tính ổn định, khả năng tích hợp và đơn giản hóa mạch xử lý tín hiệu. Trước hết, cảm biến đầu ra số không yêu cầu bổ sung bộ chuyển đổi tương tự-số (ADC) rời. Trong khi cảm biến analog cần mạch ADC để số hóa tín hiệu trước khi xử lý, cảm biến digital thực hiện trực tiếp quá trình này trong nội bộ mạch tích hợp. Điều này giúp giảm đáng kể độ phức tạp trong thiết kế phần cứng, tiết kiệm diện tích PCB, và giảm chi phí linh kiện. Hơn nữa, tín hiệu tương tự dễ bị suy giảm do nhiễu điện từ, đặc biệt khi truyền qua dây dẫn dài hoặc trong môi trường nhiễu cao như hệ thống đeo gần cơ thể người. Ngược lại, tín hiệu số được truyền dưới dạng xung nhị phân có khả năng chống nhiễu tốt hơn, đảm bảo độ chính xác trong truyền dẫn và làm tăng độ tin cậy của hệ thống. Ngoài ra, các cảm biến kỹ thuật số hiện đại thường tích hợp sẵn các khối chức năng bổ trợ như: bộ lọc thông thấp phần cứng (hardware low-pass filter) giúp loại bỏ nhiễu tần số cao, bộ đệm FIFO (*first-in-first-out*) để giảm yêu cầu đọc dữ liệu liên tục từ MCU, và thậm chí cả bộ hiệu chuẩn nội tại tự động (auto-calibration unit). Những tính năng này góp phần giảm đáng kể khối lượng tính toán cần thiết từ phía vi điều khiển, từ đó tiết kiệm năng lượng hệ thống và cải thiện hiệu suất vận hành, đặc biệt trong các ứng dụng thời gian thực và nền tảng tính toán giới hạn như thiết bị đeo.

### **Chuẩn giao tiếp (Communication Interface):**

Xét về yếu tố giao tiếp giữa cảm biến và vi điều khiển, chuẩn I<sup>2</sup>C được lựa chọn là phương án tối ưu cho hệ thống theo dõi tư thế ngủ. Ưu điểm lớn nhất của I<sup>2</sup>C nằm ở thiết kế đơn giản – chỉ yêu cầu hai đường tín hiệu (SCL và SDA) cho toàn bộ giao tiếp dữ liệu. Điều này giúp tiết kiệm số chân phần cứng trên vi điều khiển và đặc biệt phù hợp trong

các thiết bị đeo cá nhân, nơi không gian mạch in (PCB) và tài nguyên cổng GPIO đều rất hạn chế. Trong khi đó, chuẩn SPI dù có ưu điểm về tốc độ truyền dữ liệu cao hơn và khả năng truyền song song hiệu quả, lại đòi hỏi ít nhất bốn dây kết nối (SCLK, MISO, MOSI, CS) và thường tiêu tốn nhiều chân vi điều khiển hơn. Hơn nữa, với đặc thù của bài toán giám sát tư thế ngủ – vốn không yêu cầu tốc độ truyền dữ liệu cao do tín hiệu thay đổi chậm và tần số lấy mẫu thấp (chỉ từ 25–50 Hz) – thì lợi thế của SPI không thực sự cần thiết. Từ các phân tích trên, có thể kết luận rằng I<sup>2</sup>C là giao thức phù hợp nhất cho hệ thống, không chỉ đảm bảo khả năng truyền dữ liệu ổn định mà còn giúp đơn giản hóa thiết kế phần cứng, giảm thiểu mức tiêu thụ năng lượng và diện tích PCB – những yếu tố then chốt trong thiết kế thiết bị đeo có kích thước và tài nguyên hạn chế.

### Tần số lấy mẫu (Sampling Rate):

Các chuyển động liên quan đến thay đổi tư thế khi ngủ thường mang tính chậm rãi, diễn ra trong thời gian dài và chủ yếu xuất hiện trong giai đoạn ngủ không chuyển động mắt nhanh (NREM). Khác với các ứng dụng đo rung động công nghiệp hoặc phát hiện va chạm – vốn đòi hỏi tần số lấy mẫu cao (trên 100 Hz) để ghi nhận các thay đổi nhanh, thì bài toán phân loại tư thế ngủ chỉ yêu cầu theo dõi các chuyển động có tần số thấp, thường nằm trong khoảng dưới 1–2 Hz. Theo định lý Nyquist, để thu thập chính xác các tín hiệu có tần số tối đa  $f_{max}$ , hệ thống cần lấy mẫu ở tần số tối thiểu là  $2f_{max}$ . Với giả định biên độ dao động tư thế xảy ra ở dải tần 0.3–3 Hz, tần số lấy mẫu 10 Hz là đủ để đảm bảo không xảy ra hiện tượng aliasing và vẫn ghi nhận được đầy đủ thông tin về các thay đổi tư thế theo thời gian. Ngoài ra, việc lựa chọn mức 10 Hz mang lại nhiều lợi ích thiết thực cho hệ thống nhúng: giúp giảm lưu lượng dữ liệu đầu ra, tiết kiệm bộ nhớ lưu trữ, giảm tần suất đánh thức vi điều khiển, từ đó tối ưu thời lượng pin – yếu tố then chốt trong các thiết bị đeo theo dõi giấc ngủ hoạt động liên tục suốt đêm.

**Tiêu thụ điện năng** là một trong những yếu tố then chốt trong thiết kế hệ thống giám sát tư thế ngủ, đặc biệt khi mục tiêu là triển khai trên thiết bị đeo hoạt động liên tục suốt đêm mà không gây phiền toái cho người dùng. Do đó, cảm biến cần hỗ trợ chế độ *ultra-low power*, với mức dòng tiêu thụ dưới 10  $\mu$ A ở chế độ chờ (*standby*) hoặc khi vận hành ở tần số thấp. Khả năng tiết kiệm năng lượng này không chỉ kéo dài thời lượng pin, mà còn giảm thiểu yêu cầu về dung lượng pin và trọng lượng thiết bị, qua đó nâng cao trải nghiệm sử dụng và mức độ chấp nhận của người dùng trong môi trường thực tế.

Ngoài ra, cảm biến được lựa chọn cũng cần đảm bảo **kích thước vật lý nhỏ gọn** để

dễ dàng tích hợp vào các PCB có không gian hạn chế. Việc tích hợp sẵn các tính năng hỗ trợ như bộ phát hiện chuyển động (*motion detection*), bộ lọc phần cứng (*hardware filtering*) hay tự hiệu chỉnh (*auto-calibration*) cũng là những lợi thế quan trọng. Các chức năng này giúp giảm độ phức tạp của hệ thống tổng thể, tiết kiệm tài nguyên xử lý trên vi điều khiển, đồng thời cải thiện khả năng phản hồi thời gian thực của hệ thống, đặc biệt trong các ứng dụng yêu cầu xử lý tại chỗ như *TinyML* trên thiết bị đeo.

Từ các phân tích trên, có thể xác định rằng cảm biến gia tốc lý tưởng cho bài toán giám sát tư thế ngủ cần thỏa mãn đồng thời các tiêu chí: dải đo động nhỏ (từ  $\pm 2g$  đến  $\pm 4g$ ), đầu ra kỹ thuật số để đơn giản hóa phần cứng và giảm nhiễu, hỗ trợ giao tiếp qua giao thức I<sup>2</sup>C để tiết kiệm chân kết nối, tiêu thụ năng lượng cực thấp nhằm duy trì hoạt động liên tục suốt đêm, và có khả năng tích hợp cao nhằm giảm tải tính toán cho hệ thống vi điều khiển.

Dựa trên các tiêu chí trên, luận văn lựa chọn hai dòng cảm biến tiêu biểu do hãng **STMicroelectronics** sản xuất là **LIS3DH** và **LSM6DS3** để triển khai thực nghiệm. Đây là hai cảm biến phổ biến trong các thiết bị đeo và ứng dụng IoT y sinh nhờ sự cân bằng tốt giữa hiệu năng, độ ổn định và tính khả thi triển khai thực tế.

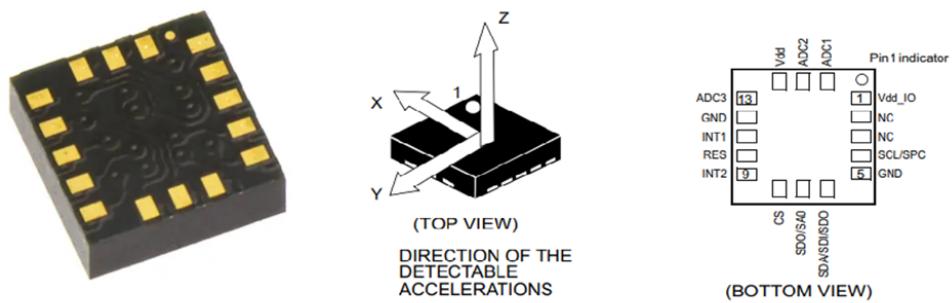
Cả hai cảm biến đều đáp ứng các yêu cầu kỹ thuật sau:

- Hỗ trợ dải đo động từ  $\pm 2g$  đến  $\pm 16g$ , cho phép linh hoạt cấu hình theo nhu cầu đo lường, với độ phân giải từ 12 đến 16 bit.
- Tín hiệu đầu ra dạng số (digital), hỗ trợ giao tiếp qua I<sup>2</sup>C và SPI, tương thích với đa số vi điều khiển phổ biến.
- Hỗ trợ chế độ tiết kiệm năng lượng (*ultra-low power*), với mức tiêu thụ dòng dưới  $10 \mu A$  trong các chế độ nghỉ hoặc đo chu kỳ thấp.
- Có khả năng lập trình ngưỡng phát hiện chuyển động, tích hợp bộ lọc thông thấp và bộ đếm tín hiệu nội bộ, giúp giảm tải xử lý phần mềm.
- Kích thước vật lý nhỏ gọn (khoảng  $3 \times 3$  mm), dễ tích hợp trên PCB của thiết bị đeo cá nhân.

Trong đó, cảm biến **LIS3DH** nổi bật nhờ cấu hình linh hoạt, độ nhạy tốt và đã được ứng dụng hiệu quả trong nhiều nghiên cứu theo dõi chuyển động chậm và phân loại tư

thế ngũ. Trong khi đó, cảm biến **LSM6DS3** là một cảm biến kết hợp (IMU – Inertial Measurement Unit), tích hợp thêm con quay hồi chuyển (gyroscope), cung cấp khả năng mở rộng nghiên cứu trong tương lai như nhận diện vi chuyển động hoặc phân tích tư thế ba chiều chi tiết hơn.

Vì vậy, cả hai cảm biến được đưa vào thử nghiệm nhằm đánh giá toàn diện các yếu tố gồm: độ nhạy, mức độ nhiễu, khả năng nhận diện tư thế và mức tiêu thụ năng lượng trong điều kiện thực tế sử dụng tại nhà. Hình 2.1 minh họa hình dạng thực tế và sơ đồ chân kết nối của cảm biến **LIS3DH**.



Hình 2.1: Cảm biến gia tốc LIS3DH và sơ đồ chân kết nối

### 2.1.2. Vi xử lý

Với sự phát triển vượt bậc và đa dạng của công nghệ 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, [86] các tác giả đã sử dụng máy tính đơn Raspberry Pi kết hợp các điện trở cảm biến lực để phát hiện 4 tư thế ngủ với sự lấy nhãn từ video theo dõi người bệnh trong suốt quá trình lấy mẫu. Kwasnicki và cộng sự đã phát triển hệ thống ngủ có thể đeo (wearable sleep system) 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 InvenSense ITG-3200, Honeywell HMC5843 để đo từ trường xác định 99.5% chính xác 4 tư thế ngủ [62]. Tuy nhiên, các thiết bị vẫn yêu cầu một nguồn năng lượng khiếu cho tính liên tục bị hạn chế đáng kể. I.Yun và cộng sự đã phát triển thiết bị theo dõi tư thế ngủ của trẻ nhỏ sử dụng vi xử lý ATmega328P-PU cùng module Bluetooth kết hợp cảm biến gia tốc ADXL335 được đặt trên bụng đã nhưng lựa chọn về mặt cấu hình thiết bị và chế tạo ra mạch cung cấp năng lượng cho những thành phần cần thiết [56]. Từ đó, giảm thiểu đáng kể mức tiêu thụ năng lượng và vẫn giữ nguyên

độ chính xác nhưng khá bất tiện cho trẻ nhỏ. 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 (Nordic Semiconductor) – một SoC thuộc dòng ARM Cortex-M4F, hỗ trợ truyền thông không dây thông qua giao thức Bluetooth Low Energy (BLE). Vi điều khiển này đảm nhiệm đồng thời cả việc lấy mẫu dữ liệu từ cảm biến gia tốc ba trục LIS2DH12 (STMicroelectronics) với tần số 100 Hz và truyền dữ liệu không dây theo thời gian thực [28, 80]. Trong nghiên cứu của Vũ Hoàng Diệu và cộng sự, 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 [81]. 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 và an toàn dữ liệu trong điều kiện mất kết nối mạng.

Tuy nhiên, qua phân tích các nghiên cứu trên có thể thấy rằng phần lớn các cấu hình phần cứng hiện tại hoặc có chi phí triển khai cao, hoặc tiêu tốn năng lượng, hoặc gặp giới hạn trong khả năng tích hợp mô hình học máy tại thiết bị. Do đó, việc lựa chọn một kiến trúc vi xử lý vừa đảm bảo hiệu suất xử lý tín hiệu sinh lý thời gian thực, vừa tối ưu năng lượng và có khả năng triển khai mô hình TinyML là cần thiết. Trong số các kiến trúc hiện nay, dòng ARM Cortex-M4 nổi bật nhờ tính cân bằng giữa hiệu năng, mức tiêu thụ năng lượng thấp và khả năng hỗ trợ xử lý tín hiệu số, phù hợp với các hệ thống đeo được trong theo dõi tư thế ngủ.

