ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA HÓA HỌC

Nguyễn Đức Phong

ỨNG DỤNG HỌC MÁY VÀ HỌC SÂU TRONG NHẬN DẠNG, PHÂN LOẠI ĐỐI TƯỢNG VÀ PHÂN TÍCH ĐỒNG THỜI, KHÔNG XỬ LÝ MẪU

Khóa luận tốt nghiệp đại học hệ chính quy

Ngành Hóa dược

(Chương trình đào tạo chất lượng cao)

Hà Nội - 2024

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA HÓA HỌC

Nguyễn Đức Phong

ỨNG DỤNG HỌC MÁY VÀ HỌC SÂU TRONG NHẬN DẠNG, PHÂN LOẠI ĐỐI TƯỢNG VÀ PHÂN TÍCH ĐỒNG THỜI, KHÔNG XỬ LÝ MẪU

Khóa luận tốt nghiệp đại học hệ chính quy

Ngành Hóa dược

(Chương trình đào tạo chất lượng cao)

Cán bộ hướng dẫn: PGS.TS Tạ Thị Thảo

NCS Nguyễn Đức Thanh

Hà Nội - 2024

LỜI CẢM ƠN

Với lòng biết ơn sâu sắc nhất em xin gửi đến các thầy, các cô ở Khoa Hóa học– Trường Đại học Khoa học Tự nhiên đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Em xin chân thành cảm ơn PGS.TS Tạ Thị Thảo và NCS Nguyễn Đức Thanh đã tận tâm hướng dẫn và tạo mọi điều kiện thuận lợi cho em hoàn thành bài khóa luận này.

Em cũng xin chân thành cảm ơn các thầy cô giảng dạy tại Bộ môn Hóa phân tích, các anh chị nghiên cứu sinh và các bạn sinh viên trong bộ môn đã giúp đỡ em trong quá trình học tập và nghiên cứu.

Bài khóa luận không tránh khỏi những thiếu sót là điều chắc chắn em rất mong nhận được những ý kiến đóng góp quý báu của thầy, cô và các bạn để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Hà Nội, ngày 29 tháng 5 năm 2024

Sinh viên

Nguyễn Đức Phong

MỤC LỤC

DANH MỤC CHỮ VIẾT TẮT

Tiếng Việt	Tiếng Anh	Viết tắt
Mạng nơron nhân tạo	Artificial Neural Networks	ANN
Cấu tử chính	Principal Components	PC
Phân tích thành phần chính	Principal component analysis	PCA
Mạng nơ ron nhân tạo kết hợp phân tích thành phần chính	Principal component analysis- Artificial Neural Networks	PCA-ANN
Máy véc tơ hỗ trợ	Support vector machine	SVM
Mạng nơ ron tích chập	Convolutional Neural Networks	CNN
Mạng nơ ron tích chập một chiều	One dimensional Neural Networks	1D-CNN
Bình phương tối thiểu nghịch đảo	Inverse least square	ILS
Bình phương tối thiểu riêng phần	Partial least square	PLS
Hồi quy cấu tử chính	Principalcomponent regression	PCR
Bình phương tối thiểu từng phần	Partial Least Square	PLS
Hồi quy bình phương tối thiểu từng phần	Partial Least Square Regression	PLSR
Lấy mẫu quá mức thiểu số	Synthetic Minority Over-sampling	SMOTE
Trung bình bình phương sai số	Mean squared error	MSE
Căn bậc hai trung bình bình phương sai số	Root mean squared error	RMSE

DANH MỤC BẢNG

DANH MỤC HÌNH

MỞ ĐẦU

Trí tuệ nhân tạo (AI) đã trải qua những bước tiến vượt bậc trong vài năm gần đây, đặc biệt là vào năm 2023. Không chỉ giữ vững vị thế là một lĩnh vực nghiên cứu học thuật, AI còn thể hiện rõ ràng tầm ảnh hưởng của mình trong thực tế, từ việc tăng tốc quy trình công nghệ thông tin, tăng cường khả năng phân tích dữ liệu đến việc đóng góp vào những bước tiến quan trọng trong y học. Các thành tựu này đã mở ra những cơ hội mới và thay đổi cách con người tiếp cận và giải quyết các vấn đề phức tạp trong nhiều lĩnh vực.

Ở Việt Nam và trên thế giới, sự phát triển của các phương pháp và thiết bị phân tích hiện đại mở ra khả năng thu thập lượng lớn thông tin về mẫu một cách dễ dàng. Các thiết bị sắc ký, quang phổ và cộng hưởng từ hạt nhân cung cấp rất nhiều dữ liệu về một mẫu đo, đồng nghĩa với việc dữ liệu trở nên phong phú hơn nhưng cũng phức tạp hơn, gây khó khăn cho người phân tích. Các thuật toán đa biến trong AI, học máy, cung cấp khả năng xử lý các loại dữ liệu vô cùng phức tạp và nhiều chiều, từ dữ liệu phổ, dữ liệu ảnh, dữ liệu âm thanh, … Điều này giúp các nhà nghiên cứu có thể phân tích và xử lý dữ liệu nhanh chóng, chính xác hơn, mở ra những khả năng mới trong nghiên cứu và ứng dụng.

Việt Nam đang phát triển kinh tế - xã hội, yêu cầu cao về chất lượng và an toàn thực phẩm, mỹ phẩm, dược phẩm. Bên cạnh hàng hóa có thương hiệu, vẫn tồn tại hàng kém chất lượng, không rõ nguồn gốc, không đảm bảo an toàn. Những mặt hàng này thường có giá rẻ và bị trà trộn bởi tiểu thương để đánh lừa người tiêu dùng. Nạn hàng giả, hàng kém chất lượng là vấn đề nhức nhối, gây ảnh hưởng lớn đến sức khỏe, tài chính người tiêu dùng và niềm tin vào thị trường. Điều này làm giảm uy tín của các nhà sản xuất, kinh doanh chân chính. Chính vì lẽ đó, nhu cầu định danh và phân loại các mặt hàng thực phẩm, dược phẩm dựa trên tác dụng, nguồn gốc, vv… để phục vụ cho việc kiểm định, giúp bảo vệ quyền lợi, an toàn của khách hàng và uy tín của các nhà sản xuất, hộ kinh doanh là vô cùng cần thiết. Hiện nay, những việc đánh giá thường dựa trên cảm quan của chuyên gia khiến cho việc đánh giá phụ thuộc rất nhiều vào kinh nghiệm và kiến thức của con người, điều đó làm mất đi tính khách quan của việc kiểm nghiệm và phân loại sản phẩm.

Bên cạnh nhu cầu phân loại, việc xác định hàm lượng của một hoạt chất có trong thực phẩm và dược phẩm …thường phải qua một quá trình phân tích phức tạp gômg tách chiết, làm giàu (nếu cần) sau đó đo tín hiệu và tính được hàm lượng của hoạt chất qua phương pháp hồi quy. Việc này gây ra vấn đề về thời gian và chi phí, nhất là trong thời đại hàng giả, hàng nhái, hàng kém chất lượng tràn lan như hiện nay.

Trong thời đại dữ liệu lớn, các nghiên cứu về thuốc sử dụng dữ liệu phổ là tương đối phổ biến, lượng dữ liệu lớn và phong phú đã giúp cho các nghiên cứu về định lượng các thành phần trong thuốc sử dụng học máy và học sâu phát triển một cách mạnh mẽ.

Ở Việt Nam, những năm gần đây, ứng dụng chemometrics trong nhận dạng, phân loại đối tượng phân tích cũng như phân tích đồng thời các chất sử dụng trực tiếp tín hiệu phân tích trên thiết bị hoặc từ bộ số liệu kết quả phân tích thường được tiến hành với sự hỗ trợ của phần mềm Matlab. Tuy nhiên, phần mềm này có giá thành đắt, nếu không có sự hỗ trợ của các đơn vị đào tạo hoặc thương mại thì người dùng thường sử dụng phiên bản bẻ khóa, không chính thống. Vì vậy, khi nghiên cứu phát triển cơ sở dữ liệu hoặc phát triển thiết bị sẽ khó có khả năng mở rộng và liên kết với các phần mềm khác khiến cho việc thương mại hóa các sản phẩm trở nên khó khăn. Vì vậy, nhu cầu đặt ra cần sử dụng các ngôn ngữ lập trình mã nguồn mở như Python, R, Java… với giá thành miễn phí, có các thư viện hỗ trợ xử lý dữ liệu chất lượng khiến cho việc mở rộng và thương mại hóa trở nên dễ dàng hơn bao giờ hết.

Với những lý do trên, để đóng góp vào xu hướng ứng dụng trí tuệ nhân tạo vào hóa phân tích, đề tài khóa luận "ỨNG DỤNG HỌC MÁY VÀ HỌC SÂU TRONG NHẬN DẠNG, PHÂN LOẠI ĐỐI TƯỢNG VÀ PHÂN TÍCH ĐỒNG THỜI, KHÔNG XỬ LÝ MẪU" được tiến hành nhằm triển khai việc ứng dụng học máy và học sâu trong phân loại đối tượng và phân tích đồng thời, phân tích không xử lý mẫu thông qua detector là camera chụp ảnh vùng Vis với ngôn ngữ lập trình mã nguồn mở Python trên cơ sở một số bộ số liệu đã công bố của cùng nhóm nghiên cứu cũng như các bộ số liệu có sẵn được phép tham khảo trên internet. Nội dung nghiên cứu gồm:

i) Định lượng đồng thời 3 kháng sinh trong mẫu thuốc bằng các thuật toán học máy
ii) Định lượng đồng thời 12 kháng sinh bằng các thuật toán học máy và học sâu
iii) Phân tích hình ảnh xác định nhanh độ đường của quả cam nguyên trạng
iv) Phân loại các đặc điểm của xoài bằng các thuật toán học máy và học sâu

CHƯƠNG 1. TỔNG QUAN

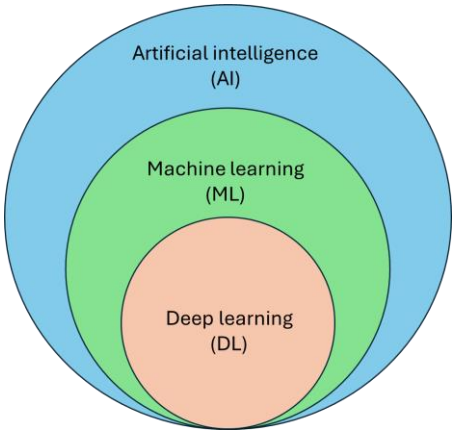
1.1 Khái niệm về trí tuệ nhân tạo, học máy và học sâu

Trí tuệ nhân tạo (AI): là một lĩnh vực khoa học máy tính rộng lớn tập trung vào việc tạo ra các tác nhân thông minh, có thể suy nghĩ, học hỏi và hành động tự chủ. Mục tiêu của AI là tạo ra các máy móc có thể mô phỏng các khả năng nhận thức của con người như học tập, lý luận, giải quyết vấn đề, nhận thức, lập kế hoạch, di chuyển và thao tác với môi trường xung quanh.

Học máy (Machine Learning): là một nhánh con của AI, tập trung vào việc cho phép máy móc học hỏi từ dữ liệu mà không cần được lập trình rõ ràng. Các thuật toán học máy có thể phân tích dữ liệu, nhận dạng mô hình và đưa ra dự đoán hoặc quyết định dựa trên những gì đã học được.

Học sâu (Deep Learning): là một nhánh con của học máy sử dụng các biến thể của mạng nơ-ron nhân tạo (artificial neural networks) để học hỏi từ dữ liệu, cho phép mạng học hỏi các mô hình phức tạp từ dữ liệu và đưa ra dự đoán chính xác hơn so với các phương pháp học máy truyền thống.

Có thể nhận biết giữa AI, machine learning và deep learning qua hình 1.1 như sau:



Hình 1.1 Mối quan hệ của AI, học máy và học sâu

AI, Machine Learning và Deep Learning có nhiều ứng dụng trong nhiều lĩnh vực khác nhau của đời sống. Nhìn chung, trí tuệ nhân tạo được phát triển theo hai hướng chính, dùng máy tính để bắt chước quá trình xử lý của con người và thiết kế những máy tính thông minh độc lập với cách suy nghĩ của con người. Hướng phát triển đầu tiên, có thể gọi là AI hẹp hay AI yếu (weak AI, narrow AI) là việc tạo ra các mô hình trí tuệ nhân

tạo chỉ làm một công việc cụ thể, hay nói cách khác là tạo ra các mô hình giống như các "chuyên gia" trong từng lĩnh vực cụ thể nhằm thay thế con người. Ví dụ như hệ thống chơi cờ Deep Blue của IBM đã đánh bại nhà vô địch thế giới Garry Kasparov vào năm 1997 [8], hay trong cùng lĩnh vực trò chơi, mô hình AlphaGo được Google phát triển đã đánh bại Lee Sedol - kỳ thủ cờ vây người Hàn Quốc, đương kim vô địch cờ vây thế giới vào năm 2016 [31]. Trong lĩnh vực y tế, mô hình CURIAL AI là mô hình trí tuệ nhân tạo đầu tiên trên thế giới có thể chẩn đoán bệnh nhân mắc COVID 19 dựa trên dữ liệu bệnh án với độ chính xác trên 90% [28], mô hình NLST với khả năng chẩn đoán ung thư phổi dựa trên dữ liệu ảnh chụp CT của bệnh nhân [4]. Hiện nay, các mô hình ngôn ngữ lớn tạo sinh và đọc hiểu văn bản (GPT, LLAMA) và các mô hình tạo sinh hình ảnh và video (DALL-E, Stable Diffusion) dựa trên cấu trúc Transformer được phát triển với một tốc độ nhanh chóng mặt, điều đó mở ra nhiều cơ hội to lớn nhưng cũng đầy thách thức với nhân loại.

1.2 Các thuật toán học máy, học sâu

1.2.1 Các thuật toán học máy

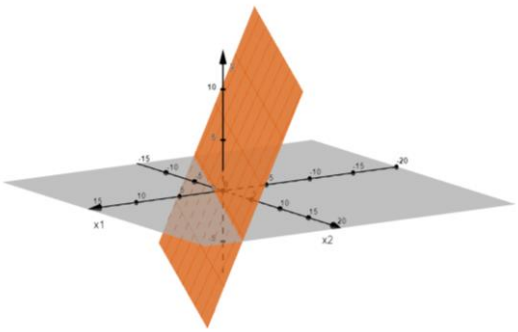
1.2.1.1 Thuật toán hồi quy đa biến tuyến tính (MLR)

Với mô hình hồi quy đơn biến, khi xem xét sự ảnh hưởng của một yếu tố tới yếu tố còn lại, tức là đánh giá sự tương quan giữa một biến phụ thuộc đối với một biến độc lập, hàm tuyến tính $\mathrm{y} = \mathrm{ax} + \mathrm{b}$ (trong đó *a* và *b* lần lượt là tham số) được sử dụng. Trong trường hợp có nhiều hơn một biến độc lập thì có thể xây dựng được hàm tuyến tính như sau:

$$\mathrm{y} = a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n + \mathrm{b}$$

Tương tự như hồi quy đơn biến tuyến tính, các giá trị $a_1, a_2, a_3, \ldots a_n$ là các giá trị độc dốc hoặc gọi là trọng số, *b* là giá trị trị số hay giao điểm

Trong mô hình hồi quy tuyến tính thì hàm số tuyến tính tuân theo một đường thẳng do biến x và biến y cấu tạo nên một mặt phẳng 2 chiều, tuy nhiên trong hồi quy đa biến, mỗi biến độc lập và biến phụ thuộc sẽ được coi là một chiều không gian, từ đó hàm số đa biến tuyến tính có thể tạo thành một đường thẳng, một mặt phẳng hoặc một siêu phẳng (hyperplane). Ví dụ về mặt phẳng của hàm số trong không gian ba chiều được biểu thị trên hình 1.2:



Hình 1.2 Mặt phẳng của hàm $y = -2x_1 + x_2 + 2$

Trong hóa học, đặc biệt là hóa phân tích, phương pháp hồi quy đa biến tuyến tính giúp giải quyết các bài toán xác định đồng thời nhiều cấu tử cùng có mặt trong hỗn hợp mà không cần tách loại trước khi xác định. Về nguyên tắc chỉ cần xây dựng dãy dung dịch chuẩn có mặt tất cả các cấu tử cần xác định với nồng độ biết trước trong hỗn hợp (các biến độc lập $x_1, x_2, x_3, \dots x_n$), đo tín hiệu phân tích của các dung dịch này dưới dạng một hay nhiều biến phụ thuộc $y_1, y_2, y_3, \dots y_k$) và thiết lập mô hình toán học mô tả quan hệ giữa tín hiệu đo và nồng độ các chất phân tích trong hỗn hợp. Dựa trên mô hình này có thể tìm được nồng độ của các cấu tử trong cùng dung dịch định phân khi có tín hiệu phân tích của dung dịch đó.

