UBND TỈNH NINH BÌNH
TRƯỜNG ĐẠI HỌC HOA LƯ

BÁO CÁO KẾT QUẢ THỰC HIỆN
NHIỆM VỤ KHOA HỌC VÀ CÔNG NGHỆ CẤP CƠ SỞ

ỨNG DỤNG MÁY HỌC TRONG VIỆC GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ TRONG NHÀ

Chủ nhiệm nhiệm vụ: Đặng Thị Thu Hà
Đơn vị: Khoa Ngoại ngữ - CNTT

NINH BÌNH, 2025

UBND TỈNH NINH BÌNH

TRƯỜNG ĐẠI HỌC HOA LƯ

BÁO CÁO KẾT QUẢ THỰC HIỆN
NHIỆM VỤ KHOA HỌC VÀ CÔNG NGHỆ CẤP CƠ SỞ

ỨNG DỤNG MÁY HỌC TRONG VIỆC GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ TRONG NHÀ

Chủ nhiệm nhiệm vụ: Đặng Thị Thu Hà

Đơn vị: Khoa Ngoại ngữ - CNTT

Các thành viên: Đào Sỹ Nhiên, Khoa Ngoại ngữ – CNTT

Nguyễn Tất Thắng, Khoa Ngoại ngữ – CNTT

Bùi Thị Tuyết, Phòng Đào tạo – Khoa học

Xác nhận của Chủ tịch HĐ nghiệm thu

(họ tên, chữ ký)

Đặng Thị Thu

Chủ nhiệm nhiệm vụ

(họ tên, chữ ký)

Đặng Thị Thu Hà

NINH BÌNH, 2025

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH VẼ, BIỂU ĐỒ

BẢNG KÝ HIỆU VÀ CHỮ VIẾT TẮT *(NẾU CÓ).*

Từ viết tắt	Giải thích (Tiếng Việt)
Ace-Met	Hỗn hợp Acetone và Methanol
ADC	Bộ chuyển đổi tín hiệu tương tự sang tín hiệu số
AE	Mô hình mã hóa tự động (cũng là *aes*)
AI	Trí tuệ nhân tạo
ANN	Mạng nơ-ron nhân tạo
BP	Giải thuật lan truyền ngược
CNN	Mạng nơ-ron tích chập
CNN-LDA	Mạng lai giữa CNN và LDA
DL	Học sâu
DLM	Mô hình học sâu
DNN	Mạng nơ-ron sâu
DT	Cây quyết định
E-nose	Mũi điện tử
EN	Mũi điện tử
IAQ	Chất lượng không khí trong nhà
ID3	Thuật toán xây dựng cây quyết định
IoT	Internet vạn vật
KNN	K-láng giềng gần nhất
LDA	Thuật toán phân loại
LN	Hồi quy Tuyến tính
LSTM	Bộ nhớ ngắn hạn dài
MAE	Sai số tuyệt đối trung bình (hoặc Hàm mất mát MAE)
MEMS	Công nghệ hệ thống cơ điện tử vi mô (Được nhắc đến trong ngữ cảnh cảm biến)
ML	Học máy

MLP	Mạng truyền ngược đa lớp
MOS	Cảm biến bán dẫn oxit kim loại
MSE	Sai số bình phương trung bình
MTL	Mô hình học đa nhiệm
NLP	Xử lý ngôn ngữ tự nhiên
PCA	Phân tích thành phần chính
PEN	Mũi điện tử (Tên thương mại của thiết bị AIRSENSE, Germany)
R^2	Điểm R-square
RF	Rừng ngẫu nhiên
ReLU	Đơn vị tuyến tính được chỉnh lưu (Hàm kích hoạt)
RH	Độ ẩm
RMSE	Căn bậc hai sai số bình phương trung bình
RNN	Mạng nơ-ron hồi quy
SVC	Phân loại Vector Hỗ trợ
SGD	Thuật toán Gradient Descent ngẫu nhiên
SVM	Máy vector hỗ trợ
SVR	Hồi quy Vector Hỗ trợ
TCN	Mạng nơ-ron tích chập thời gian
VOCs	Hợp chất hữu cơ dễ bay hơi
WHO	Tổ chức Y tế Thế giới
XGBoost Regressor	Thuật toán hồi quy tăng cường gradient

TRANG THÔNG TIN KẾT QUẢ NGHIÊN CỨU

Nghiên cứu tập trung vào bài toán giám sát chất lượng không khí trong nhà, một vấn đề mang tính cấp thiết ảnh hưởng trực tiếp đến sức khỏe và năng suất lao động, đặc biệt trong các không gian kín. Các phương pháp giám sát truyền thống thường đắt đỏ, cồng kềnh và thiếu linh hoạt, trong khi các hệ thống mũi điện tử (E-nose) hiện nay gặp khó khăn khi phân tích hỗn hợp khí do tín hiệu cảm biến chồng chéo và tính phi tuyến cao.

Mục tiêu của đề tài là ứng dụng các mô hình học máy (Machine Learning – ML) tiên tiến để khắc phục những hạn chế trên, thông qua việc phát triển một hệ thống E-nose tích hợp cảm biến khí Nano có khả năng phân loại và hồi quy nồng độ khí độc hại trong nhà với độ chính xác cao.

Tính mới và sáng tạo của nghiên cứu thể hiện ở việc đề xuất và tối ưu các mô hình học máy xử lý tín hiệu chuỗi thời gian đa chiều từ cảm biến, giúp mô tả hiệu quả mối quan hệ phi tuyến giữa tín hiệu và nồng độ khí. Kết quả thực nghiệm cho thấy mô hình Mạng nơ-ron nhân tạo (ANN) đạt hiệu suất vượt trội: độ chính xác 0.9911 trong bài toán phân loại 8 lớp khí hỗn hợp, và giá trị $R^2 = 0.95$, RMSE = 0.21, MAE = 0.10 trong bài toán hồi quy, tốt hơn các mô hình Random Forest và Support Vector Classifier.

Về đóng góp, nghiên cứu cung cấp giải pháp giám sát thông minh, chi phí thấp, có khả năng cảnh báo sớm nguy cơ ô nhiễm không khí, phục vụ bảo vệ sức khỏe cộng đồng. Đồng thời, kết quả đề tài góp phần nâng cao chất lượng đào tạo và tạo môi trường nghiên cứu ứng dụng Trí tuệ nhân tạo cho sinh viên Trường Đại học Hoa Lư. Giải pháp có tiềm năng ứng dụng rộng rãi trong giám sát môi trường trong nhà, văn phòng và khu công nghiệp, góp phần vào phát triển kinh tế - xã hội bền vững và chuyển đổi số trong lĩnh vực môi trường.

MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu

Chất lượng không khí trong nhà (Indoor Air Quality - IAQ) đóng vai trò quan trọng đối với sức khỏe và hiệu quả làm việc, đặc biệt trong các môi trường kín như nhà ở, văn phòng, hay trường học. Tuy nhiên, các phương pháp truyền thống để giám sát IAQ thường gặp hạn chế về chi phí, độ chính xác, và khả năng triển khai trên diện rộng. Trong những năm gần đây, công nghệ máy học (Machine Learning) đã mở ra hướng đi mới với khả năng phân tích dữ liệu cảm biến phức tạp, dự đoán nồng độ khí ô nhiễm và phát hiện các bất thường trong thời gian thực.

Năm 2021, nghiên cứu của Denglong Ma cùng cộng sự sử dụng PEN 3 electronic nose (AIRSENSE, Germany) gồm 10 cảm biến MOS (Metal Oxide Semiconductor) để phân loại (đơn khí) 10 khí VOCs, dữ liệu gồm 135 mẫu [1]. Một mô hình học sâu (DLM) đã được đề xuất để cải thiện độ chính xác của mảng cảm biến. Kết quả cho thấy mô hình có dữ liệu từ mảng cảm biến phân loại VOC chính xác hơn so với mô hình chỉ có một cảm biến. Hơn nữa, một mạng DLM đơn giản đã được đào tạo để phân loại VOC với độ chính xác là 92%. Sau đó dữ liệu được chuẩn hóa trước khi xây dựng mô hình, điều này đã nâng cao độ chính xác dự đoán lên 96% đối với DLM sau khi so sánh với SVM và VGG-19.

Rabeb Faleh và cộng sự năm 2023 đã đề xuất một mạng lai CNN-LDA để phân loại (hỗn hợp 2 khí) khí ô nhiễm gồm mùi nước hoa, khói thuốc và hỗn hợp của chúng [2]. Một tập dữ liệu khí công khai lấy từ 7 cảm biến MOS đã áp dụng. Kết quả cho thấy độ tin cậy của mô hình lai CNN-LDA, đạt được độ chính xác thử nghiệm cao nhất với phân loại 3 lớp là 93%, so với các mô hình CNN và LDA riêng lẻ với độ chính xác phân loại lần lượt là 90% và 83%.

Trong nghiên cứu của Wangze Ni và cộng sự năm 2024, sử dụng đa cảm biến gồm 8 cảm biến thương mại MOS để phân loại và hồi quy nồng độ (đơn khí) 12 khí VOC, bộ dữ liệu 180 mẫu [3]. Xây dựng mô hình MTL, thành phần chính của mô hình chủ yếu được tạo thành từ các mạng nơ-ron tích chập, nhấn mạnh các đặc điểm không gian của các chuỗi. Việc tích hợp một lớp bộ nhớ ngắn hạn dài đảm bảo duy trì phân tích đặc điểm thời gian của dữ liệu đầu vào, do đó nâng cao hiệu suất dự đoán

Mô hình được đào tạo đạt được độ chính xác là 95,31% và điểm R2 là 0,9510 cho các tác vụ phân loại và hồi quy.

Từ các nghiên cứu, có thể thấy rằng việc ứng dụng máy học trong giám sát chất lượng không khí trong nhà là một hướng đi tiềm năng, được nhiều nhà nghiên cứu trên thế giới quan tâm nhờ khả năng cải thiện độ chính xác trong phân tích dữ liệu và dự đoán nguy cơ ô nhiễm. Tuy nhiên, tại Việt Nam, lĩnh vực này vẫn còn hạn chế, đặc biệt về việc ứng dụng trí tuệ nhân tạo vào giám sát chất lượng không khí trong nhà

2. Tính cấp thiết của nhiệm vụ KH&CN

Chất lượng không khí trong nhà đang trở thành một vấn đề ngày càng quan trọng trong cuộc sống hiện đại, đặc biệt là trong các đô thị đông dân cư. Do các phương pháp truyền thống để giám sát chất lượng không khí thường thiếu tính linh hoạt, chi phí cao, và khó tích hợp vào hệ thống di động. Máy học kết hợp với cảm biến khí trong giám sát khí là một phương pháp tiên tiến được sử dụng để theo dõi và phân tích các loại khí trong môi trường. Cảm biến khí Nano đã trở thành một lựa chọn hứa hẹn để giám sát khí. Các cảm biến khí có kích thước nhỏ, giá thành thấp, tiêu thụ năng lượng ít, và có khả năng phát hiện đo lường nhanh chóng các thành phần hóa học của không khí, các chất khí độc hại như khí CO, CO_2, H_2, H_2S, NO, NO_2, CH_4, VOCs, và các hạt bụi siêu nhỏ (PM2.5) - những thành phần quyết định chất lượng không khí [4][5]. Tuy nhiên, đơn cảm biến khí không đo lường đồng thời nhiều loại khí khác nhau. Trong thực tế rất khó cô lập hoàn toàn các khí cần đo, không khí thường chứa hỗn hợp khí [6][7]. Vì vậy đa cảm biến đã được thiết kế để có thể sử dụng trong nhiều môi trường khác nhau, từ công nghiệp đến môi trường sống, do có khả năng phát hiện nhiều chất khí có thể tồn tại. Dữ liệu đo được từ đa cảm biến đối với hỗn hợp khí phức tạp, đan chéo không tuyến tính [8][9] . Cảm biến khí có một hạn chế đáng chú ý là độ chọn lọc thấp nên những giải pháp thông thường không phát hiện được các loại khí, hỗn hợp khí có sự hiện diện của những khí nào và nồng độ chính xác của từng khí có trong hỗn hợp là bao nhiêu [10].

Mũi điện tử (EN), được thiết kế để mô phỏng hệ thống khứu giác của con, được sử dụng để phát hiện và nhận diện các hợp chất bay hơi khác nhau [11][12].. Dựa trên khứu giác sinh học, một EN di động điển hình bao gồm ba thành phần: một

mảng cảm biến khí có phản ứng chồng chéo, một mạch xử lý tín hiệu trước(ví dụ, bộ chuyển đổi tín hiệu tương tự sang tín hiệu số (ADC) hoặc bộ phân tích trở kháng) và một hệ thống nhận dạng [13][14] . Đặc điểm chi phí thấp, di động và không xâm lấn của hệ thống EN đã khiến nó trở thành công cụ không thể thiếu trong lĩnh vực cảm biến khí. Kết quả là, EN đã được ứng dụng rộng rãi trong nhiều lĩnh vực như công nghiệp thực phẩm, quản lý sức khỏe, chẩn đoán bệnh, kiểm soát chất lượng nước và không khí, và phát hiện rò rỉ khí độc [15][16].

Tuy nhiên, các EN hiện nay gặp khó khăn trong việc phân tích các hỗn hợp khí, đặc biệt khi các loại khí có sự tương tác hóa học với nhau làm thay đổi tín hiệu nhận được từ cảm biến. Điều này gây ra sai lệch trong quá trình phân tích và làm giảm độ chính xác của mô hình. Khi áp dụng cho hỗn hợp khí, hiệu suất của mô hình giảm rõ rệt do không thể giải quyết được sự chồng chéo tín hiệu.

Đề tài sẽ tập trung vào giải quyết các hạn chế trên bằng đề xuất các giải pháp và mô hình học máy phù hợp để xử lý các tín hiệu chuỗi thời gian ngắn đa chiều phi tuyến từ đa cảm biến hiệu quả nhằm nâng cao độ chính xác trong việc giám sát các khí ô nhiễm trong nhà.

Việc ứng dụng công nghệ máy học (Machine Learning) trong phân tích và dự đoán chất lượng không khí không chỉ là một giải pháp hiện đại mà còn là bước tiến quan trọng trong lĩnh vực trí tuệ nhân tạo (AI). Nó góp phần tối ưu hóa quá trình giám sát môi trường, đáp ứng nhu cầu ngày càng tăng về các giải pháp công nghệ thông minh trong quản lý và bảo vệ môi trường. Mặt khác nó còn mở ra cơ hội nghiên cứu và thực hành cho sinh viên ngành Công nghệ Thông tin Trường Đại học Hoa Lư trong các học phần về Trí tuệ nhân tạo, từ đó nâng cao chất lượng đào tạo và kết nối kiến thức lý thuyết với thực tiễn.

Với những lý do trên, có thể khẳng định việc thực hiện đề tài là một nhiệm vụ cấp thiết và mang lại nhiều lợi ích lâu dài cho Trường Đại học Hoa Lư. Do đó, chúng tôi lựa chọn “Ứng dụng máy học trong việc giám sát chất lượng không khí trong nhà” làm chủ đề nghiên cứu cho nhiệm vụ khoa học và công nghệ năm 2025 của mình.

3. Mục tiêu của nhiệm vụ KH&CN

Mục tiêu của đề tài là áp dụng các mô hình máy học vào việc giám sát các khí độc hại trong nhà.

4. Đối tượng và phạm vi nghiên cứu của nhiệm vụ KH&CN

4.1. Đối tượng nghiên cứu

- Nghiên cứu về mũi điện tử, bộ dữ liệu
- Nghiên cứu các công cụ, ngôn ngữ để thử nghiệm chương trình
- Nghiên cứu các mô hình máy học.

4.2. Phạm vi nghiên cứu

- Ngôn ngữ lập trình cho mô hình: Python
- Bộ dữ liệu: hỗn hợp khí độc hại trong nhà
- Mô hình, thuật toán học nông, học sâu

5. Cách tiếp cận và phương pháp nghiên cứu

5.1. Cách tiếp cận

Tiếp cận từ lý thuyết -> ứng dụng vào bài toán thực tiễn -> giải pháp công nghệ

- Nghiên cứu về bài toán phân loại và hồi quy nồng độ khí
- Nghiên cứu các công cụ để thử nghiệm huấn luyện mô hình.

5.2. Phương pháp nghiên cứu

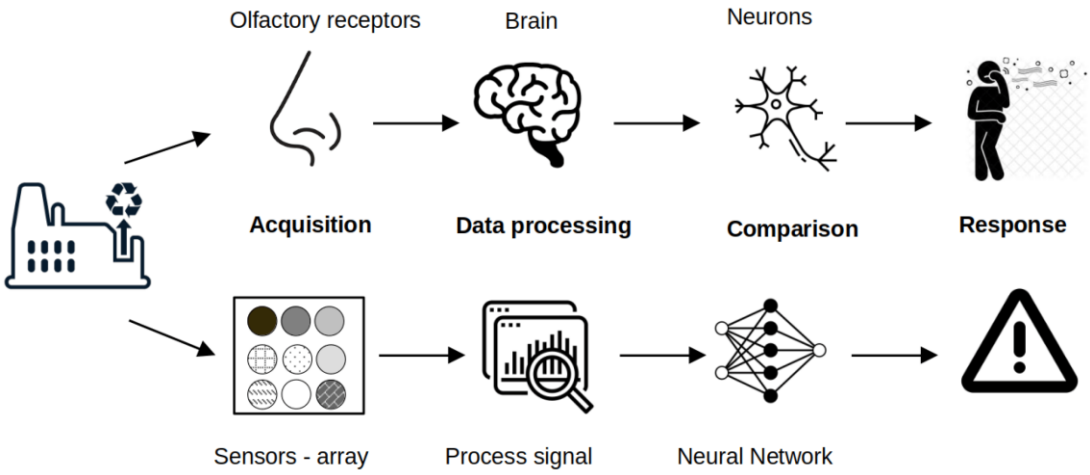
Phương pháp nghiên cứu lý thuyết: nghiên cứu về mũi điện tử, cảm biến, máy học truyền thống, học sâu, các công cụ thử nghiệm chương trình

Phương pháp thực nghiệm: xây dựng các mô hình và huấn luyện tối ưu hoá

CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN MŨI ĐIỆN TỬ VÀ HỌC MÁY

Chương này trình bày cơ sở lý thuyết và bối cảnh nghiên cứu liên quan đến mũi điện tử và các phương pháp học máy, nhằm xây dựng nền tảng cho những nội dung phân tích và phát triển mô hình ở các chương tiếp theo. Phần đầu của chương này cung cấp một cái nhìn tổng quát và chi tiết về khái niệm, cấu tạo, nguyên lý hoạt động, cũng như các hướng nghiên cứu và ứng dụng của mũi điện tử. Mục tiêu là xây dựng nền tảng lý thuyết và thực tiễn để hiểu rõ bài toán nghiên cứu phát triển mũi điện tử, với trọng tâm đặc biệt là sử dụng đa cảm biến khí Nano trong giám sát chất lượng không khí trong nhà [17][18][12][19]. Phần tiếp theo cấp tổng quan về học máy và học sâu, từ các mô hình cơ bản đến các ứng dụng đa dạng, đồng thời trình bày quy trình xây dựng mô hình dựa trên học máy để giải quyết bài toán dự báo và phân tích dữ liệu khí. Qua đó, chương 1 đóng vai trò định hướng, làm rõ mối liên hệ giữa công nghệ cảm biến tiên tiến và các thuật toán học máy hiện đại, tạo nền tảng khoa học vững chắc cho các nghiên cứu tiếp theo.

1.1. Khái niệm mũi điện tử



Hình 1.1. Mô hình các đa cảm biến khí kết hợp học máy được so sánh như "mũi điện tử" trong việc giám sát các khí khác nhau [*Sensors* 2022, *22*(4), 1510].

Mũi điện tử (EN) là một thiết bị được thiết kế để mô phỏng hệ thống khứu giác của con người, nhằm phát hiện và nhận diện các loại khí/mùi hoặc hợp chất hữu cơ dễ bay hơi (VOCs) [11]. Thiết bị này sử dụng một mảng cảm biến hóa học để phản

ứng với các phân tử khí/mùi, tạo ra các tín hiệu đặc trưng như Hình 1.1. Các tín hiệu này sau đó được xử lý bằng các thuật toán học máy hoặc học sâu để phân loại, định lượng hoặc dự đoán nồng độ của các khí cụ thể.

1.2. Nguyên lý cơ bản của phân tích khí bằng mũi điện tử

Mũi điện tử hoạt động dựa trên việc phát hiện và phân tích các đặc tính hóa học của khí thông qua mảng cảm biến khí và công nghệ xử lý tín hiệu, với các bước cơ bản như sau:

- Thu thập mẫu khí
- Phát hiện tín hiệu bằng mảng cảm biến
- Xử lý tín hiệu
- Phân tích mẫu bằng thuật toán

Mẫu tín hiệu được phân tích bởi các thuật toán học máy hoặc học sâu:

* Phân loại (Classification): Xác định loại khí trong mẫu.

* Hồi quy (Regression): Dự đoán nồng độ của từng thành phần khí.

* Phát hiện bất thường (Anomaly Detection): Nhận diện sự bất thường trong khí thải hoặc môi trường.

Các mô hình phổ biến bao gồm: KNN, SVM, ANN, CNN, hoặc mô hình học chuyển giao.

- Đưa ra kết quả: kết quả cuối cùng có thể bao gồm: Loại khí hoặc hỗn hợp khí; Nồng độ từng thành phần khí; Phát hiện các khí bất thường hoặc vượt ngưỡng an toàn.

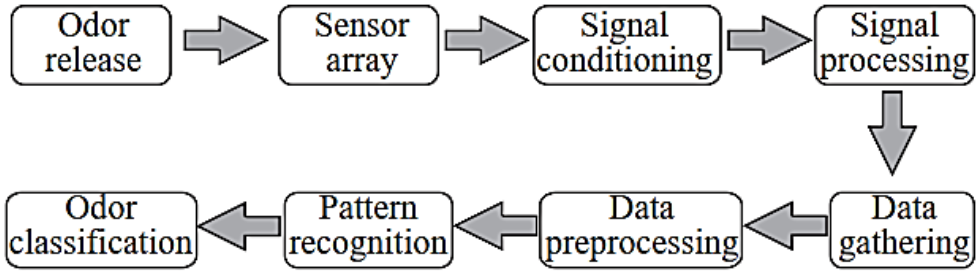
Tóm lại

Nguyên lý hoạt động của mũi điện tử dựa vào sự tương tác của khí với mảng cảm biến, kết hợp với công nghệ xử lý tín hiệu và các thuật toán phân tích dữ liệu, nhằm tái hiện khả năng nhận biết mùi của con người một cách chính xác và đáng tin cậy.

1.3. Cấu tạo mũi điện tử

EN bao gồm cả thành phần phần cứng và phần mềm như được mô tả ngắn gọn trong ứng dụng phân loại mùi ở Hình 2. Ban đầu, khí xung quanh được giải phóng được hấp thụ bởi mảng cảm biến. Việc phát hiện tín hiệu đầu vào diễn ra theo sự thay đổi về điện áp, dòng điện, tần số, các thông số điện trở tùy thuộc vào loại thành phần

trong mảng cảm biến. Vì các loại cảm biến riêng biệt thường được sử dụng trong mảng cảm biến, nên các tín hiệu thu được phải được xử lý trước để hiểu đúng những thay đổi vật lý đó và sau đó được xử lý để số hóa chúng nhằm tạo thành một tập dữ liệu. Do đó, các tín hiệu cảm nhận được được xử lý thích hợp, ví dụ, khuếch đại, lọc hoặc chuyển đổi, để dễ dàng sử dụng trong các giai đoạn tiếp theo[22].



Hình 1.2. Mô tả về EN bao gồm cả thành phần phần cứng và phần mềm

Các tín hiệu đã xử lý sau đó được phân tích theo các thuộc tính cụ thể của chúng trong giai đoạn thu thập dữ liệu. Sau đó, dữ liệu đủ được thu thập từ các tín hiệu này và dữ liệu thu được được xử lý trước theo các yêu cầu của thuật toán nhận dạng mẫu được sử dụng. Cuối cùng, mùi được phân loại theo giai đoạn nhận dạng mẫu.

**** Khái niệm và nguyên tắc hoạt động của cảm biến khí***

+ Khái niệm cảm biến khí

Cảm biến khí là thiết bị dùng để phát hiện và đo lường nồng độ của các loại khí cụ thể trong môi trường. Chúng chuyển đổi tín hiệu từ khí được phát hiện thành tín hiệu điện để xử lý hoặc phân tích. Những cảm biến này thường được sử dụng trong các lĩnh vực như giám sát môi trường, công nghiệp, y tế, nông nghiệp và các ứng dụng an toàn.

+ Nguyên tắc hoạt động của cảm biến khí

Cảm biến khí hoạt động dựa trên nguyên lý chuyển đổi sự thay đổi vật lý hoặc hóa học khi tiếp xúc với khí mục tiêu thành tín hiệu điện.

**** Phân loại cảm biến khí***

Cảm biến khí có thể được phân loại dựa trên nguyên lý hoạt động, loại khí phát hiện, hoặc phương pháp chế tạo. Dưới đây là một số phân loại phổ biến:

+ Cảm biến quang học (Optical)

+ Cảm biến sóng âm bề mặt (Surface Acoustic Wave)

+ Cảm biến điện hóa (Electrochemical)

+ Cảm biến nhiệt

+ Cảm biến hồng ngoại

+ Cảm biến bán dẫn (Semiconductor Gas SENsors)

Nguyên lý: Khi khí mục tiêu tiếp xúc với bề mặt của vật liệu bán dẫn (thường là oxit kim loại như SnO_2, ZnO), xảy ra phản ứng hóa học, làm thay đổi điện trở của cảm biến.