Kiến trúc ARM có nhiều dòng vi xử lý khác nhau, được phát triển và nâng cấp liên tục nhằm đáp ứng nhu cầu đa dạng trong lĩnh vực công nghệ nhúng. Trong đó, dòng Cortex-M thuộc kiến trúc ARMv7 đã trở thành nền tảng phổ biến cho các hệ thống nhúng sử dụng vi điều khiển nhờ vào hiệu suất cao, khả năng mở rộng và mức tiêu thụ năng lượng tối ưu. Dòng Cortex-M bao gồm nhiều phiên bản như Cortex-M0, Cortex-M0+, Cortex-M1, Cortex-M3, Cortex-M4 và Cortex-M7, mỗi phiên bản được thiết kế để phục vụ cho các mức độ yêu cầu hiệu năng khác nhau [87]. Các vi xử lý thuộc họ Cortex-M chủ yếu được ứng dụng trong các hệ thống nhúng thời gian thực, nơi yêu cầu sự cân bằng giữa hiệu suất xử lý, tiêu thụ năng lượng và chi phí. Một số vi xử lý ARM khác, không thuộc họ Cortex-M, được sử dụng trong các thiết bị hiệu suất cao như điện thoại thông

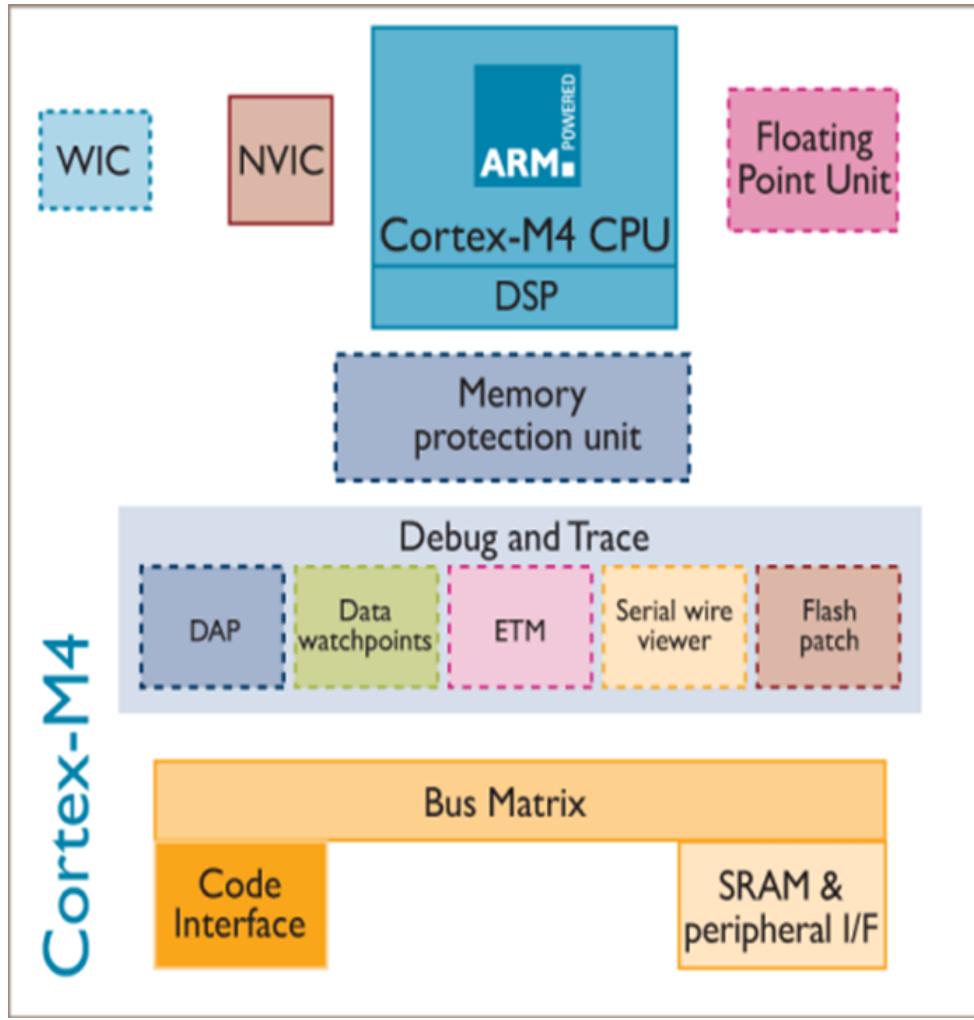
minh và máy tính bảng, vốn yêu cầu cấu hình phần cứng mạnh hơn và khả năng xử lý đa tác vụ cao hơn. Theo tài liệu [88], vi xử lý Cortex-M4 là một bộ xử lý 32-bit sử dụng kiến trúc tập lệnh rút gọn (RISC), được xây dựng theo kiến trúc Harvard, trong đó bus dữ liệu và bus lệnh được tách biệt nhằm tối ưu hiệu suất truy xuất bộ nhớ. Vi xử lý này hỗ trợ đầy đủ cả tập lệnh Thumb-1 (16-bit) và Thumb-2 (hỗn hợp 16/32-bit), mang lại sự linh hoạt trong mã hóa lệnh và tiết kiệm không gian bộ nhớ chương trình.

Về hiệu năng, Cortex-M4 đạt từ 1,25 đến 1,95 DMIPS/MHz (Dhrystone Million Instructions Per Second per MHz), cho thấy khả năng xử lý hiệu quả trong các ứng dụng nhúng yêu cầu độ chính xác và độ phản hồi thời gian thực cao. Bên cạnh đó, vi xử lý hỗ trợ tối đa 240 tín hiệu ngắn, bao gồm cả ngắn không thể bị chặn (Non-Maskable Interrupts – NMI), cùng khả năng cấu hình từ 8 đến 256 mức ưu tiên ngắn, giúp hệ thống hoạt động ổn định trong môi trường có nhiều sự kiện cạnh tranh đồng thời. Ngoài ra, hiện nay ứng dụng trí tuệ nhân tạo (AI) tại thiết bị biên (Edge AI) đang ngày càng phổ biến, đặc biệt trong các lĩnh vực như nhà thông minh, thiết bị đeo, giám sát an ninh và công nghiệp 4.0. Với khả năng xử lý tín hiệu số (DSP) và hỗ trợ các mạng nơ-ron nhỏ gọn, các vi xử lý Cortex-M, đặc biệt là dòng Cortex-M4, đang được khai thác để triển khai các mô hình học sâu nhẹ (tinyML) ngay trên vi điều khiển [89][90].

Kết nối bus được mô tả trong Hình 2.2 cho phép truyền dữ liệu đồng thời trên nhiều bus khác nhau, đồng thời cung cấp khả năng quản lý truyền dữ liệu hiệu quả, chẳng hạn như sử dụng bộ đệm ghi và điều khiển hướng bit hoạt động (bit-banding). Hệ thống cũng có thể bao gồm các cầu bus (bus bridges) nhằm kết nối nhiều loại bus vào một mạng duy nhất sử dụng chung không gian bộ nhớ. Ngoài ra, bộ xử lý được trang bị hệ thống hỗ trợ gỡ lỗi tích hợp, bao gồm khả năng kiểm soát gỡ lỗi, thiết lập điểm ngắn (breakpoint) chương trình và điểm theo dõi dữ liệu (watchpoint). Khi xảy ra sự kiện gỡ lỗi, hệ thống có thể tạm dừng trạng thái hoạt động của lỗi xử lý để phục vụ việc phân tích và xử lý lỗi.

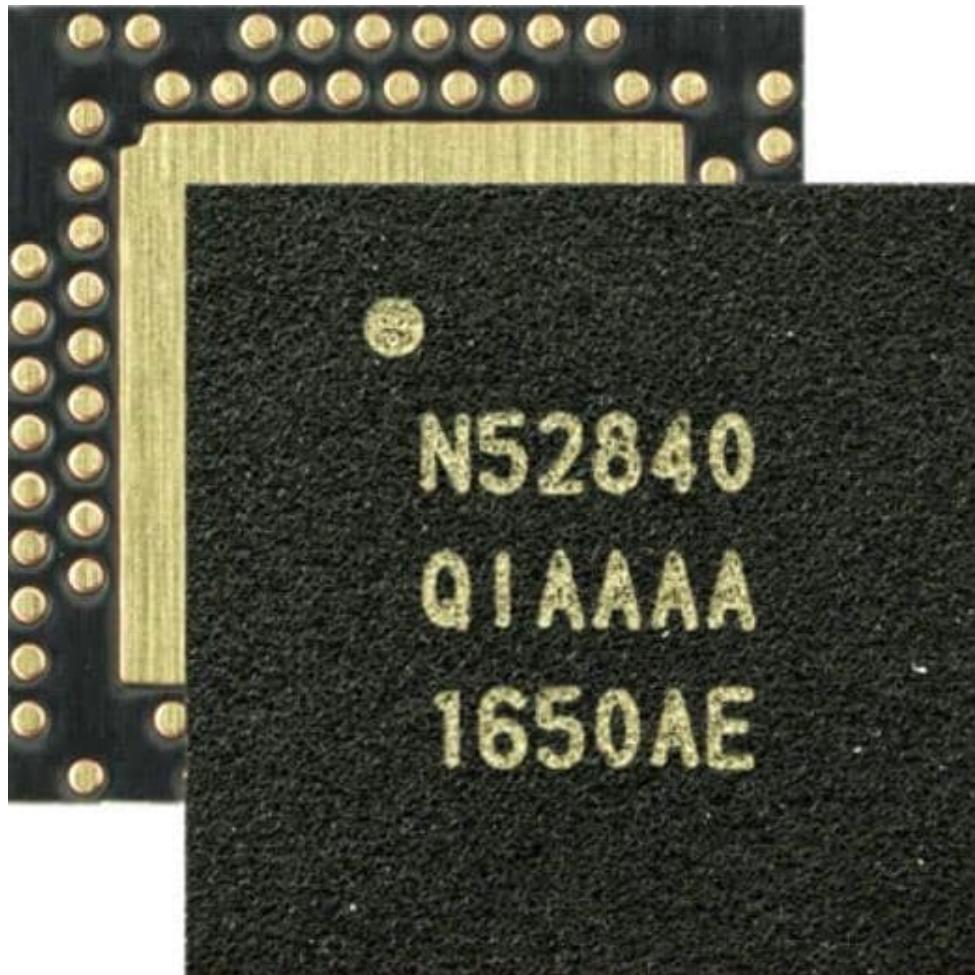
Bên cạnh đó, kiến trúc Cortex-M4 tích hợp Bộ điều khiển ngắn vectored lồng nhau (Nested Vectored Interrupt Controller – NVIC) với khả năng hỗ trợ lên đến 240 tín hiệu yêu cầu ngắn, bao gồm cả ngắn không chặn được (NMI). NVIC hỗ trợ xử lý ngắn lồng nhau một cách tự động bằng cách so sánh mức ưu tiên giữa các yêu cầu ngắn với mức ưu tiên hiện tại đang được xử lý.

Đối với các ứng dụng yêu cầu tiết kiệm năng lượng, hệ thống còn được trang bị bộ đánh thức ngắn (Wake-up Interrupt Controller – WIC), cho phép đưa bộ vi điều khiển



Hình 2.2: Thành phần chính của vi điều khiển Cortex-M4

vào chế độ nghỉ bằng cách tắt hầu hết các thành phần không cần thiết, đồng thời duy trì khả năng đánh thức hệ thống khi phát hiện một yêu cầu ngắn. Ngoài ra, cơ chế bảo vệ bộ nhớ cũng được tích hợp nhằm đảm bảo an toàn cho hệ thống, ví dụ như chỉ cho phép truy cập đọc tại một số vùng bộ nhớ hoặc ngăn người dùng truy cập vào các vùng dữ liệu đặc quyền của hệ điều hành hoặc ứng dụng hệ thống.



Hình 2.3: Nordic Semiconductor NRF52840

Sau quá trình khảo sát và so sánh các dòng vi xử lý phổ biến, tác giả lựa chọn nRF52840 (Nordic Semiconductor) làm nền tảng phần cứng cho hệ thống đề xuất, nhờ vào các ưu điểm nổi bật như kích thước nhỏ, tiêu thụ năng lượng thấp và tích hợp sẵn giao tiếp Bluetooth Low Energy (BLE). Đây là vi xử lý cao cấp nhất trong dòng nRF52, thuộc loại hệ thống trên một vi mạch (System-on-Chip – SoC), được thiết kế chuyên biệt cho các ứng dụng không dây tầm ngắn và tiết kiệm năng lượng [91].

**nRF52840** tích hợp bộ thu phát đa giao thức 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 dâu phẩy động (FPU). Vi xử lý này được trang bị bộ nhớ 1 MB Flash và 256 KB RAM, hỗ trợ chuẩn Bluetooth 5.3 cùng khả năng giao tiếp đa giao thức (multiprotocol), cho phép cải thiện tốc độ, phạm vi truyền và độ tin cậy của kết nối không dây. Hệ thống bảo mật tích hợp đầy đủ, bao gồm các tính năng mã hóa phần cứng, đáp ứng yêu cầu khắt khe về bảo vệ dữ liệu. 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: tối đa hai giao diện I2C, bốn SPI master, ba SPI slave, bốn kênh PWM hỗ trợ EasyDMA, cùng với năm bộ định thời 32-bit, phù hợp cho các ứng dụng đòi hỏi xử lý thời gian thực chính xác. Tất cả các đặc điểm trên khiến nRF52840 trở thành lựa chọn lý tưởng cho các hệ thống nhúng đeo được tích hợp AI nhẹ và kết nối không dây thông minh.

Ngoài ra, nRF52840 hỗ trợ một hệ sinh thái phần mềm mạnh mẽ, bao gồm SDK của Nordic và nền tảng TensorFlow Lite for Microcontrollers, giúp rút ngắn thời gian phát triển và triển khai hệ thống TinyML. Thiết bị còn sở hữu khả năng quản lý năng lượng linh hoạt, tương thích tốt với nguồn pin hoặc USB.

### **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, Bluetooth Low Energy (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.

Ngược lại, các công nghệ truyền thông không dây khác đều bộc lộ những hạn chế khó khắc phục trong bối cảnh thiết bị theo dõi giấc ngủ cá nhân. Wi-Fi, tuy có ưu thế về băng thông và khả năng truy cập trực tiếp Internet, lại tiêu thụ năng lượng cao và mất nhiều thời gian để khởi tạo kết nối, điều này đi ngược với yêu cầu tiết kiệm pin và hoạt động liên tục của thiết bị đeo cỡ nhỏ. Bluetooth cổ điển (BR/EDR), 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, trong khi hệ thống theo dõi tư thế ngủ chỉ cần trao đổi các gói dữ liệu nhỏ giọt. ZigBee, mặc dù hiệu quả trong các mạng cảm biến nhà thông minh nhờ cấu trúc mesh, lại thiếu sự hỗ trợ trên điện thoại thông minh và đòi hỏi gateway riêng, gây bất tiện cho mô hình thiết bị cá nhân. Trong khi đó, LoRa/LoRaWAN nổi bật

với khả năng truyền xa và tiết kiệm năng lượng, nhưng tốc độ dữ liệu rất thấp (chỉ vài kbps), không thể đáp ứng yêu cầu truyền tải gần thời gian thực trong giám sát giấc ngủ.