Các bước triển khai

Trước tiên có thể thấy nếu sử dụng nhiều biến độc lập $x_1, x_2, x_3, \dots x_n$ để dự đoán nhiều biến phụ thuộc $y_1, y_2, y_3, \dots y_k$ và các quan sát được tiến hành k lần, có thể mô hình hóa $x_1, x_2, x_3, \dots x_n$ và $y_1, y_2, y_3, \dots y_k$ về dạng ma trận như sau:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \quad Y = \begin{bmatrix} y_{11} & \cdots & y_{1k} \\ \vdots & \ddots & \vdots \\ y_{m1} & \cdots & y_{mk} \end{bmatrix}$$

Từ đó quy về phương trình có dạng $Y = A \cdot X + B$. Trong đó $Y(m \times k), A(n \times k), X(m \times n), B(m \times 1)$ đều là các ma trận, và việc giải bài toán hồi quy đa biến chính là việc tìm ra các ma trận A và B phù hợp nhất.

Việc cùng lúc tìm ra ma trận A và B sử dụng các phép toán đơn giản là một việc tương đối khó khăn, do thực tế ma trận B có dạng giống ma trận Y và chính là một ẩn cần tìm, vậy nên có thể gộp ma trận B vào trong ma trận A và thêm cột giá trị 1 vào bên phải ma trận X như sau

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nk} \\ b_{(n+1)1} & \cdots & b_{(n+1)k} \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & \cdots & x_{1n} & 1_{1(n+1)} \\ \vdots & \ddots & \vdots & \vdots \\ x_{m1} & \cdots & x_{mn} & 1_{m(n+1)} \end{bmatrix}$$

Từ đó, ma trận X mới sẽ có dạng $(m \times (n+1))$ và ma trận A mới có dạng $((n+1)\times k)$ nên có thể chuyển phương trình hồi quy về dạng đơn giản hơn là:

$$Y = X \cdot A$$

Để giải phương trình trên, chúng ta sẽ sử dụng phương pháp bình phương tối thiểu thông thường (Ordinary least square) để tìm được ma trận A theo công thức:

$$A = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$$

1.2.1.2 Thuật toán phân tích thành phần chính (PCA)

Trong không gian dữ liệu nhiều chiều, thuật toán PCA đi tìm những chiều không gian (đặc trưng) mới sao cho trên những chiều không gian mới này, chiều không gian đầu tiên sẽ có phương sai lớn nhất. Trong trường hợp một số chiều đầu tiên sẽ có phương sai xấp xỉ giống nhau, thì có thể nói rằng thuật toán PCA sẽ đi tìm một phép xoay trục tọa độ để được một hệ trục tọa độ mới sao cho trong hệ mới này, thông tin của dữ liệu sẽ tập trung chủ yếu ở một số thành phần, phần còn lại chứa ít thông tin hơn có thể được lược bỏ [32].

Khi đó, một tập hợp các biến liên quan với nhau ban đầu trong tập số liệu được chuyển thành tập hợp các biến không liên quan và được sắp xếp theo thứ tự giảm độ biến thiên hay phương sai. Những biến không liên quan này là sự kết hợp tuyến tính các biến ban đầu. Dựa trên phương sai do mỗi biến mới gây ra có thể loại bỏ bớt các biến phía cuối dãy mà chỉ mất ít nhất thông tin về các số liệu thực ban đầu. Bằng cách này sẽ giảm được kích thước của tập số liệu trong khi vẫn có thể giữ nguyên thông tin.

Các chiều không gian mới này được gọi là các cấu tử chính (principal components-PC), các cấu tử chính này chính là các vector riêng (eigenvectors) của tập dữ liệu ban đầu. Việc tính toán các vector riêng sẽ dựa vào ma trận hiệp phương sai của tập dữ liệu.

Bằng phương pháp PCA có thể chuyển tập số liệu gồm n chiều (n cột trong ma trận) ban đầu thành tập số liệu có kích thước nhỏ hơn gồm p chiều tương ứng với số PC.

Như vậy, trong quá trình tính toán tìm các thành phần chính, đã có sự quay thứ cấp của thành phần chính nhằm giúp cho việc quan sát tốt hơn và thu gọn các phương sai từ biến độc lập vào thành phần đơn giản đồng thời hiểu rõ hơn về số liệu gốc.

Các bước triển khai

- Tìm ra vector kỳ vọng của toàn bộ tập dữ liệu và chuẩn hóa tập dữ liệu

$$\overline{x} = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \hat{x} = x_n - \overline{x}$$

- Đặt $\hat{X} = [\widehat{x_1}, \widehat{x_2}, \dots \widehat{x_n}]$ là ma trận dữ liệu đã được chuẩn hóa, từ đó tiến hành tính ma trận hiệp phương sai:

$$S = \frac{1}{N}\hat{X}.\hat{X}^T$$

- Tính toán các vector riêng và trị riêng của ma trận hiệp phương sai
- Chọn K vector riêng tương ứng với K giá trị riêng lớn nhất để xây dựng ma trận U_K có các cột tạo thành hệ trực giao, từ đó tạo thành không gian con có K chiều dữ liệu
- Chiếu dữ liệu đã chuẩn hóa $\hat{X}$ xuống không gian con vừa tạo, từ đó sẽ có được tọa độ mới của các điểm dữ liệu chính là tập dữ liệu mới Z sau khi giảm về K chiều

$$Z = U_K^T \cdot \hat{X}$$

Lưu ý: việc truy tìm các vector riêng, trị riêng và xây dựng không gian con hoàn toàn phụ thuộc vào dữ liệu của tập huấn luyện, tập kiêm tra chỉ phóng chiếu các điểm dữ liệu về không gian con đã được xây dựng.

1.2.1.3 Thuật toán bình phương tối thiểu từng phần (PLS)

Thuật toán PLS là phương pháp đa biến dùng để mô hình hoá mối quan hệ giữa biến độc lập X và biến phụ thuộc Y. PLS mô hình hoá cả 2 biến X và Y đồng thời để tìm ra biến ẩn (latent variables- LVs) trong X mà từ đó sẽ đoán được biến ẩn trong Y. Số tối ưu các biến ẩn có thể được ước đoán bằng dùng thuật toán đánh giá chéo (cross-validation) hoặc tập số liệu kiểm tra riêng biệt [12].

Mục đích của PLS là mô hình hoá X sao cho có thể đoán được thông tin trong Y. PLS sẽ tối ưu hoá giá trị đồng phương sai (covariance) giữa ma trận X và Y. Hai ma trận

X và Y được phân tích thành hai ma trận trị số (score matrices) T, U và ma trận trọng số (loading matrices) P và Q.

Nói cách khác phương pháp PLS khác với các phương pháp hồi qui khác ở chỗ nó thích hợp cho những tập số liệu có số thí nghiệm ít hơn số biến và sự tương quan giữa các biến độc lập và có tính chất cộng tính cao.

Mục đích của PLS cũng là giảm số biến và tạo ra các cấu tử không liên quan sau đó biểu diễn phương trình bình phương tối thiểu với những cấu tử này.

Các bước triển khai

Giả sử ma trận X và Y đều đã được chuẩn hóa, tiến hành tính toán từng cấu từ một như sau:

Bước 1: lấy vector u_0 là một vector bất kỳ khác 0 có dạng giống một cột trong ma trận Y

Bước 2: tính w'(loading trong X) bằng công thức $w' = (u'u)^{-1}u'X$

Bước 3: chuẩn hóa w': $w' = \frac{w'}{||w'||}$

Bước 4: tính t (score trong X) bằng công thức $t = Xw$

Bước 5: tính q' (loading trong Y) bằng công thức $q' = (t't)^{-1}t'Y$

Bước 6: Chuẩn hóa q': $q' = \frac{q'}{||q'||}$

Bước 7: tính u (score trong Y) bằng công thức $u = Yq$

Từ bước 2 đến bước 7 là một vòng lặp liên tục cho đến khi hội tụ, kiểm tra sự hội tụ bằng cách kiểm tra t ở bước 4 có giống với t ở các vòng lặp cũ hay không, nếu giống với t ở các vòng lặp trước đó thì hội tụ và vòng lặp dừng lại, còn không thì tiếp tục. (Trong trương Y chỉ có một biến thì có thể bỏ qua bước 5 đến 7 bằng cách coi q bằng 1). Sau khi đạt sự hội tụ thì tiến hành thêm các bước như sau:

Bước 8: Tính toán p' (là loading thực của X) bằng công thức $p' = (t't)^{-1}t'X$

Bước 9: Chuẩn hóa p': $p' = \frac{p'}{||p'||}$

Bước 10: Hiệu chỉnh giá trị của t và w':

$$t = t\left|\left|t\right|\right|$$
$$w' = w'\left|\left|w'\right|\right|$$

Bước 11: Tìm hệ số hồi quy b của phương trình $\hat{u} = bt$

$$b = (t't)^{-1}t'u$$

Bước 12: Tính toán ma trận phần dư E và F:

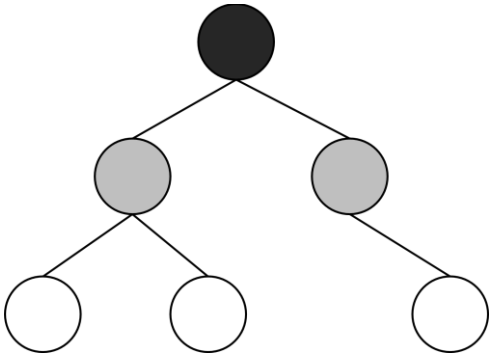
$$E_h = E_{h-1} - t_h p_h; X = E_0$$
$$F_h = F_{h-1} - b_h t_h q_h'; Y = F_0$$

Từ đây, tiến hành quay ngược lại bước 1 để tính các cấu tử chính tiếp theo, sau khi tính toán được cấu tử chính đầu tiên thì X ở các bước 2, 4 và 8 sẽ thay bằng E_h còn Y ở các bước 5 và 7 sẽ được thay bằng F_h

1.2.1.4 Thuật toán cây quyết định (Decision tree)

Thuật toán cây quyết định là thuật toán sử dụng cấu trúc mô hình cây phân cấp dùng để phân loại các lớp của đối tượng với biến đầu vào là các giá trị đặc trưng của đối tượng đó. Cây quyết định sẽ tạo ra các điều kiện phù hợp để phân loại được các đối tượng trong tập train, và từ đó sau khi có được các quyết định sẽ sử dụng chúng để dự đoán các đối tượng trong tập test. Cây quyết định có thể được ứng dụng trong cả bài toán hồi quy và bài toán phân loại [25].

Cấu trúc của cây quyết định được biểu diễn như hình 1.3:



Hình 1.3 Sơ đồ cây quyết định nhị phân

Mỗi node trong cây quyết định tương ứng với một điều kiện, hoặt một đặc trưng, tính chất, mỗi đương kết nối từ node này sang node khác được gọi là cành-biểu hiện cho

việc có thỏa mãn điều kiện của node mà cành bắt đầu nối hay không. Node đầu tiên được gọi là node gốc (root), các node cuối mà không có cành nối ra được gọi là lá. Trong cây quyết định, mỗi lá sẽ biểu thị cho các lớp. Dữ liệu đầu vào sẽ được đi qua cây quyết định từ node gốc qua các cành, qua các node (điều kiện) trung gian rối tới được lá của nó, đây chính là lúc đối tượng đó được phân loại.

Để hình dung một cách dễ dàng, mô hình cây quyết định sẽ tạo ra các đường biên, chính là các điều kiện để phân loại dữ liệu, nếu đầu vào của dữ liệu có nhiều đặc trưng, mô hình sẽ tạo ra các siêu phẳng để phân biệt các lớp. Việc xây dựng cây quyết định thường sử dụng các thuật toán phổ biến là ID3, C4.5 hoặc CART [25].

Các bước triển khai

Bước 1: Lựa chọn tiêu chí (điều kiện) phân loại:

Với hồi quy, tiêu chí được sử dụng là sai số trung bình MSE, còn đối với bài toán phân loại, các tiêu chí thường được sử dụng là $gini$ và $entropy$ trong đó:

- Chỉ số $gini$ đo lường xác suất phân loại sai của một phần tử được chọn ngẫu nhiên nếu nó được phân loại theo phân phối các nhãn trong nút đó:

$$gini(p) = 1 - \sum_{i=1}^{C} (p_i)^2$$

$$gini\ index = gini(p) - \sum_{i=1}^{K} \frac{m_k}{M} gini(c_k)$$

Với $gini(p)$ là chỉ số gini ở node cha, K là số node con được tách ra, $gini(c_k)$ là chỉ số gini ở node con thứ k, M là số phần tử ở node p, m_k là số phần tử node con thứ k.

- Entropy đo lường sự rối loạn thông tin trong một tập dữ liệu, giá trị entropy càng cao thì dữ liệu càng bị vẩn đục, nếu entropy càng thấp thì dữ liệu càng tinh khiết.

$$entropy = -\sum_{i=1}^{n} p_i \,.\, log(p_i)$$

Bước 2: Xây dựng cây:

Bắt đầu với nút gốc, chứa toàn bộ tập dữ liệu.

Chọn đặc trưng tốt nhất để chia dữ liệu dựa trên tiêu chí chia đã chọn. Điều này bao gồm việc tính toán tiêu chí cho mỗi đặc trưng và chọn đặc trưng giảm thiểu nó.

Chia dữ liệu thành các tập con dựa trên đặc trưng đã chọn.

Tạo các nút con cho từng tập con và lặp lại quy trình cho từng nút con cho đến khi gặp một trong các điều kiện dừng. Các điều kiện dừng có thể bao gồm chiều sâu tối đa, số mẫu tối thiểu trong một nút hoặc ngưỡng độ tinh khiết.

Bước 3: Cắt tỉa (Tùy chọn)

Cắt tỉa là một kỹ thuật được sử dụng để giảm kích thước cây quyết định bằng cách loại bỏ các nhánh không cung cấp thông tin đáng kể. Điều này có thể giúp ngăn chặn tình trạng quá khớp.

Cắt tỉa có thể được thực hiện bằng các phương pháp khác nhau, như cắt tỉa dựa trên chi phí phức tạp.

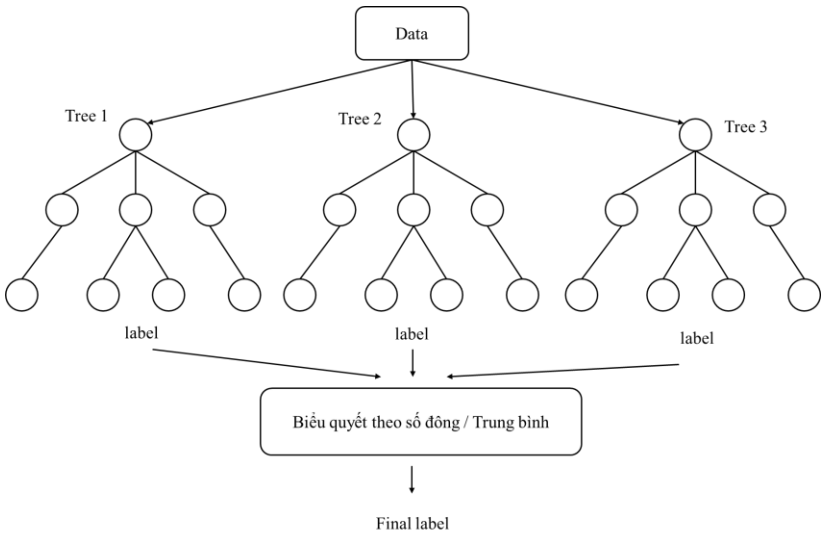
Bước 4: Dự đoán

Để thực hiện dự đoán, điều hướng cây quyết định bắt đầu từ nút gốc.

Theo đường đi của các điều kiện (đặc trưng) cho đến khi bạn đến một nút lá, và nhãn đa số (cho phân loại) hoặc giá trị trung bình (cho hồi quy) của các mẫu huấn luyện trong nút lá đó là dự đoán.