Ứng dụng: Phát hiện khí độc như CO, NO_2, hoặc khí dễ cháy như CH_4.

Tóm lại

Cảm biến khí có nhiều loại với nguyên lý hoạt động và ứng dụng khác nhau, được lựa chọn tùy thuộc vào yêu cầu cụ thể như loại khí cần phát hiện, môi trường hoạt động, và độ nhạy cần thiết.

Việc lựa chọn loại cảm biến phù hợp phụ thuộc vào yêu cầu cụ thể về độ nhạy, độ chọn lọc, môi trường hoạt động, và ngân sách của ứng dụng. Cảm biến quang học là lý tưởng cho các ứng dụng cần độ chính xác cao, trong khi cảm biến bán dẫn và xúc tác phù hợp hơn cho các ứng dụng chi phí thấp.

1.4. Cảm biến khí Nano

Cảm biến khí nano là một loại cảm biến sử dụng vật liệu nano (như dây nano, màng mỏng nano) để phát hiện và phân tích các khí trong môi trường. Các cảm biến này có khả năng phát hiện các khí độc hại, khí ô nhiễm hoặc khí có mùi đặc trưng với độ nhạy và tốc độ phản hồi cao nhờ vào diện tích bề mặt lớn và tính chất đặc biệt của vật liệu nano.

* Ưu điểm của cảm biến khí nano: độ nhạy cao, tốc độ phản hồi nhanh, kích thước nhỏ gọn, tiết kiệm năng lượng, giá thành thấp.

* Nhược điểm của cảm biến khí nano: độ chọn lọc thấp, ảnh hưởng của môi trường, khả năng bền vững, khó sản xuất đồng đều.

1.5. Bài toán phát triển mũi điện tử để giám sát chất lượng không khí trong nhà sử dụng cảm biến khí Nano

+ Mục tiêu của bài toán

Mục tiêu là phát triển một hệ thống mũi điện tử sử dụng **đa cảm biến khí Nano** để giám sát chất lượng không khí trong nhà, giúp phân loại và hồi quy chính xác nồng độ các khí độc hại ảnh hưởng đến sức khỏe con người..

+ Đặc điểm của bài toán

- Đầu vào (Input): Dữ liệu từ đa cảm biến khí Nano

Đa cảm biến khí Nano sẽ nhận diện các loại khí độc hại có mặt trong không khí như: CO_2 (carbon dioxide), CO (carbon monoxide), NH_3 (ammonia), NO_2 (nitrogEN dioxide), VOCs (Volatile Organic Compounds) - các hợp chất hữu cơ dễ bay hơi, O_3 (ozone), các mùi và các khí khác.

Cảm biến khí Nano có đặc điểm nhạy bén, khả năng phân biệt khí ở nồng độ thấp, và độ ổn định cao trong các điều kiện môi trường thay đổi như nhiệt độ và độ ẩm.

- **Quá trình xử lý:**

- ***Thu thập và tiền xử lý dữ liệu****:* Thu thập dữ liệu từ các cảm biến khí Nano, loại bỏ nhiễu và làm sạch dữ liệu cảm biến, đảm bảo tính chính xác và ổn định của dữ liệu.
- ***Chuẩn hóa và hiệu chuẩn dữ liệu***: Các giá trị từ cảm biến khí Nano có thể có đơn vị khác nhau, cần chuẩn hóa để đảm bảo đồng nhất và dễ dàng xử lý. Sử dụng các phương pháp như chuẩn hóa min-max hoặc chuẩn hóa z-score.
- ***Trích xuất đặc trưng***

Các đặc trưng cần thiết từ dữ liệu cảm biến khí Nano sẽ được trích xuất để phục vụ cho việc phân tích và nhận diện khí. Ví dụ, sử dụng các phương pháp như:

* Phân tích thành phần chính (PCA) để giảm số lượng đặc trưng nhưng vẫn giữ lại thông tin quan trọng.

* Các đặc trưng thống kê như trung bình, phương sai, và đặc trưng tần số.

* Các đặc trưng liên quan đến sự biến đổi tín hiệu qua thời gian.

* Các kỹ thuật để tăng cường dữ liệu

- ***Phân tích và nhận diện khí****:*

Áp dụng các mô hình học máy (machine learning) và học sâu (deep learning) để phân loại và nhận diện các khí có mặt trong không khí:

- **Đầu ra (Output)**

- Phân loại các khí độc hại hoặc hỗn hợp khí độc hại
- Hồi quy nồng độ của các khí độc hại hoặc hỗn hợp khí độc hại

1.6. Tổng quan về học máy và học sâu

"Khả năng học của máy tính mà không cần lập trình rõ ràng cho một nhiệm vụ cụ thể" được định nghĩa là học máy (ML) [20] và được đặt ra vào năm 1959 [21].

Học máy (tiếng Anh: machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể.

Một số định nghĩa khác về học máy như sau:

✓ Một quá trình nhờ đó một hệ thống cải thiện hiệu suất (hiệu quả hoạt động) của nó

✓ Một quá trình mà một chương trình máy tính cải thiện hiệu suất của nó trong một công việc thông qua kinh nghiệm

✓ Việc lập trình các máy tính để tối ưu hóa một tiêu chí hiệu suất dựa trên các dữ liệu ví dụ hoặc kinh nghiệm trong quá khứ.

Một chương trình máy tính cần các quy tắc, luật lệ để có thể thực thi được một tác vụ nào đó như dán nhãn cho các email là thư rác nếu nội dung email có chứa từ khoá "quảng cáo". Nhưng với học máy, các máy tính có thể tự động phân lại các thư rác thành mà không cần chỉ trước bất kỳ quy tắc nào cả. Có thể hiểu đơn giản là nó giúp cho máy tính có được cảm quan và suy nghĩ được như con người. Nói một cách khác, học máy là phương pháp vẽ các đường thể hiện mối quan hệ của tập dữ liệu. Ví dụ như đường ngăn cách 2 loại dữ liệu cho nhãn khác nhau, đường thể hiện xu hướng của giá nhà phụ thuộc vào diện tích và trí hay các đường phân cụm dữ liệu.

Biểu diễn một bài toán học máy

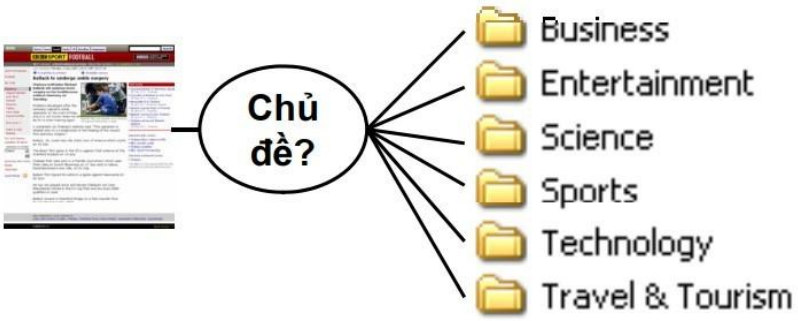
Học máy = Cải thiện hiệu quả một công việc thông qua kinh nghiệm

✓ Một công việc (nhiệm vụ): *T*

✓ Các tiêu chí đánh giá hiệu năng (đánh giá mô hình): *P*

✓ Thông qua (sử dụng) kinh nghiệm: *E*

Ví dụ 4.2: Phân loại các trang Web như Hình 1.3.



Hình 1.3. Phân loại trang web

- ✓ *T*: Phân loại các trang Web theo các chủ đề đã định trước
- ✓ *P*: Tỷ lệ (%) các trang Web được phân loại chính xác
- ✓ *E*: Một tập các trang Web, trong đó mỗi trang Web gắn với một chủ đề tương ứng

Học sâu (deep learning) là một tập hợp con của học máy (machine learning), tập trung vào việc xây dựng và huấn luyện mạng nơ-ron nhiều lớp, được gọi là mạng nơ-ron sâu (DNN – Deep neural networks) để chúng có thể tự động học, hiểu dữ liệu, mô phỏng khả năng ra quyết định phức tạp của bộ não con người.

Mô hình học sâu có thể nhận diện nhiều hình mẫu phức tạp trong hình ảnh, văn bản, âm thanh và các dữ liệu khác để tạo ra thông tin chuyên sâu và dự đoán chính xác. Đây là một sự kết hợp giữa toán học và khoa học thần kinh. Kết quả của nó mang lại cực kỳ to lớn, có thể coi là khởi nguyên của nhiều ngành công nghiệp mới. Tại thời điểm này, hầu hết các công ty lớn trong ngành công nghệ thông tin lẫn các ngành khác như ô tô, điện tử đều đang tập trung phát triển và ứng dụng kỹ thuật học sâu cho bài toán của mình. Ví dụ như AlphaGo của Google đã chiến thắng nhà vô địch cờ vây Lee Sedol vào tháng 3 năm 2016. Tính năng nhận diện khuôn mặt khá chính xác của Facebook được triển khai vào năm 2016. Trợ lý ảo Siri của Apple được giới thiệu từ năm 2006. Xe tự lái của Google được thử nghiệm chính thức trên đường phố vào năm 2015, ChatGPT năm 2022…

1.7. Các mô hình học máy cơ bản

1.7.1. Mô hình có giám sát (Supervised Learning)

Mô hình có giám sát yêu cầu dữ liệu huấn luyện có gán nhãn, trong đó mỗi đầu vào (input) đều được liên kết với một đầu ra mong muốn.

Đặc điểm:

Dữ liệu: Gồm tập hợp các cặp đầu vào - đầu ra (X,Y), trong đó X là dữ liệu đầu vào và Y là nhãn tương ứng.

Mục tiêu: Học một ánh xạ từ X→Y sao cho có thể dự đoán chính xác Y mới từ X chưa thấy.

Ví dụ ứng dụng:

Hồi quy (Regression): Dự đoán nồng độ khí (giá trị liên tục) từ dữ liệu cảm biến.

Phân loại (Classification): Phân loại loại khí (ví dụ: NH3, H2S) dựa trên đặc trưng cảm biến.

Thuật toán phổ biến:

Hồi quy tuyến tính (Linear Regression).

Mạng nơ-ron nhân tạo (Artificial Neural Networks).

Máy vector hỗ trợ (Support Vector Machines).

Rừng ngẫu nhiên (Random Forest).

Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), RecurrENt Neural Networks (RNN) và Long Short-Term Memory (LSTM).

1.7.2. Mô hình bán giám sát (Semi-Supervised Learning)

Mô hình bán giám sát sử dụng cả dữ liệu gán nhãn và chưa gán nhãn trong quá trình huấn luyện.

Đặc điểm: Dữ liệu: Chỉ một phần nhỏ dữ liệu đầu vào được gán nhãn, phần lớn còn lại chưa được gán nhãn.

Mục tiêu: Tận dụng dữ liệu chưa gán nhãn để cải thiện độ chính xác của mô hình.

Ví dụ ứng dụng: Dự đoán nồng độ khí khi chỉ có một số mẫu được gán nhãn và các mẫu khác không rõ giá trị chính xác. Phát hiện rò rỉ khí bằng cách sử dụng dữ liệu cảm biến chưa gán nhãn từ các tình huống thực.

Thuật toán phổ biến: Self-training (Tự học), Co-training (Học hợp tác), Semi-Supervised Support Vector Machines (S3VM), mạng nơ-ron bán giám sát (Semi-Supervised Neural Networks), GENerative Adversarial Networks (GANs)

1.7.3. Mô hình không giám sát (Unsupervised Learning)

Mô hình không giám sát không yêu cầu dữ liệu gán nhãn mà chỉ làm việc với dữ liệu đầu vào (X).

Đặc điểm:

Dữ liệu: Chỉ có đầu vào mà không có nhãn đầu ra.

Mục tiêu: Tìm kiếm mẫu, cấu trúc ẩn, hoặc mối quan hệ trong dữ liệu.

Ví dụ ứng dụng:

Phân cụm (Clustering): Nhóm các loại khí khác nhau dựa trên tín hiệu cảm biến mà không cần biết trước loại khí cụ thể.

Giảm chiều dữ liệu (DimENsionality Reduction): Tóm tắt các đặc trưng cảm biến quan trọng.

Phát hiện bất thường (Anomaly Detection): Phát hiện các mẫu dữ liệu bất thường, chẳng hạn khí độc đột ngột.

Thuật toán phổ biến: K-means Clustering, Gaussian Mixture Models (GMM), Principal component Analysis (PCA), T-SNE (T distributed Stochastic Neighbor Embedding), autoencoders (aes)

1.7.4. Một số mô hình cơ bản

a) Cây quyết định

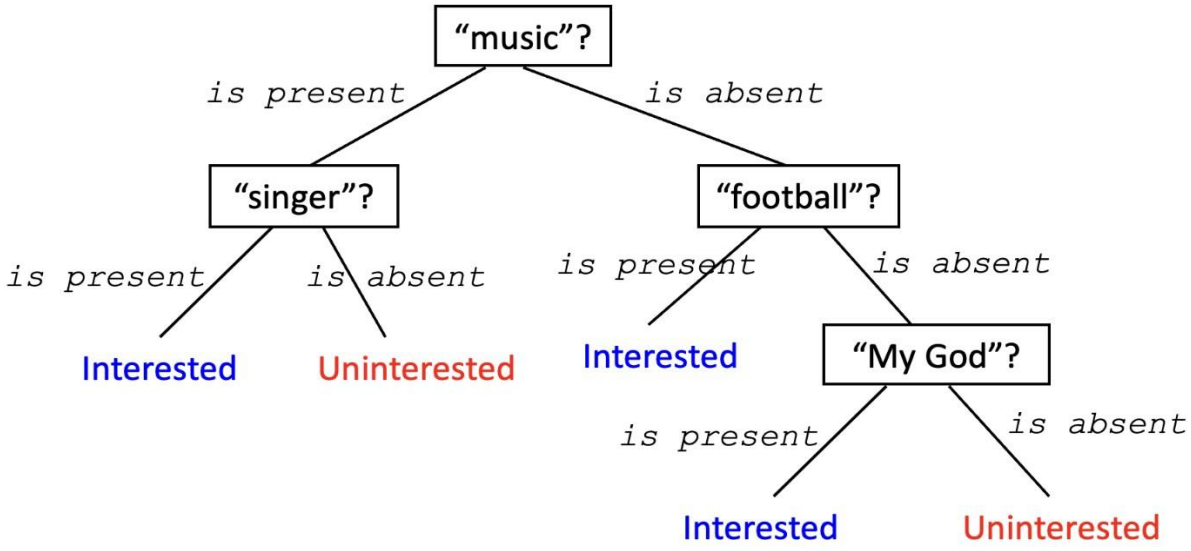
** Giới thiệu*

Đối với bài toán học có giám sát, chúng ta thường cần tìm hàm y^* từ một tập dữ liệu

D đã có nhãn. Do không biết hàm y^*, nên ta thường chọn một lớp mô hình H để xấp xỉ nó. Ở đây chúng ta sẽ tìm hiểu một cách tiếp cận hoàn toàn khác. Ta sẽ không chọn trước hình dáng của hàm, mà sẽ dùng cấu trúc cây (tree structure) để biểu diễn các hàm. Mỗi cây sẽ biểu diễn một hàm cụ thể. Như vậy việc học sẽ đi tìm một cây mà có thể xấp xỉ y^* tốt. Những cây như thế được gọi là *Cây quyết định (Decision tree).*

Cụ thể hơn, cây quyết định là một cấu trúc dạng cây trong đó mỗi nút trong biểu diễn một thuộc tính cần kiểm tra giá trị đối với các mẫu, mỗi nhánh từ một nút sẽ tương ứng với một giá trị có thể của thuộc tính gắn với nút đó và mỗi nút lá biểu diễn một lớp hay là một dự đoán cuối cùng (Hình 1.4). Mục đích của ta là

tạo ra một mô hình dự đoán giá trị của biến đích bằng cách học các luật quyết định IF-THEN đơn giản được suy ra từ các thuộc tính của dữ liệu. Một cây quyết định học được sẽ dự đoán một mẫu dữ liệu bằng cách duyệt cây từ nút gốc đến một nút lá, trong đó nhãn lớp hay giá trị gắn với nút lá đó sẽ được dùng để dự đoán. Như vậy, một cây sẽ biểu diễn một hàm phân loại hoặc hồi qui nào đó.



[("music" is present) ∧ ("singer" is present)] ∨

[("music" is absent) ∧ ("football" is present)] ∨

[("music" is absent) ∧ ("football" is absent) ∧ ("My God" is present)]

Hình 1.4. Ví dụ về DT. DT được dùng để biểu diễn tri thức về thói quen xem các chương trình truyền hình của một người.

Các mô hình cây trong đó biến đích lấy một tập giá trị rời rạc được gọi là cây phân loại; trong các cấu trúc cây này, mỗi đường đi (path) từ nút gốc đến một nút lá sẽ tương ứng với một kết hợp (conjunction) của các kiểm tra giá trị thuộc tính (attribute tests). Cây quyết định (bản thân nó) chính là một phép tuyển (disjunction) của các kết hợp (conjunctions) này. Cây quyết định trong đó biến đích lấy các giá trị liên tục (thường là số thực) được gọi là cây hồi quy. Cây quyết định là một trong những thuật toán học máy phổ biến nhất nhờ tính dễ hiểu và đơn giản của chúng.

** ID3*

Ý tưởng

Trong phần này, chúng ta sẽ làm quen với một thuật toán xây dựng cây quyết định ra đời từ rất sớm. ID3 (Iterative Dichotomiser 3) do Ross Quinlan đề xuất năm 1986, là một thuật toán xây dựng/học cây quyết định được áp dụng cho các bài toán phân loại mà tất cả các thuộc tính đều có kiểu định danh/phạm trù.

Trong ID3, chúng ta cần xác định thứ tự của thuộc tính cần được xem xét tại mỗi bước. Với các bài toán có nhiều thuộc tính và mỗi thuộc tính có nhiều giá trị khác nhau, việc tìm được cây tối ưu thường là không khả thi. Thay vào đó, một phương pháp đơn giản thường được sử dụng là tại mỗi bước, một thuộc tính tốt nhất sẽ được chọn ra dựa trên một tiêu chuẩn nào đó. Việc chọn ra thuộc tính tốt nhất ở mỗi bước như thế này được gọi là cách chọn tham lam (greedy). Cách chọn này có thể không phải là tối ưu, nhưng trực giác cho chúng ta thấy rằng cách làm này sẽ gần với cách làm tối ưu. Ngoài ra, cách làm này khiến cho bài toán cần giải quyết trở nên đơn giản hơn.

ID3 thực hiện tìm kiếm tham lam trên không gian các cây quyết định để xây dựng một cây quyết định theo chiến lược top-down, bắt đầu từ nút gốc. Cụ thể:

- Ở mỗi nút, chọn thuộc tính kiểm tra tốt nhất - là thuộc tính có khả năng phân loại tốt nhất đối với các mẫu học gắn với nút đó.
- Tạo mới một cây con của nút hiện tại cho mỗi giá trị có thể của thuộc tính kiểm tra, và tập học sẽ được tách ra thành các tập con tương ứng với cây con vừa tạo.

Quá trình phát triển cây quyết định sẽ tiếp tục cho đến khi:

- Cây quyết định phân loại hoàn toàn các mẫu học, hoặc
- Tất cả các thuộc tính đã được sử dụng

Trong quá trình xây dựng một cây quyết định, với mỗi thuộc tính được chọn, ta chia dữ liệu tại nút đang xét vào các nút con tương ứng với tất cả các giá trị có thể của thuộc tính đó rồi tiếp tục áp dụng phương pháp này cho mỗi nút con. Do vậy, mỗi thuộc tính chỉ được phép xuất hiện tối đa 1 lần đối với bất kỳ một đường đi nào trong cây.

Để thực hiện ID3, tại nút đang xét, chúng ta cần chọn được thuộc tính có khả năng phân loại tốt nhất đối với các mẫu học gắn với nút đó để chia dữ liệu vào các nút con của nó. Bằng trực giác, một phép phân chia là tốt nhất nếu dữ liệu trong mỗi nút con hoàn toàn thuộc vào một lớp–khi đó nút con này có thể được coi là một nút lá, tức ta không cần phân chia thêm nữa. Nếu dữ liệu trong các nút con vẫn lẫn vào các lớp khác nhau theo tỉ lệ lớn, ta coi rằng phép phân chia đó chưa thực sự tốt. Từ nhận xét này, ta cần có một hàm số đo độ thuần khiết (purity) của một phép phân chia. Hàm số này sẽ cho giá trị thấp nhất nếu dữ liệu trong mỗi nút con nằm trong cùng một lớp (tinh khiết nhất), và cho giá trị cao nếu mỗi nút con có chứa dữ liệu thuộc nhiều lớp khác nhau. Entropy là một độ đo trong Lý thuyết thông tin được sử dụng để đo mức độ hỗn tạp của một tập và Entropy=0 nếu tất cả các mẫu thuộc cùng một lớp. Vì vậy, thuộc tính có khả năng phân loại tốt nhất đối với các mẫu học gắn với nút đang xét là thuộc tính sao cho nếu dùng thuộc tính đó để phân chia, entropy sẽ giảm đi một lượng lớn nhất. Chúng ta gọi Information Gain của một thuộc tính đối với một tập S các mẫu dữ liệu là độ đo dùng để đo mức độ giảm Entropy hay lượng thông tin (trung bình) bị mất nếu chia S theo tất cả các giá trị có thể của thuộc tính đó.

Information Gain của thuộc tính A đối với tập S được tính như sau:

$$Gain(\boldsymbol{S},A) = Entropy(\boldsymbol{S}) - \sum_{v \in Values(A)} \frac{|\boldsymbol{S}_v|}{|\boldsymbol{S}|} Entropy(\boldsymbol{S}_v)$$

trong đó *Values*(*A*) là tập các giá trị có thể của thuộc tính A, và

$S_v = \{x : x \in S$ và thuộc tính A trong x có giá trị là $v\}$

Trong công thức trên, thành phần thứ 2 thể hiện giá trị Entropy sau khi tập S được phân chia bởi các giá trị của thuộc tính A.

Chiến lược tìm kiếm của giải thuật ID3

ID3 thực hiện tìm kiếm tham lam trên không gian các cây quyết định để xây dựng một cây quyết định theo chiến lược top-down, bắt đầu từ nút gốc. Do vậy, ID3 chỉ đảm bảo tìm được lời giải tối ưu cục bộ (locally optimal solution) chứ không đảm bảo tìm được lời giải tối ưu tổng thể (globally optimal solution). Đặc biệt, một khi một thuộc tính được chọn là thuộc tính kiểm tra cho một nút, ID3

không bao giờ cân nhắc lại lựa chọn này. ID3 luôn chọn cây quyết định phù hợp đầu tiên tìm thấy trong quá trình tìm kiếm của nó và ưu tiên các cây quyết định đơn giản (chiều cao cây thấp) trong đó một thuộc tính có giá trị Information Gain càng lớn thì sẽ là thuộc tính kiểm tra của một nút càng gần nút gốc.

Ví dụ

Để mọi thứ được rõ ràng hơn, chúng ta cùng xem ví dụ với dữ liệu huấn luyện được cho trong Bảng dưới đây. Bảng dữ liệu này được lấy từ cuốn sách D Mining: Practical Machine Learning Tools and Techniques, trang 11. Đây là một bảng dữ liệu được sử dụng rất nhiều trong các bài giảng về cây quyết định. Bảng 1.1. dữ liệu này mô tả mối quan hệ giữa thời tiết trong 14 ngày (bốn cột đầu, không tính cột id) và việc một người chơi (không chơi) tennis (cột cuối cùng). Nói cách khác, ta phải dự đoán giá trị ở cột cuối cùng nếu biết giá trị của bốn cột còn lại.

Theo giải thuật ID3, tại nút gốc, chúng ta cần xác định được thuộc tính nào trong số Outlook, Temperature, Humidity, Wind nên được chọn là thuộc tính kiểm tra? Chúng ta hãy tính giá trị Information Gain của 4 thuộc tính trên đối với tập học *S* các mẫu từ bảng dữ liệu trên. Cụ thể, chúng ta sẽ tính Gain(*S*,Wind)?

Bảng 1.1. Dữ liệu về thời tiết

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Thuộc tính Wind có 2 giá trị có thể là Weak và Strong. S = {9 ví dụ lớp Yes và 5 ví dụ lớp No}. Gọi

S_{Weak} = {6 ví dụ lớp Yes và 2 ví dụ lớp No: giá trị của thuộc tính Wind là Weak}, S_{Strong} = {3 ví dụ lớp Yes và 3 ví dụ lớp No: giá trị của thuộc tính Wind là Strong}.

Như vậy, chúng ta có:

$$\begin{aligned} Gain(\boldsymbol{S}, Wind) &= Entropy(\boldsymbol{S}) - \sum_{v \in \{Strong, Weak\}} \frac{|\boldsymbol{S}_v|}{|\boldsymbol{S}|} Entropy(\boldsymbol{S}_v) \\ &= Entropy(\boldsymbol{S}) - \frac{8}{14} Entropy(\boldsymbol{S}_{Weak}) - \frac{6}{14} Entropy(\boldsymbol{S}_{Strong}) \\ &= 0.94 - \frac{8}{14} \times 0.81 - \frac{6}{14} \times 1 \\ &= 0.048 \end{aligned}$$

Tương tự, chúng ta tính giá trị Information Gain của 3 thuộc tính còn lại Outlook, Temperature, Humidity:

- Gain(*S*, Outlook) = ... = 0.246
- Gain(*S*, Temperature) = ... = 0.029
- Gain(*S*, Humidity) = ... = 0.151
- Và Gain(*S*, Wind) = ... = 0.048

Vì vậy, Outlook được chọn là thuộc tính kiểm tra cho nút gốc!

b. Rừng ngẫu nhiên (Random Forests)

Ý tưởng

Trong thuật toán Cây quyết định, khi xây dựng cây quyết định nếu để độ sâu tùy ý thì cây sẽ phân loại đúng hết các dữ liệu trong tập luyện dẫn đến mô hình có thể dự đoán tệ trên tập kiểm thử, khi đó mô hình bị overfitting, hay nói cách khác là mô hình có high variance.