**Tóm lại**, so với các giao thức không dây khác, BLE đạt sự cân bằng tối ưu giữa năng lượng, tốc độ, độ trễ và tính phổ biến. Đây là cơ sở để luận văn lựa chọn BLE làm chuẩn kết nối chính, bảo đảm vừa duy trì độ chính xác của dữ liệu cảm biến, vừa kéo dài thời lượng pin, đồng thời tạo thuận lợi cho việc triển khai thực tế trên quy mô công đồng.

	nRF52805	nRF52810	nRF52811	nRF52820	nRF52832	nRF52833	nRF52840	nRF5340
Bluetooth 5.3	X	X	X	X	X	X	X	X
Bluetooth 2 Mbps	X	X	X	X	X	X	X	X
Bluetooth Long Range				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 2.4: Các kiểu kết nối không dây trong họ chip nRF52

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. Với ưu thế tiêu thụ năng lượng tối thiểu nhưng vẫn đảm bảo tốc độ truyền phù hợp, BLE đặc biệt thích hợp cho các thiết bị y sinh hoạt động liên tục bằng pin có dung lượng hạn chế. BLE hiện được hỗ trợ phổ biến trên hầu hết các hệ điều hành như iOS, Android, macOS, Windows 10 và Linux, cũng như trong các thiết bị di động hiện

đại.

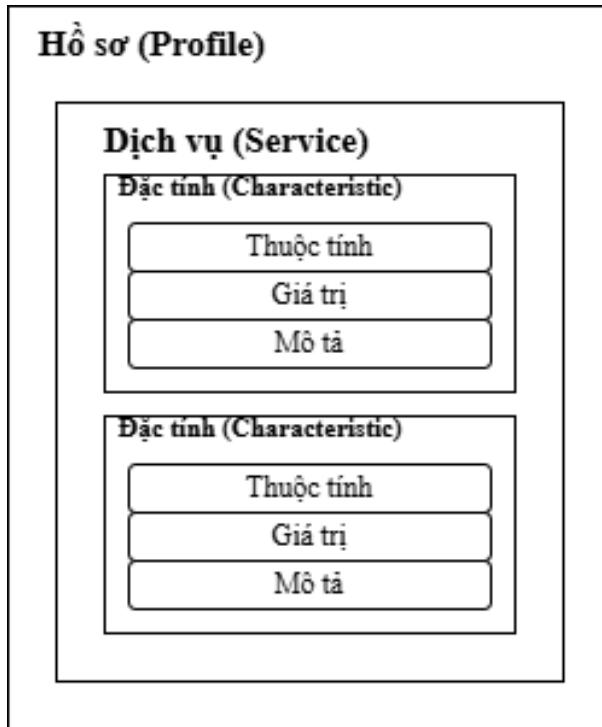
Về mặt bảo mật, BLE tích hợp các cơ chế mã hóa và xác thực nhằm đảm bảo tính bí mật, toàn vẹn và riêng tư của dữ liệu truyền qua mạng. Công nghệ này đã trở thành một phần tiêu chuẩn trong hầu hết các thiết bị di động hiện đại như smartphone, máy tính bảng, và laptop, đồng thời được hỗ trợ đầy đủ trên các hệ điều hành phổ biến bao gồm iOS, Android, macOS, Windows 10 và Linux. 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. Một trong những cải tiến nổi bật là chế độ 2 Mbps, cho phép tăng gấp đôi tốc độ truyền lý thuyết, tương ứng với thông lượng thực tế lên đến 1.4 Mbps. Quan trọng hơn, chế độ này còn giúp giảm đáng kể mức tiêu thụ năng lượng – cụ thể là giảm một nửa năng lượng tiêu thụ trên mỗi bit dữ liệu – từ đó kéo dài thời gian hoạt động của thiết bị hoặc cho phép sử dụng các nguồn năng lượng nhỏ và chi phí thấp hơn [92].

Bên cạnh đó, tính năng Advertising Extensions (mở rộng quảng cáo) đã cách mạng hóa cơ chế phát sóng của BLE. Các gói quảng cáo giờ đây có thể chứa lượng dữ liệu gấp 8 lần so với phiên bản trước, cho phép truyền tải các khối dữ liệu lớn hơn mà không cần thiết lập kết nối ngay lập tức. Đồng thời, các gói quảng cáo có thể được xâu chuỗi để tạo thành các tập tin quảng cáo phức hợp. Tính năng lựa chọn kênh được tối ưu hóa giúp tăng cường độ ổn định và khả năng chống nhiễu trong các môi trường có mật độ thiết bị cao. Đặc biệt, chế độ Long Range mở rộng đáng kể phạm vi truyền thông của BLE, cho phép các thiết bị duy trì kết nối trong toàn bộ không gian của một ngôi nhà thông minh hoặc trong các ứng dụng IoT công nghiệp quy mô vừa và nhỏ.

BLE tổ chức logic giao tiếp dựa trên mô hình GATT (Generic Attribute Profile). 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) – nơi mỗi đặc tính được định danh bằng UUID 16-bit hoặc 128-bit, với quyền truy cập như chỉ đọc, chỉ ghi, hoặc hỗ trợ thông báo (notify).

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



Hình 2.5: Cấu trúc của GATT

lưới nhiều-nút (many-to-many), sử dụng BLE làm lớp truyền tải vật lý. Mỗi nút trong mạng có thể đóng vai trò chuyển tiếp (relay), cho phép dữ liệu lan truyền đến các vùng rộng hơn theo mô hình phân tán – phù hợp với các ứng dụng IoT quy mô lớn như nhà thông minh, chiếu sáng công nghiệp hoặc giám sát phân tán. Trong mạng Mesh, các gói dữ liệu có thể được đóng gói qua advertising packet hoặc qua các giao tiếp GATT tùy tình huống sử dụng.

Các profile BLE là tập hợp các dịch vụ được chuẩn hóa bởi Bluetooth SIG hoặc định nghĩa tùy chỉnh, ví dụ như dịch vụ UART tùy chỉnh gồm hai đặc tính RX và TX, tương ứng với kênh nhận và truyền.

#### 2.1.4. Thiết bị thực nghiệm

Việc lựa chọn nền tảng phần cứng thử nghiệm có ý nghĩa quyết định đối với toàn bộ nghiên cứu, bởi nó không chỉ ảnh hưởng đến độ tin cậy của dữ liệu và khả năng huấn luyện mô hình mà còn định hình hướng phát triển thiết bị trong các giai đoạn tiếp theo. Sau khi xác định được loại cảm biến và vi xử lý phù hợp, luận văn triển khai song song hai nền tảng kit thương mại nhằm đáp ứng hai mục tiêu khác nhau: thu thập dữ liệu huấn

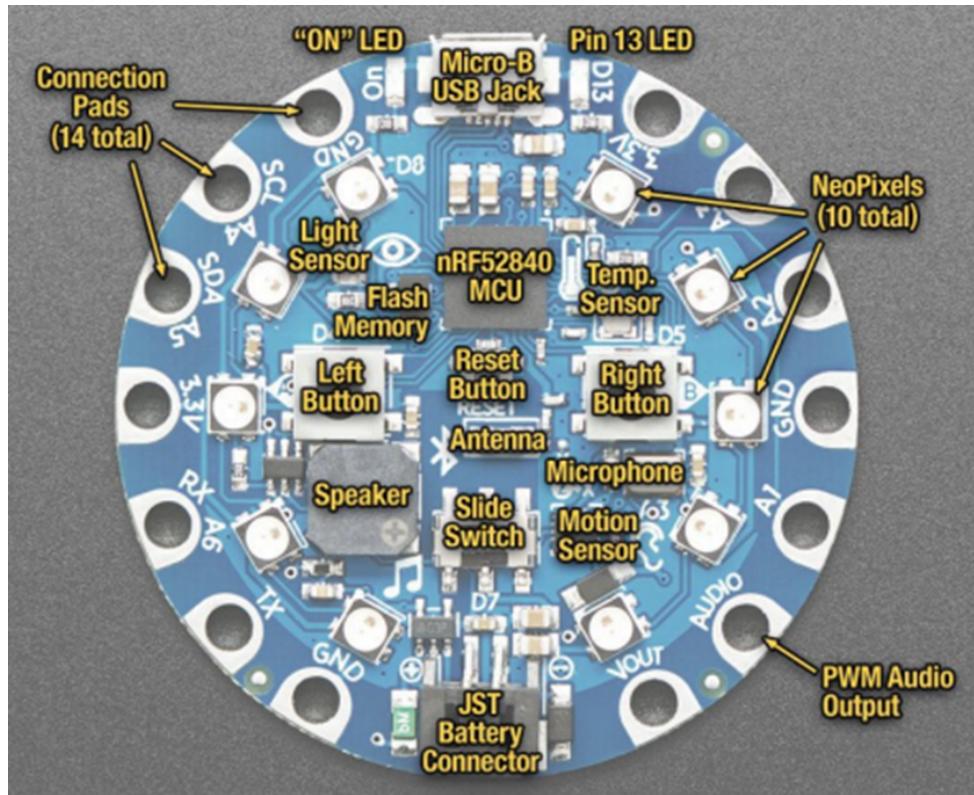
luyện và đánh giá khả năng triển khai mô hình học máy nhẹ trên thiết bị đeo thực tế.

Việc sử dụng đồng thời hai nền tảng mang lại ý nghĩa khoa học quan trọng. Một mặt, nó giúp kiểm chứng mô hình trên các điều kiện phần cứng khác nhau qua đó làm rõ khả năng tổng quát hóa và tính ổn định của hệ thống. Mặt khác, cách tiếp cận này tránh được sự lệ thuộc vào một cấu hình duy nhất, chứng minh rằng mô hình có thể tái triển khai và thích ứng trên nhiều nền tảng phần cứng cùng họ vi xử lý. Nhờ đó, nghiên cứu không chỉ dừng lại ở việc xây dựng một nguyên mẫu cụ thể, mà còn góp phần khẳng định tính khả chuyển và độ tin cậy của giải pháp trong các ứng dụng thực tiễn lâu dài.

Thứ nhất, bộ kit *Adafruit Playground* được lựa chọn cho giai đoạn thu thập dữ liệu và thực nghiệm sơ bộ. Kit này tích hợp sẵn cảm biến gia tốc MEMS LIS3DH, cho phép đo gia tốc theo ba trục không gian (X, Y, Z) với độ chính xác cao. Cảm biến được kết nối với vi điều khiển thông qua giao thức SPI, trong đó chân chọn thiết bị (CS) được gán tại chân số 8 và đầu ra ngắn (IRQ) tại chân số 7 (IRQ #4). Theo cấu trúc bố trí tiêu chuẩn, trục X được định hướng theo chiều giác USB, trục Y hướng sang bên trái, trong khi trục Z vuông góc với mặt phẳng thiết bị. Với chi phí chỉ khoảng 25 USD/kit [93], nền tảng này mang lại sự cân bằng hợp lý giữa tính kinh tế và độ tin cậy, cho phép triển khai nhanh chóng hệ thống thu thập dữ liệu trong giai đoạn đầu.

Thứ hai, để kiểm chứng tính khả thi trong triển khai mô hình tối ưu và đánh giá hiệu năng thực tế của hệ thống đeo, luận văn sử dụng thêm bo mạch *Arduino Nano 33 BLE Sense*. Đây là nền tảng hiện đại dựa trên vi xử lý nRF52840 (ARM Cortex-M4F), hỗ trợ Bluetooth Low Energy (BLE) và tích hợp sẵn nhiều cảm biến (IMU, microphone, nhiệt độ, độ ẩm, ...). Đặc biệt, bo mạch này tương thích trực tiếp với *TensorFlow Lite for Microcontrollers* [94], nhờ đó có thể triển khai và đánh giá trực tiếp các mô hình TinyML trong môi trường nhúng.

Đáng chú ý, bên cạnh việc tận dụng các bộ kit thương mại, nhóm nghiên cứu đồng thời phát triển một phiên bản mạch phần cứng tùy chỉnh dựa trên những phân tích ở các chương trước. Hướng tiếp cận này vừa bảo đảm tiến độ triển khai trong giai đoạn thử nghiệm, vừa đặt nền tảng cho việc xây dựng một thiết bị nhúng chuyên dụng, tối ưu hơn về chi phí, hiệu năng và khả năng tích hợp trong các ứng dụng giám sát tư thế ngủ và hỗ trợ chẩn đoán ngưng thở tắc nghẽn khi ngủ.