1.2.1.5 Thuật toán rừng ngẫu nhiên (Random forest)

Thuật toán rừng ngẫu nhiên sẽ tạo ra nhiều cây quyết định sử dụng nhiều mẫu con khác nhau. Các mẫu con này sẽ được sắp xếp và chọn lựa một cách ngẫu nhiên, đó là lý do từ random xuất hiện trong cái tên random forest. Không chỉ các mẫu được lấy ngẫu nhiên, mà các đặc trưng cũng được lấy ngẫu nhiên vào trong các mẫu con, kỹ thuật lấy mẫu này được gọi là Bootstraping. Sau khi có được các giá trị phân loại từ các cây, mô hình sẽ tiến hành bầu cử chọn ra lớp chiếm đa số sẽ là lớp cuối cùng. Việc này giúp cho mô hình có phương sai nhỏ hơn so với mô hình cây quyết định, giải quyết được vấn đề sai số lớn trong thuật toán cây quyết định [6]. Sơ đồ của một cấu trúc rừng ngẫu nhiên được hiển thị trên hình 1.4:



Hình 1.4 Sơ đồ mô hình rừng ngẫu nhiên

Các bước triển khai

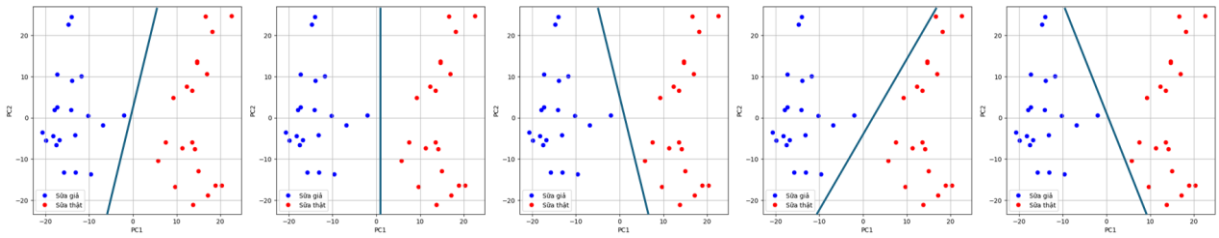
Bước 1: Lấy n bộ dữ liệu con từ bộ dữ liệu ban đầu sử dụng phương pháp Bootstraping. Lấy k đặc trưng ngẫu nhiên từ bộ dữ liệu ban đầu

Bước 2: Xây dựng các cây quyết định sử dụng bộ dữ liệu con từ bước 1.

Bước 3: Tiến hành lấy bầu cử nhãn xuất hiện nhiều lần nhất với bài toán phân loại, lấy trung bình với bài toán hồi quy.

1.2.1.6 Thuật toán máy vector hỗ trợ (SVM)

Trong không gian nhiều chiều, nếu các điểm dữ liệu thuộc các nhãn giống nhau nằm cùng về một khu và cách xa các điểm nhãn khác, tức là có thể phân cách tuyến tính (Linearly separable), mô hình máy vector hỗ trợ (support vector machine - SVM) đưa ra một đề xuất đó là tính toán một đường thẳng có thể phân chia các điểm dữ liệu một cách hoàn toàn, tuy nhiên khoảng cách từ các điểm dữ liệu xa nhất trong mỗi lớp là giống nhau, đồng thời khoảng cách này phải đạt giá trị tối đa. Khoảng cách mà đó được gọi là lề (margin), margin đều nhau chứng tỏ rằng đường phân chia là công bằng, margin đạt giá trị lớn nhất chứng tỏ sự phân chia càng rạch ròi. Các cách phân chia dữ liệu trong không gian được biểu thị trên hình 1.5 (dữ liệu sữa giả - sữa thật):



Hình 1.5 Một số cách phân chia dữ liệu trong không gian hai chiều

Nói tóm lại, trong không gian N chiều, thuật toán SVM chính là việc tìm ra siêu phẳng giúp cho việc phân chia điểm dữ liệu là hoàn hảo nhất và margin của nó đạt giá trị lớn nhất [10].

Với bài toán mà dữ liệu linear separable, thuật toán SVM được sử dụng gọi là hard margin SVM. Khi dữ liệu không thực sự linear separable, thuật toán SVM vẫn có thể sử dụng được để giải quyết bài bài toán bằng cách sử dụng soft margin SVM và kernel SVM

Các bước triển khai

Bài toán phân loại nhị phân hard-margin SVM được triển khai như sau:

Bước 1: Gán nhãn điểm dữ liệu: Tập dữ liệu gồm các điểm dữ liệu được biểu diễn bởi các vector đặc trưng: $x_1, x_2, \dots x_n$ trong đó mỗi điểm dữ liệu được gán cho một lớp nhãn tương ứng: $y_1, y_2, \dots y_n$ [23].

Bước 2: Tìm siêu phẳng tối ưu: Mục tiêu của SVM là tìm ra siêu phẳng có dạng $y = w^T.x + b$, nơi w là vector trọng số (weight) và b là trị số (bias).Siêu phẳng này phải cách biên tối ưu nhất (lớp gần nhất) mà không gây lỗi xác định nào. Điều này đồng nghĩa với việc tối ưu hoá margin (khoảng cách từ biên đến điểm dữ liệu gần nhất). Sau khi có được diêu phẳng, khoảng cách của điểm dữ liệu (x_n, y_n) trong dữ liệu được tính theo công thức $\frac{y_n(w^T.x_n+b)}{||w||_2}$.

Bước 3: Tối ưu hóa hàm mục tiêu: Để đạt được siêu phẳng tối ưu, cần tìm các giá trị trọng số và trị số tối ưu bằng cách tối đa khoảng cách lề của mô hình, đồng thời cực tiểu hóa hàm mất mát:

$$L(w,b) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} max(0,1 - y_i.(w^T.x_i + b))$$

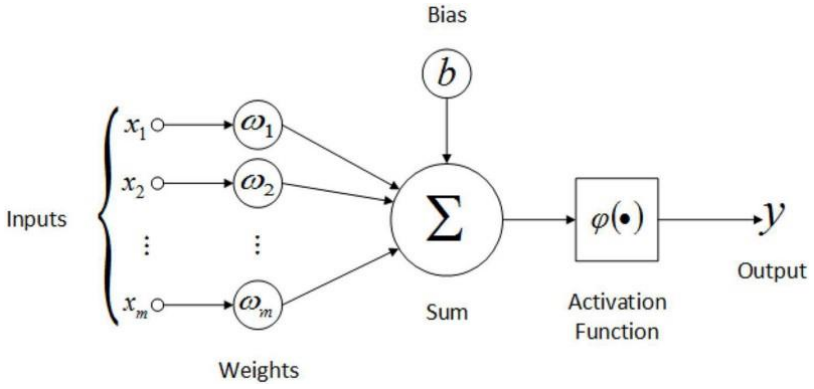
Bước 4: Tiến hành cực tiểu hóa hàm tối ưu bằng phương pháp CVXOPT nhằm tìm ra w và b, từ đó có thể tiến hành phân loại điểm dữ liệu bằng cách tính nhãn của điểm dữ liệu với đầu vào là x_n được tính bằng công thức:

$$class(x) = sgn(w^T.x_n + b) \ (\geq 0 \ là \ class \ A, < 0 \ là \ class \ B)$$

1.2.1.7 Mạng nơ ron nhân tạo (ANN)

Mạng nơron nhân tạo là một mô phỏng xử lý thông tin, được nghiên cứu từ hệ thống thần kinh của sinh vật, trong đó một mô hình toán học được tạo ra giống như bộ não để xử lý thông tin. ANN giống như con người, được học bởi kinh nghiệm, lưu những kinh nghiệm hiểu biết và sử dụng trong những tình huống phù hợp

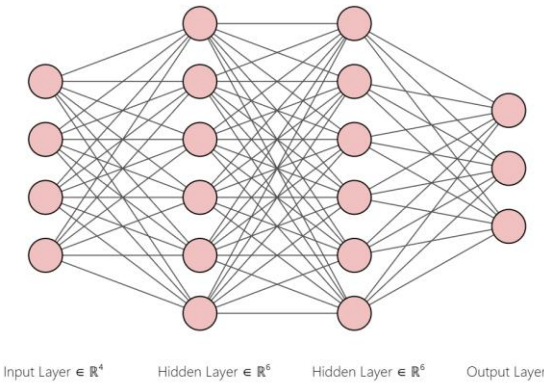
Mạng nơ ron nhân tạo là một sự tập hợp phức tạp của các nơ ron nhân tạo, cấu trúc chung của một nơ ron nhân tạo được hiển thị trên hình 1.6:



Hình 1.6 Cấu trúc của một nơ ron nhân tạo

Nơron này sẽ hoạt động như sau: giả sử có N dữ liệu đầu vào (inputs), nơron sẽ có N trọng số (weights) tương ứng với N đường truyền đầu vào. Nơron sẽ lấy giá trị đầu vào thứ nhất, nhân với trọng số trên đường vào thứ nhất, lấy giá trị đầu vào thứ hai nhân với trọng số của đường vào thứ hai v.v..., rồi lấy tổng của tất cả các kết quả thu được. Đường truyền nào có trọng số càng lớn thì tín hiệu truyền qua đó càng lớn, như vậy có thể xem trọng số là đại lượng tương đương với synapse trong nơron sinh học, hàm y tương đương với axon. Nếu tổng này lớn hơn một ngưỡng giá trị nào đó thì đầu ra của nơron sẽ ở mức tích cực .

Mạng nơ ron cơ bản thường có cấu trúc được hiển thị trên hình 1.7:



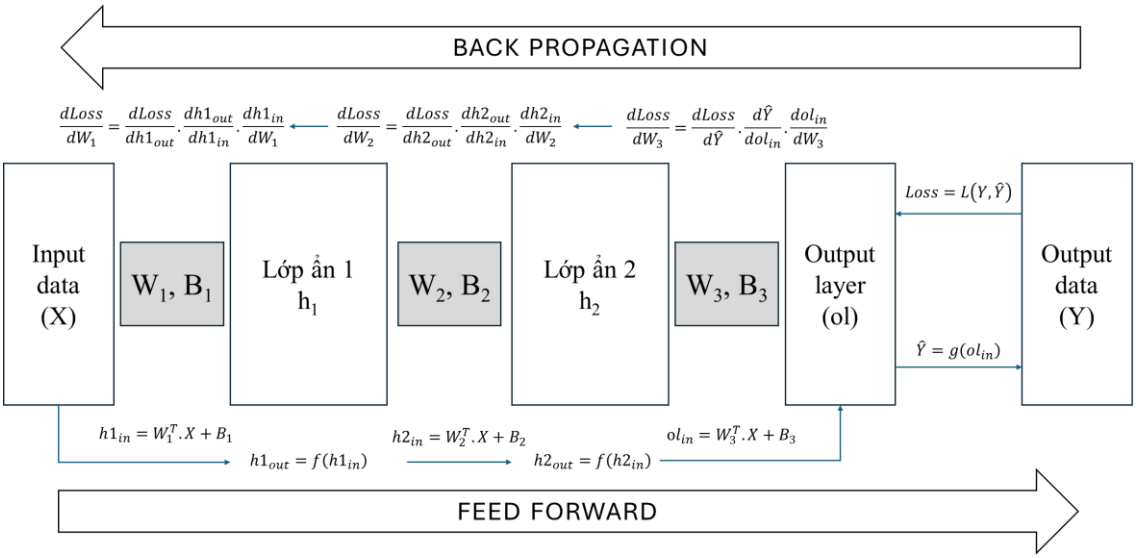
Hình 1.7 Cấu trúc phổ biến của mạng nơ ron nhân tạo (ANN)

Trong đó lớp bên trái ngoài cùng được gọi là lớp đầu vào (Input layer) có số nơ ron – nodes tương ứng với số lượng đặc trưng của dữ liệu đầu vào, các lớp ở giữa được gọi là các lớp ẩn (hidden layer) sẽ có nhiệm vụ học sự phức tạp của dữ liệu (hình thái dữ liệu, tính tuyến tính, tính phi tuyến tính). Lớp bên phải ngoài cùng được gọi là lớp đầu ra hay lớp xuất (output layer), số node của lớp ngoài cùng tương ứng với số giá trị mà mô hình tính toán ra. Có thể hiểu rằng các đường nối giữa các nodes với nhau tương ứng với một giá trị trọng số và một giá trị trị số, việc dữ liệu đầu vào đi qua mô hình chính là việc mô hình tính toán giá trị

$$layer\ output = f(nodes\ output) = f(w.x + b)$$

Với f ở đây chính là hàm kích hoạt, một số hàm kích hoạt thường được sử dụng bao gồm linear, sigmoid, tanh, relu,... Trong đó phổ biến nhất hiện nay chính là hàm relu, do khả năng tính toán nhanh nhưng vẫn có thể bảo đảm khả năng học sự phi tuyến tính của dữ liệu. Các hàm kích hoạt sẽ đứng ở lớp của các lớp ẩn để áp dụng sau khi dữ liệu được đưa đến các lớp ẩn. Ở lớp ngoài cùng, hàm kích hoạt sẽ phụ thuộc vào bản chất bài toán mà đang xử lý (bài toán hồi quy f sẽ là hàm linear, bài toán phân loại nhị phân sẽ là sigmoid với một nodes ở lớp ngoài cùng, bài toán phân loại đa lớp sẽ là hàm softmax).

Do có rất nhiều đường nối giữa các node của các lớp với nhau nên các giá trị w và b thường được tổng hợp lại và xử lý dưới dạng ma trận W và B. Tiến trình dữ liệu đi qua mô hình ANN được thể hiện qua hình 1.8 như sau:



Hình 1.8 Quy trình học tập dữ liệu của thuật ANN

Mô hình ANN sẽ có 2 quá trình chính: quá trình thứ nhất là tính toán ra $\hat{Y}$ là quá trình áp dụng các phép nhân ma trận và hàm kích hoạt, dữ liệu được truyền qua các lớp từ trái sang phải (feed forward); quá trình thứ 2 là học và cập nhật các giá trị W và B của từng lớp bằng cách sử dụng chain rule dựa vào đạo hàm của giá trị mất mát (hàm mất mát tùy thuộc vào hàm kích hoạt ở lớp ngoài cùng) – goi là lan truyền ngược (backpropagation).

Các bước triển khai

Bước 1: Cấu hình mạng ANN (số lớp ẩn, số nơ ron mỗi lớp ẩn, hàm kích hoạt, hệ số học máy, số bước học, vv...). Khi mạng được tạo thì giá trị các W và B ở mỗi lớp được cho một cách ngẫu nhiên.

Bước 2: Tính toán $\hat{Y}$ bằng feed forward, tình giá trị mất mát giữa Y và $\hat{Y}$ bằng hàm mất mát

Bước 3: Tính giá trị đạo hàm của mất mát theo W và B mỗi lớp bằng chain rule

Bước 4: Cập nhật các giá trị W và B mỗi lớp bằng thuật toán tối ưu hóa (gradient descent, adam, RMSprop, genetic algorithm, ...)

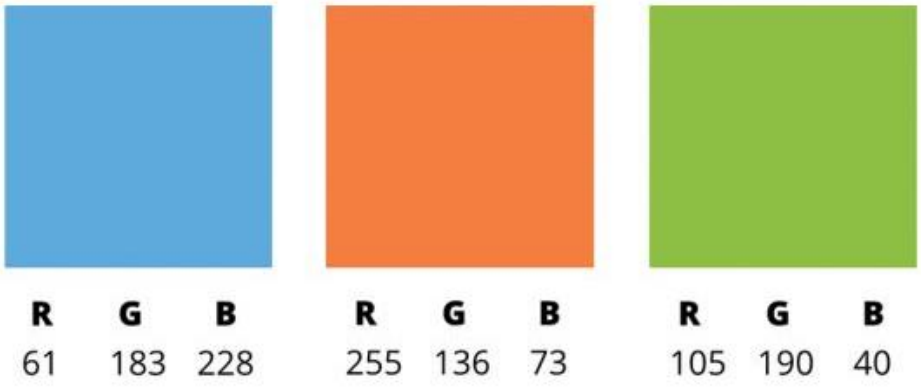
Bước 5: Với giá trị W và B mới cập nhật, lặp lại bước 2 cho tới khi hết số bước học.

Bước 6: Sử đụng mô hình để hồi quy/phân loại dữ liệu trong tập test.