Thuật toán Random Forests gồm nhiều cây quyết định, mỗi cây quyết định đều có những yếu tố ngẫu nhiên:

1. Lấy ngẫu nhiên dữ liệu để xây dựng cây quyết định.
2. Lấy ngẫu nhiên các thuộc tính để xây dựng cây quyết định.

Do mỗi cây quyết định trong thuật toán học Random Forests không dùng tất cả dữ liệu luyện, cũng như không dùng tất cả các thuộc tính của dữ liệu để xây dựng cây nên mỗi cây có thể sẽ dự đoán không tốt, khi đó mỗi mô hình cây quyết định không bị overfitting mà có thế bị underfitting, hay nói cách khác là mô hình có high bias. Tuy nhiên, kết quả cuối cùng của thuật toán Random Forests lại tổng hợp từ nhiều cây quyết định, thế nên thông tin từ các cây sẽ bổ sung thông tin cho nhau, dẫn đến mô hình có low bias và low variance, hay mô hình có kết quả dự đoán tốt.

Random Forest là một thuật toán học máy phổ biến thuộc về phương pháp học có giám sát. Nó có thể được sử dụng cho cả bài toán Phân loại và Hồi quy trong Học máy. Nó dựa trên khái niệm học kết hợp, là một quá trình kết hợp nhiều bộ phân loại để giải quyết một vấn đề phức tạp và để cải thiện hiệu suất của mô hình.

Rừng ngẫu nhiên cũng là một thuật toán học máy linh hoạt, dễ sử dụng, tạo ra kết quả tuyệt vời ngay cả khi không điều chỉnh siêu tham số. Nó cũng là một trong những thuật toán được sử dụng nhiều nhất, do tính đơn giản và đa dạng của nó. Như tên gọi của phương pháp học máy này, "Rừng ngẫu nhiên là một bộ phân loại chứa một số cây quyết định trên các tập con khác nhau của tập dữ liệu cũng như các tập con thuộc tính khác nhau và lấy giá trị dự đoán trung bình để cải thiện độ chính xác dự đoán của tập dữ liệu đó."

Thuật toán học Random Forests

Giả sử tập dữ liệu huấn luyện D gồm có n mẫu dữ liệu (sample) và mỗi dữ liệu có d thuộc tính (feature). Chúng ta học K cây quyết định như sau:

1. Tạo K cây, mỗi cây được sinh ra như sau:
   (a) Xây dựng một tập con Di bằng cách lấy ngẫu nhiên (có trùng lặp) từ D.
   (b) Học cây thứ i từ Di như sau:
   (c) Tại mỗi nút trong quá trình xây dựng cây:
      i. chọn ngẫu nhiên một tập con các thuộc tính
      ii. phân nhánh cây dựa trên tập thuộc tính đó.

(d) Cây này sẽ được sinh ra với cỡ lớn nhất, không dùng cắt tỉa.

2. Mỗi phán đoán về sau thu được bằng cách lấy trung bình các phán đoán từ tất cả các cây.

Để xây dựng một tập con Di từ D, chúng ta lấy ngẫu nhiên n dữ liệu từ tập dữ liệu huấn luyện với kĩ thuật Bootstrapping, hay còn gọi là lấy mẫu ngẫu nhiên có lặp lại (random sampling with replacement), có nghĩa là, chúng ta lấy mẫu được 1 mẫu dữ liệu thì không bỏ dữ liệu đấy ra mà vẫn giữ lại trong tập dữ liệu ban đầu, rồi tiếp tục lấy mẫu cho tới khi có đủ n mẫu dữ liệu. Khi dùng kĩ thuật này thì tập n dữ liệu mới có thể có những mẫu dữ liệu bị trùng nhau.

Do quá trình xây dựng mỗi cây quyết định đều có yếu tố ngẫu nhiên (random) nên kết quả là các cây quyết định trong thuật toán Random Forests có thể khác nhau.

Thuật toán Random Forests sẽ bao gồm nhiều cây quyết định, mỗi cây được xây dựng dùng thuật toán Cây quyết định trên tập dữ liệu khác nhau và dùng tập thuộc tính khác nhau. Sau đó kết quả dự đoán của thuật toán Random Forests sẽ được tổng hợp từ các cây quyết định đã xây dựng được.

Khi dùng thuật toán Random Forests, chúng ta nên chú ý đến các siêu tham số như:

số lượng cây quyết định sẽ xây dựng, số lượng thuộc tính dùng để xây dựng cây...

c. Máy véctơ hỗ trợ (SVM)

Máy vectơ hỗ trợ (Support vector machine - SVM) được đề xuất bởi V. Vapnik và các đồng nghiệp của ông vào những năm 1970s ở Nga, và sau đó đã trở nên nổi tiếng và phổ biến vào những năm 1990s. SVM là một phương pháp phân lớp tuyến tính (linear classifier), với mục đích xác định một siêu phẳng (hyperplane) để phân tách hai lớp của dữ liệu. Ví dụ: lớp có nhãn dương (positive) và lớp có nhãn âm (negative). Sau đó, các hàm nhân (kernel functions), cũng được gọi là các hàm biến đổi (transformation functions), được dùng cho các trường hợp phân lớp phi tuyến. SVM có nền tảng lý thuyết chặt chẽ, được xây dựng dựa trên lý thuyết toán học và thống kê.

Về mặt thực nghiệm, SVM thường được coi là một phương pháp tốt đối với những bài toán phân lớp có không gian rất nhiều chiều. Điều này có lợi ích đặc biệt khi dữ liệu đặc trưng của các đối tượng là một tập rất lớn các thuộc tính. Đặc biệt, SVM đã được biết đến là một trong số các phương pháp phân lớp tốt nhất đối với các bài toán phân lớp văn bản (text classification). Trong phân loại văn bản, dữ liệu thường có số chiều lớn, với mỗi từ hoặc đặc trưng là một chiều. SVM hiệu quả trong việc xử lý không gian chiều cao này và có khả năng tạo ra siêu phẳng phân loại tốt giữa các danh mục văn bản khác nhau.

** Mô hình hóa*

Đầu tiên, xem xét bài toán phân loại nhị phân có tập huấn luyện gồm r điểm (x_i,y_i) với $i = \{1,...,r\}$ trong đó $x_i = (x_{i1},x_{i2},...,x_{in})^T$ là vector n chiều, $y_i \in \{-1,1\}$ là lớp của điểm x_i. Ở đây, chúng ta có hai lớp: lớp dương ($y_i = 1$) và lớp âm ($y_i = -1$).

Giả thiết rằng dữ liệu ban đầu của chúng ta là có thể phân tách tuyến tính, chúng ta muốn tìm một siêu phẳng có thể phân tách các lớp âm và lớp dương. Siêu phẳng có dạng:

$$f(\mathbf{x}) = \langle w,x\rangle + b \tag{23}$$

trong đó, w là vector trọng số, b là hệ số bias, $\langle w,x\rangle$ là tích vô hướng của x và w. Siêu phẳng phân tách 2 lớp dương và âm phải thỏa mãn ràng buộc:

$$y_i = \begin{cases} +1, & \text{if} \langle \mathbf{x}_i, \mathbf{w}\rangle + b \geq +1 \\ -1, & \text{if} \langle \mathbf{x}_i, \mathbf{w}\rangle + b \leq -1 \end{cases} \tag{24}$$

Siêu phẳng (H_0) phân tách lớp dương và lớp âm có dạng: $\langle w,x\rangle + b = 0$. H_0 còn được gọi là ranh giới quyết định (decision boundary). Thực tế, khi dữ liệu học có thể phân tách tuyến tính, có vô số các siêu phẳng phân tách (Hình 36). Câu hỏi đặt ra là siêu phẳng phân tách nào là tốt nhất? SVM lựa chọn mặt siêu phẳng phân tách có lề (margin) lớn nhất. Lý thuyết học máy đã chỉ ra rằng *một mặt siêu phẳng phân tách như thế sẽ tối thiểu hóa giới hạn lỗi (phân lớp) mắc phải (so với mọi siêu phẳng khác).*

Chúng ta sẽ xem xét chi tiết khái niệm cực đại hóa mức lề. Giả sử rằng tập dữ liệu huấn luyện có thể phân tách được một cách tuyến tính. Xét một quan sát của lớp dương $(x^+,1)$ và một quan sát của lớp âm $(x^-,-1)$ gần nhất đối với siêu

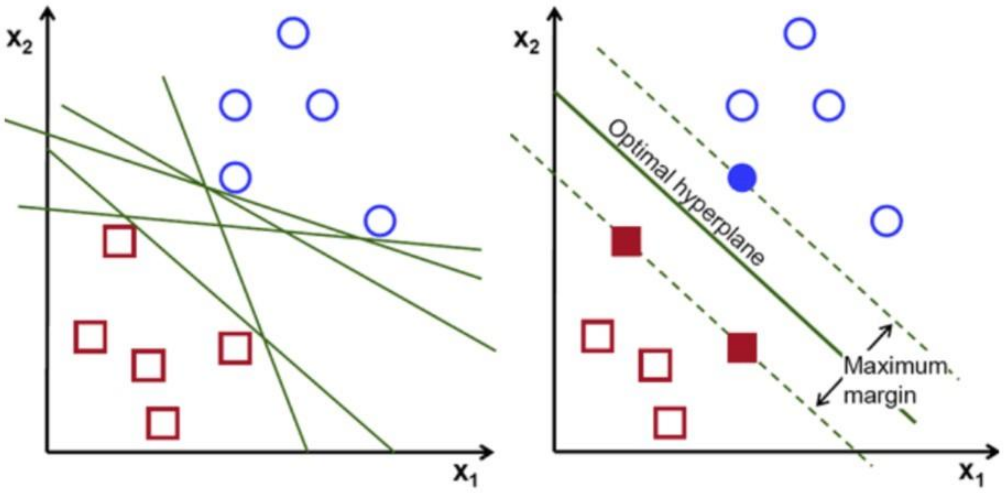
phẳng phân tách (H_0)($\langle w,x\rangle+b = 0$). Định nghĩa 2 siêu phẳng lề song song với nhau:

- H_+ đi qua x^+, và song song với H_0
- H_- đi qua x^-, và song song với H_0

Phương trình mô tả cho H_+ và H_- được viết như sau:

$H_+ : \langle w,x\rangle+b = 1$

$H_- : \langle w,x\rangle+b = -1$



Hình 1.5. Hình minh họa các siêu phẳng phân tách và siêu phẳng phân tách tối ưu khi cựu đại hóa mức lề.

Mức lề (margin) là khoảng cách giữa 2 siêu phẳng lề H_+ và H_-. Chúng ta có một số khái niệm:

- d_+ là khoảng cách giữa H_+ và H_0
- d_- là khoảng cách giữa H_- và H_0
- ($d_+ + d_-$) là mức lề

Trong không gian vectơ, **khoảng cách** từ một điểm x_i đến siêu phẳng $\langle w\cdot x\rangle+b = 0$ là:

$$\frac{|\langle \mathbf{w}\cdot \mathbf{x}_i\rangle + b|}{||\mathbf{w}||}$$

trong đó, $||\mathbf{w}||$ là độ dài của vector $\mathbf{w}$:

$$||\mathbf{w}|| = \sqrt{\langle \mathbf{w} \cdot \mathbf{w} \rangle} = \sqrt{\sum_{i=1}^{n} \mathbf{w}_i^2}$$

Áp dụng công thức trên, chúng ta có thể tính được d_+: khoảng cách từ x^+ đến ($\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$):

$$d^+ = \frac{|\langle \mathbf{w} \cdot \mathbf{x}^+ \rangle + b|}{||\mathbf{w}||} = \frac{|1|}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||}$$

Tương tự, chúng ta tính toán d_-: khoảng cách từ x^- đến ($\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$):

$$d^- = \frac{|\langle \mathbf{w} \cdot \mathbf{x}^- \rangle + b|}{||w||} = \frac{|-1|}{||w||} = \frac{1}{||w||}$$

Từ đó, mức lề có thể tính được như sau:

$$margin = d_+ + d_- = \frac{2}{||\boldsymbol{w}||}$$

Học SVM tương đương với giải quyết **bài toán cực tiểu hóa có ràng buộc**.

Chúng ta phải cực tiểu hóa: $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$ với điều kiện:

$$\begin{cases} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1, \text{ nếu } y_i = 1 \\ \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1, \text{ nếu } y_i = -1 \end{cases}$$

Bài toán này tương đương với cực tiểu hóa: $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$ với điều kiện: $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1$

($\forall i = 1..r$).

**Tối ưu hóa có ràng buộc*

Trong phần này, chúng ta xem xét lại vấn đề tối ưu hóa có ràng buộc và áp dụng cho bài toán học mức lề cực đại trong SVM.

Bài toán cực tiểu hóa có ràng buộc đẳng thức: Cực tiểu hóa $f(x)$, với điều kiện $g(x)=0$.

Điều kiện cần để x_0 là một lời giải:

$$\begin{cases} \frac{\partial}{\partial x}(f(x)+\alpha g(x))\Big|_{x=x_0} = 0 \\ g(x) = 0 \end{cases}$$

với α là một hệ số nhân (multiplier) Lagrange. Trong trường hợp có nhiều ràng buộc

đẳng thức $g_i(x)=0$ ($i=1..r$), cần một hệ số nhân Lagrange cho mỗi ràng buộc:

$$\begin{cases} \frac{\partial}{\partial x}(f(x)+\sum_{i=0}^{r}\alpha_i g_i(x))\Big|_{x=x_0} = 0 \\ g_i(x) = 0 \end{cases}$$

Trong khi đó, bài toán cực tiểu hóa có các ràng buộc bất đẳng thức: Cực tiểu hóa $f(x)$, với các điều kiện $g_i(x) \leq 0$.

Điều kiện cần để x_0 là một lời giải:

$$\begin{cases} \frac{\partial}{\partial x}(f(x)+\sum_{i=0}^{r}a_i g_i(x))\Big|_{x=x_0} = 0 \quad \text{với } \alpha_i \geq 0 \\ g_i(x) \leq 0 \end{cases}$$

Hàm: $L = f(x)+\sum_{i=1}^{r}\alpha_i g_i(x)$ được gọi là hàm Lagrange.

Học SVM được quy về giải bài toán cực tiểu hóa có rằng buộc bất đẳng thức. Biểu thức Lagrange

$$L_p(w,b,\alpha) = \frac{\langle \mathbf{w}\cdot\mathbf{w}\rangle}{2} - \sum_{i=1}^{r}\alpha_i[y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) - 1]$$

trong đó $\alpha_i \geq 0$ là các hệ số nhân Lagrange.

Lý thuyết tối ưu chỉ ra rằng một lời giải tối ưu cho (30) phải thỏa mãn các điều kiện nhất định, được gọi là **các điều kiện Karush-Kuhn-Tucker** (là các điều kiện cần, nhưng không phải là các điều kiện đủ). Các điều kiện Karush-Kuhn-Tucker đóng vai trò trung tâm trong cả lý thuyết và ứng dụng của lĩnh vực tối ưu có ràng buộc. Tập điều kiện Karush-Kuhn-Tucker cho bài toán tối ưu (30):

$$\frac{\partial L_p}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{r} \alpha_i y_i \mathbf{x}_i = 0$$

$$\frac{\partial L_p}{\partial b} = -\sum_{i=1}^{r} \alpha_i y_i = 0$$

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 \geq 0, \forall \mathbf{x}_i \quad (i = 1..r)$$

$$\alpha_i \geq 0, \quad i = 1..r$$

$$\alpha_i [y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] = 0, \quad i = 1..r$$

Trong đó, (33) chính là tập các ràng buộc ban đầu. Điều kiện bổ sung (35) chỉ ra rằng chỉ những ví dụ (điểm dữ liệu) thuộc các mặt siêu phẳng lề (H_+ và H_-) mới có $\alpha_i \geq 0$ bởi vì với những ví đụ đó thì $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 = 0$. Những ví dụ này được gọi là các vectơ hỗ trợ. Trong đó, đối với các ví dụ khác thì $\alpha_i = 0$.

Trong trường hợp tổng quát, các điều kiện Karush-Kuhn-Tucker là *cần* đối với một lời giải tối ưu, nhưng *chưa đủ*. Tuy nhiên đối với SVM, bài toán cực tiểu hóa có hàm mục tiêu lồi (convex) và các ràng buộc tuyến tính, thì các điều kiện Karush-KuhnTucker là cần và đủ đối với một lời giải tối ưu. Nhưng giải quyết bài toán tối ưu này vẫn là một nhiệm vụ khó khăn, do sự tồn tại của các ràng buộc bất đẳng thức. Phương pháp Lagrange giải quyết bài toán tối ưu hàm lồi dẫn đến một bài toán **đối ngẫu (dual)** của bài toán tối ưu. Bài toán đối ngẫu dễ giải quyết hơn so với bài toán tối ưu **ban đầu (primal)**.

Để thu được biểu thức đối ngẫu từ biểu thức ban đầu trong SVM, chúng ta gán giá trị bằng 0 đối với các đạo hàm bộ phận của biểu thức Lagrange trong (30) đối với các biến ban đầu w và b. Sau đó, áp dụng các quan hệ thu được đối với biểu thức Lagrange. Nghĩa là áp dụng các biểu thức (31) và (32) vào biểu thức Lagrange ban đầu (30) để loại bỏ các biến ban đầu w và b, chúng ta sẽ thu được biểu thức đối ngẫu L_D:

$$L_d(\alpha) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

Cả hai biểu thức L_P và L_D đều là các biểu thức Lagrange. Chúng đều dựa trên cùng một hàm một tiêu – nhưng với các ràng buộc khác nhau. Lời giải tìm được, bằng cách cực tiểu hóa L_P hoặc cực đại hóa L_D.

Bài toán tối ưu đối ngẫu cực đại hoá: $L_d(\alpha) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$ với điều kiện:

$$\begin{cases} \sum_{i=1}^{r} \alpha_i y_i = 0 \\ \alpha_i \geq 0, \forall i = 1..r \end{cases}$$

Đối với hàm mục tiêu là hàm lồi và các ràng buộc tuyến tính, giá trị cực đại của L_D xảy ra tại cùng các giá trị của w, b và α_i giúp đạt được giá trị cực tiểu của L_P. Giải bài toán trên, ta thu được các hệ số nhân Lagrange α_i (các hệ số α_i này sẽ được dùng để tính w và b). Giải bài toán trên cần đến các phương pháp lặp (để giải quyết bài toán tối ưu hàm lồi bậc hai có các ràng buộc tuyến tính). Chi tiết các phương pháp này nằm ngoài phạm vi của bài giảng.

Sau khi giải bài toán tối ưu đối ngẫu thu được α, chúng ta có thể tính được các giá trị w^* và b^*. Gọi SV (support vectors) là tập các vectơ hỗ trợ. SV là tập con của tập r các ví dụ huấn luyện ban đầu và thỏa mãn $\alpha_i > 0$ với các vector hỗ trợ $\mathbf{x}_i$. Đồng thời, $\alpha_i = 0$ với các vector không phải vector hỗ trợ $\mathbf{x}_i$. Sử dụng biểu thức (31), ta có thể tính được giá trị w^*

$$w^* = \sum_{i=1}^{r} \alpha_i y_i \mathbf{x}_i = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \mathbf{x}_i$$

vì $\forall \mathbf{x}_i \notin SV$: $\alpha_i = 0$. Mặt khác, sử dụng biểu thức (35) và (bất kỳ) một vectơ hỗ trợ x_k, ta có: $\alpha_k[y_k(\langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle + b^*) - 1] = 0$. Chú ý rằng $\alpha_k > 0$ với mọi vector hỗ trợ x_k. Vì

1

vậy: $y_k(\langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle + b^*) - 1 = 0$. Từ đây, ta được: $b^* = \_\_ - \langle \mathbf{w}^* \cdot \mathbf{x}_k \rangle$. Từ đó, chúng ta thu

y_k

được ranh giới quyết định phân lớp:

$$f(\mathbf{x}) = \langle \mathbf{w}^* \cdot \mathbf{x} \rangle + b^* = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b* = 0$$

Khi làm việc với một ví dụ cần phân lớp z, chúng ta cần tính giá trị:

$$sign(\langle \mathbf{w}^* \cdot \mathbf{z} \rangle + b^*) = sign(\sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{z} \rangle + b*)$$

Nếu biểu thức (3.3.2) trả về giá trị 1, thì ví dụ z được phân vào lớp có nhãn dương (positive); ngược lại, được phân vào lớp có nhãn âm (negative). Việc phân lớp này chỉ phụ thuộc vào các vectơ hỗ trợ và chỉ cần giá trị tích vô hướng (tích trong) của 2 vectơ (chứ không cần biết giá trị của 2 vectơ đấy).

** Multi-class SVM*

Multi-class SVM là một dạng SVM được sử dụng cho bài toán phân loại đa lớp (multi-class classification). SVM gốc chỉ xử lý được bài toán phân loại nhị phân (binary classification). Để xử lý các tập dữ liệu có nhiều hơn hai lớp, Multi-class SVM sử dụng các phương pháp như One-vs-All (OvA) hoặc One-vs-One (OvO). Trong phương pháp OvA, mỗi lớp được so sánh với tất cả các lớp khác để tạo ra một bài toán phân loại nhị phân riêng biệt. Trong phương pháp OvO, một bài toán phân loại nhị phân được tạo ra cho mỗi cặp lớp khác nhau. Sau đó, kết quả của các bài toán nhị phân được kết hợp để đưa ra dự đoán cuối cùng cho bài toán phân loại đa lớp.

- Một-đối-nhiều (One-vs-Rest hoặc One-vs-All, OvR/OvA):

• Trong phương pháp này, đối với mỗi lớp trong tập dữ liệu, một SVM nhị phân được huấn luyện để phân biệt giữa lớp đó và tất cả các lớp còn lại. Nếu có *k*k lớp trong dữ liệu, sẽ cần *k*k bộ phân loại SVM nhị phân.

• Khi dự đoán, mỗi bộ phân loại sẽ đưa ra một điểm số hoặc một giá trị dự đoán. Lớp có điểm số cao nhất hoặc xác suất cao nhất sẽ được chọn làm kết quả cuối cùng.

• Phương pháp này dễ triển khai và thường cho kết quả tốt đối với các bài toán có số lớp không quá lớn.

- Một-đối-một (One-vs-One, OvO):

• Trong phương pháp này, một SVM nhị phân được huấn luyện cho mỗi cặp lớp có thể có trong tập dữ liệu. Với *k*k lớp, số lượng bộ phân loại SVM cần thiết là *k*(*k*−1)22k(k−1).

• Khi dự đoán, mỗi bộ phân loại sẽ thực hiện phân loại giữa hai lớp, và lớp nào nhận được nhiều "phiếu bầu" nhất sẽ được chọn làm kết quả cuối cùng.

• Phương pháp này có thể yêu cầu nhiều bộ phân loại hơn nhưng có xu hướng chính xác hơn, đặc biệt khi các lớp có nhiều sự chồng chéo.

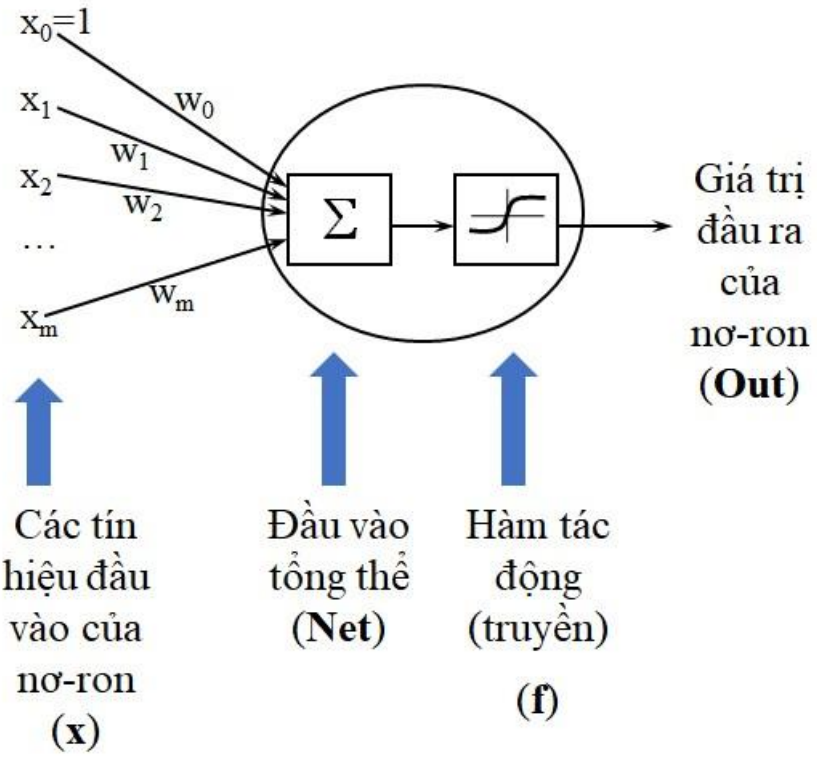
Ví dụ, nếu chúng ta có 3 lớp (A, B, C) trong bài toán phân loại đa lớp, phương pháp OvA sẽ tạo ra 3 bài toán nhị phân riêng biệt: A vs. (B, C), B vs. (A, C), C vs. (A, B). Mỗi bài toán nhị phân sẽ xác định xem một điểm dữ liệu thuộc lớp nào. Kết quả của các bài toán nhị phân sau đó được kết hợp để xác định lớp cuối cùng cho một điểm dữ liệu.