Hình 2.6: Cấu trúc các thành phần trên Circuit Playground

## 2.2. Hệ thống thu thập, xử lý, lưu trữ dữ liệu

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/ngắt 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

---

```
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

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)

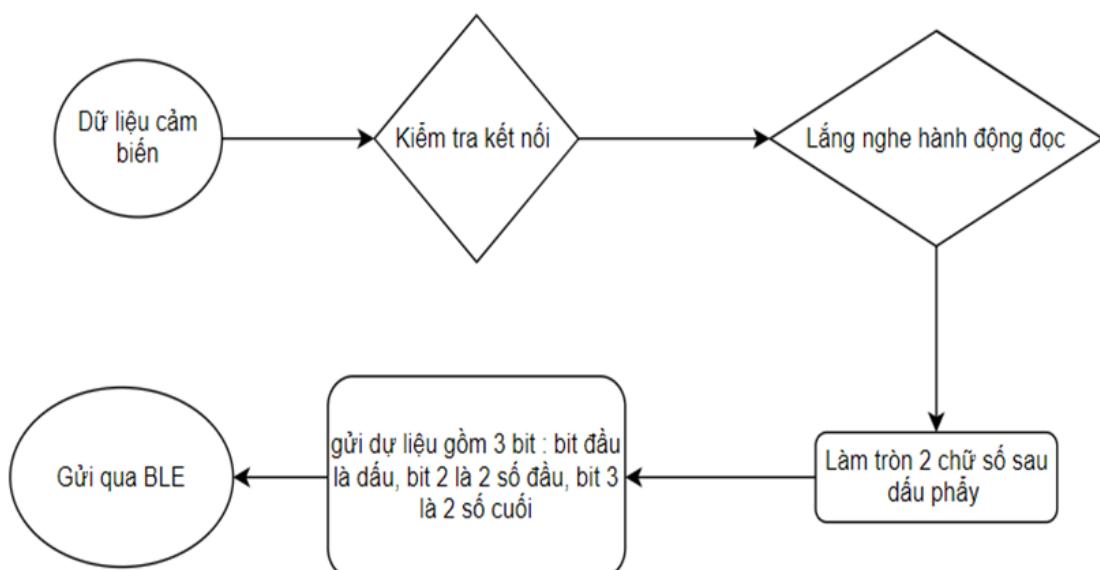
CHR\_PROPS\_NOTIFY gửi thông báo không xác nhận (bit 4)

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



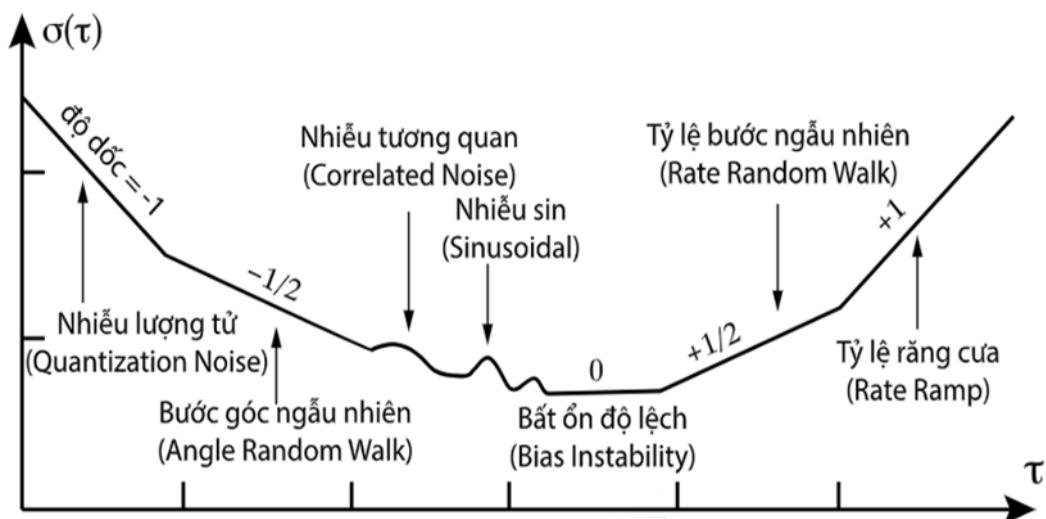
Hình 2.7: Lưu đồ luồng gửi thông tin BLE

Luồng xử lý dữ liệu BLE được minh họa tại Hình 2.7. 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 gia 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.

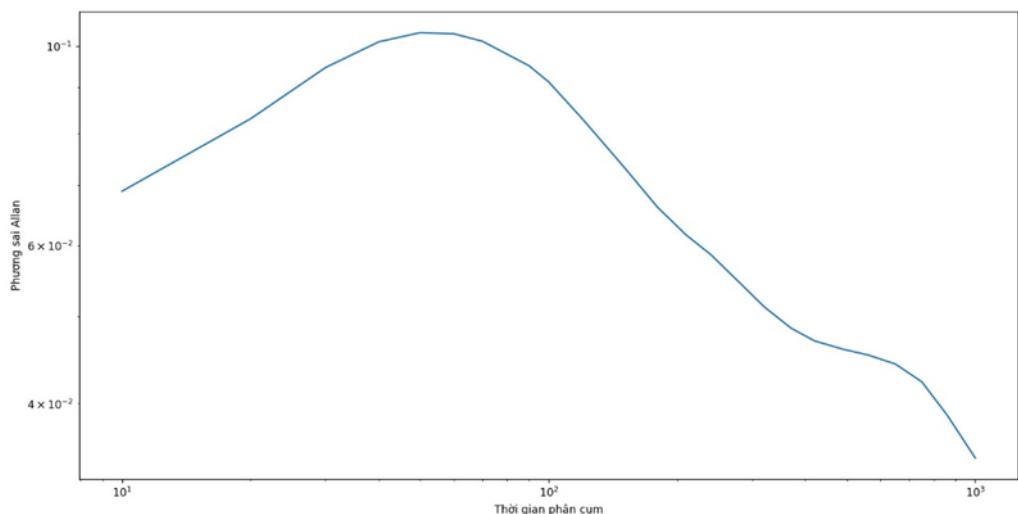


Hình 2.8: Minh họa kết quả phân tích đường cong Allan

**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 [95]. 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 2.11 cho thấy nhiễu chiếm ưu thế là nhiễu lượng tử (quantization noise), đặc trưng bởi hệ số góc tương ứng trong đồ thị.



Hình 2.9: Biểu đồ phương sai Allan của trục X

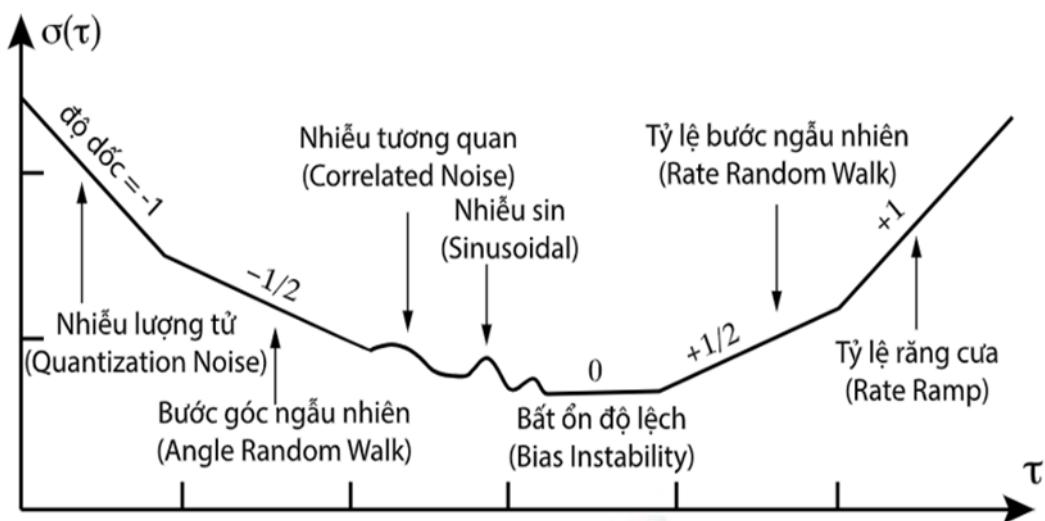
### 2.2.3. Hiệu chuẩn cảm biến

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.

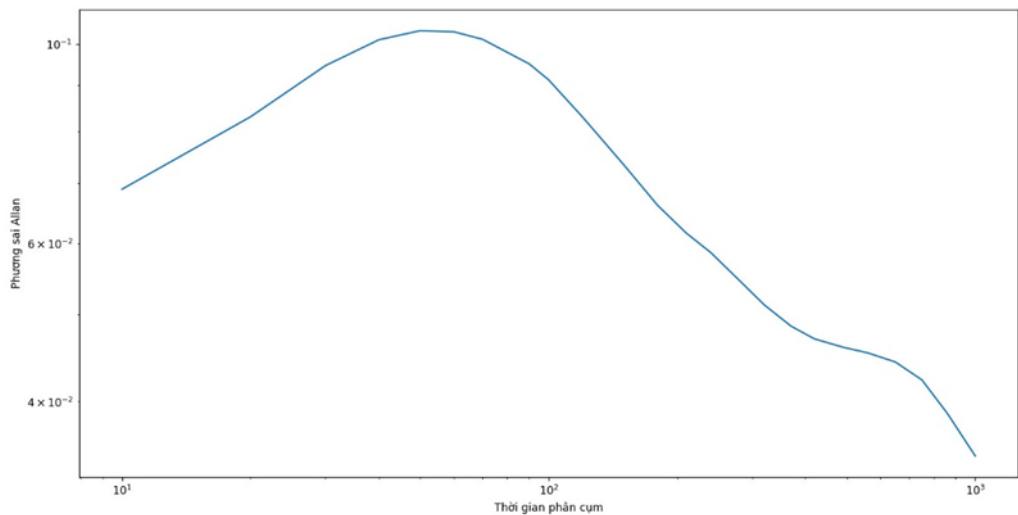
**Hiệu chuẩn sai số hệ thống.** Sai số hệ thống bao gồm độ lệch offset, sai số độ nhạy và sai số giao thoa trực (cross-axis). Để hiệu chỉnh, tác giả sử dụng gia tốc trọng trường như một tín hiệu chuẩn có giá trị tuyệt đối ổn định. Khi đặt cảm biến ở trạng thái tĩnh và xoay lần lượt để các trục X, Y, 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 2.10: 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 [95], một công cụ kinh điển trong phân tích ổn định tín hiệu và được ứng dụng rộng rãi trong lĩnh vực quán 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 2.11 cho thấy thành phần nhiễu trội 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 2.11: 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 [74]. 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.

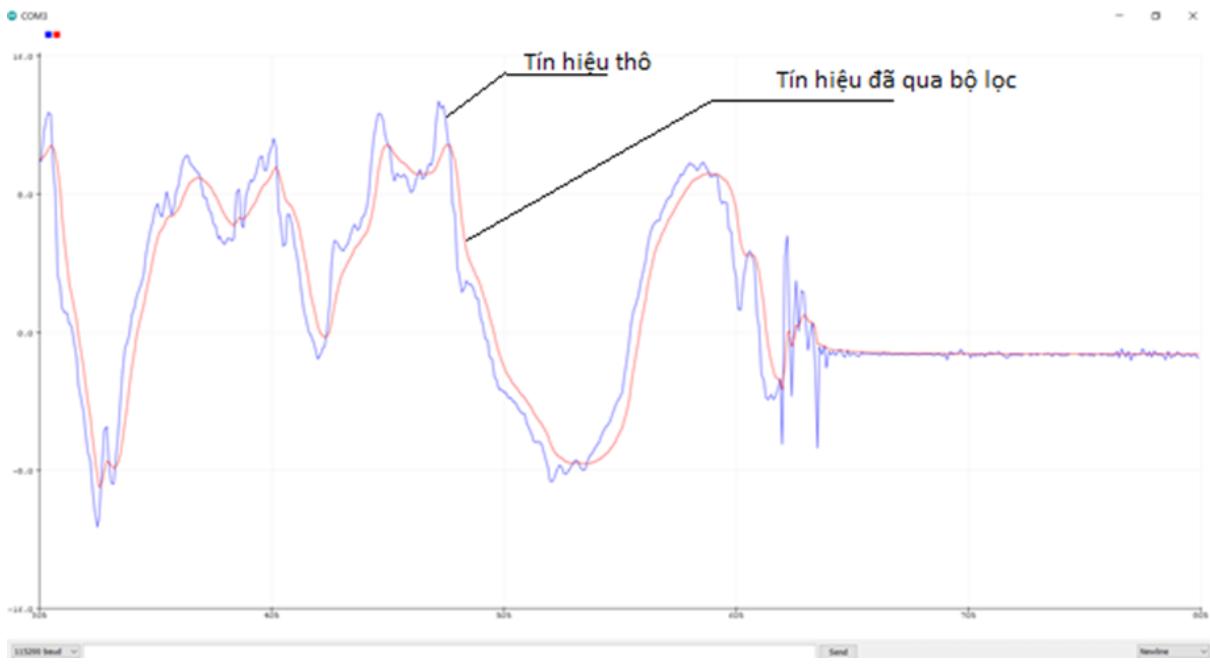
**Lý do lựa chọn Kalman.** So với moving average (dễ gây trễ tín hiệu) hay Butterworth low-pass (cần bậc lọc cao, tốn tài nguyên), Kalman phù hợp hơn cho hệ thống nhúng vì ba lý do: (i) hoạt động theo cơ chế cập nhật đệ quy, tiết kiệm bộ nhớ và chi phí tính toán; (ii) cho phép cân bằng động giữa độ tin 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 2.12 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



Hình 2.12: Kết quả bộ lọc Kalman cho dữ liệu trục X của cảm biến gia tốc

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:

**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.

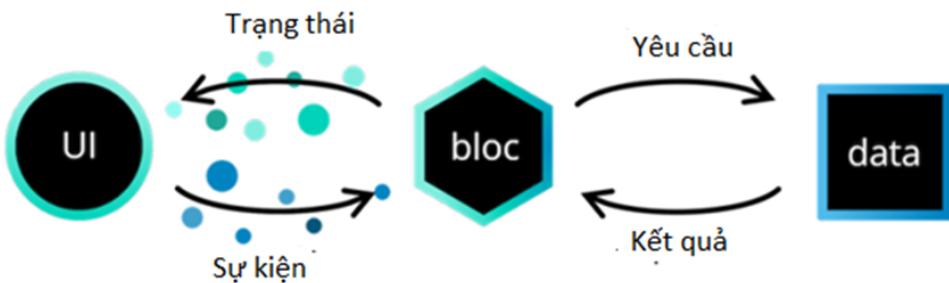
Ứng dụng được thiết kế xoay quanh ba nhóm chức năng chính như minh họa trong Bảng 2.1. Trước hết, nhóm **bảo mật** tập trung vào các thao tác đăng nhập, xác thực và khôi phục tài khoản, nhằm đảm bảo an toàn dữ liệu và bảo vệ quyền riêng tư của người dùng. Tiếp theo, nhóm **chức năng chung** bao gồm các 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.

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

## 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 2.13.



Hình 2.13: Cấu trúc kiến trúc BLoC trong ứng dụng Flutter

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 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 thô 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

---