1.2.2 Các thuật toán học sâu

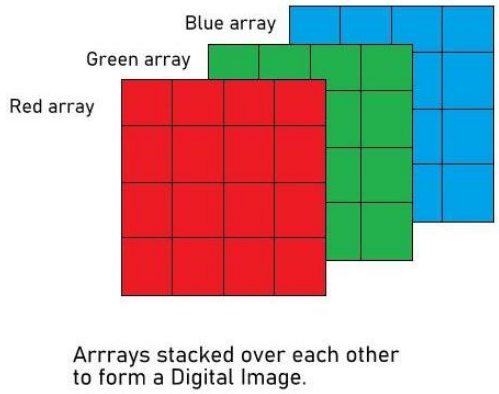
1.2.2.1 Tensor dữ liệu

Về dữ liệu dạng ảnh, một bức ảnh là một tập hợp của các điểm ảnh (pixels). Với một bức ảnh màu, mỗi pixel sẽ tương ứng với một tập hợp gồm 3 phần tử, mỗi phần tử đại diện cho một giá trị kênh màu R, G, B. Các giá trị này dao động trong khoảng từ 0 đến 255. Với một bức ảnh xám – grayscale, hay có tên thân thuộc hơn là ảnh đen trắng, mỗi pixel sẽ chỉ có một giá trị trong khoảng 0-255 (càng về 0 thì càng đen, càng gần 255 thì càng trắng). Ví dụ về một giá giá trị RGB trong ảnh được biểu thị trên hình 1.9:



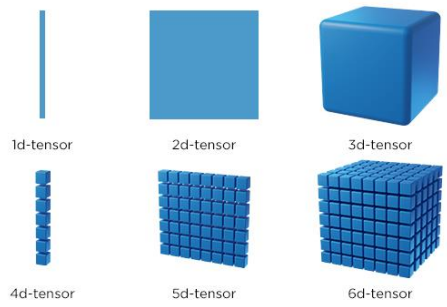
Hình 1.9 Một số giá trị RGB của pixel ảnh

Với mỗi kênh màu của một bức ảnh có kích cỡ mxn sẽ tương ứng với một ma trận số có kích cỡ mxn, như vậy để một bức ảnh có thể hiện thị hết các màu sẽ là một sự chồng chéo của 3 ma trận mxn (mỗi ma trận tương ứng với một kênh màu). Dữ liệu ảnh gồm ba kênh màu được biểu thị ở hình 1.10:



Hình 1.10 Dữ liệu ảnh ba chiều gồm ba kênh màu

Tensor là một dạng biểu diễn của dữ liệu nhiều chiều. Một ma trận được coi như là một tensor hai chiều, là sự sắp đặt liền kề của các vector – tương ứng với tensor một chiều. Khi xếp nhiều ma trận lên nhau thì lúc đó dữ liệu sẽ có dạng giống như hình hộp chữ nhật, tương ứng với một tensor 3 chiều. Một số dạng tensor cơ bản được thể hiện trong hình 1.11:

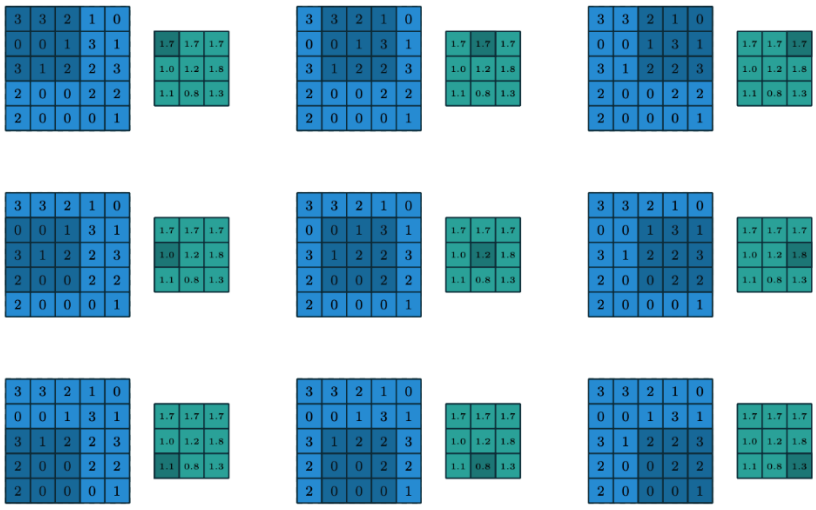


Hình 1.11 Một số dạng tensor cơ bản

1.2.2.2 Mạng tích chập hai chiều (2D-CNN)

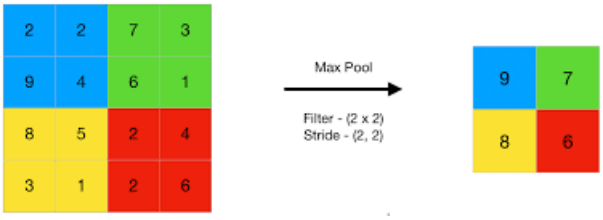
Nếu sử dụng một tập các bức ảnh khác nhau, sau đó dàn trải ra thành vector thì mỗi đặc trưng của vector ảnh thực chất không giống với một vector, do mỗi ảnh sẽ có pixel màu tại các vị trí khác nhau nên việc coi mỗi pixel tương ứng với một giá trị đặc trưng là không thực tế. Tuy nhiên, nếu nhìn vào một bức ảnh, màu của những pixel cạnh nhau thường gần giống nhau, nếu lấy một pixel bất kỳ tại một tọa độ nào đó, thì những pixel láng giềng của nó sẽ có màu sắc khá tương đồng với pixel đã chọn. Điều đó cho thấy rằng các đặc trưng thực sự của một bức ảnh sẽ không có tính liền kề như đối với vector mà thường có tính không gian rời rạc. Bắt nguồn từ ý tưởng đó, thuật toán tích chập ra đời.

Nếu có một cửa sổ trượt (ma trận) trượt qua từng pixel của bức ảnh, sau đó nhân từng giá trị của cửa sổ trượt với giá trị tương ứng của bức ảnh xong cộng lại hết với nhau, sẽ chuyển được dữ liệu của bức ảnh cũ sang một bức ảnh mới, phép tích chập hai chiều trên ảnh được mô tả trên hình 1.12:



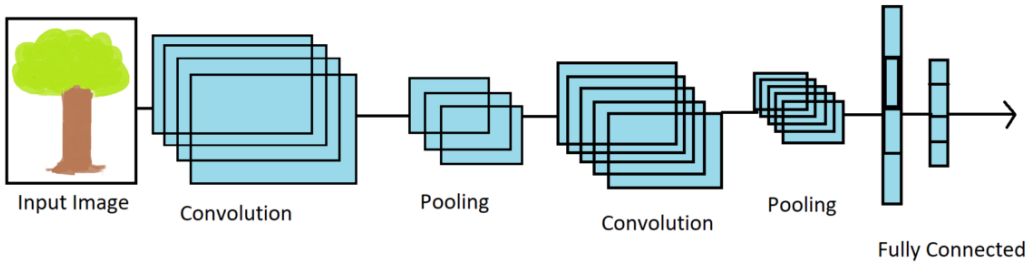
Hình 1.12 Mô hình hóa phép tích chập trên một ma trận

Thực chất, việc sử dụng cửa số trượt (thường được gọi là kernel) là nhằm trích xuất các đặc trưng của bức ảnh – chính là việc xử lý các đặc trưng không gian đã được nói ở trên. Ngoài ra còn một kĩ thuật nữa được gọi là pooling, tức là lấy pixel có giá trị cao nhất trong một nhóm pixel ra nhằm giảm số lượng giữ liệu mà vẫn giữ được thông tin quan trọng của bức ảnh, hình 1.13 mô tả phép max-pooling trên một ma trận:



Hình 1.13 Mô hình hóa phép pooling trên một ma trận

Một mô hình mạng nơ ron tích chập sẽ được cấu thành từ 2 phần, phần đầu tiên sẽ là phần xuất đặc trưng (bao gồm các lớp tích chập và các lớp pooling chồng lên nhau), phần thứ hai là phần mạng nơ rơn nhân tạo (Fully connected layer), hai phần này được nối với bằng bằng một flatten layer (Lớp dàn trải tensor ảnh về dạng vector).



Hình 1.14 Cấu trúc mô hình mạng tích chập với dữ liệu đầu vào là ảnh

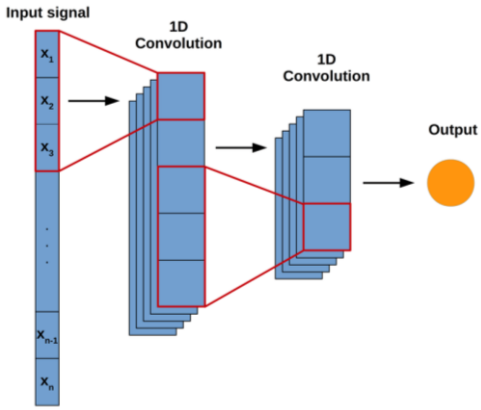
Ở đầu ra của các lớp convolution hay pooling cũng có các hàm kích hoạt tương tự như mô hình ANN. Đồng thời các kernel ban đầu cũng đều được tạo ra ngẫu nhiên, sau đó được học bởi mô hình sử dụng các thuật toán tối ưu như Gradient descent, Adam, vv...

Nói chung, với dữ liệu dạng ảnh, đầu tiên mô hình CNN sẽ sử dụng các lớp tích chập nhằm trích xuất các đặp điểm của bức ảnh như độ sáng, màu sắc, đường nét, vv... Sau đó sử dụng lớp pooling để dữ lại các giá trị quan trọng nhất, từ đó có thể đưa các đặc trưng có tính chất không gian về các giá trị trong vector rồi phân loại [21].

Một điểm yếu của CNN nói chung và các thuật toán học sâu nói riêng đó là số lượng tham số của thuật toán là rất lớn, do vậy cần rất nhiều mẫu để thuật toán có thể học được tập dữ liệu, nếu mô hình trở nên quá phức tạp so với dữ liệu thì rất dễ gây ra hiện tượng overfitting

1.2.2.3 Mạng tích chập một chiều (1D-CNN)

Trong hóa học nói chung và hóa phân tích nói riêng, dữ liệu phổ thường có sự phức tạp khi có số lượng đặc trưng lớn, đồng thời do việc khó khăn trong thực nghiệm và đo mẫu nên số lượng mẫu dữ liệu thường không lớn, trong khi với những bài toán phân loại ảnh bằng CNN thì thường yêu cầu dữ liệu lên tới hàng nghĩn bức ảnh, từ đó một mô hình mới dựa trên mô hình CNN gọi là 1D-CNN. Cấu trúc mô hình 1D-CNN được thể hiện trong hình 1.15 như sau:



Hình 1.15 Cấu trúc mô hình mạng tích chập một chiều

Mô hình 1D CNN hoạt động với dữ liệu dạng tín hiệu. Thông thường với mô hình CNN thì dữ liệu đầu vào của mỗi mẫu là 2D, tương ứng với dạng của các kernel cũng là 2D. Tuy nhiên với mô hình 1D-CNN thì dữ liệu đầu vào sẽ là vector, đó là lý do gọi mô hình là 1D-CNN. Các lớp convolution và pooling tương đối giống như CNN, chí có điều nó hoạt động với dữ liệu một chiều, do đó lượng tham số được giảm bớt đi khá nhiều so với mạng CNN cơ bản, khiến cho mô hình 1D-CNN vẫn hoạt động tốt với số lượng dữ liệu ít mà vẫn học được độ phức tạp của dữ liệu do cơ chế trích xuất đặc trưng [18].

Các bước triển khai

Bước 1: Cấu hình mô hình CNN

Bước 2: Tính toán $\hat{Y}$ bằng feed forward, tình giá trị mất mát giữa Y và $\hat{Y}$ bằng hàm mất mát

Bước 3: Cập nhật các giá trị trọng số và trị số, các kernel (còn gọi là K) bằng các thuật toán tối ưu

Bước 4: Với các giá trị tham số mới, quay lại lặp lại bước 2 và bước 3 cho tới hết số bước học

Bước 5: Sử đụng mô hình để hồi quy/phân loại dữ liệu trong tập test.

1.3 Các phương pháp tiền xử lý dữ liệu

1.3.1 Điều chỉnh tỷ lệ (Minmax Scaling)

Để giúp cho các thuật toán học máy và học sâu hội tụ nhanh hơn, dữ liệu đầu vào thường chỉ giao động trong một khoảng giá trị chung, thường là (-1,1) hoặc (0,1) sử dụng công thức sau cho từng cột dữ liệu:

$$X_{new} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}}$$

1.3.2 Bình thường hóa dữ liệu (Standard Scaling)

Không chỉ thay đổi khoảng giá trị, việc bình thường hóa dữ liệu còn khiến cho dữ liệu được biến đổi sao cho gần giống nhất với một phân phối chuẩn, việc này khiến các thuật toán hồi quy và phân loại hoạt động tốt hơn.

$$X_{new} = \frac{X_{old} - X_{mean}}{\sigma}$$

1.3.3 Làm mịn dữ liệu bằng Savitzky-Golay

Thuật toán Savitzky-Golay là một kỹ thuật lọc tín hiệu số được sử dụng để làm mịn dữ liệu và bảo toàn các đặc tính quan trọng như đỉnh và độ rộng của các tín hiệu trong khi giảm nhiễu. Phương pháp này đặc biệt hữu ích trong các ứng dụng phân tích phổ và xử lý tín hiệu nơi mà việc bảo toàn các đặc tính của tín hiệu gốc là quan trọng.

Phương pháp này sử dụng kỹ thuật nội suy đa thức trong một cửa sổ trượt để làm mịn dữ liệu. Để thực hiện việc này, một đa thức bậc thấp được khớp với các điểm dữ liệu trong một khoảng cửa sổ di chuyển qua toàn bộ tập dữ liệu. Giá trị trung bình của các điểm dữ liệu trong khoảng cửa sổ này được tính toán và thay thế giá trị trung tâm của cửa sổ đó [14].

Công thức ngắn gọn

Với tập dữ liệu có dạng $y(i)$ với $i = 1,2,3 \dots N$. Đầu tiên cần :

- Chọn bậc của đa thức (p): Đây là bậc của đa thức được sử dụng để khớp dữ liệu trong mỗi cửa sổ.
- Chọn kích thước cửa sổ (2m+1): Đây là số lượng điểm dữ liệu trong mỗi cửa sổ, với m là số lượng điểm dữ liệu ở mỗi phía của điểm trung tâm.

Công thức tổng quát của bộ lọc Savitzky-Golay là:

$$y'(i) = \sum_{j=-m}^{m} C_j \, . \, y(i+j)$$

Trong đó:

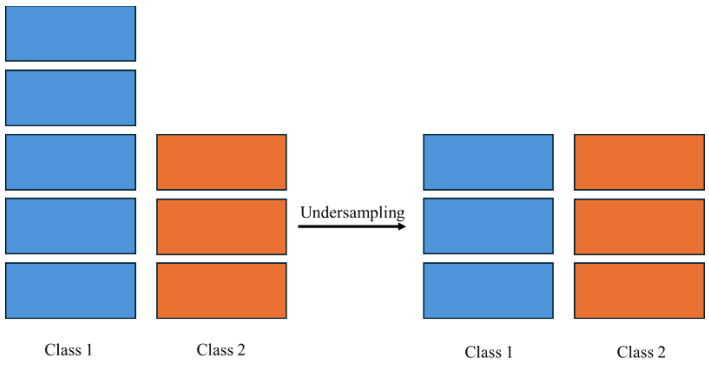
- $y'(i)$ là giá trị min tại vị trí i
- $y(i+j)$ là giá trị ban đầu tại vị trí i+j
- C_j là hệ số Savitzky-Golay được tính toán trên bậc của đa thức p và kích thước cửa sổ $2m+1$

1.3.4 Thuật toán cân bằng dữ liệu

Một vấn đề thường gặp trong các bài toán phân loại đó là mất cân bằng dữ liệu, đó là khi một lớp dữ liệu có số lượng mẫu lớn hơn rất nhiều so với các lớp dữ liệu. Điều này gây ra việc các mô hình phân loại sẽ có xu hướng thiên vị hơn cho lớp dữ liệu có nhiều mẫu hơn, khiến cho các mô hình trở nên không khách quan trong việc phân loại [17]. Các hướng giải quyết phổ thông nhất thường sẽ cố làm cho dữ liệu ở các lớp trở nên đồng đều hơn, sử dụng các thuật toán cân bằng dữ liệu.

1.3.4.1 Downsampling dữ liệu

Downsampling là kỹ thuật khiến cho các lớp dữ liệu có số mẫu đồng đều nhau bằng cách cắt giảm số mẫu ở các lớp có số mẫu lớn hơn một cách ngẫu nhiên. Mô hình hóa thuật toán downsampling dữ liệu được thể hiện trên hình 1.16:



Hình 1.16. Thuật toán downsampling

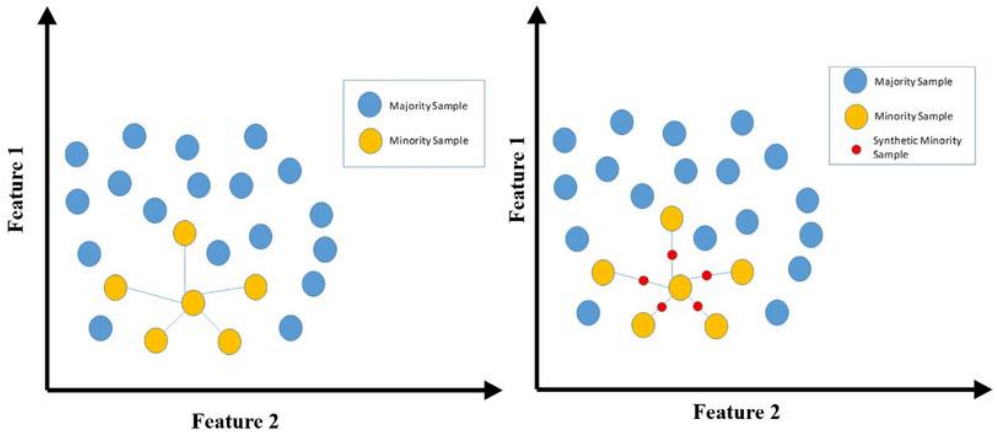
Việc cắt giảm dữ liệu ở các lớp có số mẫu lớn hơn khiến cho dữ liệu đồng đều hơn, nhưng đồng thời cũng có thể làm vô tình mất đi các mẫu quan trọng, nên việc sử dụng thuật toán undersampling thường được áp dụng khi toàn bộ tập dữ liệu có số mẫu khá lớn.