Phương pháp OvO sẽ tạo ra một số lượng bài toán nhị phân bằng số lớp nhân với số lớp trừ đi 1, tức là ở ví dụ trên sẽ tạo ra 3 bài toán nhị phân: A vs. B, A vs. C và B vs. C. Mỗi bài toán nhị phân sẽ quyết định lớp của một điểm dữ liệu. Lớp cuối cùng cho một điểm dữ liệu sẽ được xác định bằng cách đếm số lượng phiếu bầu cho mỗi lớp từ các bài toán nhị phân và chọn lớp có số phiếu bầu cao nhất.

c. Mạng nơron nhân tạo

Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) là một mô hình tính toán lấy cảm hứng từ cách hệ thống nơ-ron trong não người hoạt động. Mục tiêu của mạng nơ-ron nhân tạo là mô phỏng khả năng học và xử lý thông tin giống như con người. Một mạng nơ-ron nhân tạo bao gồm các đơn vị xử lý cơ bản được gọi là nơ-ron, và các kết nối giữa chúng được trọng số hóa. Mỗi nơ-ron nhận đầu vào, xử lý thông tin bằng cách áp dụng một hàm kích thích, và sau đó tạo ra đầu ra như Hình 1.6. Quá trình này được lặp lại qua nhiều lớp nơ-ron để tạo ra một mô hình có khả năng học từ dữ liệu.

ANN có thể được xem như một cấu trúc xử lý thông tin một cách phân tán và song song ở mức cao. ANN có khả năng học (learn), nhớ lại (recall), và khái quát hóa (generalize) từ các dữ liệu học. Khả năng của một ANN phụ thuộc vào: Kiến trúc (topology) của mạng nơ-ron, đặc tính đầu vào/ra của mỗi nơ-ron, thuật toán học (huấn luyện) và dữ liệu học.



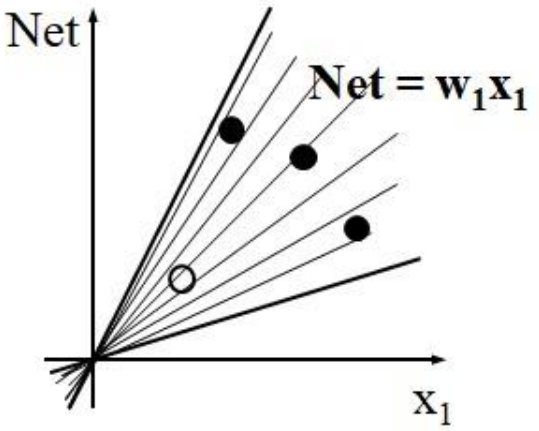
Hình 1.6. Kiến trúc của một nơ-ron

** Kiến trúc của một nơ-ron*

Cấu trúc và hoạt động của một nơ-ron bao gồm những thành phần như sau (Hình 43):

- Các tín hiệu đầu vào (input signals) của nơ-ron ($x_i, i = 1..m$)
- Mỗi tín hiệu đầu vào xi gắn với một trọng số w_i
- Trọng số điều chỉnh (bias) w_0 (với $x_0 = 1$)
- Đầu vào tổng thể (Net input) là một hàm tích hợp của các tín hiệu đầu vào –

Net(w,x)



Hình 1.7. Hàm Net không thể phân tách được hai lớp khi không sử dụng bias.

- Hàm tác động/truyền (Activation/transfer function) tính giá trị đầu ra của nơ-ron – $f(Net(w,x))$
- Giá trị đầu ra (Output) của nơ-ron: $Out = f(Net(w,x))$

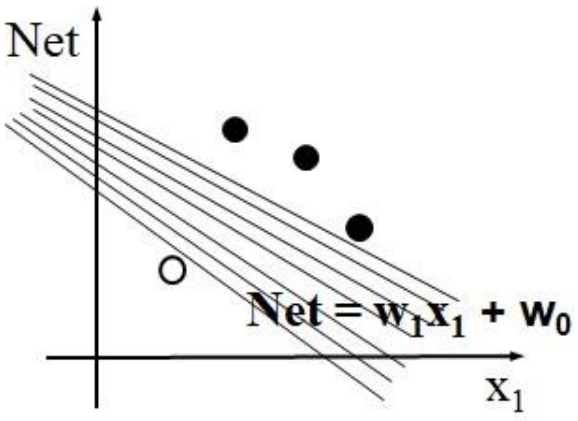
Đầu vào tổng thể (net input) thường được tính toán bởi một hàm tuyến tính

$$\begin{aligned} Net &= w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n \\ &= w_0 \cdot 1 + \sum_{i=1}^{m} w_ix_i \\ &= \sum_{i=0}^{m} w_ix_i \end{aligned}$$

Trọng số điều chỉnh (bias) w_0 giúp mở rộng hàm tuyến tính đầu vào tổng thể (net input) theo dữ liệu đầu vào thay vì chỉ là họ hàm tuyến tính đi qua gốc tọa độ. Ví dụ, chúng ta xét dữ liệu trên không gian một chiều như Hình 1.7, Hình 1.8. Họ các hàm Net=w_1x_1 không thể phân tách được các ví dụ thành 2 lớp (two classes) 44 nhưng họ các hàm Net=$w_1x_1+w_0$ có thể phân tách được 45.

Hàm kích thích (activation function), còn được gọi là hàm truyền (transfer function), là một phần quan trọng trong mỗi nơ-ron nhân tạo (ANN). Nhiệm vụ của hàm kích thích là định rõ cách mà đầu vào được chuyển đổi thành đầu ra của

nơ-ron trong mạng. Các hàm kích thích khác nhau có ảnh hưởng đến khả năng học tập và hiệu suất của mô hình. Lựa chọn hàm kích thích phụ thuộc vào đặc điểm của vấn đề cụ thể và cách mà



Hình 1.8. Hàm Net không thể phân tách được hai

Dưới đây là một số hàm kích thích phổ biến được sử dụng trong mạng nơ-ron nhân tạo:

Giới hạn cứng: Còn được gọi là hàm ngưỡng (threshold function). Giá trị đầu ra lấy một trong 2 giá trị 1 hoặc 0 phụ thuộc theo giá trị ngưỡng θ như sau:

$$Out(Net) = HL(Net, \theta)$$
$$= \begin{cases} 1 & , \text{if } Net > \theta \\ 0 & , \text{otherwise} \end{cases}$$

Đầu ra của hàm có thể thay đổi khi nhận giá trị 1 hoặc −1 theo hàm dấu như sau:

$$Out(Net) = HL2(Net,\theta)$$
$$= sign(Net,\theta)$$

Tuy hàm giới hạn cứng đơn giản, dễ tính toán, nhưng chúng không liên tục, không có đạo hàm tại ngưỡng.

Hàm tác động logic ngưỡng: Còn được gọi là hàm tuyến tính bão hòa (saturating linear function). Hàm logic ngưỡng kết hợp của 2 hàm tác động: tuyến tính và giới hạn chặt. Công thức chi tiết của hàm logic ngưỡng như sau:

$$Out(Net) = tl(Net, \alpha, \theta)$$

$$= \begin{cases} 0 & \text{if} Net < -\theta \\ \alpha(Net + \theta) & \text{, if} -\theta \le Net \le \frac{1}{\alpha} - \theta \\ 1 & \text{, if} Net > \frac{1}{\alpha} - \theta \end{cases}$$

$$= \max(0, \min(1, \alpha(Net + \theta)))$$

trong đó, tham số θ xác định độ dốc của khoảng tuyến tính. Hàm logic ngưỡng Liên tục, nhưng không có đạo hàm.

Hàm tác động Sigmoid: Hàm sigmoid giúp chuyển đổi đầu vào thành giá trị nằm trong khoảng (0,1). Nó thường được sử dụng trong các lớp đầu ra của mô hình để thực hiện phân loại nhị phân. Công thức chi tiết của hàm Sigmoid như sau:

$$Out(Net) = sf(Net, \alpha, \theta)$$

$$= \frac{1}{1 + e^{-\alpha(Net + \theta)}}$$

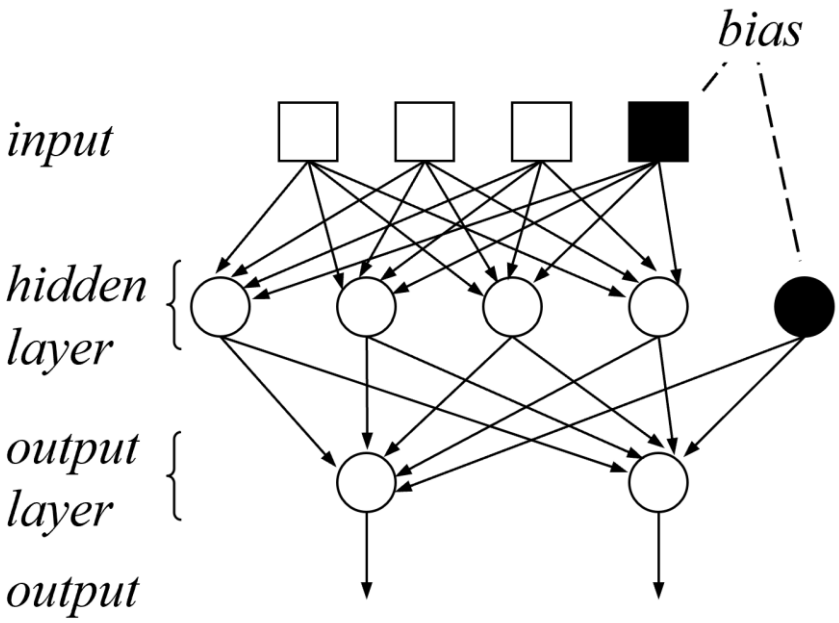
tham số θ xác định độ dốc. Giá trị đầu ra trong khoảng (0,1). Hàm sigmoid liên tục và đạo hàm liên tục. Đạo hàm của một hàm sigmoid được biểu diễn bằng một hàm của chính nó. Bởi vì có nhiều khoảng đầu vào làm cho đầu ra của hàm sigmoid gần 0 hoặc 1, nên đạo hàm của hàm sigmoid dễ thu được giá trị nhỏ, gần 0 (vanishing gradient). Vì vậy, sử dụng hàm sigmoid có thể dẫn đến hiện tượng mạng không được cập nhật tham số trong quá trình học.

Hàm tác động Hyperbolic tangent: Tương tự như sigmoid, hàm tanh giúp chuyển đổi đầu vào thành giá trị nằm trong khoảng (−1,1). Hàm này thường được sử dụng trong các lớp ẩn của mô hình. Công thức chi tiết của hàm như sau:

Out

$$(\text{Net}) = \tanh(\text{Net}, \alpha, \theta) = \frac{1 - e^{-\alpha(\text{Net}+\theta)}}{1 + e^{-\alpha(\text{Net}+\theta)}} = \frac{2}{1 + e^{-\alpha(\text{Net}+\theta)}} - 1$$

Tham số α xác định độ dốc. Tính chất của hàm này tương tự như hàm sigmoid: Liên tục, có đạo hàm liên tục và đạo hàm của một hàm tanh có thể được biểu diễn bằng một hàm của chính nó.



Hình 1.9. Kiến trúc của mạng nơ-ron. Một ANN với một tầng ẩn gồm 3 tín hiệu vào, 2 giá trị đầu ra. Tổng cộng, có 6 neurons (4 ở tầng ẩn và 2 ở tầng đầu ra).

Hàm ReLU (Rectified Linear Unit):

$$f(x) = max(0,x)$$

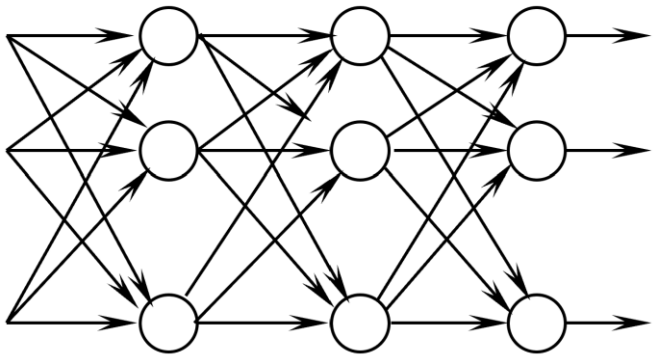
ReLU giữ nguyên giá trị dương của đầu vào và đưa về 0 nếu giá trị âm. Nó đã trở thành một lựa chọn phổ biến cho các lớp ẩn, vì nó giúp giảm vấn đề biến mất đạo hàm và có tính chất kích thích sự học tập của mạng.

** Kiến trúc mạng ANN*

Kiến trúc của một ANN Hình 1.11 được xác định bởi các thành phần sau:

- Số lượng các tín hiệu đầu vào và đầu ra
- Số lượng các tầng
- Số lượng các nơ-ron trong mỗi tầng
- Số lượng các liên kết đối với mỗi nơ-ron
- Cách thức các nơ-ron (trong một tầng, hoặc giữa các tầng) liên kết với nhau

Một ANN phải có: Một tầng đầu vào (input layer), một tầng đầu ra (output layer) và có thể có hoặc không nhiều tầng ẩn (hidden layer(s)). Trong đó, một tầng (layer)



Hình 1.10. Kiến trúc của mạng nơ-ron đầy đủ

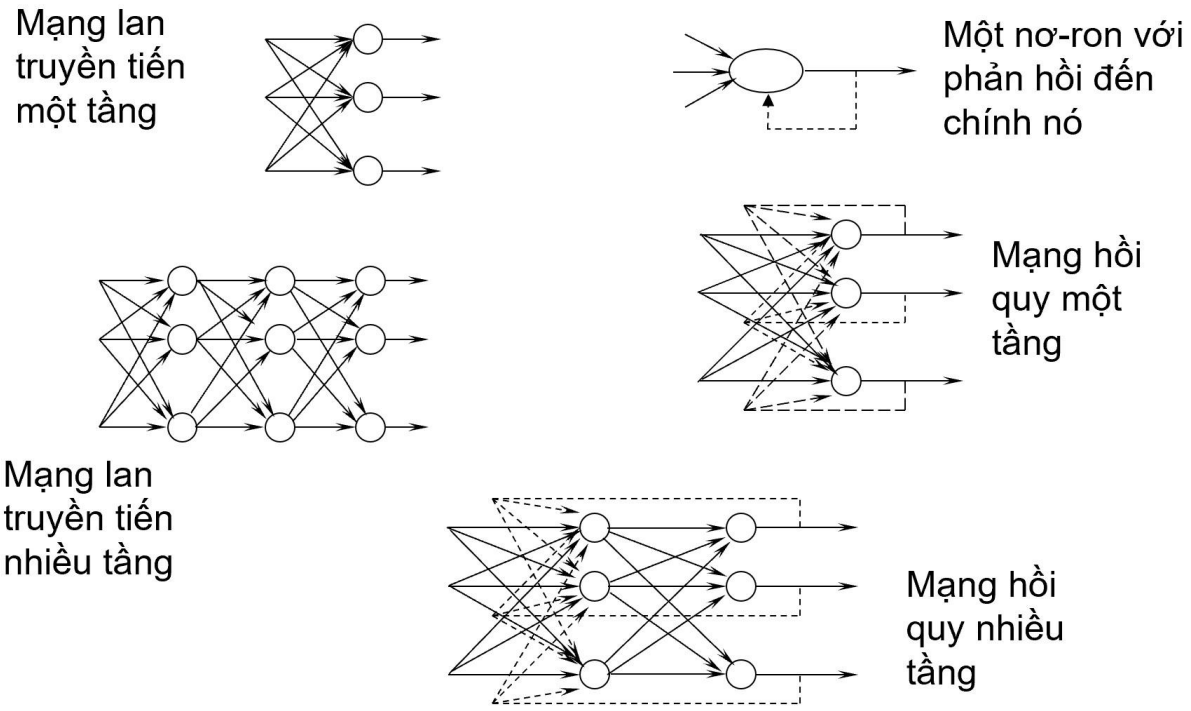
chứa một nhóm các nơ-ron. Tầng ẩn (hidden layer) là một tầng nằm ở giữa tầng đầu vào (input layer) và tầng đầu ra (output layer). Các nút ở tầng ẩn (hidden nodes) không tương tác trực tiếp với môi trường bên ngoài (của mạng nơ-ron). Hình 1.9 minh họa kiến trúc mạng gồm một tầng ẩn.

Một số khái niệm kiến trúc mạng phổ biến được trình bày như sau:

- Một ANN được gọi là liên kết đầy đủ (fully connected) (Hình 1.10) nếu mọi đầu ra từ một tầng liên kết với mọi nơ-ron của tầng kế tiếp.
- Một ANN được gọi là mạng lan truyền tiến (feed-forward network) nếu không có bất kỳ đầu ra của một nút là đầu vào của một nút khác thuộc cùng tầng (hoặc thuộc một tầng phía trước).
- Khi các đầu ra của một nút liên kết ngược lại làm các đầu vào của một nút thuộc cùng tầng (hoặc thuộc một tầng phía trước), thì đó là một mạng phản hồi (feedback network)
  - Nếu phản hồi là liên kết đầu vào đối với các nút thuộc cùng tầng, thì đó là phản hồi bên (lateral feedback).
  - Các mạng phản hồi có các vòng lặp kín (closed loops) được gọi là các mạng hồi quy (recurrent networks).

*Mạng perceptron

Mạng perceptron là một mô hình nơ-ron nhân tạo đơn giản nhất được đề xuất bởi Frank Rosenblatt vào những năm 1950. Nó là một dạng cơ bản của mạng nơ-ron chỉ



Hình 1.11. Một số kiến trúc của mạng nơ-ron

gồm duy nhất một nơ-ron. Mô hình này được sử dụng chủ yếu để thực hiện tác vụ phân loại nhị phân. Biểu diễn của mạng perceptron

$$Out = sign(Net(w,x)) = sign(\sum_{j=0}^{m} w_j x_j)$$

Đối với một ví dụ x, giá trị đầu ra của perceptron là 1, nếu $Net(w,x) > 0$ và −1 trong trường hợp còn lại.

Với một tập các ví dụ học $D = (x,d)$ với x là đầu vào, d là giá trị đầu ra mong muốn (−1 hoặc 1), quá trình học của perceptron nhằm xác định một vectơ trọng số cho phép perceptron sinh ra giá trị đầu ra chính xác (−1 hoặc 1) cho mỗi ví dụ học. Với một ví dụ học x được perceptron phân lớp chính xác, thì vectơ trọng số w không thay đổi. Trong trường hợp khác thì:

- Nếu $d = 1$ nhưng perceptron lại sinh ra −1 ($Out = -1$), thì w cần được thay đổi sao cho giá trị $Net(w,x)$ tăng lên.
- Nếu $d = -1$ nhưng perceptron lại sinh ra 1 ($Out = 1$), thì w cần được thay đổi sao cho giá trị $Net(w,x)$ giảm đi.

Algorithm 1 Perceptron_batch(D, η)

1: Initialize w ($w_i \leftarrow$ an initial (small) random value)

2: **repeat**

3:$\Delta w \leftarrow 0$

4:**for** each training instance $(x,d) \in D$ **do**

5:Compute the real output value *Out*

6:**if** ($Out \neq d$) **then**

7: $\Delta w \leftarrow \Delta w + \eta(d - Out)x$

8: **end if**

9:**end for**

10:$w \leftarrow w + \Delta w$

11: **until** all the training instances in D are correctly classified

12: **return** w

Thuật toán Alg.1 mô tả quá trình học của mạng perceptron. Giải thuật học cho perceptron được chứng minh là hội tụ (converge) nếu các ví dụ học là có thể phân tách tuyến tính (linearly separable) và sử dụng một tốc độ học η đủ nhỏ. Tuy nhiên, giải thuật học perceptron có thể không hội tụ nếu như các ví dụ học không thể phân tách tuyến tính (not linearly separable). Trong khi một perceptron chỉ có thể biểu diễn một hàm phân tách tuyến tính (linear separation function), một mạng nơ-ron nhiều tầng (multi-layer NN) được học bởi giải thuật lan truyền ngược (Back Propagation - BP) có thể biểu diễn một hàm phân tách phi tuyến phức tạp (highly non-linear separation function).

** Huấn luyện mạng nơ-ron tổng quát*

Có hai kiểu học phổ biến trong các mạng nơ-ron nhân tạo: Học tham số (Parameter learning) và học cấu trúc (Structure learning). Mục tiêu của học tham số (Parameter learning) là thay đổi thích nghi các trọng số (weights) của các liên kết trong mạng nơron. Trong khi đó, mục tiêu của học cấu trúc là thay đổi thích nghi cấu trúc mạng, bao gồm số lượng các nơ-ron và các kiểu liên kết giữa chúng. Hai kiểu học này có thể được thực hiện đồng thời hoặc riêng rẽ. Trong nội dung mô học này, chúng ta sẽ chỉ xét việc học tham số.

Huấn luyện một mạng nơron (khi cố định kiến trúc) chính là việc học các trọng số w của mạng từ tập học D. Thông thường, chúng ta đưa việc học tham số mạng về bài toán cực tiểu hoá một hàm lỗi thực nghiệm:

$$L(\mathbf{w}) = \frac{1}{|\mathbf{D}|} \sum_{x \in \mathbf{D}} loss(d_x, out(x))$$

trong đó *out*(*x*) là đầu ra của mạng, với đầu vào *x* có nhãn tương ứng là d_x; *loss* là một hàm đo lỗi phán đoán. Với hồi quy hàm lỗi bình phương thường được sử dụng. Với phân loại, hàm cross-entropy thường được sử dụng để đo sự sai khác của kết quả đầu ra của mạng và giá trị mong muốn thực tế. Để cập nhật tham số, nhiều phương pháp lặp dựa trên Gradient: Backpropagation, SGD, Adam, AdaGrad, etc.

Xét một ANN có *n* nơ-ron đầu ra, chúng ta sử dụng lỗi bình phương cho bài toán hồi quy. Đối với một ví dụ học (*x*,*d*), giá trị lỗi học (training error) gây ra bởi vectơ trọng số (hiện tại) *w*:

$$E_x(w) = \frac{1}{2} \sum_{i=1}^{n} (d_i - out_i)^2$$

Hàm lỗi gây ra bởi vectơ trọng số (hiện tại) *w* đối với toàn bộ tập học *D*:

$$E_D(w) = \frac{1}{|D|} \sum_{x \in D} E_x(w)$$

Gradient của *E* (ký hiệu là ∇E) là một vectơ:

$$\nabla E(\mathbf{w}) = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_N} \right)$$

trong đó *N* là tổng số các trọng số (các liên kết) trong mạng. Gradient ∇E xác định hướng gây ra việc tăng nhanh nhất (steepest increase) đối với giá trị lỗi *E*. Trong khi đó, hướng gây ra việc giảm nhanh nhất (steepest decrease) là hướng ngược với gradient của *E*:

$$\delta W = -\eta \cdot \nabla E(w)$$

$$\Delta W_i = -\eta \frac{\partial E}{\partial w_i}, \forall i = 1..N$$

Giải thuật học lan truyền ngược BP (backpropagation) được sử dụng để học các trọng

---

Algorithm 2 Gradient_descent_incremental(D, η)

1: Initialize w ($w_i \leftarrow$ an initial (small) random value)
2: **repeat**
3:**for** each training instance $(x,d) \in D$ **do**
4:Compute the network output
5:**for** each weight component w_i **do**
6: $w_i \leftarrow w_i - \eta \left(\frac{\partial E_x}{\partial w_i}\right)$
7: **end for**
8:**end for**
9: **until** stopping criterion satisfied
10: **return** w

---

số của một mạng nơ-ron nhiều tầng với cấu trúc mạng cố định (các nơ-ron và các liên kết giữa chúng là cố định) và đối với mỗi nơ-ron, hàm tác động phải có đạo hàm liên tục (hoặc có chiến lược giải quyết tại một vài điểm không tồn tại đạo hàm). Giải thuật BP áp dụng chiến lược gradient descent (Alg.2) trong quy tắc cập nhật các trọng số để cực tiểu hóa lỗi (khác biệt) giữa các giá trị đầu ra thực tế và các giá trị đầu ra mong muốn, đối với các ví dụ học.

Giải thuật học lan truyền ngược tìm kiếm một vectơ các trọng số (weights vector) giúp cực tiểu hóa lỗi tổng thể của hệ thống đối với tập học. Giải thuật BP bao gồm 2 giai đoạn (bước):

- Giai đoạn lan truyền tiến tín hiệu (Signal forward). Các tín hiệu đầu vào (vectơ các giá trị đầu vào) được lan truyền tiến từ tầng đầu vào đến tầng đầu ra (đi qua các tầng ẩn).
- Giai đoạn lan truyền ngược lỗi (Error backward). Căn cứ vào giá trị đầu ra mong muốn của vectơ đầu vào, hệ thống tính toán giá trị lỗi. Bắt đầu

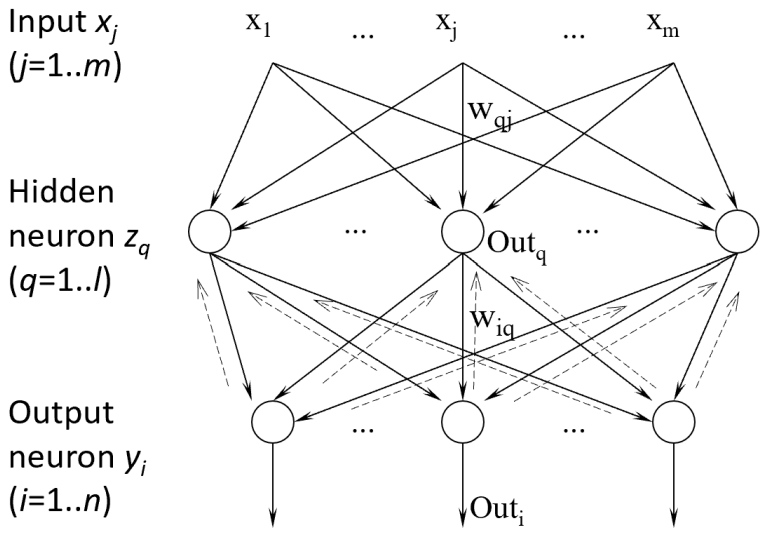
từ tầng đầu ra, giá trị lỗi được lan truyền ngược qua mạng, từ tầng này qua tầng khác (phía trước), cho đến tầng đầu vào. Việc lan truyền ngược lỗi (error back-propagation) được thực hiện thông qua việc tính toán (một cách truy hồi) giá trị gradient cục bộ của mỗi nơ-ron.