```
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 serviceAccclerometer;
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

```
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"

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.

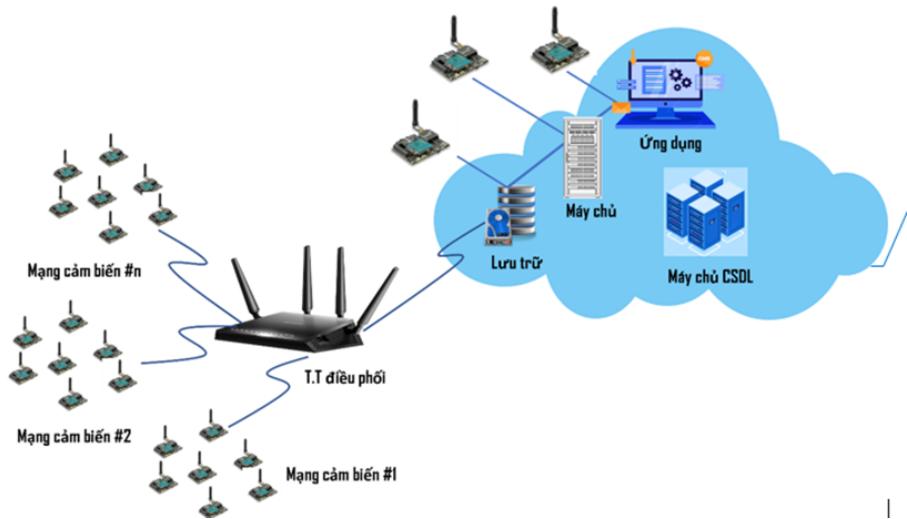
Hệ thống RAG kết hợp khả năng truy vấn ngôn ngữ nghĩa từ tập văn bản lớn (document retrieval) và khả năng sinh văn bản linh hoạt từ mô hình ngôn ngữ lớn (LLM), từ đó cung cấp các câu trả lời chính xác, có căn cứ và dễ hiểu cho người dùng không chuyên.

**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 mật khẩu. Dữ liệu người dùng (câu hỏi, chỉ số BMI, lịch sử cảm biến) được liên kết thông 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 hoá 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

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 khôi chức năng chính: **máy chủ xử lý**, **tầng bộ nhớ đệm (cache)**, và **cơ sở dữ liệu lưu trữ**.



Hình 2.14: Mô hình tích hợp giữa mạng cảm biến và cấu trúc dữ liệu đám mây

**(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.

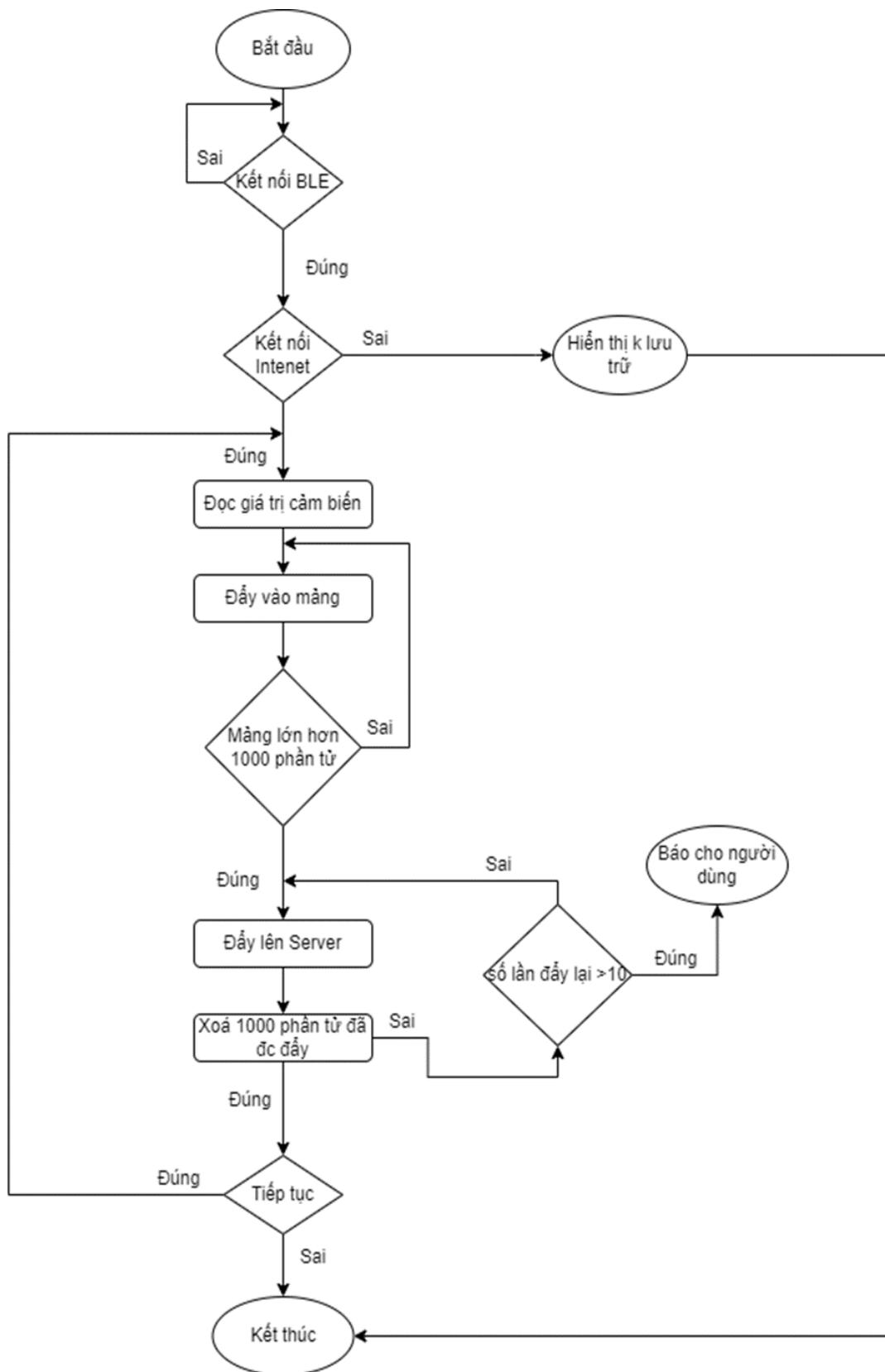
**Mở rộng về sau**, khi khối lượng dữ liệu ngày càng gia tăng và đa dạng hoá về

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.

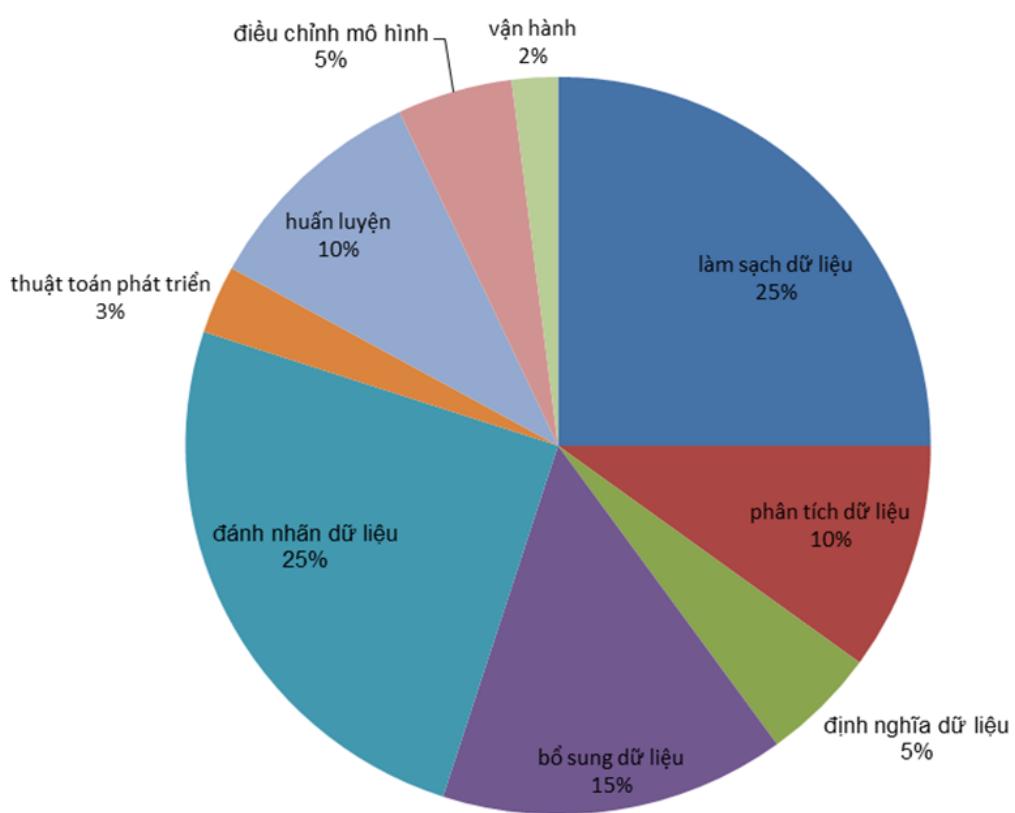
Lưu đồ thuật toán lưu trữ dữ liệu được thể hiện trong Hình 2.15, bao gồm hai nhánh xử lý nhằm đảm bảo tính liên tục dịch vụ và độ tin cậy dữ liệu. (i) Khi người dùng không có kết nối mạng, hệ thống vẫn duy trì liên kết BLE và hiển thị tín hiệu thời gian thực; dữ liệu được lưu tạm ở phía thiết bị/ứng 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.



Hình 2.15: Lưu đồ thuật toán lưu trữ dữ liệu cảm biến

## 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 2.16: Phân bổ thời gian thực hiện đối với dự án học máy

Hình 2.16 trình bày phân bổ 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 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] [96]:

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

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 [23]. 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:

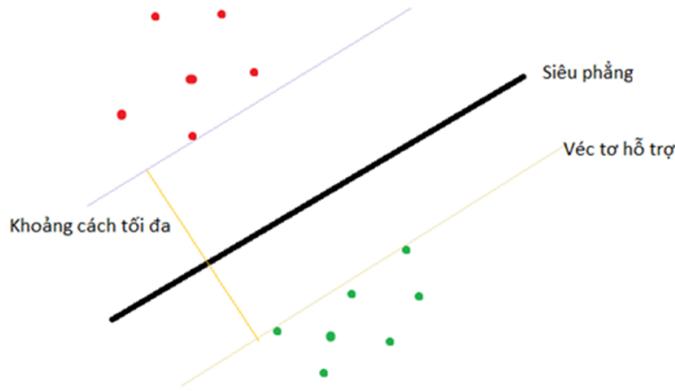
$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \forall i \quad (2.3)$$

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 2.17 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 2.17: 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.

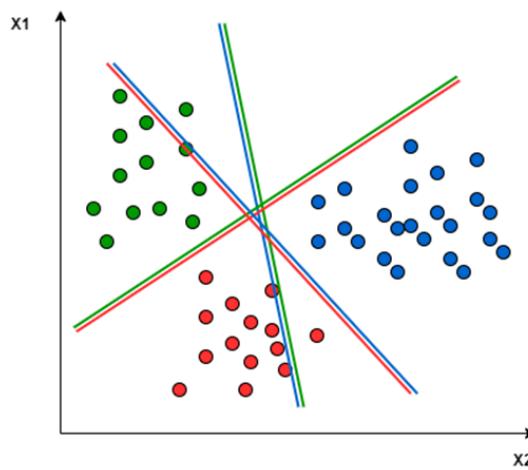
Một khái niệm quan trọng trong SVM là **biên (margin)** – khoảng cách giữa mặt siêu phẳng và các điểm dữ liệu gần nhất thuộc hai lớp. Mặt siêu phẳng tối ưu là mặt phẳng có biên lớn nhất, và chỉ những điểm nằm gần sát biên mới ảnh hưởng đến việc xác định mặt siêu phẳng, được gọi là **các vector hỗ trợ (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 2.18 và Hình 2.19.

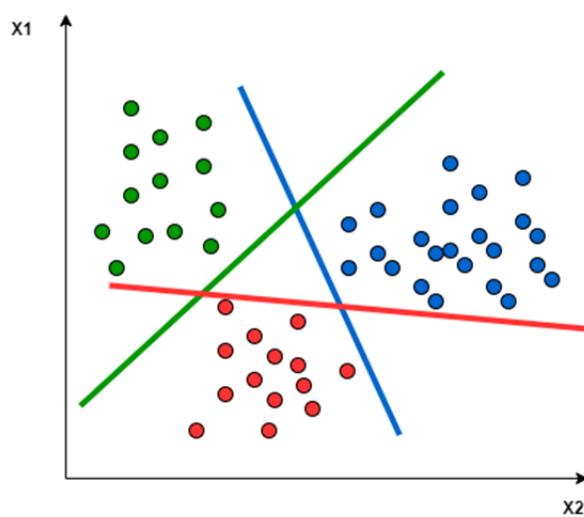
**One-vs-One (OvO):** Trong phương pháp này, một mô hình SVM được huấn luyện cho mỗi cặp lớp. Với  $K$  lớp, tổng cộng  $\frac{K(K-1)}{2}$  mô hình con được huấn luyện. Mỗi mô hình học cách phân biệt giữa hai lớp cụ thể và bỏ qua các lớp còn lại. Trong quá trình dự đoán, một cơ chế bỏ phiếu (voting) 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 2.18: Chiến lược phân loại đa lớp bằng phương pháp One-vs-One



Hình 2.19: 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)** [97]. 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 [26].

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) [98]. 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 [99].

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**

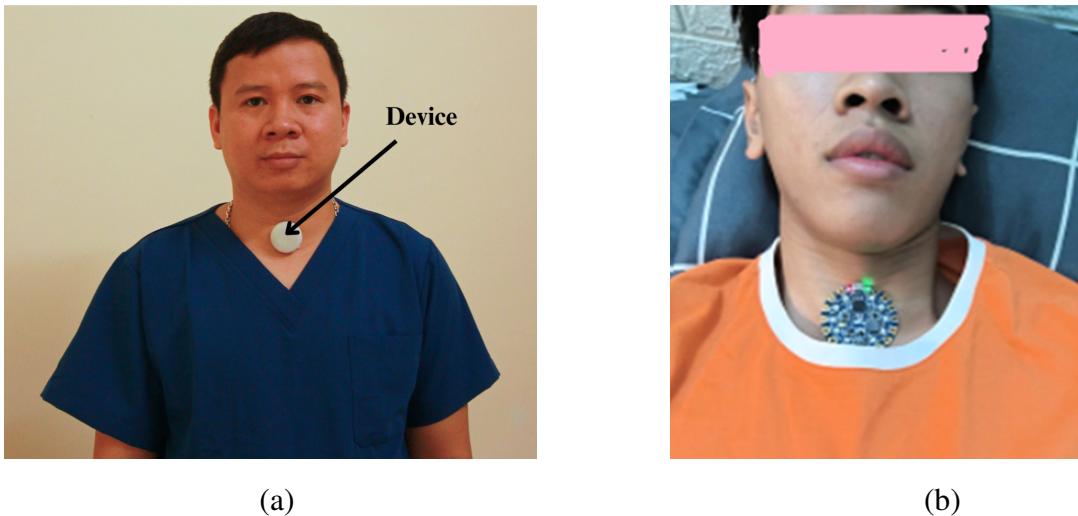
Trong phần này, tác giả trình bày các kết quả đạt được từ hai giai đoạn chính: (i) thu thập và xử lý dữ liệu cảm biến từ các tư thế ngủ khác nhau; (ii) huấn luyện mô hình học máy và triển khai mô hình tối ưu lên vi điều khiển nhằm đánh giá tính khả thi của giải pháp trên thiết bị biến.

### **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)

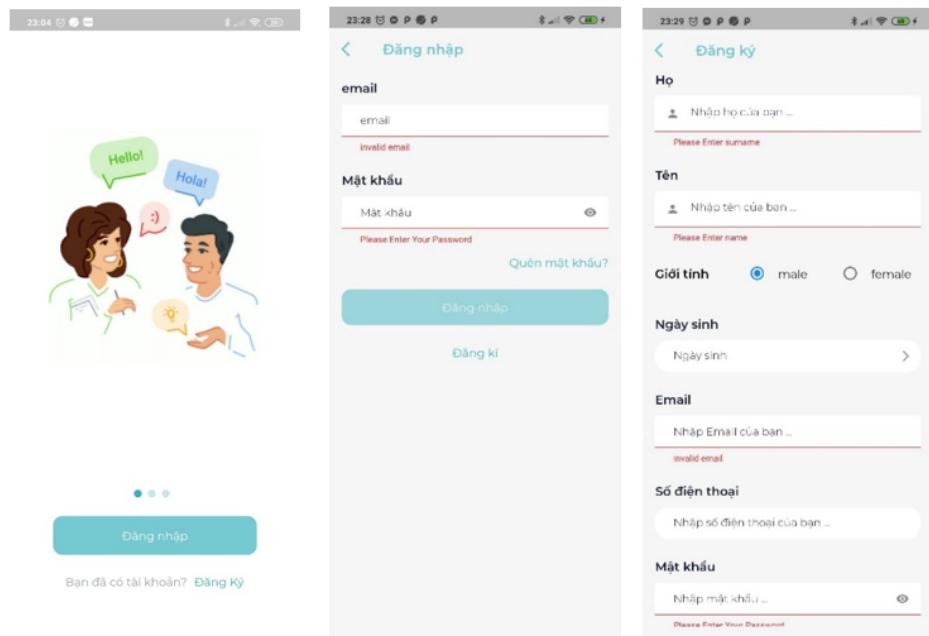
Hình 3.1: Hệ thống thử nghiệm: (a) vị trí đặt thiết bị cảm biến; (b) minh họa thực nghiệm thực tế trong tư thế nằm.

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.

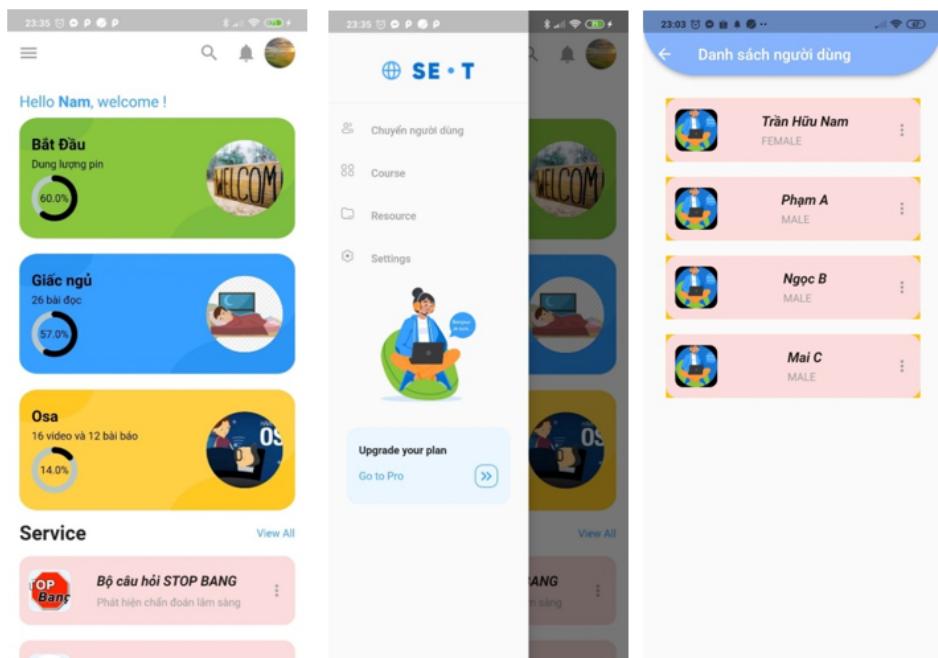
Một trong các nhiệm vụ chính mà tác giả thực hiện là phát triển ứng dụng di động phục vụ cho quá trình thu thập, hiển thị và xử lý dữ liệu. Dựa trên phản hồi từ nhóm nghiên cứu, tư vấn khoa học của Thầy PGS.TS. Mai Anh Tuấn, tư vấn y khoa của Thầy GS.TS. Dương Quý Sỹ, ứng dụng được thiết kế với tiêu 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

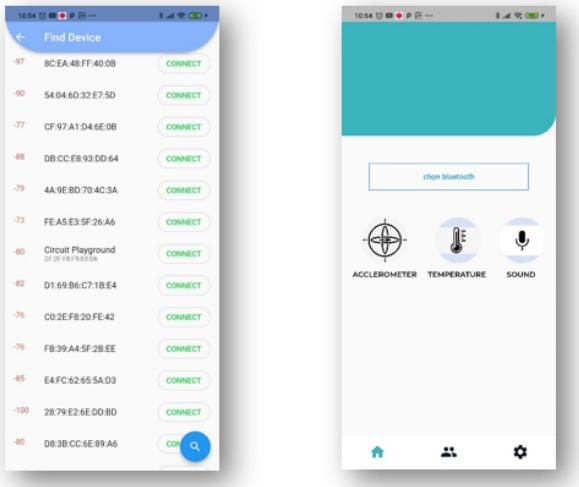


Hình 3.2: Giao diện chức năng đăng ký và đăng nhập



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,

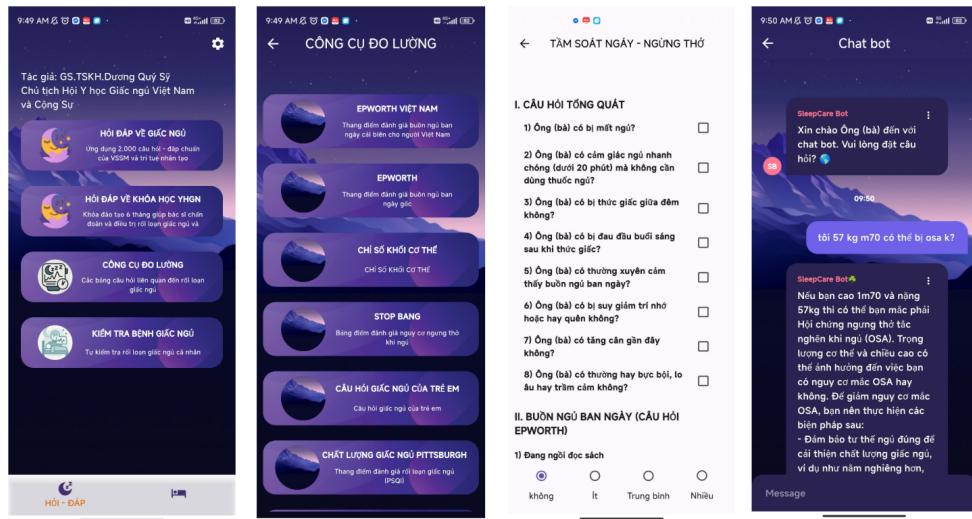


Hình 3.4: Giao diện màn hình danh sách BLE và chi tiết các dịch vụ kết nối với phần cứng

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

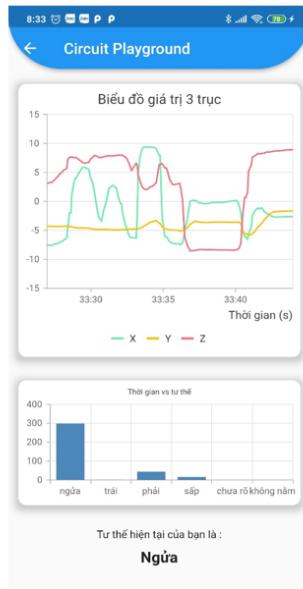
```