1.3.4.2 Upsampling dữ liệu bằng SMOTE

Ngược lại với downsampling, các thuật toán upsampling sẽ cố gắng tạo ra các điểm dữ liệu mới bằng các phương pháp nội suy dựa trên dữ liệu đã có.

Một trong các thuật toán nổi tiếng nhất của kĩ thuật này có tên là SMOTE - Synthetic Minority Over-sampling Technique. SMOTE hoạt động bằng cách tạo các mẫu tổng hợp dọc theo các đường nối các lân cận gần nhất trong không gian đặc trưng. Ý tưởng cơ bản đằng sau SMOTE là tạo các mẫu lớp thiểu số mới bằng cách thực hiện các bước nhỏ từ một trong các mẫu lớp thiểu số đến một trong k hàng xóm gần nhất của nó trong không gian đặc trưng, trong đó k là tham số của thuật toán.một trong các mẫu lớp thiểu số đến một trong k hàng xóm gần nhất của nó trong không gian đặc trưng, trong đó k là tham số của thuật toán [13].

Thuật toán tạo ra một mẫu mới bằng cách chọn ngẫu nhiên một trong k hàng xóm gần nhất và sau đó thêm một nhiễu loạn nhỏ vào vectơ đặc trưng giữa mẫu ban đầu và hàng xóm được chọn. Điều này tạo ra dữ liệu tổng hợp mới tương tự như các mẫu lớp thiểu số trong không gian đặc trưng nhưng không phải là bản sao chính xác của bất kỳ mẫu hiện có nào. Cách tạo ra dữ liệu mới được hiển thị ở hình 1.17:



Hình 1.17 Thuật toán SMOTE

1.4 Phương pháp đánh giá độ chính xác mô hình hồi quy

1.4.1 Các phép đo hồi quy

- Giá trị MSE: $MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$
  - y_i: giá trị mẫu thứ i thực thế
  - $\hat{y}_i$: giá trị mẫu thứ i được dự đoán
  - n: số mẫu

Giá trị MSE tính toán sự sai khác trung bình giữa giá trị dự đoán và giá trị thực tế, giá trị MSE càng gần 0 chứng tỏ mô hình dự đoán càng chính xác, MSE còn được ứng dụng là hàm mục tiêu của các thuật toán học máy và học sâu, khi mục đích cuối cùng của các thuật toán hồi quy là giảm thiểu giá trị MSE tới nhỏ nhất.

- Giá trị RMSE: $RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$

Cũng giống như giá trị MSE, RMSE cũng tính toán trung bình sai khác của giá trị dự đoán và giá trị thực tế, tuy nhiên do đã khử được bậc hai nên sai số tính toán không bị phóng đại, điều đó giúp cho thuật toán RMSE đáng tin cậy hơn khi biểu thị sai số.

- Hệ số xác định R^2: $R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}$
  - $\overline{y}$: giá trị trung bình của biến mục tiêu

Hệ số xác định R^2 cho biết tỉ lệ phương sai của biến mục tiêu được giải thích bởi các biến độc lập trong mô hình hồi quy tuyến tính. Giá trị của hệ số xác định nằm trong khoảng từ 0 đến 1. Nếu giá trị của hệ số xác định gần bằng 1, tức là mô hình hồi quy tuyến tính giải thích được một phần lớn sự biến thiên của biến mục tiêu. Trong trường hợp giá trị của hệ số xác định gần bằng 0, mô hình hồi quy tuyến tính không giải thích được sự biến thiên của biến mục tiêu và cho thấy mô hình không phù hợp.

- Phần trăm sai số: $\%Error = \left|\frac{\hat{y} - y}{y}\right| . 100\%$

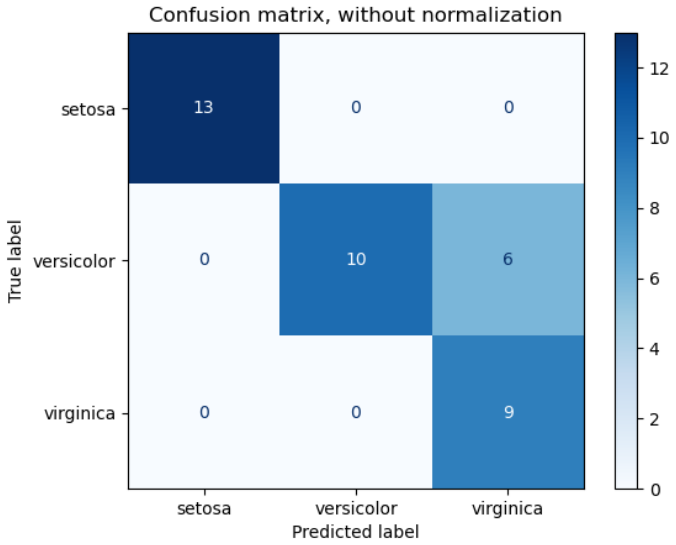
Giá trị phần trăm sai số đo đạc sai số tương đối giữa giá trị dự đoán và giá trị thực thế, không như MSE hay RMSE thì giá trị phần trăm sai số được tính trên từng giá trị thay vì lấy trung bình cộng. Phần trăm sai số càng gần 0 thì sai số càng thấp chứng tỏ độ chính xác của mô hình càng cao.

1.4.2 Các phép đo phân loại

- Độ chính xác: Độ $chính\ xác = \frac{Số\ mẫu\ đoán\ đúng}{Tổng\ số\ mẫu\ dự\ đoán}$

Độ chính xác cho biết tỉ lệ số mẫu đoãn đúng của mô hình trên toàn bộ dữ liệu, giá trị của độ chính xác càng gần 1 chứng tỏ mô hình phân loại tốt, càng gần 0 thì ngược lại

- Ma trận nhầm lẫn (confusion matrix): là một bảng dùng để đánh giá hiệu suất của mô hình phân loại, cho biết mức độ chính xác của các dự đoán. Bảng này so sánh giữa các dự đoán của mô hình và kết quả thực tế, giúp xác định số lần mô hình dự đoán đúng hay sai cho từng loại cụ thể. Trục tung của ma trận đại diện cho các giá trị thực tế, trong khi trục hoành của ma trận đại diện cho các giá trị dự đoán. Ví dụ về ma trận nhầm lẫn được biểu thị ở hình 1.18 sau:



Hình 1.18 Ma trận nhầm lẫn của phép phân loại ba lớp

Các giá trị nằm trên đường chéo chính của ma trận nhầm lẫn chính là số mẫu dự đoán đúng của các lớp phân loại, trong khi đó tất cả các giá trị nằm lệch khỏi đường chéo đều là số mẫu dự đoán sai.

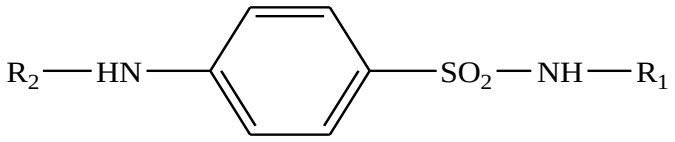
1.5 Ứng dụng học máy và và học sâu trong định lượng các chất kháng sinh trong thuốc

1.5.1 Tổng quan về các nhóm thuốc kháng sinh nghiên cứu

1.5.1.1 Nhóm thuốc kháng sinh Sulfamid

Cấu tạo chung của nhóm Sulfamid

Các sulfamid kháng khuẩn là dẫn chất của p- aminobenzensulfonamid, có công thức cấu tạo chung là:



Hình 1.19 Công thức cấu tạo chung của nhóm Sulfamid

Trong đó thường gặp R2 là H, và cũng chỉ khi R2 là H thì sulfamid mới có hoạt tính kháng khuẩn, khi R2 ≠H, thì chất đó là tiền thuốc. R1 có thể là mạch thẳng, dị vòng. Tuy nhiên, nếu R1 là dị vòng thì hiệu lực kháng khuẩn mạnh hơn, thông thường là các dị vòng 2 – 3 dị tố. Khi R1 và R2 đều là gốc hidro thì thu được sulfamid là có cấu tạo đơn giản nhất (sulfanilamid) [22].

Công thức cấu tạo của một số sulfamid phổ biến được hiển thị trong phụ lục đi kèm.

Tính chất vật lý và hóa học của nhóm thuốc Sulfamid

- Tính chất vật lý

Sulfamid ở dạng tinh thể màu trắng hoặc màu vàng nhạt trừ prontosil, không mùi, thường ít tan trong nước, benzen, chloroform. Sulfamid tan trong dung dịch acid vô cơ loãng và hydroxyd kiềm (trừ sulfaguanidin) [22]. Các sulfamid có các thông số xác định về: độ chảy, phổ IR, phổ UV (do có chứa nhân thơm).

- Tính chất hóa học

Hầu hết các Sulfamid đều có tính chất lưỡng tính [22]:

- Tính acid (trừ sulfaguanidin): do có H ở N- amid linh động
- Tính bazơ: Có tính kiềm do có nhóm amin thơm tự do, nên tan trong dung dịch acid.

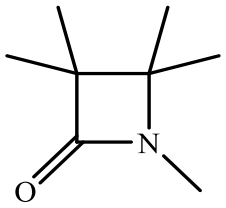
Các sulfamid có thể tham gia phản ứng diazo hoá do có nhóm amin thơm tự do (có thể tham gia phản ứng ghép đôi với 2-naphtol/kiềm để cho sản phẩm màu đỏ da cam).

Tác dụng với một số muối kim loại (CuSO4, CoCl2) tạo thành phức màu tủa với Cu2+, Co2+ đặc trưng cho từng sulfamid, nên thường được dùng để phân biệt các sulfamid với nhau. Đốt khô trong ống nghiệm, sulfamid bị phân huỷ, để lại cặn có màu điển hình cho từng sulfamid, ví dụ, đốt sulfanilamid sẽ giải phóng amoniac và cho cặn màu xanh tím.

1.5.1.2 Nhóm thuốc kháng sinh họ β- lactam

Cấu tạo chung của nhóm β- lactam

Các kháng sinh mà phân tử có cấu trúc azetidin-2-on (vòng β-lactam)- một amid vòng 4 cạnh. Gồm các nhóm : penicillin, cephalosporin, monobactam, cacbapenem. Trong đó 2 nhóm sử dụng phổ biến và lớn nhất là penicillin và cephalosporin [11]. Công thức cấu tạo của nhóm Azetidin-2-on được thể hiện ở hình 1.20:

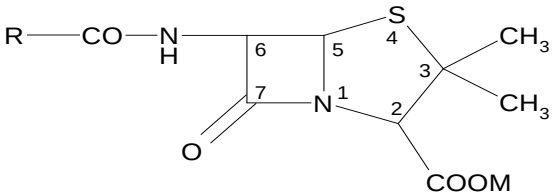


azetidin-2-on

Hình 1.20 Công thức cấu tạo của nhóm Azetidin-2-on (beta-lactam)

Nhóm các penicillin:

Các penicillin đều có cấu trúc cơ bản gồm 2 vòng: vòng thiazolidin, vòng β-lactam [5]. Công thức cấu tạo các kháng sinh penicillin được thể hiện trong hình 1.21:



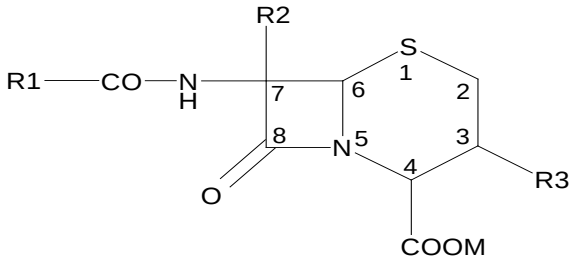
Hình 1.21 Công thức cấu tạo các kháng sinh penicillin

Dựa vào khổ kháng khuẩn, chia các cephalosporin thành 4 thế hệ. Các cephalosporin thế hệ trước tác dụng trên vi khuẩn gram dương mạnh hơn, nhưng trên gram âm yếu hơn thế hệ sau.

Công thức cấu tạo của một số penicillin được hiểu thị trong phụ lục đi kèm.

Nhóm các cephalosporin.

Các cephalosporin cấu trúc chung gồm 2 vòng: vòng β-Lactam 4 cạnh gắn với 1 dị vòng 6 cạnh, những cacbon bất đối có cấu hình 6R, 7R. Công thức cấu tạo kháng sinh cephalosporin được thể hiện trong hình 1.22:



Hình 1.22 Công thức cấu tạo kháng sinh cephalosporin

Dựa vào khổ kháng khuẩn, chia các cephalosporin thành 4 thế hệ. Các cephalosporin thế hệ trước tác dụng trên vi khuẩn gram dương mạnh hơn, nhưng trên gram âm yếu hơn thế hệ sau. Công thức cấu tạo của các hoạt chất trong nhóm cephalosporin được hiểu thị trong phụ lục đi kèm.

Tính chất vật lý và hóa học của nhóm thuốc β- lactam

- Tính chất vật lý:

Các β-lactam thường ở dạng bột kết tinh màu trắng, dạng axit ít tan trong nước, dạng muối natri và kali dễ tan; tan được trong metanol và một số dung môi hữu cơ phân cực vừa phải. Tan trong dung dịch axit và kiềm loãng do đa phần chứa đồng thời nhóm –COOH và –NH2 [11].

Cực đại hấp thụ chủ yếu do nhân phenyl, tùy vào cấu trúc khác làm dạng phổ thay đổi (đỉnh phụ, vai, sự dịch chuyển sang bước sóng ngắn hoặc dài, giảm độ hấp thụ).

- Tính chất hóa học:

Các β-lactam là các axit với nhóm –COOH có pKa= 2.5-2.8 tùy vào cấu trúc phân tử. Trong môi trường axit, kiềm, β-lactamase có tác dụng phân cắt khung phân tử, mở vòng β-lactam làm kháng sinh mất tác dụng.

Các penicillin là các acid khó tan trong nước, dạng muối natri hoặc kali dễ tan dùng để pha thuốc tiêm.

Các cephalosporin có vòng β-lactam đều kém bền do cộng hưởng amid không tồn tại. Cộng hưởng amid mất đi chủ yếu còn do vòng cộng hưởng "en-amin". Cộng hưởng này mạnh lên khi R3 hút điện tử.

So với các penicillin thì các cephalosporin bền với acid hơn, tuy nhiên do cộng hưởng "en-amin" mà phản ứng cộng hợp ái điện tử vẫn xảy ra tại trung tâm(-), kéo theo cắt đứt đường nối đôi amid. Đó là quá trình thủy phân acid, chất đầu tiên tạo thành là acid cephalosporic.

1.5.2. Một số ứng dụng của các mô hình học máy trong phân tích kháng sinh

Trong lĩnh vực y khoa, học máy đã mở ra những khả năng mới trong việc phân tích và phát triển các loại kháng sinh. Các mô hình học máy có thể được huấn luyện để nhận diện các cấu trúc phân tử có tiềm năng làm kháng sinh mới, giúp rút ngắn thời gian nghiên cứu và phát triển. Ví dụ, một nghiên cứu đã sử dụng mô hình học sâu để phân tích cơ sở dữ liệu lớn về các hợp chất hóa học, từ đó tìm ra các hợp chất mới có khả năng chống lại vi khuẩn kháng thuốc. Mô hình này không chỉ giúp phát hiện các kháng sinh tiềm năng mà còn có thể dự đoán độc tính và tác dụng phụ của chúng, qua đó hỗ trợ quá trình thử nghiệm lâm sàng. M.R. Hormozi-Nezhad và cộng sự đã đề xuất một chiến lược mới dựa trên sử dụng AuNP ở các điều kiện đệm khác nhau làm thành phần cảm biến màu nhằm phân tích phát hiện nhiều loại kháng sinh trong môi trường nước [15]. Hay một phương pháp mới dựa trên quang phổ FTIR kết hợp với các thuật toán học máy giúp xác định các chủng đa kháng thuốc gây ra từ các loài E. coli [20]. Ngoài ra, học máy còn được áp dụng trong việc tối ưu hóa liều lượng kháng sinh, nhằm đạt hiệu quả điều trị cao nhất mà vẫn đảm bảo an toàn cho bệnh nhân. Các thuật toán có thể phân tích dữ liệu bệnh án để đề xuất liều lượng phù hợp dựa trên đặc điểm cá nhân của từng bệnh nhân, từ đó giảm thiểu nguy cơ phát triển kháng thuốc. Sự kết hợp giữa học máy và y học chắc

chắn sẽ tiếp tục mở ra những cơ hội mới trong việc chống lại các bệnh nhiễm trùng và cải thiện sức khỏe cộng đồng.