Chúng ta sử dụng mạng nơ-ron 3 tầng (Hình 1.12) để minh họa giải thuật học BP như sau:

- m tín hiệu đầu vào x_j (j = 1..m), l nơ-ron tầng ẩn z_q (q = 1..l), n nơ-ron đầu ra y_i

  (i = 1..n)

- w_{qj} là trọng số của liên kết từ tín hiệu đầu vào x_j tới nơ-ron tầng ẩn z_q



Hình 1.12. Quá trình lan truyền tiến (nét liền) và lan truyền ngược (nét đứt) trong học mạng ANN.

- w_{iq} là trọng số của liên kết từ nơ-ron tầng ẩn z_q tới nơ-ron đầu ra y_i
- Out_q là giá trị đầu ra (cục bộ) của nơ-ron tầng ẩn z_q
- Out_i là giá trị đầu ra của mạng tương ứng với nơ-ron đầu ra y_i

Bước lan truyền tiến: Đối với mỗi ví dụ học x, vectơ đầu vào x được lan truyền từ tầng đầu vào đến tầng đầu ra. Mạng sẽ sinh ra một giá trị đầu ra thực tế (actual output) *Out* (là một vectơ của các giá trị Out_i, i = 1..n). Nơ-ron z_q ở tầng ẩn sẽ nhận được giá trị đầu vào tổng thể (*netinput*) bằng:

$$Net_q = \sum_{j=1}^{m} w_{qj} x_j$$

và sinh ra một giá trị đầu ra (cục bộ) bằng:

$$Out_q = f\left(\sum_{j=1}^{m} w_{qj} x_j\right)$$

trong đó $f(.)$ là hàm tác động (activation function) của nơ-ron z_q. Tiếp theo, giá trị đầu vào tổng thể (net input) của nơ-ron y_i ở tầng đầu ra:

$$Net_i = \sum_{q=1}^{l} w_{iq} Out_q = \sum_{q=1}^{l} w_{iq} f\left(\sum_{j=1}^{m} w_{qj} x_j\right)$$

Nơ-ron y_i sinh ra giá trị đầu ra (là một giá trị đầu ra của mạng):

$$Out_i = f(Net_i) = f\left(\sum_{q=1}^{l} w_{iq} Out_q\right) = f\left(\sum_{q=1}^{l} w_{iq} f\left(\sum_{j=1}^{m} w_{qj} x_j\right)\right)$$

Vectơ các giá trị đầu ra Out_i (i=1..n) chính là giá trị đầu ra thực tế của mạng, đối với vectơ đầu vào x. Từ đó, đối với mỗi ví dụ học x, các tín hiệu lỗi (error signals) do sự khác biết giữa giá trị đầu ra mong muốn d và giá trị đầu ra thực tế Out được tính toán. Các tín hiệu lỗi này được lan truyền ngược (back-propagated) từ tầng đầu ra tới các tầng phía trước, để cập nhật các trọng số (weights).

Quá trình lan truyền ngược: Xét các tín hiệu lỗi và việc lan truyền ngược của chúng với một hàm lỗi sau:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(d_i - \mathrm{Out}_i)^2 = \frac{1}{2}\sum_{i=1}^{n}[d_i - f(\mathrm{Net}_i)]^2$$
$$= \frac{1}{2}\sum_{i=1}^{n}\left[d_i - f\left(\sum_{q=1}^{l} w_{iq}\mathrm{Out}_q\right)\right]^2$$

Theo phương pháp gradient-descent, các trọng số của các liên kết từ tầng ẩn tới tầng đầu ra được cập nhật bởi

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial w_{iq}}$$

Sử dụng quy tắc chuỗi đạo hàm đối với $\partial E / \partial w_{iq}$, ta có

$$\begin{aligned}\Delta w_{iq} &= -\eta \frac{\partial E}{\partial \mathrm{Out}_i} \frac{\partial \mathrm{Out}_i}{\partial \mathrm{Net}_i} \frac{\partial \mathrm{Net}_i}{\partial w_{iq}} \\ &= \eta (d_i - \mathrm{Out}_i) f'(\mathrm{Net}_i) \mathrm{Out}_q \\ &= \eta \delta_i \mathrm{Out}_q\end{aligned}$$

Chú ý rằng dấu "–" đã được kết hợp với giá trị $\partial E/\partial Out_i$. Gọi δ_i là tín hiệu lỗi (error signal) của nơ-ron y_i ở tầng đầu ra, chúng ta có:

$$\delta_i = -\frac{\partial E}{\partial \mathrm{Net}_i} = -\frac{\partial E}{\partial \mathrm{Out}_i} \frac{\partial \mathrm{Out}_i}{\partial \mathrm{Net}_i}$$

$$= (d_i - Out_i) f'(Net_i)$$

trong đó Net_i là đầu vào tổng thể (net input) của nơ-ron y_i ở tầng đầu ra, và $f'(Net_i) = \partial f(Net_i)/\partial Net_i$. Để cập nhật các trọng số của các liên kết từ tầng đầu vào tới tầng ẩn, chúng ta cũng áp dụng phương pháp gradient-descent và quy tắc chuỗi đạo hàm.

trong đó Net_q là đầu vào tổng thể (net input) của nơ-ron z_q ở tầng ẩn, và $f'(Net_q) = \partial f(Netq)/\partial Net_q$.

Theo các công thức tính các tín hiệu lỗi δi và δq đã nêu, thì tín hiệu lỗi của một nơ-ron ở tầng ẩn khác với tín hiệu lỗi của một nơ-ron ở tầng đầu ra. Do sự khác biệt này, thủ tục cập nhật trọng số trong giải thuật BP còn được gọi là quy tắc học delta tổng quát. Tín hiệu lỗi δq của nơ-ron z_q ở tầng ẩn được xác định bởi: Các tín hiệu lỗi δ_i của các nơ-ron y_i ở tầng đầu ra (mà nơ-ron z_q liên kết tới) và các hệ số chính là các trọng số

w_{iq}.

Quá trình tính toán tín hiệu lỗi (error signals) như trên có thể được mở rộng (khái quát) dễ dàng đối với mạng nơ-ron có nhiều hơn 1 tầng ẩn.

$$\Delta w_{qj} = -\eta \frac{\partial E}{\partial w_{qj}} = -\eta \frac{\partial E}{\partial \text{Out}_q} \frac{\partial \text{Out}_q}{\partial \text{Net}_q} \frac{\partial \text{Net}_q}{\partial w_{qj}}$$

Từ công thức tính hàm lỗi $E(w)$, ta thấy rằng mỗi thành phần lỗi $(d_i - y_i)$ $(i = 1..n)$ là một hàm của Out_q:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} \left[d_i - f\left(\sum_{q=1}^{l} w_{iq} \text{Out}_q \right) \right]^2$$

Áp dụng quy tắc chuỗi đạo hàm, ta có:

$$\begin{aligned} \Delta w_{qj} &= \eta \sum_{i=1}^{n} \left[(d_i - \text{Out}_i) f'(\text{Net}_i) w_{iq} \right] f'(\text{Net}_q) x_j \\ &= \eta \sum_{i=1}^{n} \left[\delta_i w_{iq} \right] f'(\text{Net}_q) x_j = \eta \delta_q x_j \end{aligned}$$

Gọi δq là tín hiệu lỗi (error signal) của nơ-ron z_q ở tầng ẩn, chúng ta có

$$\begin{aligned} \delta_q &= -\frac{\partial E}{\partial \text{Net}_q} = -\frac{\partial E}{\partial \text{Out}_q} \frac{\partial \text{Out}_q}{\partial \text{Net}_q} \\ &= f'(\text{Net}_q) \sum_{i=1}^{n} (\delta_i) w_{iq} \end{aligned}$$

Dạng tổng quát của quy tắc cập nhật trọng số trong giải thuật BP là: $\Delta W_{ab} = \eta \delta_a x_b$ với b và a là 2 chỉ số tương ứng với 2 đầu của liên kết (b→a) (từ một nơ-ron (hoặc tín hiệu đầu vào) b đến nơ-ron a), x_b là giá trị đầu ra của nơ-ron ở tầng ẩn (hoặc tín hiệu đầu vào) b và δ_a là tín hiệu lỗi của nơ-ron a.

1.8. Các ứng dụng của học máy

Ứng dụng: Học máy có ứng dụng rộng khắp trong các ngành khoa học/sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ.

Một số ứng dụng thường thấy Xử lý ngôn ngữ tự nhiên (Natural Language Processing): xử lý văn bản, giao tiếp người – máy, … Nhận dạng (Pattern Recognition): nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision) … Tìm kiếm (Search ENgine) Chẩn đoán trong y tế: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động. Tin sinh học: phân loại chuỗi gENe, quá trình hình thành gENe/protein Vật lý: phân tích ảnh thiên văn, tác động giữa các hạt …

Phát hiện gian lận tài chính (financial fraud): gian lận thẻ tín dụng Phân tích thị trường chứng khoán (stock market analysis) Chơi trò chơi: tự động chơi cờ, hành động của các nhân vật ảo Mũi điện tử, lưỡi điện tử. Rôbốt là tổng hợp của rất nhiều ngành khoa học, trong đó học máy tạo nên hệ thần kinh/bộ não của người máy.

1.9. Xây dựng mô hình dựa trên học máy

+ Một bài toán học máy cần trải qua 03 bước chính:

✓ Chọn/đề xuất mô hình: Chọn mô hình phù hợp cho tập dữ liệu.

✓ Tìm tham số: Các mô hình có các tham số tương ứng, nhiệm vụ lúc này là tìm các tham số này sao cho phù hợp với tập dữ liệu nhất có thể.

✓ Suy luận: Sau khi có được mô hình và tham số, ta có thể dựa vào chúng để đưa ra suy luận cho một đầu vào mới nào đó.

+ Quy trình xây dựng mô hình học máy như sau:

✓ Thu thập dữ liệu: Thu thập dữ liệu để mô hình học

✓ Chuẩn bị dữ liệu: Xử lý và đưa dữ liệu về định dạng tối ưu, trích chọn đặc trưng hoặc giảm chiều dữ liệu

✓ Huấn luyện: Tại bước này, thuật toán học máy thực hiện việc học thông qua các ví dụ đã được thu thập và chuẩn bị từ hai bước trên

✓ Đánh giá: Kiểm thử mô hình để đánh giá xem chất lượng của mô hình tốt đến đâu

✓ Tinh chỉnh: Tinh chỉnh mô hình để tối ưu hiệu quả

Bất cứ một bài toán học máy nào cũng đều cần có dữ liệu để huấn luyện, ta có thể coi nó là điều kiện tiên quyết. Dữ liệu sau khi có được cần phải:

✓ **Chuẩn hoá**: Tất cả các dữ liệu đầu vào đều cần được chuẩn hoá để máy tính có thể xử lý được. Quá trình chuẩn hoá bao gồm số hoá dữ liệu, co giãn thông số cho phù hợp với bài toán. Việc chuẩn hoá này ảnh hưởng trực tiếp tới tốc độ huấn luyện cũng như cả hiệu quả huấn luyện.

✓ **Phân chia**: Việc mô hình được chọn rất khớp với tập dữ liệu đang có không có nghĩa là giả thuyết của ta là đúng mà có thể xảy ra tình huống dữ liệu thật lại không khớp. Vấn đề này trong học máy được gọi là khớp quá (*Overfitting*). Vì vậy khi huấn luyện người ta phải phân chia dữ liệu ra thành 3 loại để có thể kiểm chứng được phần nào mức độ tổng quát của mô hình. Cụ thể 3 loại đó là:

✓ **Tập huấn luyện** (*Training set*): Thường chiếm 60%, dùng để học khi huấn luyện.

✓ **Tập kiểm chứng** (*Cross validation set*): Thường chiếm 20%, dùng để kiểm chứng mô hình khi huấn luyện.

✓ **Tập kiểm tra** (*Test set*): Thường chiếm 20%, dùng để kiểm tra xem mô hình đã phù hợp chưa sau khi huấn luyện.

Lưu ý rằng, tập kiểm tra ta phải lọc riêng ra và không được sử dụng trong khi huấn luyện. Còn tập huấn luyện và tập kiểm chứng thì nên xáo trộn đổi cho nhau để mô hình của ta được huấn luyện với các mẫu ngẫu nhiên nhất có thể.

CHƯƠNG 2. ỨNG DỤNG BÀI TOÁN HỌC MÁY TRONG GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ TRONG NHÀ.

2.1. Ứng dụng bài toán học máy trong giám sát chất lượng không khí trong nhà

2.1.1. Bài toán giám sát chất lượng không khí trong nhà

Trong bối cảnh đô thị hóa nhanh chóng và sự gia tăng đáng kể các nguồn phát thải từ hoạt động sinh hoạt, sản xuất và giao thông, vấn đề ô nhiễm không khí ngày càng trở nên nghiêm trọng, không chỉ ở môi trường ngoài trời mà còn trong không gian trong nhà. Chất lượng không khí trong nhà có ảnh hưởng trực tiếp đến sức khỏe, hiệu suất làm việc và chất lượng cuộc sống của con người, đặc biệt là đối với những người phải làm việc hoặc sinh hoạt trong không gian kín trong thời gian dài như văn phòng, trường học, bệnh viện hay hộ gia đình.

Theo các nghiên cứu của Tổ chức Y tế Thế giới, con người dành trung bình khoảng 90% thời gian trong nhà, trong khi nồng độ các chất ô nhiễm trong nhà đôi khi có thể cao gấp 2–5 lần so với ngoài trời. Các khí độc hại thường gặp như CO_2, CO, NO_2, SO_2, NH_3, H_2S, VOCs (hợp chất hữu cơ bay hơi)… có thể phát sinh từ nhiều nguồn như sơn tường, vật liệu xây dựng, khói thuốc lá, thiết bị điện tử, hoạt động nấu nướng hoặc con người. Do đó, việc giám sát liên tục và chính xác chất lượng không khí trong nhà trở thành một nhiệm vụ thiết yếu để bảo vệ sức khỏe cộng đồng.

Tuy nhiên, việc đo lường và phân tích chất lượng không khí bằng các thiết bị truyền thống (ví dụ như máy quang phổ khối, thiết bị sắc ký khí hoặc cảm biến chuyên dụng đơn khí) thường đắt đỏ, cồng kềnh và tiêu tốn năng lượng, khiến cho việc triển khai giám sát diện rộng trở nên khó khăn. Để khắc phục hạn chế này, các hệ thống mũi điện tử (Electronic Nose – E-nose) sử dụng mạng cảm biến khí bán dẫn (MOS sensors) đã được phát triển nhằm mô phỏng khả năng nhận biết mùi của con người, từ đó nhận dạng và định lượng các khí trong hỗn hợp. Tuy nhiên, dữ liệu thu được từ các cảm biến khí này thường có nhiễu cao, phi tuyến và dễ trôi theo thời gian (sensor drift), gây khó khăn trong việc phân tích, nhận dạng mẫu hoặc ước lượng nồng độ khí.

Đây chính là lúc các bài toán học máy (Machine Learning – ML) thể hiện vai trò vượt trội. Học máy cung cấp các thuật toán có khả năng học từ dữ liệu cảm biến, tự động phát hiện mẫu đặc trưng, nhận dạng mối quan hệ phi tuyến và đưa ra dự đoán chính xác về tình trạng không khí. Các mô hình học máy như K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting, Artificial Neural Network (ANN) và các mô hình học sâu (Deep Learning) như Convolutional Neural Network (CNN) hay Autoencoder (AE) đã được áp dụng rộng rãi trong nhiều nghiên cứu nhằm phân loại loại khí, hồi quy nồng độ, dự đoán xu hướng ô nhiễm, và phát hiện bất thường (anomaly detection) trong dữ liệu giám sát môi trường.

Cụ thể, trong lĩnh vực giám sát chất lượng không khí trong nhà, học máy có thể được ứng dụng trong nhiều khía cạnh sau:

Phân loại loại khí và nguồn phát thải:

Các thuật toán phân loại giúp nhận dạng loại khí ô nhiễm (ví dụ: formaldehyde, benzene, toluene, NH_3, H_2S...) dựa trên phản hồi của các cảm biến. Việc xác định đúng loại khí giúp người quản lý môi trường trong nhà có biện pháp xử lý kịp thời.

Hồi quy và ước lượng nồng độ khí:

Các mô hình hồi quy như Linear Regression, XGBoost Regressor, ANN Regressor cho phép ước lượng nồng độ từng khí trong hỗn hợp dựa trên tín hiệu đầu vào từ mạng cảm biến. Điều này đặc biệt hữu ích khi cần xác định mức độ ô nhiễm (ppm) để so sánh với ngưỡng tiêu chuẩn an toàn của WHO hoặc EPA.

Phát hiện bất thường và cảnh báo sớm:

Các mô hình như Isolation Forest, One-Class SVM hoặc Autoencoder-based anomaly detection có thể phát hiện các giá trị cảm biến bất thường, giúp đưa ra cảnh báo sớm khi chất lượng không khí suy giảm đột ngột, tránh các rủi ro sức khỏe hoặc hỏa hoạn do khí độc tích tụ.

Dự báo xu hướng ô nhiễm:

Dựa trên chuỗi thời gian dữ liệu (time-series), các mô hình như LSTM (Long Short-Term Memory) hoặc Temporal Convolutional Networks (TCN) có thể dự đoán sự biến thiên của chất lượng không khí trong tương lai, hỗ trợ điều chỉnh hệ thống thông gió, điều hòa không khí hoặc thiết bị lọc khí.

Tối ưu hóa hệ thống cảm biến và năng lượng:

Học máy giúp lựa chọn cảm biến tối ưu (sensor selection) hoặc giảm thiểu số lượng cảm biến cần thiết trong hệ thống, qua đó giảm chi phí và tiêu thụ năng lượng mà vẫn đảm bảo hiệu quả giám sát.

Bên cạnh đó, các nghiên cứu hiện đại còn kết hợp học máy với Internet of Things (IoT) để xây dựng hệ thống giám sát chất lượng không khí thông minh. Các cảm biến IoT liên tục thu thập dữ liệu, gửi lên nền tảng điện toán đám mây (cloud), nơi các mô hình học máy được triển khai để phân tích theo thời gian thực. Người dùng có thể theo dõi tình trạng không khí trong nhà thông qua ứng dụng di động, nhận cảnh báo tự động khi nồng độ ô nhiễm vượt ngưỡng.

Hơn nữa, việc ứng dụng học sâu (Deep Learning) đang mở ra hướng tiếp cận mới, cho phép trích xuất đặc trưng phức tạp từ dữ liệu cảm biến thô. Ví dụ, mô hình Autoencoder kết hợp XGBoost có thể tự động học biểu diễn đặc trưng ẩn (latent features) giúp tăng độ chính xác khi ước lượng nồng độ khí. Các mô hình lai (hybrid models) giữa học sâu và thuật toán tăng cường cũng được chứng minh là mang lại hiệu suất vượt trội trong điều kiện dữ liệu cảm biến nhiều nhiễu.

Tóm lại, việc ứng dụng bài toán học máy trong giám sát chất lượng không khí trong nhà không chỉ góp phần nâng cao hiệu quả và độ chính xác trong đánh giá môi trường sống mà còn tạo nền tảng cho các hệ thống nhà thông minh (Smart Home) và thành phố thông minh (Smart City) trong tương lai. Nhờ khả năng tự động học và thích nghi với sự thay đổi của dữ liệu cảm biến, học máy đang trở thành công cụ cốt lõi trong việc xây dựng các giải pháp giám sát không khí bền vững, tiết kiệm và thông minh – góp phần bảo vệ sức khỏe con người và cải thiện chất lượng cuộc sống trong kỷ nguyên công nghệ số.

2.1.2. Phân loại đa lớp hỗn hợp nhiều khí

Phân loại đa lớp hỗn hợp nhiều khí là một bài toán mở rộng của phân loại đa lớp, trong đó mỗi mẫu dữ liệu đại diện cho sự pha trộn của nhiều loại khí với các nồng độ khác nhau. Mục tiêu là xác định loại hỗn hợp khí cụ thể hoặc trạng thái dựa trên các đặc điểm đo được từ cảm biến.

Trong bài toán này, mỗi mẫu được gán vào một trong k lớp, mỗi lớp tương ứng với một loại hỗn hợp khí hoặc một trạng thái môi trường đặc trưng. Các tín hiệu đầu vào thường đến từ các cảm biến khí (ví dụ: điện áp, cường độ dòng điện) hoặc đặc trưng đã được xử lý (ví dụ: giá trị trung bình, độ lệch chuẩn).

Ví dụ:

- C1 = "Hỗn hợp NH3 và CO2 ở mức thấp".
- C2 = "Hỗn hợp H2S và CO ở mức trung bình".
- C3 = "Hỗn hợp NO2, NH3, và H2 ở mức cao".

+ Đặc điểm của bài toán

- **Đa chiều:** Mỗi hỗn hợp khí được đặc trưng bởi các tín hiệu từ nhiều loại cảm biến (thường có hàng chục cảm biến).
- **Dữ liệu phức tạp:** Các tín hiệu từ cảm biến chịu ảnh hưởng của nhiễu, trôi tín hiệu, hoặc tương tác giữa các khí.
- **Phân lớp không tuyến tính:** Mối quan hệ giữa tín hiệu đầu vào và nhãn hỗn hợp khí thường không tuyến tính, đòi hỏi các mô hình phức tạp như mạng nơ-ron sâu.

+ Phương pháp phân loại đa lớp hỗn hợp khí

a) Tiền xử lý dữ liệu:

***Chuẩn hóa tín hiệu*:** Loại bỏ ảnh hưởng của các giá trị lớn nhỏ khác nhau từ các cảm biến.

***Giảm chiều dữ liệu*:** Áp dụng các phương pháp như **PCA** (Phân tích thành phần chính) để giảm số lượng đặc trưng đầu vào, làm giảm độ phức tạp của mô hình.

***Loại bỏ nhiễu*:** Sử dụng bộ lọc (median filter, wavelet dENoising) hoặc các kỹ thuật học sâu như **AutoENcoder**.

Tăng cường dữ liệu: Tạo thêm dữ liệu bằng cách dịch chuyển, thêm nhiễu, hoặc áp dụng các phép biến đổi tín hiệu

b) Mô hình học máy:

***Học máy truyền thống*:**

- **Random Forest:** Hoạt động tốt với dữ liệu đa chiều, giảm thiểu hiện tượng quá khớp.

- **SVM (Support Vector Machine):** Áp dụng phương pháp mở rộng cho phân loại đa lớp.

Học sâu:

- **Mạng nơ-ron nhân tạo (ANN):** Với các tầng ẩn đầy đủ, ANN có thể học được các mối quan hệ phi tuyến tính giữa tín hiệu cảm biến và nhãn lớp.

c) Hàm kích hoạt và đầu ra: Sử dụng hàm kích hoạt ở lớp đầu ra để tính xác suất cho từng lớp.

d) Đánh giá mô hình phân loại đa lớp hỗn hợp khí

a) Các chỉ số chính: Accuracy, F1, Precision, Recall.

b) Ma trận nhầm lẫn: Phân tích chi tiết các lỗi phân loại giữa các lớp, xác định xem mô hình có bị nhầm lẫn giữa các lớp khí tương tự hay không.

2.1.3. Dự đoán nồng độ khí/hỗn hợp khí

Dự đoán là quá trình xây dựng một mô hình y=f(X) để ước lượng giá trị liên tục của y (hồi quy nồng độ khí).

Quy trình thực hiện:

a) Tiền xử lý dữ liệu

b) Chọn/đề xuất mô hình

c) Học mô hình:

- Huấn luyện mô hình hồi quy dựa trên tập dữ liệu bao gồm các cặp đầu vào X(vector đặc trưng) và giá trị mục tiêu y (nồng độ khí thực tế).

d) Dự đoán:

- Áp dụng mô hình đã học để dự đoán giá trị nồng độ khí trên tập dữ liệu kiểm thử

e) Đánh giá mô hình dự đoán bằng các sai số: F2, MAE, MSE, RMSE, MAPE

Các thuật toán phổ biến:

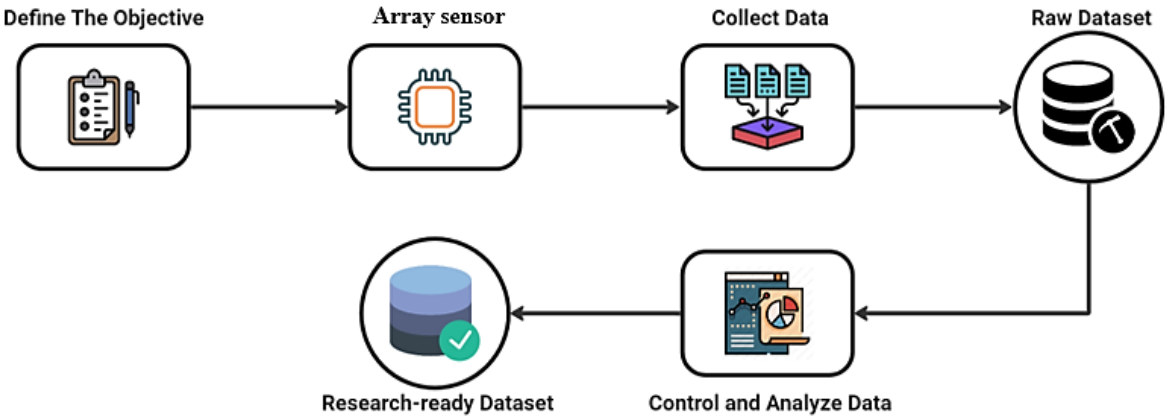
a) Học máy truyền thống: Linear Regression, Decision Tree, Random Forest, GradiENt Boosting.

b) Học sâu:

- Mạng ANN để dự đoán xu hướng phức tạp.