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.



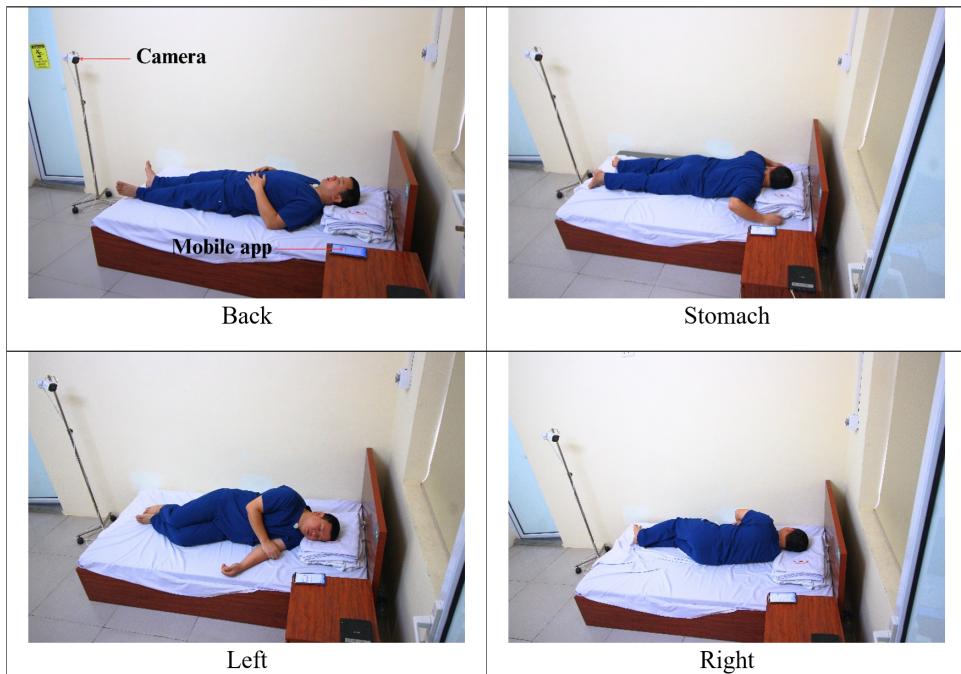
Hình 3.6: Giao diện hiển thị dữ liệu gia tốc ba trục

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.

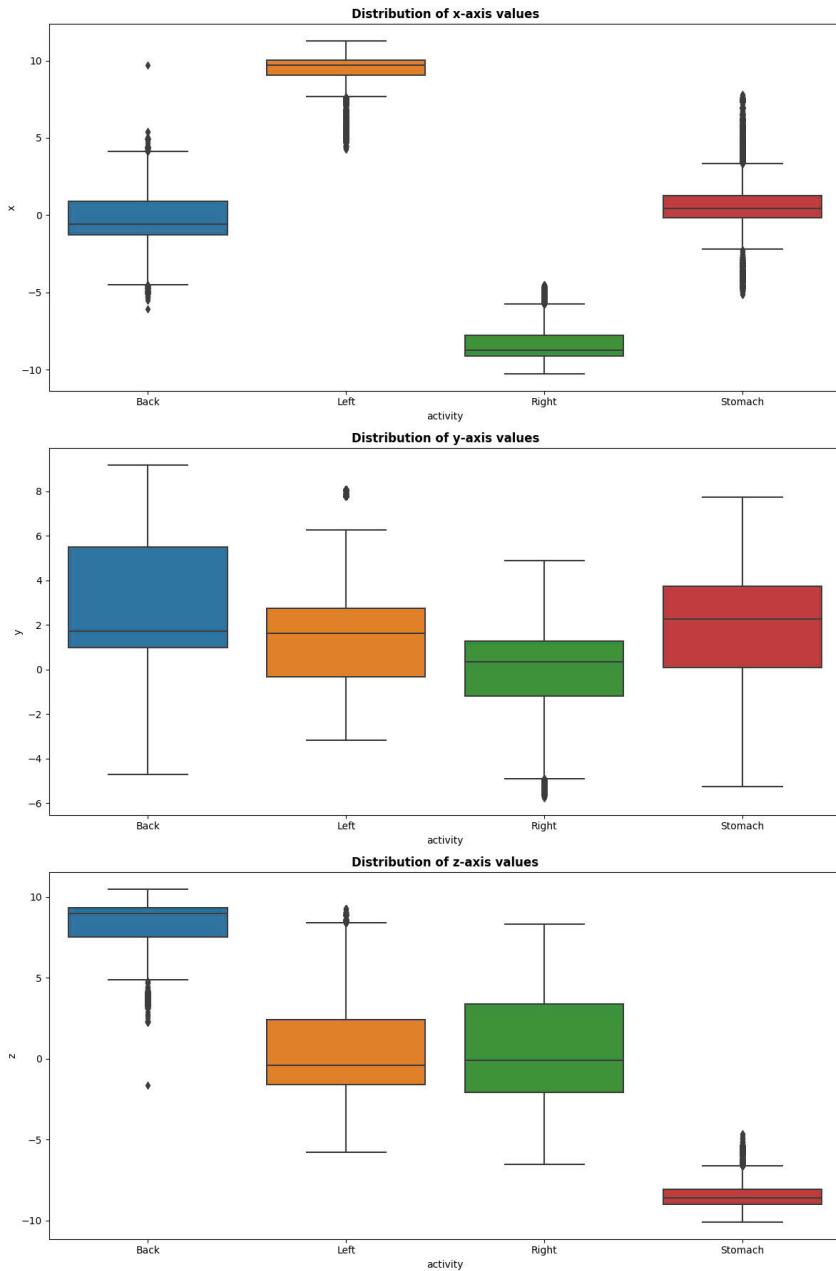
Tất cả dữ liệu thu thập từ các tình nguyện viên và tác giả đều được xuất ra định dạng CSV, bao gồm thông tin thời gian (timestamp), giá trị cảm biến trên ba trục  $x$ ,  $y$ ,  $z$ , và nhãn tư thế tương ứng (nếu có). Dữ liệu này được sử dụng làm đầu vào cho quá trình trích xuất đặc trưng và huấn luyện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

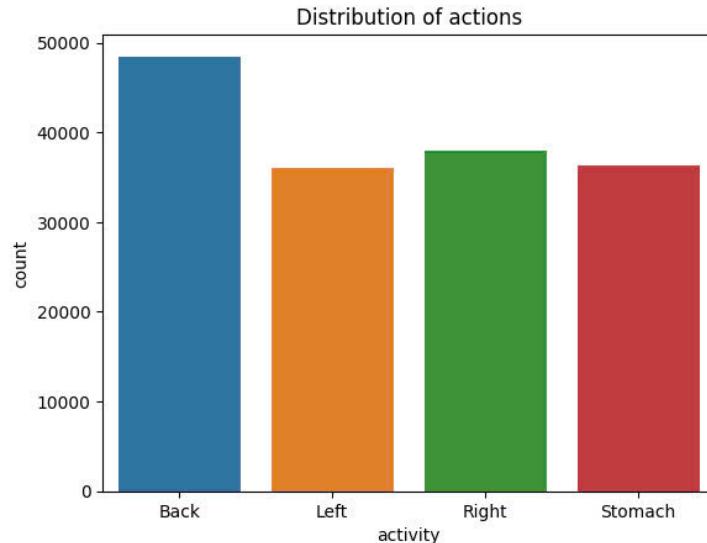
trạng thái rõ ràng. Ngoài ra, các ngưỡng thường cần hiệu chỉnh theo từng cá nhân do sự khác biệt về hình thể, kiểu vận động và vị trí gắn cảm biến.