1.6. Ứng dụng học máy và học sâu trong phân tích thực phẩm

1.6.1. Tổng quan về quả xoài và phân loại xoài

Quả xoài (*Mangifera indica*) là một loại trái cây nhiệt đới được mệnh danh là "vua của các loại trái cây", là một trong những loại trái cây nhiệt đới phổ biến và được yêu thích nhất trên thế giới. Xoài có nguồn gốc từ Nam Á, đặc biệt là từ khu vực Ấn Độ và Myanmar, nhưng ngày nay được trồng rộng rãi ở các khu vực nhiệt đới và cận nhiệt đới trên toàn cầu như Mỹ Latinh, Châu Phi, và Đông Nam Á. Với hình dạng bầu dục, màu sắc từ xanh lục khi còn non đến vàng cam rực rỡ khi chín, xoài không chỉ hấp dẫn về mặt thẩm mỹ mà còn nổi bật với hương vị ngọt ngào, thơm lừng và mọng nước.

Nhu cầu sử dụng xoài ngày càng tăng không chỉ do hương vị thơm ngon mà còn vì những lợi ích sức khỏe mà nó mang lại. Xoài có thể được tiêu thụ tươi sống như một món tráng miệng ngon lành, hay chế biến thành nhiều sản phẩm khác nhau như nước ép, sinh tố, kem, salad, mứt, và các món ăn chín. Ở nhiều quốc gia, xoài còn được dùng để làm gia vị trong các món ăn truyền thống, như món chutney ở Ấn Độ hay món gỏi xoài ở Đông Nam Á.

Thành phần hóa học của xoài

Quả xoài có nhiều thành phần hóa học đa dạng, mỗi thành phần đóng một vai trò quan trọng trong việc cung cấp giá trị dinh dưỡng và các lợi ích sức khỏe. Việc phân tích hàm lượng các thành phần này không chỉ giúp hiểu rõ hơn về giá trị dinh dưỡng của quả xoài mà còn tạo cơ sở cho các ứng dụng trong y học và công nghiệp thực phẩm.

Các thành phần của xoài có thể được chia thành ba nhóm chính: chất dinh dưỡng đa lượng (carbohydrate, protein, axit amin, lipid, chất béo và axit hữu cơ), vi chất dinh dưỡng (vitamin và khoáng chất) và chất phytochemical (hợp chất phenolic, polyphenol, sắc tố và các thành phần dễ bay hơi). Quả xoài cũng chứa carbohydrate cấu trúc như pectin và cellulose. Các axit amin chủ yếu gồm lysine, leucine, cysteine, valine, arginine, phenylalanine và methionine. Hàm lượng lipid tăng lên trong quá trình chín, đặc biệt là axit béo omega-3 và omega-6. Các sắc tố quan trọng nhất của quả xoài gồm diệp lục (a và b) và carotenoid. Các axit hữu cơ chủ yếu như axit malic và axit citric tạo nên tính

axit đặc trưng của trái cây. Các hợp chất dễ bay hơi, với những chức năng hóa học đa dạng, góp phần tạo nên hương thơm đặc trưng của xoài [19].

Hàm lượng các thành phần hóa học trong quả xoài có thể thay đổi dựa trên nhiều yếu tố như giống, điều kiện trồng trọt, giai đoạn chín và điều kiện bảo quản. Sự đa dạng và phong phú của các thành phần hóa học này không chỉ làm cho quả xoài trở thành một nguồn dinh dưỡng quý giá mà còn mang lại nhiều lợi ích sức khỏe cho con người. Việc nghiên cứu và phân tích các thành phần này không chỉ giúp tối ưu hóa việc sử dụng quả xoài trong dinh dưỡng và y học mà còn mở ra các ứng dụng mới trong công nghệ thực phẩm và dược phẩm.

Một số nghiên cứu về phân loại xoài

Các nghiên cứu về phân loại xoài đã được tiến hành rộng rãi trên ở Việt Nam và trên thế giới. Các đặc trưng vật lý của xoài như màu sắc, khối lượng , kích thước, hình dáng và mật độ quả được nghiên cứu phân loại bằng việc ứng dụng thị giác máy tính kết hợp với các thuật toán trí tuệ nhân tạo, dữ liệu ảnh trong thời gian thực được thu bằng các máy ảnh CCD và các sặc trưng vật lý được xác định là thông tin quan tọng để từ đó dự đoán độ ngọt và độ chín của xoài [30]. Việc ứng dụng phép đo phổ kết hợp với các thuật toán học máy và học sâu cũng được nghiên cứu phát triển một cách mạnh mẽ. Các loại mùi vị như ngọt, chua ngọt, chua và nhạt được tiến hành phân loại dựa trên dữ liệu phổ IR kết hợp với các thuật toán để đưa ra các kết quả phục vụ cho kiểm soát trước khi được bán ra thị trường [7]. Phổ IR cũng được kết hợp với các thuật toán liệu PLS và PCR để để định danh va phân loại giống của xoài [16]. Việc xác định độ chín của xoài đẻ phục vụ cho việc bảo quản và kiểm tra chất lượng trước khi tung ra thị trường là điều cần thiết, vì vậy có rất nhiều nghiên cứu hướng tới xây dựng hệ đo cầm tay kết hợp với phần mềm để phân loại nhanh độ chín của xoài trong phòng thí nghiệm và ngoài hiện trường [1, 24, 26].

1.6.2. Phân tích hình ảnh xác định nhanh độ đường trong quả cam Việt Nam

Trong số các loại cây ăn quả chủ lực, cam là loại cây có giá trị và phù hợp với điều kiện tự nhiên tại Việt Nam nên được trồng phổ biến khắp ba miền của nước ta, với nhiều giống bản địa quý, gắn với các vùng trồng cam nổi tiếng như: cam sành Hà Giang, cam Vinh, cam Cao Phong, cam Vân Du, cam Xoài, cam Canh ... đem lại hiệu quả kinh tế cao trong kinh tế nông nghiệp. Ở phía Bắc, trong số các vùng trồng cam nổi tiếng và

có thương hiệu, hiện nay giống cam Cao Phong, trồng nhiều tại huyện Cao Phong, tỉnh Hòa Bình (với 1.350 ha cam) và cam Vinh trồng tại tỉnh Hưng Yên (1.800 ha cam) có sản lượng rất lớn và là các vùng trồng cam đã được Cục Sở hữu trí tuệ (Bộ Khoa học và Công nghệ) cấp chứng nhận nhãn hiệu

Thành phần hóa học của cam

Cam chứa nhiều loại hóa chất bao gồm axit ascorbic (vitamin C), pectin, oxedrine, polyhydroxyphenol, axit folic, thiamine, kali, axit nicotinic, magiê, flavonoid, phenol, terpenoid, tannin, đường khử, alkaloid và saponin. Sự phân bố của các hóa chất này khác nhau giữa các phần khác nhau của quả cam. Vỏ cam thường có nồng độ cao hơn các nguyên tố như brom, canxi, xeri, kali, lanthanum, natri, rubidium và scandium. Flavedo (lớp ngoài của vỏ) chứa hàm lượng vitamin C, flavon và carotenoids cao, trong khi lớp albedo (lớp trong của vỏ) giàu phenolics, flavanone và có hoạt tính chống oxy hóa cao. Hạt của quả cam có hàm lượng sắt và kẽm cao hơn. Những hóa chất này trong cam có những lợi ích tiềm năng cho sức khỏe, bao gồm tăng cường khả năng miễn dịch, ngăn ngừa các bệnh mãn tính và tăng cường sức khỏe tổng thể.

- Trong phần ăn được của quả cam có chứa: Nước 80-90%, protid 1,3%, lipid 0,1 – 0,3%, đường 12-12,7%, vitamin C 45-61 %, acid citric 0,5-2%.
- Vỏ cam có chứa: Các hợp chất flavonoid, pectin, tinh dầu (0,5%). Tinh dầu vỏ cam, Oleum Auranti Dulcis, với tên thương phẩm là Orange oil là chất lỏng màu vàng hoặc nâu vàng, mùi thơm, vị không đắng. Thành phần chính là limonen (90%), các alcol, aldehyd (< 3%) gồm citral và decyl aldehyde.
- Hoa cam có chứa tinh dầu. Thành phần chính của tinh dầu hoa cam là limonen, linalol, methyl anthranilat (0,3%).

Một số nghiên cứu về dự đoán độ đường của cam bằng phân tích hình ảnh và thị giác máy tính

Hiện nay mới chỉ có một công bố về việc ứng dụng kỹ thuật phân tích hình ảnh và thị giác máy tính cho mục tiêu dự đoán độ đường của cam. Nghiên cứu sử dụng hệ chụp được xây dựng thủ công và điện thoại thông minh để thu thập ảnh màu của cam, kết hợp với bộ dữ liệu Orange để tiến hành trích xất các đặc trưng về màu sắc kết hợp với các thuật toán học máy vụ cho việc phân loại đối tượng thành 3 nhóm: ngọt vừa, ngọt và rất ngọt [2].

CHƯƠNG 2. THỰC NGHIỆM

2.1 Phân tích đồng thời kháng sinh bằng phương pháp phổ kết hợp với các thuật toán học máy, học sâu

2.1.1 Phân tích đồng thời Tetracycline, Penicillin G và Cephalexin trong mẫu thuốc bằng phổ UV kết hợp với các thuật toán học máy

2.1.1.1 Hóa chất

Thuốc thử và mẫu

Tetracycline Hydrochloride (TET), Penicillin G (PGP) và Cephalexin Monohydrate (CEX) được sản xuất từ Viện Kiểm nghiệm Thuốc Trung ương (Hà Nội, Việt Nam). Nước cất hai lần đã được sử dụng trong suốt nghiên cứu. Mẫu thương mại của ba viên kháng sinh này được mua từ các hiệu thuốc địa phương ở Hà Nội, Việt Nam.

Dung dịch chuẩn gốc và dung dịch chuẩn làm việc

Dung dịch chuẩn gốc của tetracyclin monohydrat, penicillin G Procaine và cephalexin monohydrat được chuẩn bị bằng cách hòa tan lượng thích hợp của từng thuốc thử phân tích trong nước tinh khiết để thu được nồng độ 200 µg/mL. Các dung dịch được bảo quản và tránh ánh sáng ở nhiệt độ 4°C. Dung dịch chuẩn làm việc được chuẩn bị hàng ngày bằng cách pha loãng thích hợp trong môi trường HCl.

Các phần thích hợp của dung dịch chuẩn gốc TET, PGP và CEX được pha loãng bằng nước cất để thu được nồng độ. Hỗn hợp của ba thành phần cũng được điều chế ở nồng độ 25 µg/mL. Các giải pháp này sau đó được quét trong phạm vi 230 nm – 350 nm.

2.1.1.2 Dụng cụ, thiết bị và phần mềm

Các phép đo quang phổ được thực hiện trên UV-1601PC (Shimadzu) được kết nối với máy tính có cài phần mềm PC UV-Win. Tất cả các phổ hấp thụ đã được lưu và sau đó xuất phần mềm UV-Win sang chương trình Microsoft Excel để xử lý thống kê. Ngôn ngữ lập trình Python phiên bản 3.11 được sử dụng để xây dựng các mô hình học máy.

2.1.1.3 Quy trình phân tích

Thiết lập hiệu chuẩn một thành phần

Khoảng nồng độ tuyến tính của TET, PGP và CEX lần lượt là 12-28 µg/mL, 5-18 µg/mL và 7-20 µg/mL. Giá trị độ hấp thụ tối đa được ghi lại ở λ max của mỗi loại thuốc (276nm đối với TET, 290nm đối với PGP và 262nm đối với CEX) so với nước cất dưới dạng mẫu trắng.

Xây dựng tập huấn luyện và tập kiểm tra

Hỗn hợp huấn luyện và xác nhận được chuẩn bị bằng cách kết hợp các dung dịch tiêu chuẩn làm việc của TET, PGP và CEX theo các tỷ lệ khác nhau trong phạm vi tuyến tính nồng độ của chúng. Năm mức nồng độ của mỗi chất phân tích đã được chọn để xây dựng cả tập huấn luyện và xác nhận. Tổng cộng có 31 hỗn hợp và 12 hỗn hợp được chuẩn bị độc lập cho các bộ huấn luyện và xác nhận tương ứng (Bảng 2.1).

Phổ hấp thụ của tất cả các hỗn hợp được ghi lại trong khoảng 230-350nm với khoảng cách 2nm**.**

Bảng 2.1 Hàm lượng TET, PGP, CEX (µg/mL) các hoạt chất trong hỗn hợp các mẫu huấn luyện và kiểm tra

Tập huấn luyện									**Tập kiểm tra**			
STT.	TET	PGP	CEX		*Mix no.*	TET	PGP	CEX	*STT.*	TET	PGP	CEX
1	25	9	6		*17*	21	17	8	*1*	19	15	6
2	17	11	6		*18*	23	17	8	*2*	25	15	6
3	21	11	6		*19*	25	17	8	*3*	19	13	8
4	25	11	6		*20*	17	9	10	*4*	21	13	8
5	19	13	6		*21*	19	9	10	*5*	25	13	8
6	23	13	6		*22*	23	9	10	*6*	17	11	10
7	17	17	6		*23*	17	15	10	*7*	19	11	10
8	19	17	6		*24*	25	9	12	*8*	25	11	10
9	21	17	6		*25*	21	11	12	*9*	17	13	6
10	23	17	6		*26*	25	11	12	*10*	17	11	8
11	25	17	6		*27*	21	15	14	*11*	21	11	8
12	25	9	8		*28*	25	15	14	*12*	23	11	12
13	19	15	8		*29*	25	13	6				

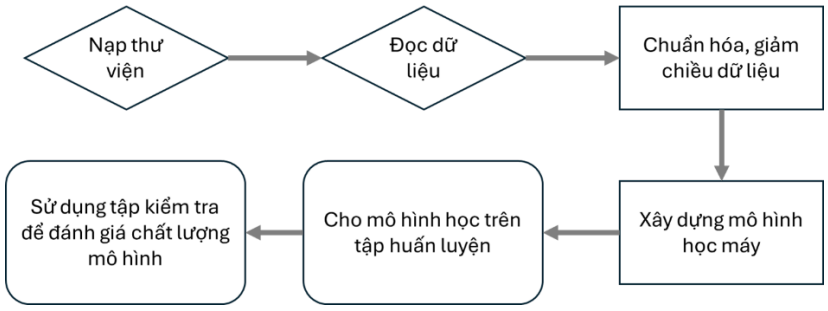
14	21	15	8		*30*	25	11	8				
15	17	17	8		*31*	17	13	8				
16	19	17	8									

Đánh giá hiệu quả của mô hình trên tập dữ liệu kiểm tra

Tiến hành xây dựng các mô hình học máy có tham số tối ưu, sử dụng tập dữ liệu huấn luyện gồm 31 mẫu để huấn luyện các mô hình đạt sai số nhỏ nhất. Kiểm tra khả năng phân tích đồng thời của các mô hình trên tập dữ liệu kiểm tra bằng các phép đánh giá R^2, RMSE và phần trăm sai số.

2.1.1.4 Lược đồ phân tích

Sau khi xây dựng được bộ dữ liệu các mẫu huấn luyện và mẫu kiểm kiểm tra, việc xử lý số liệu được tiến hành như ở hình 2.1:



Hình 2.1 Lược đồ phân tích Tetracycline, Penicillin G và Cephalexin trong mẫu thuốc bằng phổ UV kết hợp với các thuật toán học máy

2.1.2 Phân tích đồng thời một số nhóm thuốc kháng sinh bằng phổ IR sử dụng học sâu

2.1.2.1 Hóa chất

Các thông tin về chất chuẩn và tá dược (Nơi sản xuất, tiêu chuẩn, số kiểm soát, hàm lượng,...) được nêu trong trong phụ lục đi kèm.

2.1.2.2 Thiết bị, dụng cụ và phần mềm

- Cân phân tích Sartorious độ chính xác ± 0,00001g.
- Máy quang phổ hồng ngoại Agilent Technologies Cary 600 Series FTIR spectrometer, dải số sóng đo 7500-2800 cm-1. Detector nhiệt DTGS

- Bộ dụng cụ ép viên: Agilent Technologies standard sampling kit (part no: Pike - 162 - 1000).
- Thư viện phổ chuẩn: ST- Japan spectral libraries (part no: K8159 - 1000)
- Ngôn ngữ Python phiên bản 3.11

2.1.2.3 Quy trình phân tích

Bước 1: Chuẩn bị các mẫu chuẩn, mẫu kiểm tra chứa đồng thời các hoạt chất và các tá dược có tỷ lệ thay đổi trong khoảng nồng độ khảo sát sao cho tín hiệu độ hấp thụ thay đổi trong vùng tuyến tính.

Bước 2: Nghiền và trộn từng mẫu trong vòng 10 phút thu được hỗn hợp đồng nhất. Lấy hỗn hợp chất vừa đồng nhất trên trộn với KBr theo tỉ lệ đã khảo sát rồi tiến hành nghiền mịn, đồng nhất mẫu trong cối mã não trong 10 phút.

Bước 3: Lấy khoảng 15 mg lượng bột vừa nghiền được cho vào bộ ép viên để thu được viên mẫu đem đo phổ hồng ngoại trong vùng phổ từ 3000-3600 nm, ghi lại độ hấp thụ quang của từng mẫu, xuất số liệu thu được dưới dạng ASCII và chuyển toàn bộ dữ liệu vào phần mềm matlab để tính toán kết quả.