2.2. Phân tích và chuẩn hoá bộ dữ liệu

2.2. Quy trình thu thập dữ liệu



Hình 2.1. Quy trình thu thập dữ liệu

Quá trình xây dựng bộ dữ liệu cho hỗn hợp 3 loại khí VOC thường bao gồm một số bước, từ việc chọn khí VOC cần theo dõi đến thu thập và xử lý dữ liệu. Hình 2.1 cung cấp cái nhìn tổng quan về quá trình này

*Xác định mục tiêu: Mục tiêu của chúng tôi là thiết kế và phân tích bộ dữ liệu có chất lượng cao và đa dạng cho hỗn hợp VOC, bao gồm Ethanol, Acetone và Manol, nhằm mục đích giám sát chất lượng không khí trong nhà. Bộ dữ liệu này được lấy từ các cảm biến khí Nano SnO2 tiên tiến, cung cấp thông tin chi tiết và độ chính xác cao. Nó nhằm mục đích kết hợp cả phân loại và hồi quy trong học máy để cung cấp thông tin toàn diện về sự hiện diện của khí trong hỗn hợp đồng thời dự đoán và cảnh báo chính xác về nồng độ của chúng. Điều này có thể tăng cường đáng kể việc giám sát và quản lý môi trường, đặc biệt là trong môi trường có nguy cơ cao về khí độc.

* Lựa chọn mảng cảm biến: Việc lựa chọn cảm biến khí chính xác và phù hợp là rất quan trọng để đảm bảo độ tin cậy của bộ dữ liệu và hiệu suất của hệ thống giám sát chất lượng không khí trong nhà. Nó phải có khả năng phát hiện và đo lường đồng thời nhiều loại hỗn hợp khí VOC khác nhau.

*Thu thập dữ liệu: Xác định các đặc điểm cụ thể mà chúng tôi muốn thu thập từ mỗi cảm biến, chẳng hạn như nồng độ của các loại khí cụ thể, nhiệt độ, độ ẩm, áp suất, v.v. Xây dựng kế hoạch chi tiết về cách thức và thời điểm thu thập dữ liệu từ mỗi cảm biến, cũng như tần suất thu thập Xác định và thực hiện quy trình

thu thập dữ liệu, đảm bảo các thông số quan trọng như nồng độ, thời gian được ghi lại đầy đủ và chính xác. Tiến hành thu thập dữ liệu từ từng cảm biến theo kế hoạch đã vạch ra.

*Kiểm soát và phân tích dữ liệu: Áp dụng các biện pháp kiểm soát chất lượng để xác định và giảm thiểu sai sót trong quá trình thu thập dữ liệu, bao gồm kiểm tra và hiệu chuẩn cảm biến. Xử lý trước dữ liệu để loại bỏ nhiễu và xử lý các giá trị bị thiếu, bao gồm việc làm sạch và chuẩn hóa dữ liệu. Tích hợp dữ liệu từ mảng cảm biến để tạo thành bộ dữ liệu hoàn chỉnh và đồng bộ. Phân tích các thuộc tính thống kê và phân phối dữ liệu để hiểu sâu hơn về các đặc điểm của tập dữ liệu. Cung cấp cái nhìn tổng quan về tập dữ liệu, bao gồm đánh giá chất lượng, đặc điểm thống kê, đánh giá phân phối, phân tích tương quan và giải thích về tầm quan trọng của các quan sát và mối quan hệ được tìm thấy trong tập dữ liệu.

Quá trình này đảm bảo rằng bộ dữ liệu được xây dựng và quản lý chặt chẽ, đảm bảo chất lượng và độ tin cậy của dữ liệu được thu thập.

2.2.1. Phương pháp trộn hỗn hợp khí VOCs

Ba khí VOC được đo tại 9 điểm nồng độ dựa trên ngưỡng đảm bảo an toàn, được tính ra ml theo công thức ở phương pháp đo tĩnh, kết quả tính được theo Bảng 2.1. Khí Ethanol đo từ 50ppm đến 3500ppm; Khí Acetone đo từ 350ppm đến 7500ppm; Khí Methanol đo từ 50ppm đến 4000ppm

Bảng 2.1. Giá trị nồng độ của các khí

Khí	**Nồng độ**								
Ethanol (ppm)	50	100	200	350	500	1000	2000	3000	3500
Ethanol (ml)	1.3	2.6	5.2	9.1	13	26	52	78	91
Acetone (ppm)	350	500	750	1500	2500	3500	4500	5500	7500
Acetone (ml)	2.3	3.3	5.0	9.9	16.6	23.2	29.8	36.4	49.7
Methanol (ppm	50	100	200	500	1000	1500	2000	3000	4000
Methanol (ml)	0.6	1.2	2.4	6.0	12.0	18.0	24.0	35.9	47.9

Bảng 2.1 trình bày giá trị nồng độ của ba loại khí dễ bay hơi (VOCs) thường gặp gồm Ethanol, Acetone và Methanol, được biểu diễn đồng thời theo hai đơn vị: ppm (parts per million) – biểu thị nồng độ khí trong không khí, và ml – thể tích tương

ứng của dung môi lỏng được sử dụng để tạo ra nồng độ đó trong buồng thử nghiệm. Bảng dữ liệu này đóng vai trò rất quan trọng trong việc hiệu chuẩn cảm biến khí, mô phỏng phản ứng cảm biến và huấn luyện mô hình học máy cho các bài toán nhận dạng hoặc hồi quy nồng độ khí trong các hệ thống mũi điện tử (E-nose).

- Phân tích chi tiết từng loại khí

* Ethanol

Dãy nồng độ của Ethanol trong bảng trải từ 50 ppm đến 3500 ppm, tương ứng với thể tích từ 1.3 ml đến 91 ml. Có thể nhận thấy mối quan hệ giữa nồng độ và thể tích gần như tuyến tính, nghĩa là khi tăng thể tích dung môi, nồng độ khí tăng theo tỷ lệ xấp xỉ bội số. Cụ thể, khi thể tích Ethanol tăng gấp đôi (từ 1.3 ml lên 2.6 ml), nồng độ cũng tăng gấp đôi (từ 50 lên 100 ppm). Điều này phản ánh tính đồng nhất và ổn định trong phương pháp pha loãng khí – một yếu tố quan trọng đảm bảo dữ liệu huấn luyện mô hình có độ chính xác cao.

Ngoài ra, phạm vi nồng độ của Ethanol được chọn khá rộng (từ thấp đến rất cao), giúp mô hình học máy có khả năng nhận biết phản ứng cảm biến trong nhiều điều kiện khác nhau, từ mức phát hiện thấp (50 ppm) đến mức bão hòa hoặc nồng độ cao (≥ 3000 ppm). Dải nồng độ này cũng phù hợp với các nghiên cứu trong lĩnh vực đánh giá chất lượng không khí trong nhà và phát hiện hơi rượu trong thực phẩm hoặc môi trường làm việc.

* Acetone

Acetone có dải nồng độ cao hơn đáng kể so với Ethanol, dao động từ 350 ppm đến 7500 ppm, tương ứng với 2.3 ml đến 49.7 ml. Mối quan hệ giữa hai đại lượng này cũng tương đối tuyến tính, song độ dốc (tốc độ tăng nồng độ theo thể tích) cao hơn Ethanol. Điều này cho thấy độ bay hơi của Acetone mạnh hơn, và lượng nhỏ dung môi cũng có thể tạo ra nồng độ khí cao hơn trong không gian đo.

Đặc điểm này phản ánh tính chất hóa học và áp suất hơi riêng biệt của Acetone, vốn dễ bay hơi và phản ứng nhanh với cảm biến bán dẫn oxit kim loại (như SnO_2, ZnO, WO_3). Với các mức nồng độ cao (≥ 5000 ppm), dữ liệu này có thể giúp đánh giá khả năng bão hòa và giới hạn phát hiện trên của cảm biến, cũng như độ ổn định của mô hình hồi quy trong miền phi tuyến. Việc chọn phạm vi từ 350 đến 7500 ppm

cũng thể hiện mục tiêu bao phủ nhiều dải ứng dụng, từ giám sát an toàn công nghiệp (nồng độ cao) đến phân tích khí hơi trong môi trường kín.

* Methanol

Methanol có dải nồng độ thấp hơn Acetone nhưng tương đương Ethanol, trải từ 50 ppm đến 4000 ppm, tương ứng với 0.6 ml đến 47.9 ml. So sánh giữa ba khí cho thấy Methanol có tỉ lệ chuyển đổi ppm/ml cao nhất, nghĩa là chỉ cần lượng nhỏ dung môi đã tạo ra nồng độ khí đáng kể. Điều này phù hợp với đặc tính vật lý của Methanol, có áp suất hơi cao và khả năng khuếch tán mạnh hơn trong không khí so với Ethanol.

Dữ liệu của Methanol thể hiện xu hướng tăng gần tuyến tính, nhưng ở nồng độ cao (≥ 2000 ppm) có dấu hiệu phi tuyến nhẹ – thể tích tăng nhanh hơn so với nồng độ, có thể do giới hạn độ bay hơi trong buồng đo hoặc sự bão hòa một phần. Thông tin này rất hữu ích khi xây dựng mô hình hồi quy, bởi nó giúp kiểm tra khả năng mô hình thích ứng với vùng phi tuyến của dữ liệu.

- Nhận xét so sánh tổng hợp

Về phạm vi nồng độ: Acetone có dải nồng độ rộng nhất (350–7500 ppm), trong khi Ethanol và Methanol chỉ dao động từ khoảng 50 đến 3500–4000 ppm. Điều này phản ánh độ bay hơi và ứng dụng thực tế khác nhau của ba loại khí.

Về độ tuyến tính giữa ppm và ml: Cả ba khí đều cho thấy mối quan hệ gần tuyến tính ở vùng nồng độ thấp, song Methanol và Acetone có thể xuất hiện phi tuyến nhẹ ở nồng độ cao do đặc tính vật lý riêng và giới hạn bay hơi.

Về độ nhạy cảm tiềm năng của cảm biến: Với cùng một thể tích dung môi, Methanol thường tạo ra nồng độ khí cao hơn Ethanol, chứng tỏ Methanol dễ phát hiện hơn trong các hệ thống cảm biến. Trong khi đó, Acetone, với dải nồng độ rất cao, thường được sử dụng để kiểm tra ngưỡng đáp ứng tối đa và khả năng phân biệt giữa các hợp chất tương tự.

Về ứng dụng thực tiễn:

Ethanol phù hợp cho các thí nghiệm kiểm định cảm biến và các bài toán hồi quy cơ bản do đặc tính ổn định, dễ kiểm soát.

Acetone thích hợp cho các nghiên cứu về độ chọn lọc, khả năng nhận dạng và đánh giá hiệu suất cảm biến ở nồng độ cao.

Methanol hữu ích trong các bài toán đánh giá độ nhạy và giới hạn phát hiện thấp của cảm biến.

- Đánh giá tổng thể

Bảng 2.1 thể hiện sự lựa chọn thí nghiệm hợp lý, có tính đại diện và bao phủ nhiều dải nồng độ, giúp mô hình học máy được huấn luyện trên tập dữ liệu phong phú, đảm bảo khả năng tổng quát hóa và dự đoán chính xác nồng độ khí trong môi trường thực tế. Mối tương quan chặt chẽ giữa đơn vị ppm và ml chứng tỏ quá trình pha chế mẫu và hiệu chuẩn được kiểm soát tốt, tạo nền tảng tin cậy cho việc thu thập dữ liệu cảm biến. Ngoài ra, sự khác biệt rõ rệt giữa ba loại khí giúp mô hình phân biệt đặc trưng hóa học dễ dàng hơn, góp phần nâng cao hiệu quả phân loại, hồi quy và phát hiện đa khí hỗn hợp trong các ứng dụng giám sát chất lượng không khí trong nhà.

Tóm lại, bảng dữ liệu này không chỉ cung cấp thông tin định lượng về nồng độ các khí VOCs mà còn phản ánh tính khoa học và tính thực tiễn trong thiết kế thí nghiệm, đảm bảo tính toàn vẹn của dữ liệu đầu vào cho các nghiên cứu học máy trong lĩnh vực cảm biến khí và mũi điện tử thông minh.

Dựa trên 9 điểm nồng độ, chúng tôi tiến hành trộn hỗn hợp 3 khí Ethanol, Acetone và Methanol theo Bảng 2.2. Hỗn hợp đo được chia thành 8 lớp: Lớp 1 không có khí nào; Lớp 2 đo 9 điểm nồng độ của khí Ethanol, Lớp 3 đo 9 điểm nồng độ của khí Acetone; Lớp 4 đo 9 điểm nồng độ của khí Methanol; Lớp 5 đo 9 điểm nồng độ của hỗn hợp khí Ethanol và Acetone; Lớp 6 đo 9 điểm nồng độ của hỗn hợp khí Ethanol và Methanol; Lớp 7 đo 8 điểm nồng độ của hỗn hợp khí Acetone và Methanol; Lớp 8 đo 81 điểm nồng độ của hỗn hợp khí Ethanol, Acetone và Methanol. Sử dụng 2 hệ đa cảm biến ứng với mỗi hỗn hợp sẽ tạo ra độ đáp ứng của đa cảm biến trong không gian 10 chiều (tương ứng với 10 cảm biến)

Bảng 2.2. Trộn hỗn hợp 3 khí Ethanol, Acetone và Methanol

MIX	Ethanol	Acetone	Methanol	CLASS	NUM
NoGas	0	0	0	1	1
Ethanol	1	0	0	2	9
Acetone	0	1	0	3	9
Methanol	0	0	1	4	9
Eth-Ace	1	1	0	5	9
Eth-Met	1	0	1	6	9
Ace-Met	0	1	1	7	8
Eth-Ace-Met	1	1	1	8	81
SUM					**135**

Bảng 2.2 trình bày cấu trúc dữ liệu và số lượng mẫu tương ứng cho các hỗn hợp khí (gas mixtures) được tạo ra từ ba hợp chất bay hơi hữu cơ (VOCs) chính là Ethanol, Acetone và Methanol, nhằm phục vụ cho thí nghiệm nhận dạng và hồi quy trong hệ thống mũi điện tử (E-nose). Bảng này không chỉ thể hiện các dạng kết hợp khí khác nhau, mà còn mã hóa lớp (CLASS) và số lượng mẫu (NUM) tương ứng cho từng loại hỗn hợp, giúp hình thành bộ dữ liệu phân loại đa lớp có cấu trúc rõ ràng, rất phù hợp cho việc huấn luyện và đánh giá mô hình học máy.

- Mô tả cấu trúc bảng

Bảng gồm sáu cột chính:

MIX: Tên hỗn hợp hoặc loại mẫu khí (ví dụ: Ethanol đơn, hỗn hợp Eth-Ace, hay hỗn hợp ba khí Eth-Ace-Met).

Ethanol, Acetone, Methanol: Ba cột nhị phân (0 hoặc 1) biểu diễn sự có mặt (1) hoặc vắng mặt (0) của từng loại khí trong hỗn hợp.

CLASS: Nhãn lớp (class label) – mã số định danh cho từng loại mẫu hỗn hợp, dùng trong huấn luyện mô hình phân loại.

NUM: Số lượng mẫu dữ liệu thu được cho mỗi loại khí hoặc hỗn hợp, phản ánh mức độ cân bằng của dữ liệu.

- Phân tích từng loại mẫu khí

(1) NoGas – Không khí nền (CLASS 1, NUM = 1)

Mẫu NoGas là trường hợp cơ sở, không chứa bất kỳ loại khí VOC nào (Ethanol = 0, Acetone = 0, Methanol = 0). Đây là mẫu nền (baseline), được sử dụng để hiệu chỉnh tín hiệu cảm biến và xác định đáp ứng gốc khi không có khí tác động.

Số lượng mẫu chỉ có 1, thể hiện đây là mẫu chuẩn, dùng để chuẩn hóa dữ liệu trước khi so sánh với các mẫu có khí. Dù chỉ chiếm tỷ lệ nhỏ, nhưng NoGas đóng vai trò rất quan trọng trong việc xác định độ thay đổi tín hiệu $\Delta R/R_0$, giúp loại bỏ sai lệch nền và nâng cao độ chính xác khi xử lý dữ liệu cảm biến.

(2) Ethanol – Mẫu khí đơn (CLASS 2, NUM = 9)

Ở nhóm này, chỉ có Ethanol = 1, còn các khí khác bằng 0. Tổng cộng có 9 mẫu đại diện cho các mức nồng độ khác nhau của Ethanol (tương ứng với Bảng 2.1 – từ 50 đến 3500 ppm).

Điều này cho phép đánh giá phản ứng riêng biệt của cảm biến đối với Ethanol, đồng thời huấn luyện mô hình để nhận dạng chính xác đặc trưng của Ethanol trong hỗn hợp đa khí.

Các mẫu này cũng giúp phân biệt tín hiệu của Ethanol với các khí có cấu trúc hóa học tương tự như Methanol.

(3) Acetone – Mẫu khí đơn (CLASS 3, NUM = 9)

Tương tự như Ethanol, nhóm này chỉ có Acetone = 1, còn lại là 0, với 9 mẫu ở các mức nồng độ khác nhau (350–7500 ppm).

Acetone thường tạo ra tín hiệu mạnh hơn Ethanol hoặc Methanol do có độ bay hơi cao, vì vậy nhóm dữ liệu này đóng vai trò quan trọng trong việc xác định mức độ nhạy và chọn lọc của cảm biến. Các mẫu Acetone đơn khí còn giúp hiệu chỉnh mô hình hồi quy, đảm bảo hệ thống không bị chồng tín hiệu khi gặp hỗn hợp nhiều khí.

(4) Methanol – Mẫu khí đơn (CLASS 4, NUM = 9)

Nhóm này có Methanol = 1, và hai khí còn lại bằng 0. Cũng có 9 mẫu tương ứng với các nồng độ từ thấp đến cao (50–4000 ppm).

Methanol là khí có áp suất hơi cao, phản ứng nhanh với cảm biến SnO_2. Do đó, các mẫu đơn Methanol là cơ sở để phân tích sự khác biệt tín hiệu giữa các loại ancol (Ethanol và Methanol).

Việc có cùng số lượng mẫu với Ethanol và Acetone đảm bảo sự cân bằng trong dữ liệu huấn luyện đơn khí.

(5) Eth-Ace – Hỗn hợp Ethanol và Acetone (CLASS 5, NUM = 9)

Đây là nhóm hỗn hợp hai khí, trong đó Ethanol = 1, Acetone = 1, Methanol = 0. Sự kết hợp này tạo ra phản ứng phức tạp vì hai khí đều có cấu trúc tương tự và độ bay hơi cao, dẫn đến sự chồng tín hiệu (signal overlapping) trên cảm biến. 9 mẫu được tạo ra với các mức nồng độ khác nhau, giúp mô hình học máy phân biệt các tín hiệu hỗn hợp và xác định tương tác lẫn nhau giữa các thành phần khí.

Dữ liệu nhóm này đóng vai trò quan trọng trong việc đánh giá khả năng phân tách phi tuyến của mô hình phân loại, đặc biệt khi áp dụng các kỹ thuật học sâu (Deep Learning).

(6) Eth-Met – Hỗn hợp Ethanol và Methanol (CLASS 6, NUM = 9)

Nhóm này cũng gồm 9 mẫu, với Ethanol = 1, Methanol = 1, và Acetone = 0. Đây là một hỗn hợp có tính chất đặc biệt vì Ethanol và Methanol là hai ancol có phản ứng cảm biến khá giống nhau, nên việc phân biệt chúng là một bài toán thách thức cho các mô hình học máy. Tập dữ liệu này giúp đánh giá độ chọn lọc của cảm biến và khả năng phát hiện thành phần riêng lẻ trong hỗn hợp tương đồng, thường được xem là tình huống kiểm tra độ chính xác cao (high-precision test case) trong các hệ E-nose.

(7) Ace-Met – Hỗn hợp Acetone và Methanol (CLASS 7, NUM = 8)

Đây là nhóm hỗn hợp hai khí còn lại, với Acetone = 1, Methanol = 1, và Ethanol = 0. Tổng cộng có 8 mẫu, ít hơn một mẫu so với các nhóm khác, có thể do giới hạn thiết bị hoặc sự trùng lặp tín hiệu trong quá trình lấy mẫu.

Nhóm Ace-Met thường tạo tín hiệu rộng và mạnh, giúp kiểm tra khả năng của cảm biến trong việc xử lý nồng độ cao và phát hiện đa khí có tính phản ứng mạnh.

Sự thiếu 1 mẫu trong nhóm này cần được lưu ý khi huấn luyện mô hình, vì nó tạo mất cân bằng nhỏ trong dữ liệu (class imbalance).

(8) Eth-Ace-Met – Hỗn hợp ba khí (CLASS 8, NUM = 81)

Đây là nhóm phức tạp nhất và có số lượng mẫu lớn nhất (81 mẫu), bao gồm cả ba khí Ethanol, Acetone, Methanol (đều = 1). Số lượng lớn cho phép mô hình học sâu khai thác đa dạng các mức nồng độ và tỷ lệ trộn khác nhau, phản ánh các tình huống thực tế trong môi trường, nơi nhiều khí đồng thời xuất hiện. Các mẫu này có thể dùng để huấn luyện mô hình hồi quy đa đầu ra (multi-output regression) hoặc mô hình phân loại nhiều thành phần. Hỗn hợp ba khí cũng là trường hợp kiểm tra tính phi tuyến mạnh nhất, giúp đánh giá khả năng tổng quát hóa của mô hình học máy khi áp dụng vào hỗn hợp VOCs phức tạp.

- Nhận xét tổng thể về bảng dữ liệu

Tổng số mẫu (SUM) là 135, trong đó mẫu đơn khí chiếm 27 mẫu (20%), hỗn hợp hai khí chiếm 26 mẫu (19%), và hỗn hợp ba khí chiếm 81 mẫu (60%). Cấu trúc này phản ánh định hướng nghiên cứu tập trung vào hỗn hợp đa khí – một vấn đề có tính thực tiễn và độ phức tạp cao hơn so với đơn khí.

Sự phân bố dữ liệu không hoàn toàn cân bằng, tuy nhiên điều này có chủ ý nhằm tăng cường độ đa dạng của dữ liệu phức tạp (multi-gas mixtures), giúp mô hình có khả năng học tốt hơn trong môi trường thực tế.

Các cột nhị phân (0/1) giúp mã hóa dữ liệu rõ ràng, tạo thuận lợi cho việc sử dụng trong các thuật toán phân loại (như Random Forest, SVM, hay mạng nơ-ron đa lớp – MLP).

Mã lớp CLASS từ 1 đến 8 cho phép xây dựng bài toán phân loại đa lớp (multi-class classification), trong đó mỗi lớp biểu diễn một kiểu hỗn hợp cụ thể.

Nhóm Eth-Ace-Met có 81 mẫu, thể hiện nỗ lực của nhóm nghiên cứu trong việc tăng cường dữ liệu cho lớp phức tạp nhất, đồng thời cũng cho phép đánh giá khả năng khái quát hóa của mô hình trong điều kiện hỗn hợp nhiều thành phần.

- Đánh giá ý nghĩa khoa học và ứng dụng

Bảng 2.2 có giá trị rất lớn trong việc mô phỏng các tình huống cảm biến đa khí thực tế. Trong môi trường thật (như phòng thí nghiệm, nhà xưởng, hoặc không gian kín), các hợp chất VOCs hiếm khi tồn tại riêng lẻ mà thường xuất hiện đồng thời ở nhiều nồng độ khác nhau.

Do đó, cấu trúc dữ liệu này giúp:

Huấn luyện mô hình học máy phân biệt tín hiệu chồng lấn, một trong những thách thức lớn nhất của hệ thống mũi điện tử.

Đánh giá hiệu suất mô hình trên từng mức độ phức tạp: từ khí đơn → hỗn hợp hai khí → hỗn hợp ba khí.

Tăng khả năng ứng dụng thực tiễn trong các hệ thống giám sát chất lượng không khí, phát hiện rò rỉ hóa chất, hoặc phân tích hơi VOCs trong lĩnh vực y sinh.