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	$\text{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.

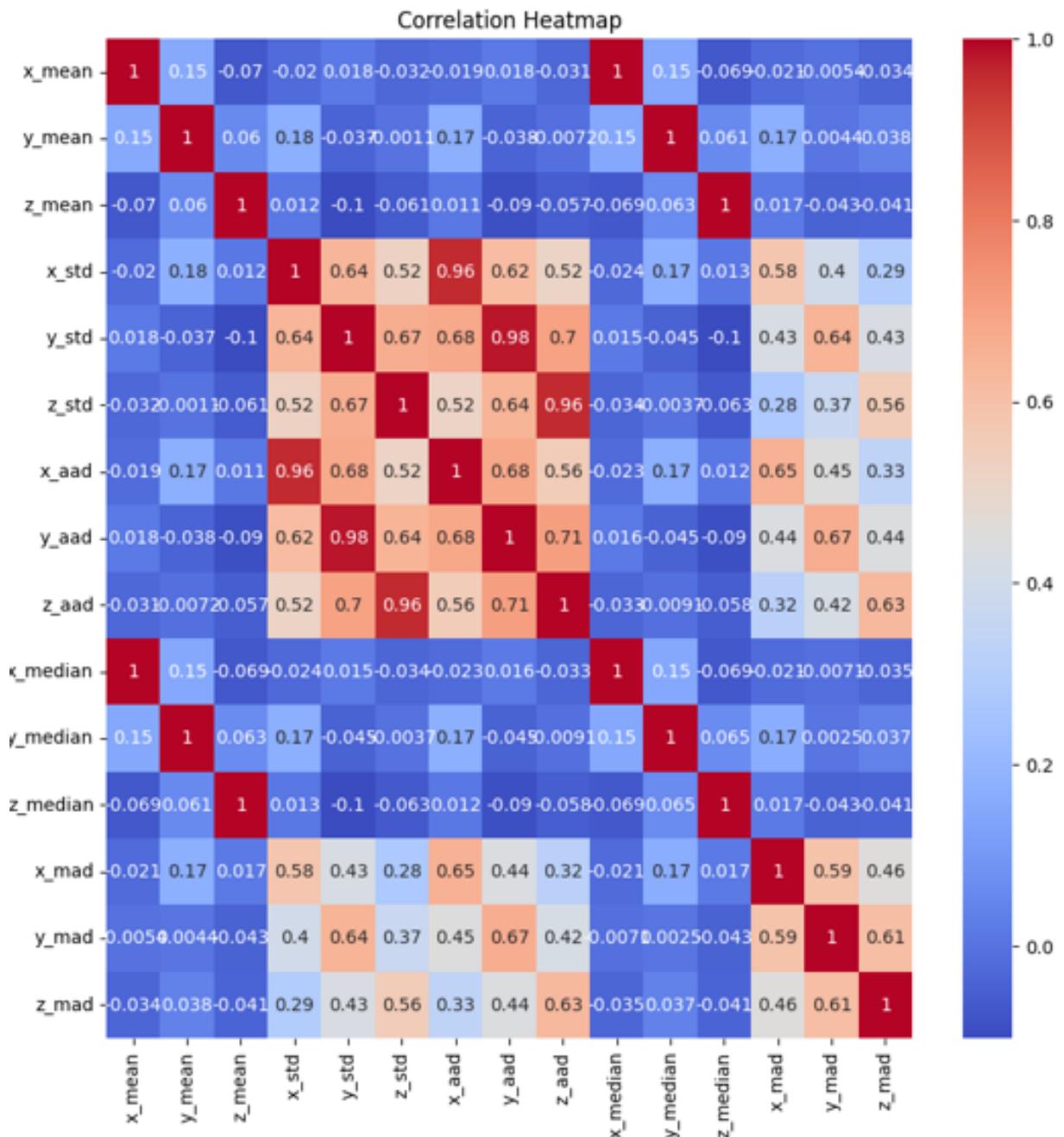
**Thứ hai**, kết quả phân tích SHAP (SHapley Additive exPlanations) ở Hình 3.11 chỉ ra rằng một số đặc trưng – đặc biệt là các đặc trưng miền thời gian trên trục  $z$  như trung bình, năng lượng, trung vị – có ảnh hưởng vượt trội đến dự đoán của mô hình. Do đó, việc ưu tiên các đặc trưng này trong kịch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

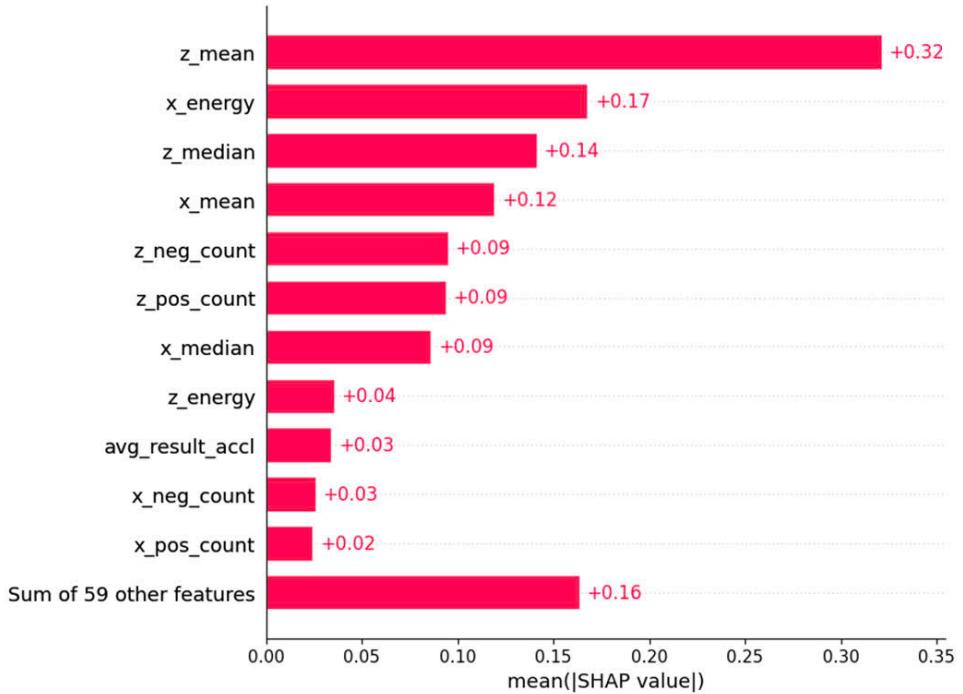
Với thiết kế kịch bản như trên, luận văn không chỉ đánh giá hiệu quả mô hình theo nhiều hướng khác nhau, mà còn hướng tới việc xác định cấu hình tối ưu giữa độ chính xác, chi phí tính toán và khả năng triển khai thực tiễn.



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

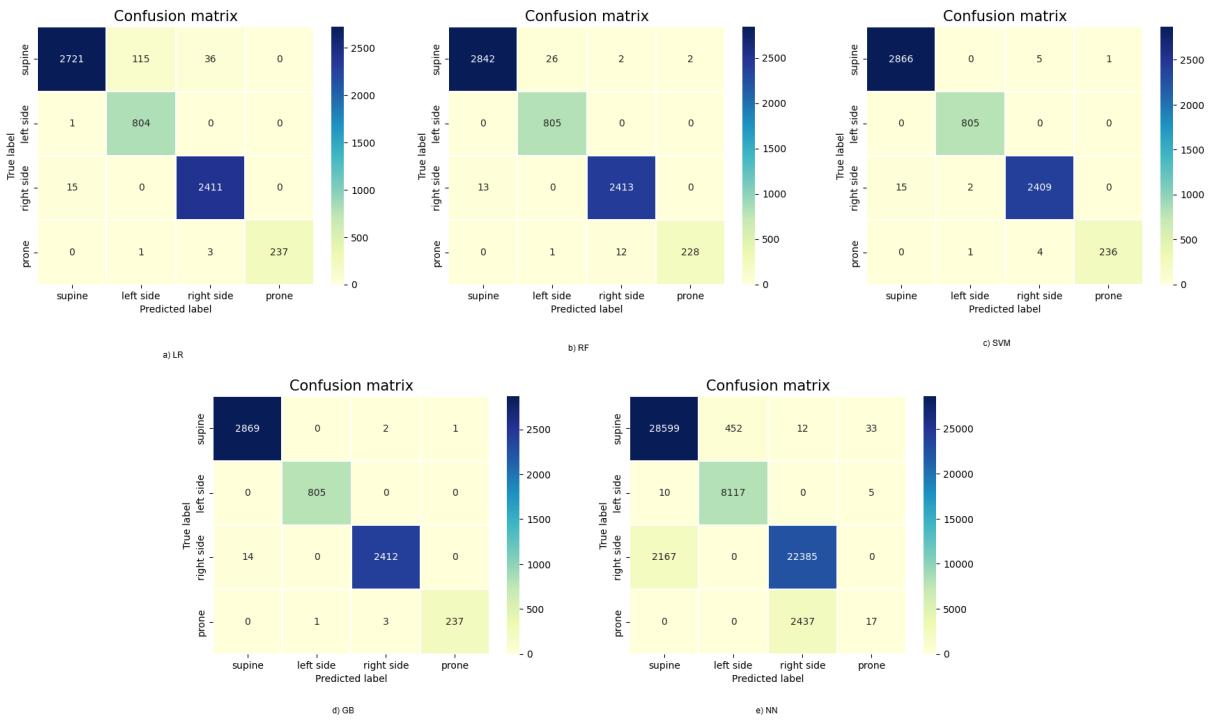
hình dựa trên cây quyết định, khả năng khai thác tốt cả quan hệ phi tuyến và sự tương tác giữa các đặc trưng. **Random Forest (RF)** mặc dù có đôi chút biến thiên, nhưng vẫn duy trì độ chính xác vượt ngưỡng 0.99 trong hầu hết kịch bản, cung cấp vai trò của các mô hình ensemble dựa trên bootstrap aggregation 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] Salari N, Gholami A, Jalili F, Zereshki E, Heidarian P, Shohaimi S **and** Mokhamadi 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>.
- [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. **in***Revue des Maladies Respiratoires*: 35.1 (2018), **pages** 14–24. ISSN: 0761-8425. DOI: <https://doi.org/10.1016/j.rmr.2017.10.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0761842517302851>.
- [4] Tregear S, Reston J, Schoelles K **and** Phillips B. Obstructive sleep apnea and risk of motor vehicle crash: systematic review and meta-analysis. **in***J Clin Sleep Med*: 5.6 (december 2009), **pages** 573–581.
- [5] Marin JM, Carrizo SJ, Vicente E **and** Agusti AG. Long-term cardiovascular outcomes in men with obstructive sleep apnoea-hypopnoea with or without treatment with continuous positive airway pressure: an observational study. **in***Lancet*:

- 365.9464 (**march** 2005), **pages** 1046–1053. DOI: 10 . 1016 / S0140 - 6736(05)71141 - 7. URL: [https://doi.org/10.1016/S0140-6736\(05\)71141-7](https://doi.org/10.1016/S0140-6736(05)71141-7).
- [6] *Ngưng thở khi ngủ do tắc nghẽn (OSA)*. Available at <https://www.msdmanuals.com/vi-vn>. 2022.
  - [7] 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). **in***Pulmonary therapy*: 11 (**january** 2025). DOI: 10 . 1007 / s41030 - 024 - 00286 - 8.
  - [8] 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. **in***Sleep 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>.
  - [9] Sabil A, Blanchard M, Trzepizur W, Goupil F, Meslier N, Paris A, Pigeanne T, Priou P, Le Vaillant M, Gagnadoux F **and** Pays de la Loire Sleep Cohort Group. Positional obstructive sleep apnea within a large multicenter French cohort: prevalence, characteristics, and treatment outcomes. **in***J Clin Sleep Med*: 16.12 (**december** 2020), **pages** 2037–2046. DOI: 10 . 5664 / jcsm . 8752. URL: <https://doi.org/10.5664/jcsm.8752>.
  - [10] Sampa Markun L.C. Clinician-Focused Overview and Developments in Polysomnography. **in***Curr Sleep Medicine*: (2020). DOI: <https://doi.org/10.1007/s40675-020-00197-5>.
  - [11] 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. **in***Sleep*: 28.4 (**april** 2005), **pages** 499–521. DOI: 10 . 1093 / sleep / 28.4.499.
  - [12] 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. **in***Frontiers in Neurology*: 14 (**february** 2023), **page** 1123227. DOI: 10 . 3389 / fneur . 2023 . 1123227.