Bước 4: Xây dựng các mô hình học máy và học sâu để tiến hành học ma trận độ hấp thụ quang của các mẫu chuẩn và mẫu kiểm tra đã chuẩn bị ở phần trên. Sử dụng Python để đọc dữ liệu, khảo sát xây dựng các mô hình với tham số tối ưu và sử dụng các mô hình này để tìm hàm lượng (%) mỗi hoạt chất trong từng mẫu. So sánh sai số tương đối của mỗi phương pháp, lựa chọn ra phương pháp tối ưu nhất để tiến hành định lượng các mẫu thực tế.

Bước 5: Tiến hành định lượng các mẫu thực tế bằng cách trộn một lượng bột mẫu với tá dược để pha loãng nồng độ hoạt chất có hàm lượng (%) nằm trong ma trận chuẩn đã xây dựng, đo phổ của các mẫu này, ghi lại phổ và sử dụng Python để để tính toán kết quả. Từ đó tiến hành tính toán hàm lượng hoạt chất trong các mẫu thuốc viên theo công thức dưới đây:

$$HL/mg\ viên = X.\frac{m_{tb}}{m_t}$$

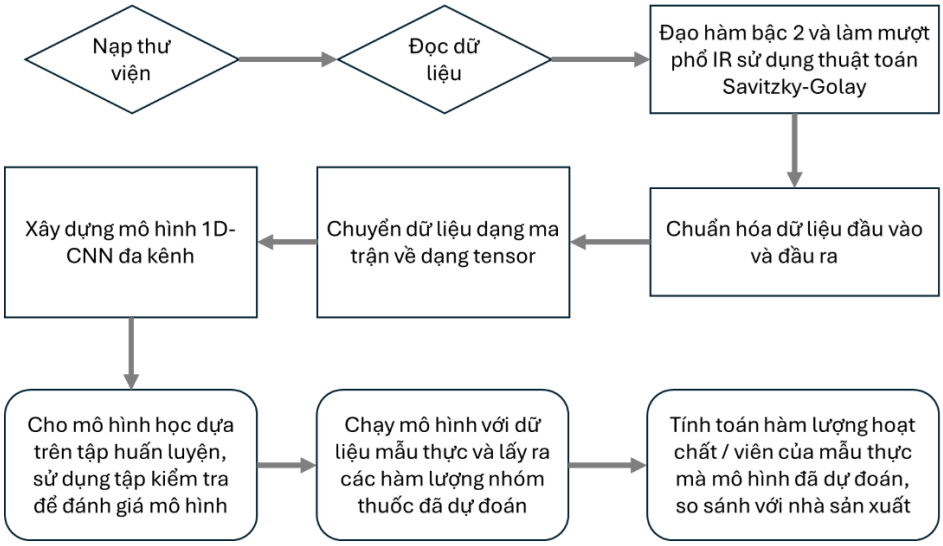
Trong đó:

- X: Lượng (mg) hoạt chất tìm được từ mô hình hồi quy đa biến

- m_t: khối lượng cân của mẫu thử (mg)
- m_{tb}: khối lượng trung bình của 1 viên của thuốc (mg)

2.1.2.4 Lược đồ phân tích

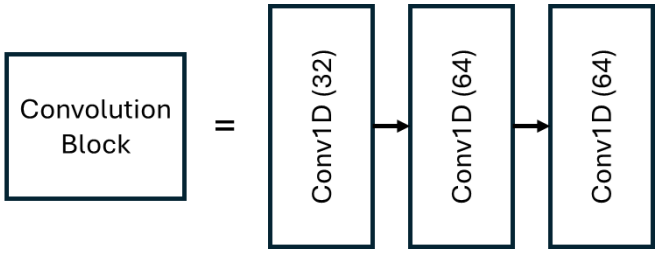
Sau được xây dựng, bộ dữ liệu phổ và hàm lượng của dữ liệu huấn luyện, dữ liệu kiểm tra, thông tin mẫu thực đều được lưu vào một file excel để tiến hành xử lý số liệu. Việc xử lý số liệu được tiến hành như ở hình 2.2:



Hình 2.2 Lược đồ phân tích đồng thời một số nhóm thuốc kháng sinh bằng phổ IR sử dụng học sâu

Mô hình 1D-CNN đa kênh: là mô hình nhận vào một tensor 1 chiều và có đầu ra gồm bốn kênh tích hợp. Với mô hình 1D-CNN thông thường, việc hồi quy với 12 hoạt chất tương ứng với 4 nhóm chất có thể khiến cho việc dự đoán thiếu đi độ chính xác, do các đặc trưng đầu ra phải chia sẻ với nhau các tham số của lớp tích chập và lớp nơ ron nhân tạo. Tuy nhiên nếu cho mỗi nhóm chất chứa các khối tích chập và nơ ron của riêng nó, việc chia sẻ dẫn đến sai số giảm sẽ không xảy ra. Cấu hình của mô hình 1D-CNN đa kênh được thể hiện trong phụ lục đi kèm.

Dữ liệu sau khi được tensor hóa sẽ được đi qua tập hợp khối tích chập (Convolution Block), tập hợp khối tích chập gồm nhiều các lớp tích chập chồng lên nhau như trong hình 2.3:



Hình 2.3 Cấu trúc một khối tích chập (Convolution block)

Kĩ thuật chồng nhiều lớp tích chập lên nhau tạo thành các khối, và sử dụng liên tiếp các khối này đã được chứng minh là đạt độ hiệu quả cao, tối đa hóa được khả năng tiếp nhận và xử lý thông tin của các mô hình học sâu, điển hình là cấu trúc VGG cho nhận dạng ảnh [27]. Sau khi dữ liệu được đi khối tích chập đầu tiên, các giá trị trung bình của nó sẽ được lấy ra trên toàn bộ dữ liệu và đưa vào từng kênh, mỗi kênh có công việc là hồi quy ra 3 hoạt chất, tổng cộng với 4 kênh sẽ dự đoán được 12 hoạt chất. Mỗi kênh đều có khối tích chập, sau đó dữ liệu được đưa vào lớp MaxPooling để lấy ra các thông tin quan trọng nhất, sau cùng được dàn trải thành vector và đưa vào lớp nơ ron nhân tạo chứa 100 nơ ron.

2.2 Xác định hàm lượng đường trong cam sử dụng thị giác máy tính và phân tích hình ảnh

2.2.1 Hóa chất, thiết bị, dụng cụ và phần mềm

- Cốc thủy tinh 100ml
- Ống đong
- Máy vắt cam
- Máy đo độ đường Brix
- Hệ chụp cam (ảnh hệ chụp được hiển thị trong phụ lục đi kèm)
- Ngôn ngữ python 3.11, sử dụng thư viện Tkinter để tạo cửa sổ giao diện.

2.2.2 Quy trình phân tích

2.2.2.1 Chụp ảnh cam

Cam được rửa sạch và lau khô, tiến hành đo cân nặng và thông tin về khối lượng quả, chu vi chiều ngang, chiều dọc quả được ghi lại. Đặt cam lên hệ chụp, mô tơ quay quay quả cam với chu kì 2 giây một lần, sau mỗi lần quay chụp quả cam một lần bằng camera, tiến hành quay và chụp 100 lần ta thu được 100 bức ảnh cam/quả.

2.2.2.2 Đo độ đường của cam

Bước 1: Tiến hành vắt quả cam và lọc vắt cam

Bước 2: Hút và nhỏ 1 giọt nước cam vắt và cho lên máy đo brix

Bước 3: Bấm start và chờ hiển thị kết quả

Bước 4: Ghi lại số đo

2.2.2.3 Huấn luyện mô hình hồi quy độ đường của cam

Bước 1: Sử dụng mô hình phân đoạn (segment) để phân biệt cam với nền trên ảnh, từ đó sử dụng kĩ thuật tách nền để tách nền ảnh cam để ảnh đầu ra chỉ có duy nhất quả cam nằm trên nền màu đen.

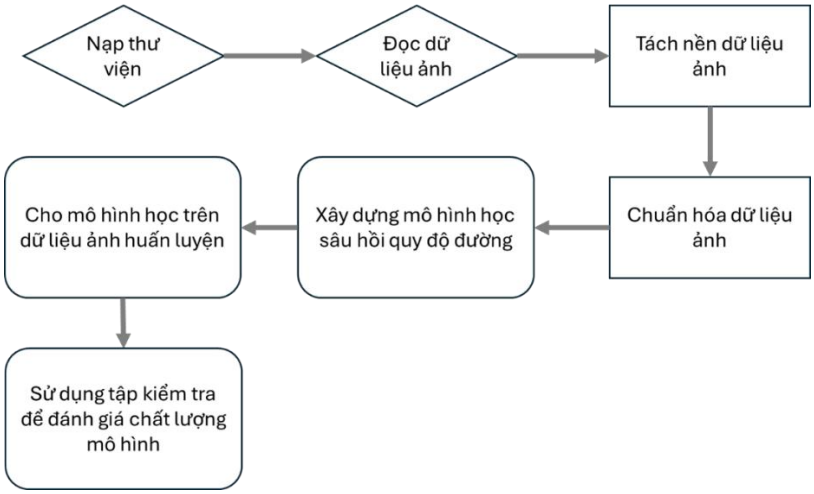
Bước 2: Trên dữ liệu ảnh đã tách nền tiến hành phân chia số lượng ảnh theo tỷ lệ 70-30, với 70% ảnh và nhãn để cho mô hình học sâu học được mối tương quan giữa hình ảnh và độ đường, còn lại 30% còn lại để kiểm tra khả năng dự đoán của mô hình.

Bước 3: Tạo mô hình và tiến hình cho mô hình học dựa trên dữ liệu huấn luyện.

Bước 4: Đánh giá mô hình trên tập kiểm tra.

2.2.3 Lược đồ phân tích

Sau khi xây dựng hệ cơ sở dữ liệu gồm các bức ảnh cam chụp được cùng với thông tin độ đường, việc phân tích và xử lý được tiến hành như hình 2.4:



Hình 2.4 Lược đồ phân tích xác định hàm lượng đường trong cam

2.3 Phân loại các đặc điểm của xoài dựa trên phổ Vis-NIR với thuật toán học sâu

(Thực nghiệm và kết quả của nghiên cứu này kế thừa từ nghiên cứu "Achieving robustness across season, location and cultivar for a NIRS model for intact mango fruit dry matter content" - N.T. Anderson, K.B. Walsh, P.P. Subedi, C.H. Hayes)

2.3.1 Lựa chọn quả

Quả của các giống xoài 'Calypso™' (Caly), 'Honey Gold' (HG), 'Keitt', 'Kensington Pride' (KP), 'Lady Grace' (LadyG), 'Lady Jane' (Lady J), 'R2E2′ và các dòng 1201, 1243 và 4069 của Chương trình Nhân giống Xoài Quốc gia Úc đã được thu thập trong bốn mùa. Các quần thể có nguồn gốc từ hai vùng đang phát triển riêng biệt ở Úc (Lãnh thổ phía Bắc và Trung tâm Queensland, lần lượt là các vùng nhiệt đới và cận nhiệt đới). Trái cây được quét ở các giai đoạn sinh lý giữa 'xanh cứng' và 'chín' (từ giai đoạn làm mềm sớm đến giai đoạn 'ăn chín') [3].

2.2.2 Thiết bị và dụng cụ

- Máy đo chất lượng sản xuất F750 (Felix Instruments, Camas, USA: sử dụng MMS1 (Máy quang phổ thu nhỏ nguyên khối; Zeiss, Oberkochen, Đức),
- Cân
- Máy bóc vỏ
- Lò sấy bằng không khí cưỡng bức (UltraFD1000, Ezidri, Beverley, Australia) [3]
- Ngôn ngữ lập trình Python phiên bản 3.11

2.2.3 Quy trình phân tích

Bước 1: Việc thu thập phổ và phân tích tham chiếu phá hủy được thực hiện trong cùng một ngày. Mỗi quả được quét hai lần ở phần rộng nhất của mỗi mặt (khoảng giữa quả) vuông góc với mặt phẳng hạt. Vị trí quét này là mô quả gần tương đương với trung bình của tất cả các mô trung bì quả.

Bước 2: Các má quả được cắt ra khỏi quả, vỏ được gọt bỏ bằng dao bào và một lõi hình trụ (đường kính 29 mm) được lấy tại vị trí thu phổ và cắt thành chiều dài 10 mm (từ phía vỏ). Mẫu cắt được chia thành bốn phần và cân, sau đó được sấy khô trong 48 giờ ở 65 °C (đến trọng lượng không đổi) trong lò sấy cưỡng bức thương mại (UltraFD1000, Ezidri, Beverley, Australia).

Bước 3: Các thông số chất khô (dry matter), nguồn gốc quả, mùa thu hoạch, nhiệt độ thu hoạch, độ chín (xanh hay chín) được ghi lại vào cùng hàng với dữ liệu phổ.

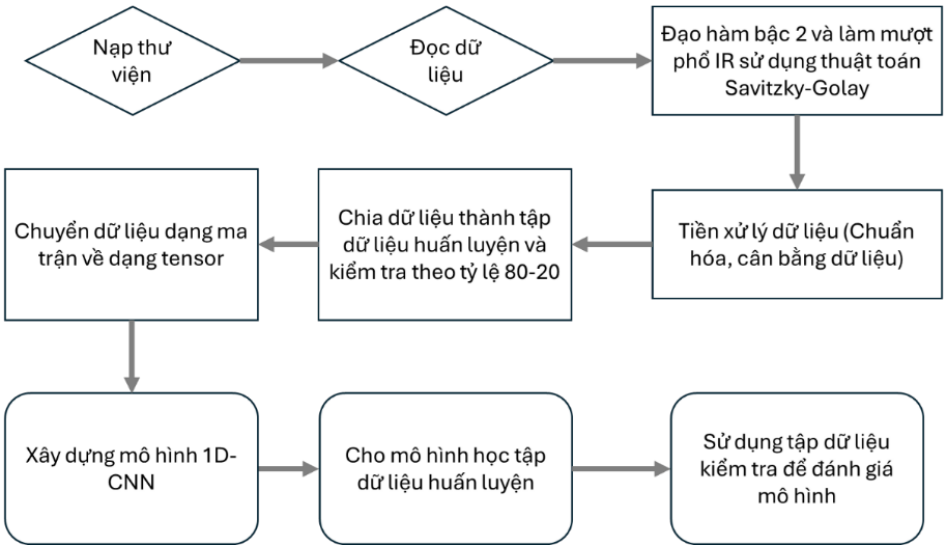
Bước 4: Chia dữ liệu theo tỷ lệ 80-20, với số lượng mẫu trong tập huấn luyện chiếm 80%, số lượng mẫu trong tập kiểm tra chiếm 20%. Để có thể đánh giá một cách khách quan, chia dữ liệu sao cho phân bố của mỗi lớp bên trong tập kiểm tra giống với bên trong tập huấn luyện nhất có thể.

Bước 5: Tạo mô hình học sâu và cho mô hình học trên dữ liệu huấn luyện.

Bước 6: Kiểm tra độ chính xác của mô hình trên tập huấn luyện bằng phép đo độ chính xác và ma trận nhầm lẫn.

2.2.4 Lược đồ phân tích

Xử dụng kết quả dữ liệu được xây dựng từ bài báo..., việc xây dựng mô hình phân loại được tiến hành như trong hình 2.5:



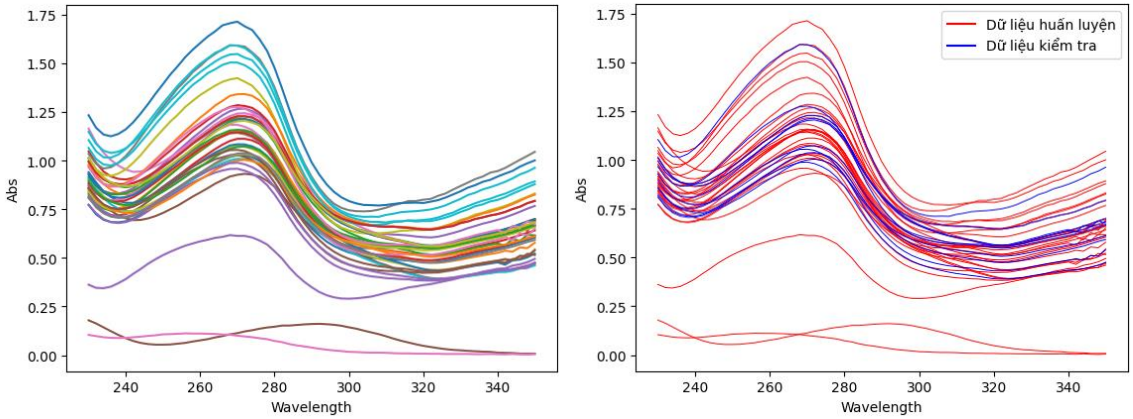
Hình 2.5 Lược đồ phân loại các đặc điểm của xoài dựa trên phổ NIR với thuật toán học sâu

CHƯƠNG 3. KẾT QUẢ VÀ THẢO LUẬN

3.1 Phân tích đồng thời kháng sinh bằng phương pháp phổ kết hợp với các thuật toán học máy, học sâu

3.1.1 Phân tích đồng thời Tetracycline, Penicillin G và Cephalexin trong mẫu thuốc bằng phổ UV kết hợp với các thuật toán học máy

Với bộ số liệu thu được (phụ lục đi kèm) theo thực nghiệm phần 2.1, sau khi sử dụng câu lệnh để đọc dữ liệu phổ từ file "LM.xlsx", thu được ma trận phổ của dữ liệu huấn luyện có kích thước 30x61 và ma trận phổ của dữ liệu kiểm tra có kích thước 10x61 được hiển trên hình 3.1:



Hình 3.1 Phổ UV của toàn bộ tập dữ liệu chứa các hoạt chất (trái), phổ của tập dữ liệu huấn luyện và tập dữ liệu kiểm tra (phải)

Phổ của tập kiểm tra nằm trong khoảng tín hiệu của tập huấn luyện, đồng thời số lượng mẫu trong tập huấn luyện ít nên để tránh hiện tượng quá khớp ưu tiên lựa chọn sử dụng các thuật toán học máy cơ bản cho việc việc định lượng đồng thời. Độ hấp thụ quang của các mẫu phân tích diao động trong khoảng từ 0 đến 1,75, có sự khác biệt nhau rất lớn nên để giúp các mô hình học máy một cách tối ưu hơn, tiến hành chuẩn hóa dữ liệu phổ sử dụng StandardScaler() trước khi áp dụng mô hình học máy.