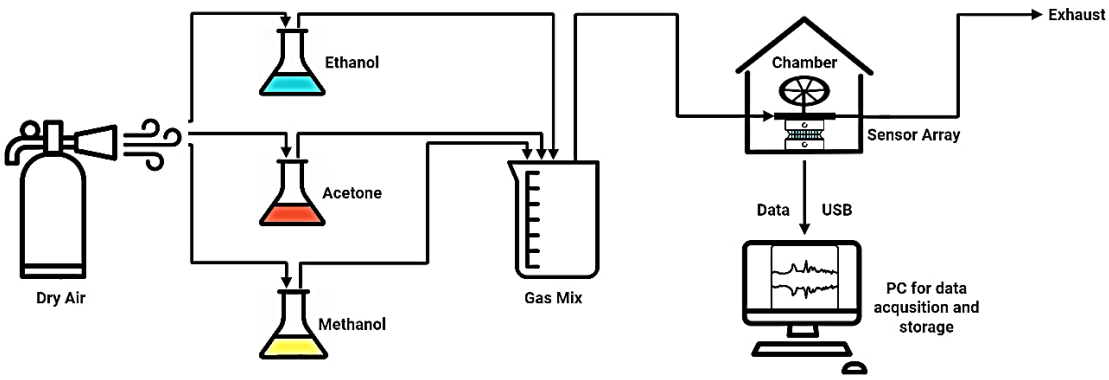
- Kết luận đánh giá

Nhìn chung, Bảng 2.2 thể hiện thiết kế thí nghiệm có tính hệ thống, khoa học và hướng đến ứng dụng thực tiễn cao. Việc bố trí các nhóm hỗn hợp theo mức độ phức tạp tăng dần, cùng với việc phân bố hợp lý số lượng mẫu, giúp tạo ra bộ dữ liệu huấn luyện và kiểm thử phong phú, phản ánh đúng bản chất của bài toán nhận dạng khí đa hợp phần.

Sự đa dạng của dữ liệu không chỉ hỗ trợ việc xây dựng mô hình phân loại chính xác, mà còn giúp kiểm định khả năng hồi quy và dự đoán nồng độ khí từng thành phần trong hỗn hợp – một bước quan trọng hướng tới hệ thống mũi điện tử thông minh có khả năng phân tích, cảnh báo và giám sát khí độc hại trong môi trường sống và công nghiệp.

2.2.2. Thí nghiệm đo hỗn hợp khí VOCs

Sơ đồ nguyên lý hệ đo và trộn khí như hình 2.2. Khí chuẩn được pha vào hỗn hợp với nồng độ cần pha loãng của các khí Ethanol, Acetone và Methanol cần đo, sau đó hỗn hợp khí cần đo được hút và xả vào buồng đo. Hệ đa cảm biến dây Nano SnO2 biến tính bề mặt Pt/Ag, tích hợp 10 cảm biến với công nghệ MEMS được đặt trong buồng đo để đáp ứng với hỗn hợp khí cần đo [22]. Hệ thống thu thập dữ liệu từ đa cảm biến được kết nối với máy tính qua chuẩn USB và được điều khiển từ máy tính thông qua chương trình ứng dụng viết trên ngôn ngữ lập trình Labview.

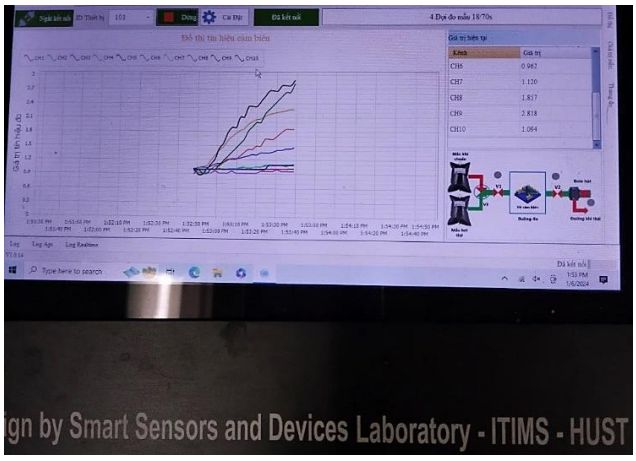


Hình 2.2. Sơ đồ nguyên lý hệ đo trộn hỗn hợp khí

. Thực hiện đo hỗn hợp khí

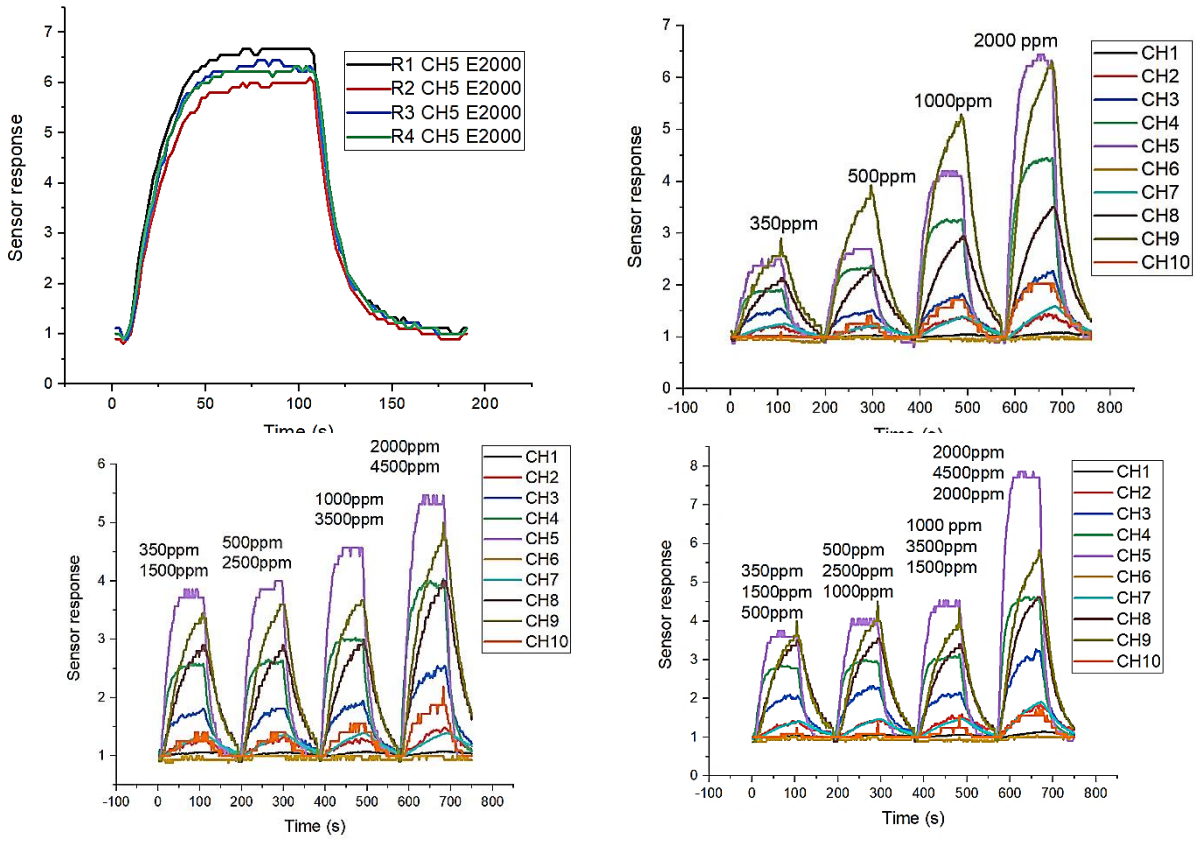
Điều kiện về môi trường lấy dữ liệu: nhiệt độ: 25 oC, áp suất: 64% RH, độ ẩm: 80-90%

Thời gian đo mẫu: thời gian chờ khởi động cảm biến 20 s, thời gian tăng nhiệt 10s, thời gian hút lấy mẫu 30s, thời gian đo mẫu 70s, thời gian kết thúc đo mẫu 80s, thời gian ghi số liệu 2s. Kiểm tra và hiệu chuẩn cảm biến trước và sau mỗi lần đo để đảm bảo độ chính xác và tin cậy.



Hình 2.3. Hình ảnh thực tế thu thập dữ liệu từ đa cảm biến khí

Ethanol ở nồng độ 2000ppm ở 4 lần đo khác nhau hiển thị trong hình 3a, có thể thấy rằng phản ứng của cảm biến ổn định trong suốt các giai đoạn này. Tổng cộng có 135 bộ dữ liệu thô được thu được. Một số dữ liệu thô đo được điển hình được hiển thị trong Hình 2.4 b–d.



Hình 2.4. (a) Đồ thị phản ứng của ứng của CH5 với Ethanol ở nồng độ 2000ppm ở 4 lần đo khác nhau.

Hình (b) Đồ thị phản ứng của 10 cảm biến với Ethanol ở 4 nồng độ

Hình (c) Đồ thị phản ứng của 10 cảm biến với hỗn hợp Ethanol-Acetone ở 4 nồng độ

Hình (d) Đồ thị phản ứng của 10 cảm biến với Ethanol-Acetone-Methanol ở 4 nồng độ.

2.2.3. Mô tả và phân tích dữ liệu

a. Mô tả dữ liệu

Dataset gồm có 135 hỗn hợp nồng độ khác nhau của 3 khí VOC Ethanol, Acetone và Methanol, mỗi hỗn hợp nồng độ được đo 4 lần. Kích thước dataset 14945 dòng, 17 cột. Mỗi hỗn hợp dữ liệu tương ứng với độ đáp ứng của 10 cảm biến theo thời gian trong mỗi lần đo.

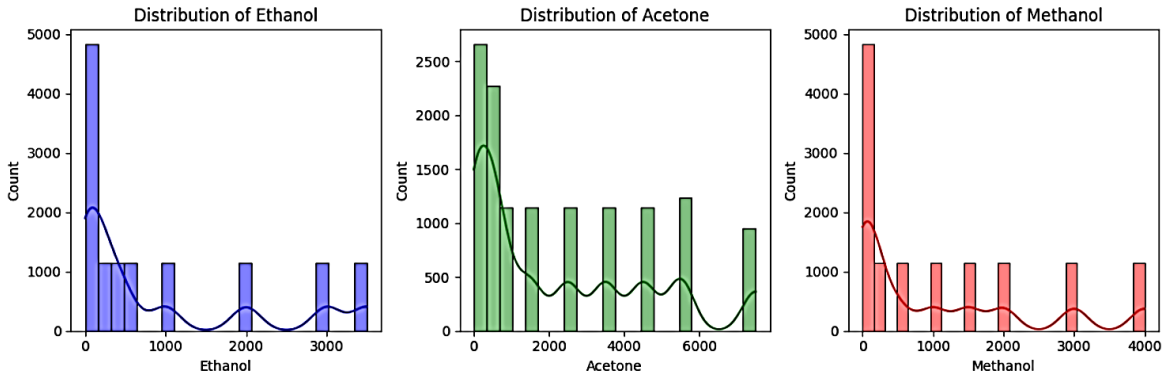
Chi tiết về dataset được mô tả chi tiết trong bảng 2.3.

Bảng 2.3. Bảng mô tả dữ liệu Data dictionary

Tên trường	Mô tả	Loại dữ liệu	Đơn vị
Ngày	Ngày đo hỗn hợp khí	Date	Date
Thời gian	Thời gian đo hỗn hợp khí	Time	Time
CH1 … CH10	Độ đáp ứng khí của cảm biến 1…10	Float	Quy hồi
MIX	Tên của hỗn hợp cần đo	Text	Text
Ethanol	Nồng độ của khí Ethanol	Integer	Quy hồi
Acetone	Nồng độ của khí Acetone	Integer	Quy hồi
Methanol	Nồng độ của khí Methanol	Integer	Quy hồi
Class	Lớp trộn hỗn hợp khí từ 1 đến 8: 0/1/2/3 khí	Integer	Phân loại

b. Phân tích đánh giá dữ liệu

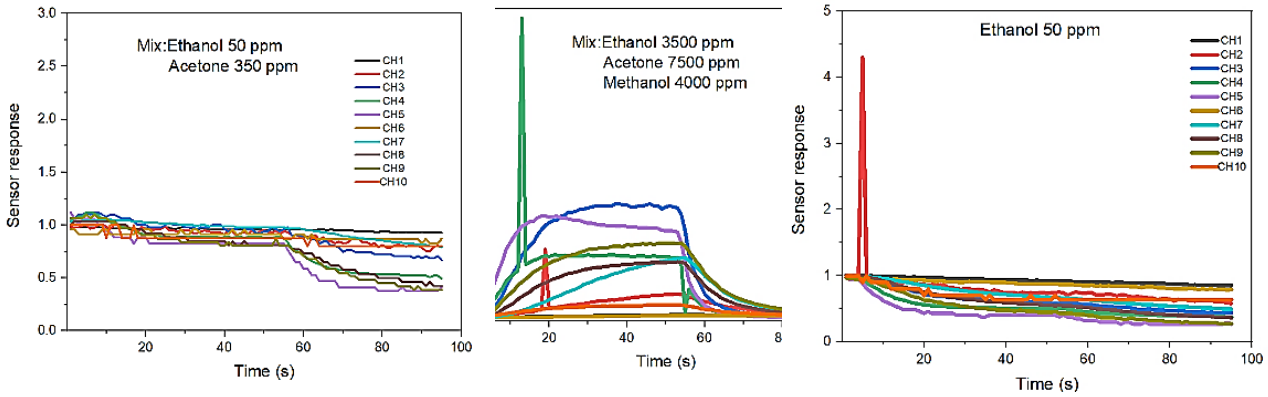
Hệ thống mảng 10 cảm biến tích hợp được đặt trong buồng đo để chứa hỗn hợp khí phục vụ phân tích. Do độ chọn lọc khác nhau của mỗi cảm biến đối với khí phân tích, việc tổng hợp phản ứng khí của từng cảm biến sẽ tạo ra "dấu vân tay điện tử" đặc trưng cho các hỗn hợp khí phân tích khác nhau. Sự phân bố của từng loại khí VOC trong tập dữ liệu tương đối cân bằng như Hình 2.5, với 108 điểm nồng độ cho Ethanol, 107 điểm nồng độ cho Acetone và 107 điểm nồng độ cho Metanol trong các hỗn hợp nồng độ VOC khác nhau.



Hình 2.5. Biểu đồ phân phối mỗi loại khí VOCs

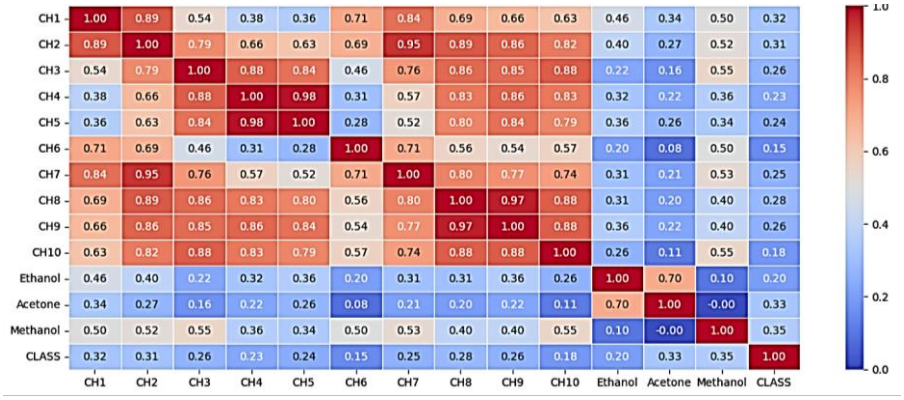
Dữ liệu bao gồm các phép đo từ 135 hỗn hợp nồng độ của 3 loại khí VOC khác nhau, với mỗi nồng độ được đo 4 lần, dẫn đến tổng cộng 135*4=540 hỗn hợp nồng

độ của 3 loại khí VOC. Để xử lý và xử lý dữ liệu null, có 1 giá trị null trong CH4 và 2 giá trị null trong CH10, giá trị này sẽ được lấp đầy bằng cách lấy giá trị trung bình của các điểm dữ liệu trước và sau. Sau đó, quy tắc Westgard được áp dụng để xác định và xử lý các điểm dữ liệu ngoài phạm vi/nhiễu như trong Hình 2.6. Nhiễu phát ra từ các cảm biến có thể là do sự thay đổi của môi trường như nhiệt độ, độ ẩm, áp suất hoặc sự mất ổn định của điện áp cảm biến cài đặt. Chúng tôi đã loại bỏ 17 mẫu hỗn hợp nồng độ khí do phản ứng cảm biến vi phạm quy tắc Westgard, chiếm 0,11% dữ liệu.



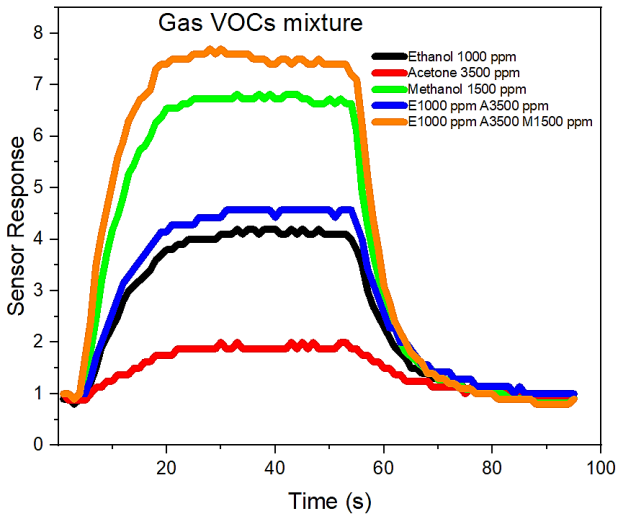
Hình 2.6. Đồ thị dữ liệu nhiễu của hỗn hợp khí VOC

Mỗi hỗn hợp khí thể hiện khả năng phản hồi tối ưu ở nhiệt độ cụ thể tương ứng với từng cảm biến. Phản hồi trung bình của cảm biến CH5 cao nhất là 3,960251, trong khi thấp nhất là CH6 ở mức 1,000193. Khả năng phản hồi của cảm biến CH6 đối với hỗn hợp khí cho thấy rất ít sự thay đổi, được phản ánh cụ thể ở giá trị trung bình là 1,000193 và phương sai là 0,065764. Các giá trị tương quan giữa các trường dữ liệu cho biết mức độ tương quan tuyến tính giữa từng cặp cảm biến. Hình 2.7 minh họa mối tương quan tuyến tính mạnh mẽ giữa các cặp cảm biến CH4 và CH5, CH8 và CH9. Metanol thể hiện mối tương quan tuyến tính tốt với CH3 và CH10, trong khi Acetone và Ethanol thể hiện mối tương quan tuyến tính yếu với các cảm biến.



Hình 2.7. Đồ thị tương quan của hỗn hợp khí VOCs

Mỗi hỗn hợp VOC có một dấu vân tay duy nhất, đan xen phi tuyến tính. Như được trình bày trong Hình 2.8, đường cong phản ứng khí đơn và hỗn hợp của cảm biến CH5 được lấy làm ví dụ. Hỗn hợp VOC bao gồm Ethanol 1000 ppm; Aceton 3500 ppm; Metanol 1500 ppm; Ethanol 1000 ppm và Aceton 3500 ppm; Ethanol 1000 ppm Acetone 3500 ppm và Metanol 1500 ppm.



Hình 2.8. Đồ thị phản hồi của cảm biến của CH5 đối với hỗn hợp khí VOCs

.

Đồ thị cảm biến cho thấy các đặc tính nhạy chéo đối với Ethanol, Acetone và Manol. Đầu ra phản hồi của cảm biến đối với khí hỗn hợp khí không bằng tổng phản ứng của cảm biến đối với hai loại khí mục tiêu, ba loại khí mục tiêu và hỗn hợp khí VOCs có đặc tính phi tuyến.

2.3. Xây dựng mô hình

Bài toán: Phân loại đa lớp, đầu vào là độ đáp ứng của 10 cảm biến đối với hỗn hợp khí từ CH1 đến CH10, đầu ra là phân loại đa lớp Class (phân loại 8 lớp từ 1..8)

2.3.1. Chuẩn bị dữ liệu

Dữ liệu thu thập được sẽ được trải qua các bước chuẩn bị, sau đó mới được phân chia thành tập huấn luyện và kiểm tra. Các bước chuẩn bị bao gồm:

✓ Lựa chọn dữ liệu sử dụng

✓ Kiểm tra kiểu dữ liệu

✓ Trực quan hóa dữ liệu

✓ Xác định các giá trị trống/null – outlier, thay thế dữ liệu trung bình của ô trước và ô sau vào các ô null

✓ Xử lý ký tự đặc biệt, ký tự không hợp lệ

✓ Làm sạch dữ liệu, lọc nhiễu

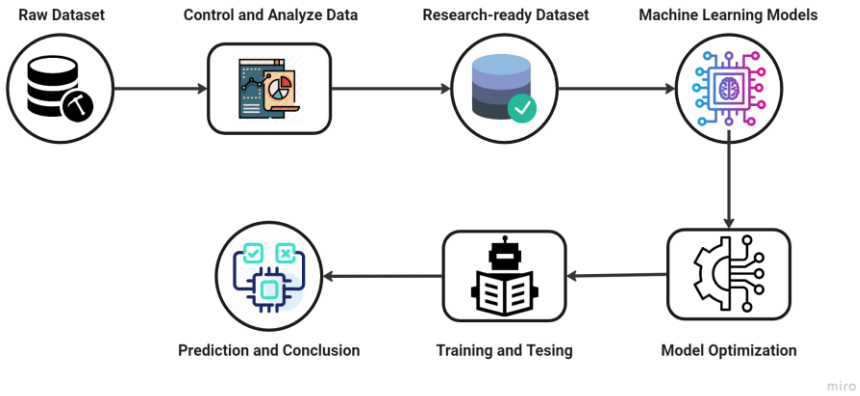
✓ Chuẩn hóa dữ liệu: Loại bỏ đặc trưng không liên quan

✓ Giảm kích thước dữ liệu: tăng tốc độ xử lý, giảm thời gian phân tích

✓ Trộn dữ liệu ngẫu nhiên và lấy mẫu phân tầng

✓ Phân chia dữ liệu phù hợp với bài toán: Để xác nhận hiệu quả của mô hình dự đoán, phần dữ liệu kiểm tra chiếm tỉ lệ 30% (4484 mẫu) trong tổng số 14945 mẫu. Tập dữ liệu huấn luyện được sử dụng để xác định các trọng số (hoặc tham số) của mô hình chứa 70% bộ dữ liệu (10461 mẫu).

2.3.2. Tinh chỉnh mô hình



Hình 2.9. Tinh chỉnh mô hình

- Dữ liệu sẵn sàng được đưa vào các mô hình học máy đã được lựa chọn cho bài toán.
- Các mô hình học máy bao gồm:
  - ✓ Support Vector Classification (SVC)
  - ✓ Random Forest (RF)
  - ✓ Multilayer Perceptron (MLP)
  - ✓ Artificial Neuron Networks (ANN)

- **Bảng 2.4. Siêu tham số tốt nhất cho 4 mô hình**

Model	Parameters	Range of values	Best value
SVC	C:	[0.001,0.01.0.1,1,10,100]	100
	Gamma:	[0.001,0.01.0.1,1,10,100]	100
	Kernel:	“linear”, “poly”, “sigmoid”, “rbf”	“rbf”
RF	max_depth:	[80,90,100,110]	110
	n_estimators:	[20,50,80,100,150,200,300]	200
	criterion:	'gini', 'entropy', 'log_loss'	'log_loss'
	max_features:	'sqrt', 'log2',None	'sqrt'
	min_samples_split:	[8,10,12]	8
	class_weight:	'balanced', 'balanced_subsample'	‘balanced'
MLP	hidden_layer_sizes:	[(64), (128), (512), (1024)]	(1024)
	activation:	‘relu’, ‘tanh’, ‘sigmoid’	‘relu’
	alpha:	[0.0001,0.001,0.01.0.1,1,10]	0.0001
	max iter:	[200,300,500,700,900,1000]	1000
ANN (one hidden layer/two hidden layers)	Node number: Node_numb1: Node_numb2:	[16,1024,32768] [16,32,64] [16,32,64]	1024 64 64
	Loss	MSE.Huber,MAE	MAE
	Optimizer	Adam, Adamax,SGD	Adam
	Batch size	[5,10,15]	5

- Sau đó các mô hình sẽ được tinh chỉnh siêu tham số và huấn luyện, kiểm tra và đưa ra các đánh giá.
- Thực nghiệm các mô hình được tiến hành trên môi trường Google Colab với ngôn ngữ lập trình Python.
- Tinh chỉnh siêu tham số là một bước quan trọng trong quy trình phát triển mô hình học máy. Bằng cách tìm ra các giá trị tối ưu cho các siêu tham số, mô hình có thể đạt được hiệu suất tốt nhất, đảm bảo độ chính xác cao, khả năng tổng quát hóa tốt và tối ưu hóa chi phí tính toán. Sau khi huấn luyện, kiểm tra và đưa ra các kết quả đánh giá cho các mô hình như Bảng 2.4. Thực nghiệm các mô hình được tiến hành trên môi trường Google Colab với ngôn ngữ lập trình Python.

2.3.3. Tính toán trong mô hình ANN

+ Đầu vào và truyền qua lớp ẩn đầu tiên

Đầu vào là một vector x=[x1,x2,…,x10]

+ Lớp ẩn đầu tiên:

Nhận đầu vào với 10 đặc trưng.

Có 64 nơ-ron.

Sử dụng hàm kích hoạt ReLU, giúp mô hình học các mối quan hệ phi tuyến giữa các đặc trưng.

Tính toán giá trị của các nơ-ron trong lớp ẩn đầu tiên:

h1=ReLU(W1x+b1)

W1: Ma trận trọng số có kích thước 64×10

b1: Vector bù (bias) có kích thước 64.

ReLU là hàm kích hoạt: ReLU(z)=max(0,z).

+ Lớp ẩn thứ hai:

Có 64 nơ-ron.

Sử dụng hàm kích hoạt ReLU, tiếp tục học các mối quan hệ phi tuyến từ đầu ra của lớp trước.