- [13] 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.
- [14] 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.
- [15] Collop NA **and** colleagues. Obstructive sleep apnea devices for out-of-center (OOC) testing: technology evaluation. *inJ Clin Sleep Med*: (2011). DOI: 10.5664/JCSM.1328.
- [16] D. Morillo, J.L. Rojas Ojeda, L.F. Crespo Foix **and** A. Jiménez. An accelerometer-based device for sleep apnea screening. *inIEEE Transactions on Information Technology in Biomedicine*: 14.2 (march 2010). Epub 2009 Jul 28, **pages** 491–499. DOI: 10.1109/TITB.2009.2027231.
- [17] Ahmet & Sumbul Yuzer. A Novel Wearable Real-Time Sleep Apnea Detection System Based on the Acceleration Sensor. *inJ Clin Sleep Med*: (2019).
- [18] 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.
- [19] 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..>
- [20] 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>.
- [21] Robin Genuer **and** Jean-Michel Poggi. Random Forests. *inApplied Mathematics and Data Science*: Springer, 2020, **pages** 33–55. DOI: 10.1007/978-3-030-56485-8\_3. URL: [https://doi.org/10.1007/978-3-030-56485-8\\_3](https://doi.org/10.1007/978-3-030-56485-8_3).

- [22] Shukun Wang, Wenping Xuan, Dan Chen, Yun Li, Rui Chen **and** Ming He. Machine Learning Assisted Wearable Wireless Device for Sleep Apnea Syndrome Diagnosis. *inBiosensors*: 13.4 (2023), **page** 483. ISSN: 2079-6374. DOI: 10.3390/bios13040483. URL: <https://doi.org/10.3390/bios13040483>.
- [23] 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>.
- [24] Padraig Cunningham **and** Sarah Delany. k-Nearest Neighbour Classifiers. *inMultiple Classifier Systems*: 54 (2007). DOI: 10.1145/3459665.
- [25] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim **and** Aboul Ella Hassanien. Linear Discriminant Analysis: A Detailed Tutorial. *inAI Communications*: 30.2 (2017), **pages** 169–190. ISSN: 0921-7126. DOI: 10.3233/AIC-170729. URL: <https://doi.org/10.3233/AIC-170729>.
- [26] 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>.
- [27] Minh Long Hoang, Guido Matrella **and** Paolo Ciampolini. Metrological evaluation of contactless sleep position recognition using an accelerometric smart bed and machine learning. *inSensors and Actuators A: Physical*: 385 (2025), **page** 116309. ISSN: 0924-4247. DOI: <https://doi.org/10.1016/j.sna.2025.116309>. URL: <https://www.sciencedirect.com/science/article/pii/S0924424725001153>.
- [28] Rawan S. Abdulsadig, Sukhpreet Singh, Zaibaa Patel **and** Esther Rodriguez-Villegas. ?Sleep Posture Detection Using an Accelerometer Placed on the Neck? *in2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*: 2022, **pages** 2430–2433. DOI: 10.1109/EMBC48229.2022.9871300.
- [29] Xu X, Mo Q, Wang Z, Zhao Y **and** Li C. Classification of Sleeping Position Using Enhanced Stacking Ensemble Learning. *inEntropy (Basel)*: 26.10 (september 2024), **page** 817. DOI: 10.3390/e26100817. URL: <https://doi.org/10.3390/e26100817>.

- [30] 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>.
- [31] Partha Pratim Ray. A review on TinyML: State-of-the-art and prospects. *inJournal of King Saud University - Computer and Information Sciences*: 34.4 (2022), **pages** 1595–1623 ISSN: 1319-1578. DOI: <https://doi.org/10.1016/j.jksuci.2021.11.019>. URL: <https://www.sciencedirect.com/science/article/pii/S1319157821003335>.
- [32] Maha S. Diab **and** Esther Rodriguez-Villegas. Embedded Machine Learning Using Microcontrollers in Wearable and Ambulatory Systems for Health and Care Applications: A Review. *inIEEE Access*: 10 (2022), **pages** 98450–98474. DOI: 10 . 1109 / ACCESS . 2022 . 3206782.
- [33] Richard B. Berry, Reena Budhiraja, Daniel J. Gottlieb, David Gozal, Conrad Iber, Vishesh K. Kapur, Carole L. Marcus, Reena Mehra, Sairam Parthasarathy, Stuart F. Quan, Susan Redline, Kingman P. Strohl, Sally L. Davidson Ward, Mary M. Tangredi **and** American Academy of Sleep Medicine. Rules for Scoring Respiratory Events in Sleep: Update of the 2007 AASM Manual for the Scoring of Sleep and Associated Events. Deliberations of the Sleep Apnea Definitions Task Force of the American Academy of Sleep Medicine. *inJournal of Clinical Sleep Medicine*: 8.5 (2012), **pages** 597–619. ISSN: 1550-9389. DOI: 10 . 5664 / jcsm . 2172. URL: <https://doi.org/10.5664/jcsm.2172>.
- [34] 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.
- [35] 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.

- [36] 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.
- [37] Khaled Al Oweidat, Ahmad A. Toubasi, Montaha AL-Iede, Manar M. Hasuneh **and** Thuraya N. Al-Sayegh. Comparing the Diagnostic Value of the Positional Obstructive Sleep Apnea Definitions. *inRespiratory Medicine*: 212 (2023), **page** 107227. ISSN: 0954-6111. DOI: 10.1016/j.rmed.2023.107227. URL: <https://www.sciencedirect.com/science/article/pii/S0954611123001154>.
- [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] 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>.
- [43] Carberry JC & Eckert DJ Osman AM Carter SG. Obstructive sleep apnea: current perspectives. *inNat Sci Sleep*: (2018). DOI: 10.2147/NSS.S124657.

- [44] Amal Osman, Sophie Carter, Jayne Carberry **and** Danny Eckert. Obstructive sleep apnea: current perspectives. **in***Nature and Science of Sleep*: 10 (january 2018), **pages** 21–34. DOI: 10.2147/NSS.S124657.
- [45] W. W. Flemons **and** W. Tsai. Quality of life consequences of sleep-disordered breathing. **in***Journal of Allergy and Clinical Immunology*: 99.2 (1997), S750–S756. DOI: 10.1016/s0091-6749(97)70123-4.
- [46] T. Mooe, S. Gullsby, T. Rabben **and** P. Eriksson. Sleep-disordered breathing: a novel predictor of atrial fibrillation after coronary artery bypass surgery. **in***Coronary Artery Disease*: 7.6 (june 1996), **pages** 475–478.
- [47] 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. **in***Archives of Internal Medicine*: 157 (1997), **pages** 1746–1752.
- [48] Johnson J Goyal M. Obstructive Sleep Apnea Diagnosis and Management. **in***Mo Med*: (2017).
- [49] Medical Advisory Secretariat. Polysomnography in patients with obstructive sleep apnea: an evidence-based analysis. **in***Ontario Health Technology Assessment Series*: 6.13 (2006). Epub 2006 Jun 1, **pages** 1–38.
- [50] Yonghwan Jeon, Kwangil Heo **and** Sungjun Kang. ?Real-Time Sleep Apnea Diagnosis Method Using Wearable Device without External Sensors? **in***2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*: IEEE, 2020, **pages** 1–3. DOI: 10.1109/PerComWorkshops48777.2020.9156177. URL: <https://doi.org/10.1109/PerComWorkshops48775.2020.9156177>.
- [51] 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. **in***IEEE 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>.

- [52] 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.
- [53] 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>.
- [54] 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>.
- [55] 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>.
- [56] M. Kim I. Yun J. Jeung. Ultra-low power wearable infant sleep position sensor. *inSensors*: 7 (2020).
- [57] D.M. Domingues, P.R. Rocha, A.C.M.V. Miachon **and others**. Sleep prediction using data from oximeter, accelerometer and snoring for portable monitor obstructive sleep apnea diagnosis. *inScientific Reports*: 14 (2024), **page** 24562. DOI: 10.1038/s41598-024-75935-8. URL: <https://doi.org/10.1038/s41598-024-75935-8>.

- [58] 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.
- [59] *Home Sleep Test (HST) Devices Comparison Guide*. Available at <https://sleepreviewmag.com/sleep-diagnostic/home-testing/home-apnea-testing/hst-devices-guide/>. **november** 2022.
- [60] L.-C.và cộng sự Jeng P.-Y.; Wang. A Wrist Sensor Sleep Posture Monitoring System: An Automatic Labeling Approach. **in***Sensors*: (2021). DOI: <https://doi.org/10.3390/s21010258>.
- [61] G Zhang Z.; Yang. Monitoring cardio-respiratory and posture movements during sleep: What can be achieved by a single motion sensor. **in***IEEE International Conference*: (2015).
- [62] Crossvà cộng sự Kwasnicki R.M. A lightweight sensing platform for monitoring sleep quality and posture: a simulated validation study. **in***Eur J Med Res*: 7 (2018). DOI: <https://doi.org/10.1186/s40001-018-0326-9>.
- [63] 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.
- [64] Kumar M Menon A. Influence of body position on severity of obstructive sleep apnea: a systematic review. **in***ISRN Otolaryngol*: ()�.
- [65] G. Delfi và cộng sự S. Akbarian. Automated Non-Contact Detection of Head and Body Positions During Sleep. **in***ISRN Otolaryngol*: (2013). DOI: 10.1155/2013/670381.
- [66] Xiao Sun, Li Qiu, Yibo Wu, Yeming Tang **and** Guohong Cao. SleepMonitor: Monitoring Respiratory Rate and Body Position During Sleep Using Smartwatch. **in***Proceedings 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>.
- [67] Vincenzo và cộng sự Natale. Monitoring sleep with a smartphone accelerometer. **in***Sleep and Biological Rhythms*: (2012).

- [68] 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.
- [69] Liqing & Xu và cộng sự Niu Weimeng & Fang. Summary of Research Status and Application of MEMS Accelerometers. *inJournal of Computer and Communications*: (2018).
- [70] Duc-Tan Tran, Thuy Nguyen, Trinh Chu Duc **and** Vu Hung. ?Cảm biến gia tốc MEMS kiểu tụ: So sánh kết quả mô phỏng và thực nghiệm? *injanuary* 2005: **pages** 465–470.
- [71] Nguyen & Vo và cộng sự Sang Vu & Thang. Human Activity Recognition and Monitoring Using Smartphones. *inInternational Conference on Biomedical Engineering in Vietnam*: (2015).
- [72] M. Rossi, D. Sala, D. Bovio **and others**. SLEEP-SEE-THROUGH: Explainable Deep Learning for Sleep Event Detection and Quantification From Wearable Somnography. *inIEEE 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>.
- [73] Alaa Sheta, Hamzah Turabieh, Tamer Thaher, 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. *inApplied Sciences*: 11.14 (2021), **page** 6622. ISSN: 2076-3417. DOI: 10.3390/app11146622. URL: <https://doi.org/10.3390/app11146622>.
- [74] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *inTransactions of the ASME-Journal of Basic Engineering*: 82.Series D (1960).
- [75] Lei Zou **and** Guang Liu. Multiscale Bidirectional Temporal Convolutional Network for Sleep Apnea Detection Based on Wearable Photoplethysmography Bracelet. *inIEEE Journal of Biomedical and Health Informatics*: 28 (2024), **pages** 1331–1340. ISSN: 2168-2194. DOI: 10.1109/JBHI.2023.3335658. URL: <https://doi.org/10.1109/JBHI.2023.3335658>.

- [76] 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>.
- [77] Thanuja Pavani và cộng sự Uday Sanku & Satti. Classifying Human Activities using Machine Learning and Deep Learning Techniques. **in***arxiv*: (2019).
- [78] Debanjan Parbat **and** Monisha Chakraborty. Multiscale Entropy Analysis of Single Lead ECG and ECG Derived Respiration for AI-Based Prediction of Sleep Apnea Events. **in***Biomedical Signal Processing and Control*: 87 (2024), **page** 105444. ISSN: 1746-8094. DOI: 10.1016/j.bspc.2023.105444. URL: <https://www.sciencedirect.com/science/article/pii/S1746809423008777>.
- [79] Xinyu Ji, Zhihao Rao, Weijie Zhang **and others**. Airline Point-of-Care System on Seat Belt for Hybrid Physiological Signal Monitoring. **in***Micromachines (Basel)*: 13.11 (2022). DOI: 10.3390/mi13111880. URL: <https://doi.org/10.3390/mi13111880>.
- [80] R.S. Abdulsadig **and** E. Rodriguez-Villegas. Sleep posture monitoring using a single neck-situated accelerometer: A proof-of-concept. **in***IEEE Access*: 11 (2023), **pages** 17693–17706. DOI: 10.1109/ACCESS.2023.3246266.
- [81] 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.
- [82] O. Elnaggar, R. Arelli, F. Coenen **and others**. An interpretable framework for sleep posture change detection and postural inactivity segmentation using wrist kinematics. **in***Scientific Reports*: 13 (2023), **page** 18027. DOI: 10.1038/s41598-023-44567-9.
- [83] 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.

- [84] T.-H. Dao, H.-Y. Hoang, V.-N. Hoang, D.-T. Tran **and** D.-N. Tran. Human activity recognition system for moderate performance microcontroller using accelerometer data and random forest algorithm. **in***EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*: 9.4 (2022), e4–e4.
- [85] 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.
- [86] 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).
- [87] Arm Ltd. *Arm Cortex-M Processor Comparison Table*. Accessed: April 6, 2025. 2025. URL: <https://developer.arm.com/documentation/102787/latest/>.
- [88] *Cortex-M4*. Available at <https://developer.arm.com/Processors/Cortex-M4>.
- [89] Ioan Lucan Orăsan, Ciprian Seiculescu **and** Cătălin Daniel Căleanu. A Brief Review of Deep Neural Network Implementations for ARM Cortex-M Processor. **in***Electronics*: 11.16 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11162545. URL: <https://www.mdpi.com/2079-9292/11/16/2545>.
- [90] Pradyumn Chaturvedi, Mahmadasraf Mulla **and** Siba Patro. ?Power Electronics Laboratory Education using ARM Cortex M4 32-bit Microcontroller? **in**december 2018: **pages** 1–6. DOI: 10.1109/PEDES.2018.8707575.
- [91] Nordic Semiconductor. *nRF52840 – Advanced Bluetooth 5, Thread and Zigbee multiprotocol SoC*. <https://www.nordicsemi.com/Products/nRF52840>. Accessed: 2025-07-28. 2025.
- [92] *Bluetooth® Core Specification Version 5.3 Feature Enhancements*. Available at <https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-version-5-3-feature-enhancements/>.
- [93] *Introducing Circuit Playground*. Available at <https://learn.adafruit.com/introducing-circuit-playground/overview>.

- [94] Arduino. *Nano 33 BLE Sense – Datasheet and Hardware Overview*. <https://docs.arduino.cc/hardware/nano-33-ble-sense/>. Accessed: 2025-07-28. 2025.
- [95] N. Djermanova M. B. Marinov B. Ganев **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..
- [96] 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).
- [97] 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>.
- [98] 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.
- [99] 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>.