Đầu ra của lớp ẩn đầu tiên, h1, là đầu vào của lớp ẩn thứ hai:

h2=ReLU(W2h1+b2)

W2: Ma trận trọng số có kích thước 64×64.

b2: Vector bù có kích thước 64.

+ Lớp đầu ra:

Có 8 nơ-ron, mỗi nơ-ron đại diện cho một lớp đầu ra.

Sử dụng hàm kích hoạt softmax để chuyển đổi các giá trị đầu ra thành xác suất, với tổng xác suất bằng 1, phù hợp cho các bài toán phân loại đa lớp.

Đầu ra của lớp ẩn thứ hai, $h2$, là đầu vào của lớp đầu ra:

y=softmax($W3h2+b3$)

$W3$: Ma trận trọng số có kích thước 8×64.

$b3$: Vector bù có kích thước 8.

softmax là hàm kích hoạt:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{8} e^{z_j}}.$$

+ **Hàm mất mát và tối ưu hóa**

Hàm mất mát:

Sử dụng hàm mất mát MAE (loss function) được truyền vào.

Sử dụng bộ tối ưu hóa (optimizer) Adam được truyền vào.

Tính toán sai số giữa giá trị dự đoán y và giá trị thực tế.

Sử dụng MAE, hàm mất mát sẽ là:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Trong đó:

yi là giá trị thực tế của mẫu thứ i

y^i là giá trị dự đoán của mô hình cho mẫu thứ i

n là tổng số mẫu

Bộ tối ưu hóa: Adam Learning rate = 0.001

Sử dụng thuật toán tối ưu hóa Adam để cập nhật trọng số dựa trên hàm mất mát.

+ Huấn luyện mô hình

Huấn luyện một mô hình ANN trên tập dữ liệu X_train và y_train qua 1500 epoch, với việc sử dụng 20% dữ liệu huấn luyện làm tập xác thực. Quá trình huấn luyện được thực hiện với các batch nhỏ mỗi batch gồm 5 mẫu

Trong quá trình thí nghiệm, sử dụng chiến lược chia tập huấn luyện và tập kiểm tra ngẫu nhiên trước khi đưa vào mô hình để tránh khớp quá mức kết hợp lấy mẫu phân tầng và xác thực chéo 5 lần lấy kết quả trung bình để đánh giá hiệu suất đào tạo của mô hình.

2.4. Kết quả so sánh các mô hình

Bảng 2.5. Đánh giá hiệu suất của các mô hình học máy

CLASS	Precision				Recall				F1-score			
	SVC	RF	ANN	MLP	SVC	RF	ANN	MLP	SVC	RF	ANN	MLP
1	1	1	1	1	1	1	1	1	1	1	1	1
2	0.99	0.99	0.97	0.96	0.99	0.98	0.99	0.97	0.99	0.99	0.98	0.97
3	0.99	0.98	0.99	0.99	0.98	1	1	0.99	0.98	0.99	0.98	0.99
4	0.97	0.98	0.98	0.96	0.97	0.97	0.99	0.87	0.97	0.98	0.96	0.91
5	0.98	0.98	0.98	0.89	0.99	0.95	0.94	0.91	0.98	0.97	0.97	0.9
6	0.96	0.99	1	0.93	0.94	0.92	0.94	0.89	0.95	0.95	0.95	0.91
7	0.97	0.99	0.95	0.81	0.92	0.95	0.96	0.91	0.95	0.97	0.99	0.86
8	0.99	0.99	0.99	0.98	0.99	1	1	0.98	0.99	0.99	0.99	0.98
Accuracy	**0.9857**	**0.9875**	**0.9911(2)** **0.9882(1)**	**0.9650**								

Bảng kết quả thể hiện hiệu suất của bốn mô hình học máy, gồm Support Vector Classifier (SVC), Random Forest (RF), Artificial Neural Network (ANN) và Multilayer Perceptron (MLP), trong bài toán phân loại tám lớp dữ liệu. Các chỉ số được sử dụng để đánh giá gồm Precision, Recall và F1-score cho từng lớp, cùng với Accuracy tổng thể của mô hình. Đây là những thước đo phổ biến dùng để phản ánh khả năng nhận dạng chính xác và tổng quát hóa của các thuật toán học máy trong bài toán phân loại.

Kết quả cho thấy ở lớp 1, tất cả các mô hình đều đạt Precision, Recall và F1-score tuyệt đối (1.0). Điều này chứng tỏ dữ liệu của lớp này có đặc trưng dễ nhận dạng và không gây nhầm lẫn với các lớp khác. Đối với lớp 2 và lớp 3, các mô hình tiếp tục duy trì hiệu suất cao, với các giá trị trung bình của Precision và Recall dao

động từ 0.96 đến 1.0. Trong đó, SVC và RF thể hiện độ ổn định tốt hơn so với MLP, phản ánh khả năng phân biệt lớp mạnh mẽ và ít xảy ra sai sót.

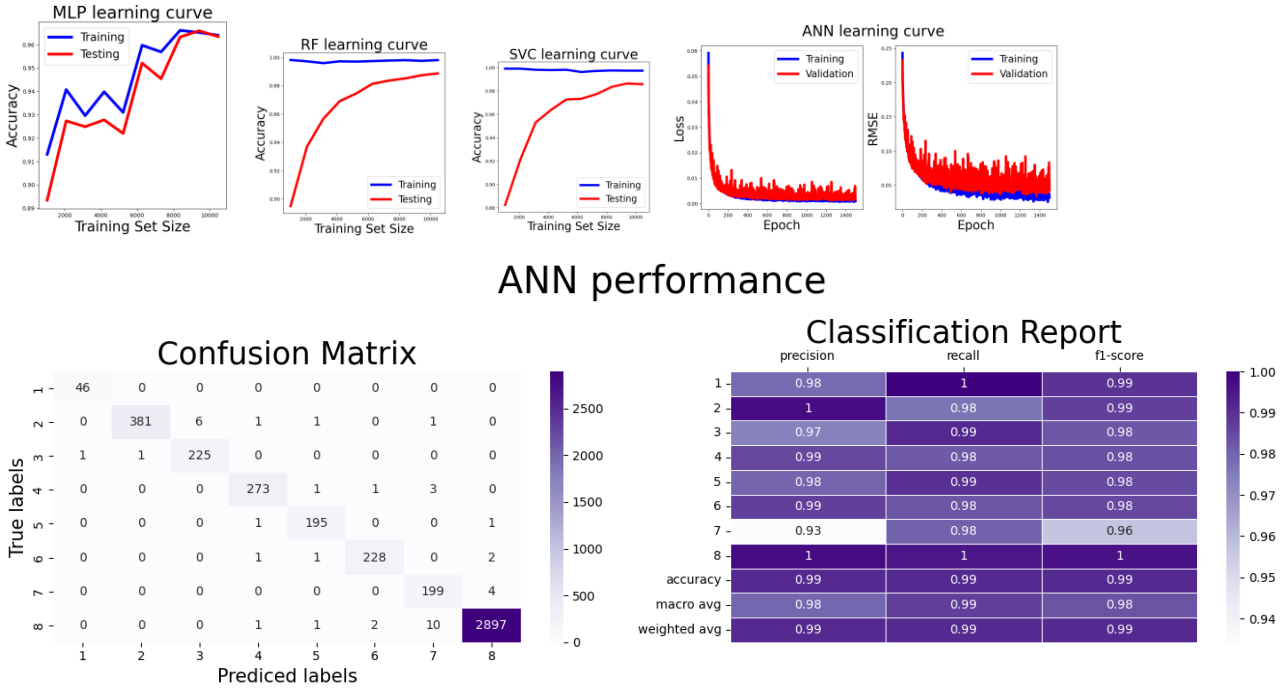
Sự khác biệt rõ ràng hơn xuất hiện từ lớp 4 đến lớp 7, khi hiệu suất của một số mô hình giảm đáng kể. Đặc biệt, MLP có xu hướng hoạt động kém ổn định, thể hiện qua Recall chỉ đạt 0.87 ở lớp 4 và F1-score chỉ còn 0.86 ở lớp 7. Ngược lại, RF và ANN vẫn duy trì được mức độ chính xác cao hơn, với F1-score dao động trong khoảng 0.95–0.98. Điều này cho thấy Random Forest và ANN có khả năng học sâu và tổng quát hóa tốt hơn, đặc biệt trong các lớp có đặc trưng phức tạp hoặc chồng chéo dữ liệu.

Ở lớp 5 và lớp 6, kết quả thể hiện sự khác biệt về cách các mô hình phản ứng với dữ liệu có mức độ nhiễu cao hơn. ANN đạt Precision tối đa 1.0 ở lớp 6, nhưng Recall giảm xuống 0.94, cho thấy mô hình có thể nhận diện chính xác các mẫu dương tính nhưng vẫn bỏ sót một số trường hợp. RF có kết quả cân bằng hơn giữa Precision và Recall, phản ánh khả năng học tốt mà không bị lệch về một phía. Đối với lớp 8, tất cả mô hình đều hoạt động gần như hoàn hảo (các giá trị trên 0.98), cho thấy dữ liệu của lớp này có đặc trưng rõ ràng và ít gây nhầm lẫn.

Xét về độ chính xác tổng thể (Accuracy), ANN đạt kết quả cao nhất với 0.9882, tiếp theo là RF (0.9875), SVC (0.9857) và cuối cùng là MLP (0.9650). Sự chênh lệch nhỏ giữa ANN và RF chứng minh cả hai mô hình đều có hiệu suất mạnh mẽ, tuy nhiên ANN có khả năng học các quan hệ phi tuyến phức tạp trong dữ liệu tốt hơn. Ngược lại, MLP mặc dù cùng họ với ANN nhưng lại cho kết quả kém hơn đáng kể, có thể do cấu trúc mạng hoặc siêu tham số chưa được tối ưu, dẫn đến hiện tượng quá khớp hoặc không hội tụ.

Tổng hợp các kết quả trên cho thấy tất cả các mô hình đều đạt hiệu suất cao (Accuracy trên 96%), chứng minh dữ liệu có tính tách biệt tốt và các đặc trưng được trích xuất có ý nghĩa. Tuy nhiên, ANN và RF vẫn nổi bật hơn nhờ sự ổn định giữa các lớp và khả năng duy trì cân bằng giữa Precision và Recall. SVC cũng là một lựa chọn tốt trong các bài toán có biên tách rõ ràng, song hiệu suất có thể giảm khi dữ liệu có mức độ phi tuyến cao. MLP, mặc dù là một biến thể sâu hơn của mạng nơ-ron, lại không đạt được hiệu quả tương đương do thiếu tối ưu hóa kiến trúc.

Nhìn chung, mô hình Artificial Neural Network (ANN) được xem là lựa chọn tối ưu nhất trong bốn mô hình được so sánh. ANN thể hiện khả năng khái quát hóa mạnh mẽ, đạt độ chính xác cao nhất và duy trì hiệu suất ổn định trên hầu hết các lớp. Tuy nhiên, Random Forest vẫn là phương án thay thế đáng tin cậy, đặc biệt trong những trường hợp cần mô hình dễ diễn giải, ổn định và ít phụ thuộc vào điều chỉnh siêu tham số phức tạp.



Hình 2.10. Đường cong học tập

Thông qua các chỉ số thống kê Precision, Recall, F1-score, hiệu suất của các mô hình DLNN, RF và MLP đối với tập kiểm tra được trình bày trong Bảng 2.5. Kết quả cho thấy, mô hình ANN với 2 lớp ẩn cho giá trị Accuracy (0.9911) cao hơn so với RF, SVC, MLP (lần lượt là 0,9875, 0,9857 và 0,9650) trên tập dữ liệu kiểm tra.

Bảng 2.6 cung cấp đánh giá so sánh các chỉ số của các thuật toán hồi quy nồng độ khí hỗn hợp. Mô hình hồi quy ANN đạt giá trị R-squared cao nhất là 0,95, thể hiện khả năng mô hình hóa và dự đoán tốt nhất. Cả MSE (0,04) và RMSE (0,21) đều thấp nhất, cho thấy khả năng duy trì sai số bình phương và độ lệch chuẩn nhỏ, nhờ đó giúp mô hình có độ ổn định dự đoán cao.

Bảng 2.6. Đánh giá hiệu năng của các mô hình học máy trong dự đoán nồng độ

Algorithm name	R-square	MSE	RMSE	MAE
Hồi quy mạng nơ ron nhân tạo ANN	0.95	0.04	0.21	0.10
Hồi quy Vector Hỗ trợ SVR	0.94	0.05	0.22	0.11
Hồi quy K-Nearest Neighbors KNN	0.93	0.07	0.23	0.08
Hồi quy Tuyến tính LN	0.61	0.61	0.44	0.38

Phân tích và đánh giá kết quả mô hình hồi quy

Bảng kết quả trên trình bày hiệu suất của bốn mô hình hồi quy phổ biến gồm Hồi quy Mạng nơ-ron nhân tạo (Artificial Neural Network – ANN), Hồi quy Vector hỗ trợ (Support Vector Regression – SVR), Hồi quy K-láng giềng gần nhất (K-Nearest Neighbors – KNN) và Hồi quy tuyến tính (Linear Regression – LN). Các chỉ số đánh giá bao gồm R-square (R^2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) và Mean Absolute Error (MAE), là những thước đo định lượng quan trọng phản ánh mức độ phù hợp, sai số và độ chính xác của mô hình khi dự đoán nồng độ khí.

Kết quả cho thấy mô hình ANN đạt hiệu suất cao nhất với $R^2 = 0.95$, MSE = 0.04, RMSE = 0.21 và MAE = 0.10. Giá trị R^2 cao chứng tỏ ANN có khả năng giải thích tới 95% sự biến thiên của dữ liệu thực tế, nghĩa là mối quan hệ phi tuyến giữa đầu vào và đầu ra đã được mô hình học rất tốt. Sai số MSE và RMSE thấp cho thấy sai lệch giữa giá trị dự đoán và giá trị thực là rất nhỏ, đồng thời MAE chỉ 0.10 phản ánh độ chính xác cao và khả năng dự đoán ổn định của mạng. Điều này hoàn toàn phù hợp với đặc tính của ANN – một mô hình có khả năng học sâu và trích xuất các đặc trưng phức tạp trong dữ liệu cảm biến, vốn thường có nhiễu và tính phi tuyến cao.

Đứng thứ hai là mô hình SVR, với $R^2 = 0.94$, MSE = 0.05, RMSE = 0.22 và MAE = 0.11. SVR cho thấy khả năng dự đoán khá tương đương ANN, mặc dù độ chính xác giảm nhẹ. Điều này có thể lý giải bởi việc SVR sử dụng hàm nhân (kernel) để ánh xạ dữ liệu vào không gian đặc trưng cao hơn, giúp mô hình hóa tốt các mối

quan hệ phi tuyến, song hiệu quả của nó phụ thuộc nhiều vào việc chọn tham số hạt nhân (kernel type, C, ε). Với dữ liệu cảm biến khí thường có biến động ngẫu nhiên và chồng chéo giữa các nồng độ, SVR vẫn chứng minh được khả năng tổng quát hóa mạnh mẽ nhưng có thể bị giới hạn trong việc học các mẫu dữ liệu phức tạp so với ANN.

Mô hình KNN regression đạt $R^2 = 0.93$, MSE = 0.07, RMSE = 0.23 và MAE = 0.08. Dù giá trị R^2 thấp hơn một chút so với ANN và SVR, nhưng MAE nhỏ nhất trong các mô hình (0.08) cho thấy KNN có khả năng dự đoán gần đúng trung bình tốt. Tuy nhiên, do bản chất KNN phụ thuộc vào khoảng cách lân cận trong không gian đặc trưng, mô hình này dễ bị ảnh hưởng bởi nhiễu trong dữ liệu và không thể học được các quan hệ phi tuyến sâu sắc. Ngoài ra, hiệu năng của KNN phụ thuộc nhiều vào số lượng hàng xóm (k) và cách chuẩn hóa dữ liệu, do đó hiệu quả có thể không ổn định khi áp dụng cho các tập dữ liệu lớn hoặc biến thiên mạnh theo thời gian.

Cuối cùng, mô hình hồi quy tuyến tính (LN) cho thấy hiệu suất thấp hơn rõ rệt với R^2 chỉ đạt 0.61, MSE = 0.61, RMSE = 0.44 và MAE = 0.38. Điều này chứng tỏ mô hình tuyến tính không phù hợp với dữ liệu cảm biến khí, vốn có quan hệ phi tuyến giữa tín hiệu đầu ra của cảm biến và nồng độ khí. Sai số lớn cho thấy mô hình không thể mô tả chính xác các đặc trưng biến thiên phức tạp trong dữ liệu, dẫn đến hiệu suất dự đoán kém. Dù vậy, mô hình LN vẫn có thể được xem như một chuẩn tham chiếu (baseline) để so sánh và đánh giá mức cải thiện của các mô hình phi tuyến tiên tiến hơn.

Tổng hợp các chỉ số cho thấy các mô hình phi tuyến (ANN, SVR, KNN) đều vượt trội hơn đáng kể so với mô hình tuyến tính trong việc dự đoán nồng độ khí. Trong đó, ANN là mô hình hiệu quả nhất, thể hiện khả năng khái quát hóa mạnh, sai số nhỏ, và độ chính xác cao nhất. SVR xếp ở vị trí thứ hai, với độ ổn định tốt và khả năng dự đoán sát thực tế. KNN tuy đơn giản hơn nhưng vẫn cho kết quả đáng khích lệ, phù hợp cho các bài toán cần tính toán nhanh và dễ triển khai. Ngược lại, LN tỏ ra không thích hợp trong các hệ thống cảm biến phức tạp đòi hỏi khả năng mô hình hóa phi tuyến cao.

Như vậy, có thể kết luận rằng mạng nơ-ron nhân tạo (ANN) là lựa chọn tối ưu cho bài toán hồi quy trong dự đoán nồng độ khí, nhờ vào khả năng học phi tuyến sâu, tự động trích xuất đặc trưng và giảm thiểu sai số hiệu quả. Sự vượt trội của ANN chứng minh tính khả thi của việc ứng dụng các mô hình học sâu trong lĩnh vực E-nose và giám sát chất lượng không khí, nơi mà mối quan hệ giữa tín hiệu cảm biến và nồng độ khí mục tiêu thường rất phức tạp và không tuyến tính

KẾT LUẬN VÀ KIẾN NGHỊ

Đề tài "Ứng dụng máy học trong việc giám sát chất lượng không khí trong nhà" đã khẳng định tiềm năng của việc kết hợp mũi điện tử (E-nose) với các mô hình máy học trong phát hiện, phân loại và dự đoán nồng độ khí ô nhiễm. Nghiên cứu cho thấy giải pháp này giúp khắc phục hạn chế của phương pháp truyền thống về chi phí, độ chính xác và khả năng triển khai, đồng thời mở ra hướng phát triển hệ thống giám sát thông minh, hỗ trợ cảnh báo sớm và bảo vệ sức khỏe cộng đồng. Đề tài cũng góp phần nâng cao chất lượng đào tạo tại Trường Đại học Hoa Lư, tạo cơ hội cho sinh viên tiếp cận công nghệ AI gắn với thực tiễn.

Nhóm nghiên cứu kiến nghị tiếp tục triển khai thực nghiệm trên quy mô lớn, mở rộng phạm vi đo với nhiều loại khí và yếu tố môi trường, đồng thời phát triển các thuật toán học sâu và phát hiện bất thường để tăng độ chính xác và khả năng ứng dụng.

DANH MỤC CÁC CÔNG TRÌNH ĐÃ CÔNG BỐ CÓ LIÊN QUAN

- Kết quả được công bố tại: 01 bài báo tại hội thảo quốc tế Scopus

Dang Thi Thu Ha, Nguyen Dinh Van, Nguyen Duc Hoa, Nguyen Van Duy, 2025, "*Design and analysis of a Volatile Organic Compounds mixture dataset for indoor air quality monitorin''g*, MMMS2024 – The Fourth International Conference on Material, Machines, and Methods for Sustainable Development, vol. 1, pp. 85-92, 2025, https://doi.org/10.1007/978-3-031-93816-0.

TÀI LIỆU THAM KHẢO

[1] D. Ma, J. Gao, Z. Zhang, and H. Zhao, “Gas recognition method based on the deep learning model of sensor array response map,” *Sensors Actuators, B Chem.*, vol. 330, Mar. 2021, doi: 10.1016/j.snb.2020.129349.

[2] R. Faleh and A. Kachouri, “A hybrid deep convolutional neural network-based electronic nose for pollution detection purposes,” *Chemom. Intell. Lab. Syst.*, vol. 237, no. February, p. 104825, 2023, doi: 10.1016/j.chemolab.2023.104825.

[3] W. Ni *et al.*, “Multi-task deep learning model for quantitative volatile organic compounds analysis by feature fusion of electronic nose sensing,” *Sensors Actuators B Chem.*, vol. 417, no. June, 2024, doi: 10.1016/j.snb.2024.136206.

[4] S. Sharma and M. Madou, “Review article: A new approach to gas sensing with nanotechnology,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 370, no. 1967, pp. 2448–2473, 2012, doi: 10.1098/rsta.2011.0506.

[5] M. Taştan and H. Gökozan, “Real-time monitoring of indoor air quality with internet of things-based e-nose,” *Appl. Sci.*, vol. 9, no. 16, 2019, doi: 10.3390/app9163435.

[6] G. Sberveglieri, G. Greco, D. Genzardi, E. Núñez-Carmona, S. Pezzottini, and V. Sberveglieri, “The Electronic Nose: Review on Sensor Arrays and Future Perspectives,” *Chem. Eng. Trans.*, vol. 95, no. March, pp. 265–270, 2022, doi: 10.3303/CET2295045.

[7] H. Nazemi, A. Joseph, J. Park, and A. Emadi, “Advanced micro-and nano-gas sensor technology: A review,” *Sensors (Switzerland)*, vol. 19, no. 6, 2019, doi: 10.3390/s19061285.

[8] D. Zhang *et al.*, “Highly sensitive BTEX sensors based on hexagonal WO3 nanosheets,” *Sensors Actuators, B Chem.*, vol. 293, no. April, pp. 23–30, 2019, doi: 10.1016/j.snb.2019.04.110.

[9] J. Fonollosa, S. Sheik, R. Huerta, and S. Marco, “Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring,” *Sensors Actuators, B Chem.*, vol. 215, no. March, pp. 618–629, 2015, doi: 10.1016/j.snb.2015.03.028.

[10] N. Joshi, V. Kushvaha, and P. Madhushri, *Machine Learning for Advanced Functional Materials*. 2023. doi: 10.1007/978-981-99-0393-1.

[11] Z. Chen, Z. Chen, Z. Song, W. Ye, and Z. Fan, "Smart gas sensor arrays powered by artificial intelligence," *J. Semicond.*, vol. 40, no. 11, 2019, doi: 10.1088/1674-4926/40/11/111601.

[12] Y. Xu, X. Zhao, Y. Chen, and W. Zhao, "Research on a mixed gas recognition and concentration detection algorithm based on a metal oxide semiconductor olfactory system sensor array," *Sensors (Switzerland)*, vol. 18, no. 10, 2018, doi: 10.3390/s18103264.

[13] D. Li, T. Lei, S. Zhang, X. Shao, and C. Xie, "A novel headspace integrated E-nose and its application in discrimination of Chinese medical herbs," *Sensors Actuators B Chem.*, vol. 221, pp. 556–563, Dec. 2015, doi: 10.1016/J.SNB.2015.06.144.

[14] T. Wang *et al.*, "Portable electronic nose system with elastic architecture and fault tolerance based on edge computing, ensemble learning, and sensor swarm," *Sensors Actuators B Chem.*, vol. 375, p. 132925, Jan. 2023, doi: 10.1016/J.SNB.2022.132925.

[15] J. Oh *et al.*, "Machine learning-based discrimination of indoor pollutants using an oxide gas sensor array: High endurance against ambient humidity and temperature," *Sensors Actuators B Chem.*, vol. 364, p. 131894, Aug. 2022, doi: 10.1016/J.SNB.2022.131894.

[16] Z. Ye, Y. Liu, and Q. Li, "Recent progress in smart electronic nose technologies enabled with machine learning methods," *Sensors*, vol. 21, no. 22, pp. 23–26, 2021, doi: 10.3390/s21227620.

[17] D. Karakaya, O. Ulucan, and M. Turkan, "Electronic Nose and Its Applications: A Survey," *International Journal of Automation and Computing*, vol. 17, no. 2. 2020. doi: 10.1007/s11633-019-1212-9.

[18] B. Szulczyński and J. Gębicki, "Currently commercially available chemical sensors employed for detection of volatile organic compounds in outdoor and indoor air," *Environ. - MDPI*, vol. 4, no. 1, pp. 1–15, 2017, doi: 10.3390/environments4010021.

[19] P. Boeker, “On ‘Electronic Nose’ methodology,” *Sensors Actuators, B Chem.*, vol. 204, pp. 2–17, 2014, doi: 10.1016/j.snb.2014.07.087.

[20] I. El Naqa and M. J. Murphy, “Machine Learning in Radiation Oncology,” *Mach. Learn. Radiat. Oncol.*, pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3.

[21] J. McCarthy and E. Feigenbaum, “Arthur L. Samuel: Pioneer in Machine Learning,” *ICGA J.*, vol. 14, no. 1, pp. 19–20, 2018, doi: 10.3233/icg-1991-14105.

[22] N. Van Duy *et al.*, “Design and fabrication of effective gradient temperature sensor array based on bilayer SnO2/Pt for gas classification,” *Sensors Actuators B Chem.*, vol. 351, no. August 2021, p. 130979, 2022, doi: 10.1016/j.snb.2021.130979.