3.1.1.1 Ứng dụng thuật toán cây quyết định và rừng ngẫu nhiên phân tích đồng thời

Mô hình cây quyết định và mô hình rừng ngẫu nhiên được khởi tạo (Tham số mô hình cây quyết định và rừng ngẫu nhiên được trình bày trong phụ lục đính kèm) sử dụng thư viện scikit-learn. Kết quả R^2 và RMSE của mô hình rừng ngẫu nhiên trên dữ liệu huấn luyện và dữ liệu kiểm tra thu được ở bảng 3.1:

Bảng 3.1 Độ chính xác của thuật toán cây quyết định và rừng ngẫu nhiên

		Mô hình cây quyết định			Mô hình rừng ngẫu nhiên		
		Tetracycline	Penicillin	Cephalexin	Tetracycline	Penicillin	Cephalexin
R^2	Train	0,93	0,78	0,81	0,98	0,90	0,91
	Test	0,86	-0,39	0,34	0,91	0,13	0,48
RMSE	Train	1,61	2,03	1,40	0,90	1,38	0,97
	Test	1,26	2,22	1,60	1,03	1,76	1,42

Nếu đánh giá theo giá trị R^2, ở cả hai mô hình giá trị R^2 của nhóm tetracycline là cao nhất, thể hiện sự đồng đều tốt ở cả tập dữ liệu huấn luyện và tập dữ liệu kiểm tra, tuy nhiên ở nhóm Cephalexin giá trị R^2 ở tập dữ liệu kiểm tra thấp một cách đáng kể, điều này thậm chí còn thể hiện rõ ràng hơn ở tập nhóm Penicillin, khi R^2 của mô hình rừng ngẫu nhiên chỉ đạt 0,13, còn đối với mô hình cây quyết định giá trị R^2 thậm chí còn nhỏ hơn 0. Nhưng điều đó chưa có nghĩa rằng hai mô hình đang xem xét tới xảy ra tình trạng quá khớp, bởi lẽ giá trị R^2 chỉ được sử dụng để biểu thị tỷ lệ biến thiên trong dữ liệu mẫu được giải thích bằng mô hình hồi quy, đặc biệt với mô hình hồi quy tuyến tính. Đối với tình huống có số mẫu nhỏ, giá trị dao động bất định, việc sử dụng R^2 là không phù hợp để đánh giá một mô hình hồi quy có tốt hay không [9]. Điều đó lý giải giá trị R^2 thấp trong trường hợp này, khi sử dụng các mô hình liên quan đến cấu trúc cây và kết hợp, khi các nguyên tắc hồi quy không tuần theo hàm tuyến tính.

Vì những lý do như trên, việc kết hợp các phép đo sai số là cần thiết để đánh giá độ hiệu quả của mô hình. Giá trị RMSE của cả hai mô hình trên tập dữ liệu kiểm tra nhìn chung đều cao hơn so với trên tập dữ liệu huấn luyện. Điều này là có thể chấp nhận vì dữ liệu tập kiểm tra chưa bao giờ xuất hiện trong quá trình học của mô hình. Một điểm đáng lưu ý là ở nhóm Tetracycline ở mô hình cây quyết định, giá trị RMSE ở tập kiểm tra còn thấp hơn so với tập huấn luyện, dù R^2 ở tập huấn luyện cao hơn.

Ngoài giá trị RMSE, có thể sử dụng phép đo phần trăm sai số để đánh giá độ phù hợp của mô hình. Giá trị hàm lượng được phân tích đồng thời và sai số của hai mô hình so với giá trị hàm lượng thực tế được thể hiện ở bảng 3.2 và bảng 3.3 sau:

Bảng 3.2 Hàm lượng (µg/mL) của tetracycline, penicillin và cephalexin khi phân tích bằng thuật toán cây quyết định và rừng ngẫu nhiên

STT	Mô hình cây quyết định			Mô hình rừng ngẫu nhiên			Hàm lượng thực" (xác định theo phương pháp HPLC)		
	Tetra cyclin e	Penic illin	Cephal exin	Tetracy cline	Penic illin	Cephal exin	Tetracy cline	Penici line	Ceph alexin
1	24,71	12,43	7,14	24,30	12,42	7,62	25,00	15,00	6,00
2	19,25	11,25	7,75	18,90	12,58	7,56	19,00	13,00	8,00
3	19,86	16,43	8,29	21,20	13,64	8,18	21,00	13,00	8,00
4	24,71	12,43	7,14	24,56	13,00	7,86	25,00	13,00	8,00
5	19,25	11,25	7,75	18,02	10,28	9,06	17,00	11,00	10,00
6	19,25	11,25	7,75	18,68	12,58	8,06	19,00	11,00	10,00
7	24,71	12,43	7,14	23,90	13,44	7,30	25,00	11,00	10,00
8	19,25	11,25	7,75	18,34	12,18	7,26	17,00	13,00	6,00
9	24,71	12,43	7,14	24,02	13,80	7,68	25,00	17,00	6,00
10	23,00	12,20	12,80	22,80	13,00	12,96	25,00	11,00	12,00

(*: hàm lượng của các chất trong mẫu tự tạo gồm hoạt chất và tá dược)

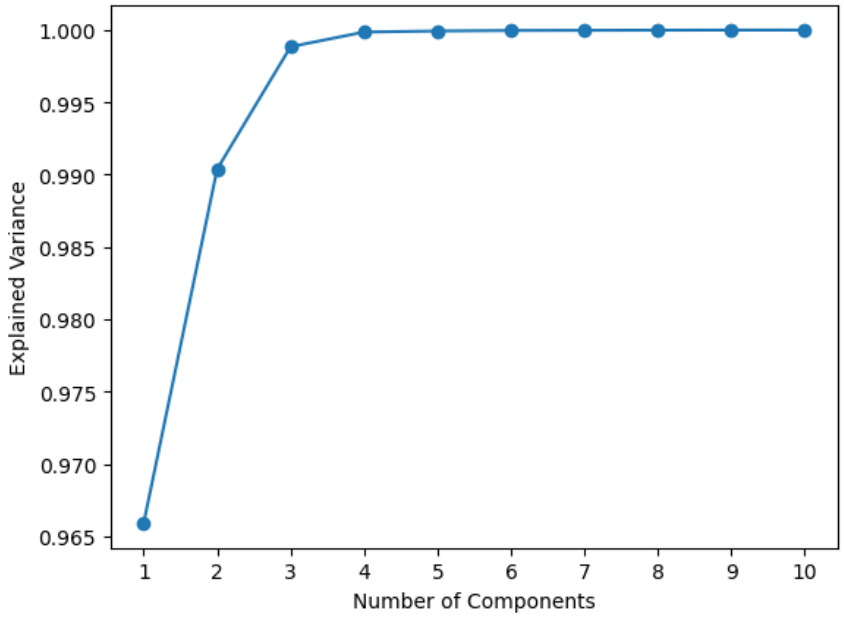
Bảng 3.3 Sai số tương đối (%) của các chất phân tích được xác định bằng thuật toán cây quyết định và rừng ngẫu nhiên

	Mô hình cây quyết định			Mô hình rừng ngẫu nhiên		
STT	Tetracycline	Peniciline	Cephalexin	Tetracycline	Peniciline	Cephalexin
1	1,14	17,14	19,05	2,80	17,20	**27,00**
2	1,32	13,46	3,13	0,53	3,23	5,50
3	5,44	**26,37**	3,57	0,95	4,92	2,25
4	1,14	4,40	10,71	1,76	0,00	1,75
5	13,24	2,27	**22,50**	6,00	6,55	9,40
6	1,32	2,27	**22,50**	1,68	14,36	19,40
7	1,14	12,99	**28,57**	4,40	**22,18**	**27,00**
8	13,24	13,46	**29,17**	7,88	6,31	**21,00**
9	1,14	**26,89**	19,05	3,92	18,82	**28,00**
10	8,00	10,91	6,67	8,80	18,18	8,00

Giá trị sai số tương đối biểu thị sai khác của kết quả tìm được theo mô hình và kết quả biết trước bằng mô hình PCR và PLSR lớn hơn 20% tập trung ở các mẫu 6, 7, 8, 9, đặc biệt là cái sai số lớn thường tập trung vào nhóm Cephalexin. Điều này chứng tỏ việc định lượng đồng thời kết hợp Cephalexin với các nhóm chất khác là tương đối khó khăn. Tỷ lệ có sai số lớn hơn 20% của mô hình cây quyết định là 6/10 mẫu còn của mô hình rừng ngẫu nhiên là 4/10 mẫu. Có thể kết luận rằng mô hình cây quyết định và rừng ngẫu nhiên chưa thực sự phù hợp để định lượng đồng thời 3 thành phần Tetracyclin, Penicillin và Cephalexin trong mẫu thuốc.

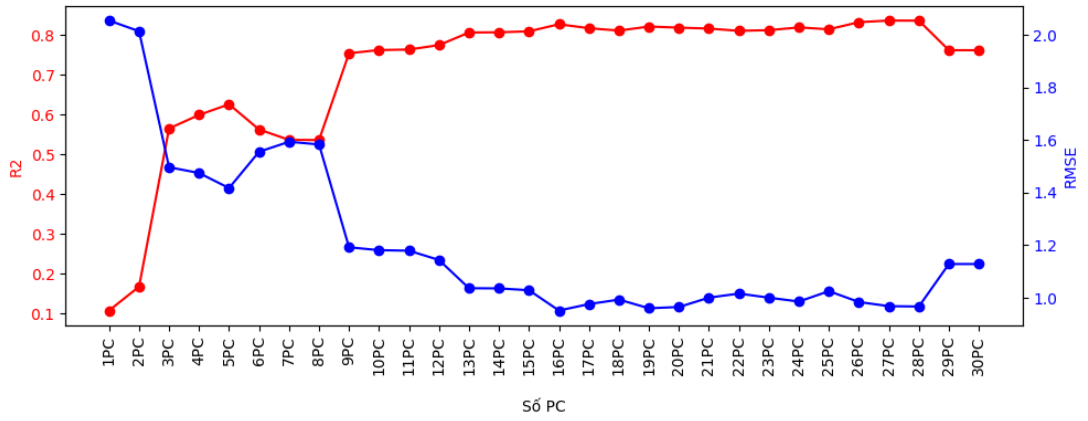
3.1.1.2 Phân tích đồng thời sử dụng thuật toán PCR và PLS

Khảo sát giá trị tổng % phương sai giải thích của dữ liệu phổ tập huấn luyện sử dụng thuật toán PCA, kết quả thu được ở hình 3.2:

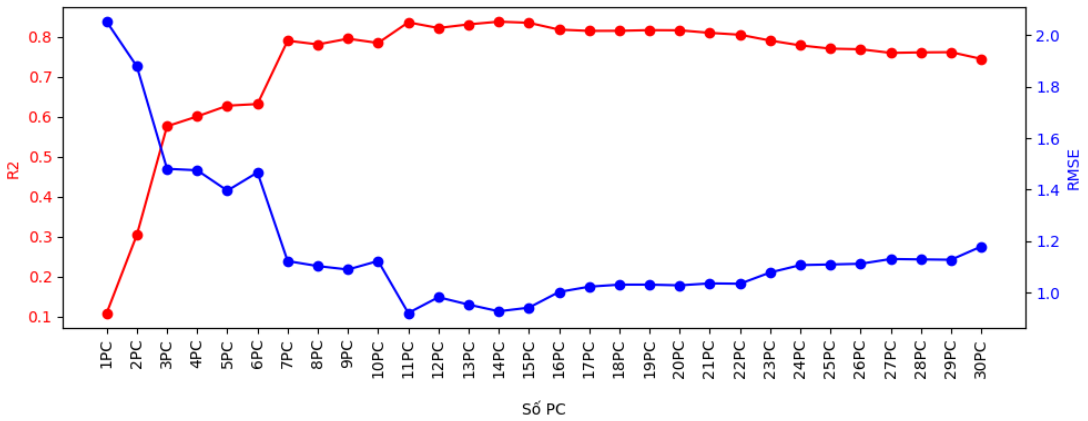


Hình 3.2 Giá trị tổng % phương sai giải thích của dữ liệu theo từng số cấu tử chính

Ở ngay PC đầu tiên, giá trị tổng % phương sai giải thích của tập dữ liệu huấn luyện đã lớn hơn 96,5%, như vậy việc giảm chiều dữ liệu có thể nói là xảy ra suôn sẻ kể cả khi chọn một cấu tử chính và việc lựa chọn số cấu tử chính sẽ tùy vào nhu cầu và kích thước mẫu của tập dữ liệu. Tuy nhiên việc lựa chọn số cấu tử chính của mô hình hồi quy không nên chỉ dựa vào giá trị tổng % phương sai giải thích khi giá trị đó chỉ giải thích cho khả năng giữ lại lượng thông tin sau khi giảm chiều. Để xác định khả năng hồi quy tốt nhất ứng với từng số cấu tử chính, cần lựa chọn sử dụng các phép đo sai số khi mô hình chạy từng PC. Kết quả khảo sát số PC của mô hình PCR và mô hình PLS được biểu thị ở hình 3.3 và hình 3.4:



Hình 3.3 Ảnh hưởng của số cấu tử chính (PC) lên giá trị R^2 và RMSE của mô hình PCR



Hình 3.4 Ảnh hưởng của số cấu tử chính (PC) lên giá trị R^2 và RMSE của mô hình PLSR

Trong thuật toán PCR, giá trị RMSE đạt mức tối thiểu tại 16 thành phần chính (PC). Đối với thuật toán PLSR, giá trị RMSE nhỏ nhất đạt được khi sử dụng 11 thành phần chính. Do đó, mô hình PCR sẽ được xây dựng với 16 thành phần chính, trong khi mô hình PLSR sẽ sử dụng 11 thành phần chính. Hàm lượng của hai mô hình PCR và PLSR được định lượng đồng thời và phần trăm sai số so với giá trị hàm lượng thực tế được thể hiện trong bảng 3.4 và bảng 3.5:

Bảng 3.4 Hàm lượng (µg/mL) của tetracycline, penicillin và cephalexin khi phân tích bằng thuật toán PCR và PLSR

STT	Mô hình PCR	Mô hình PLSR	Thực tế

	Tetracycline	Penicillin	Cephalexin	Tetracycline	Penicillin	Cephalexin	Tetracycline	Penicillin	Cephalexin
1	25,28	16,55	4,99	25,40	16,60	5,09	25,00	15,00	6,00
2	18,70	12,72	6,55	18,78	12,91	6,39	19,00	13,00	8,00
3	18,69	12,94	7,36	19,27	13,05	7,41	21,00	13,00	8,00
4	23,81	13,18	7,60	24,36	13,03	7,99	25,00	13,00	8,00
5	16,65	11,69	9,56	16,74	12,04	9,68	17,00	11,00	10,00
6	18,86	10,75	9,38	18,56	11,27	9,54	19,00	11,00	10,00
7	25,52	11,80	8,31	25,23	11,59	8,40	25,00	11,00	10,00
8	19,34	12,91	4,96	19,39	12,87	5,22	17,00	13,00	6,00
9	23,87	16,87	7,13	23,57	17,38	7,41	25,00	17,00	6,00
10	24,77	11,06	13,27	24,48	11,34	12,96	25,00	11,00	12,00

Bảng 3.5 Sai số tương đối (%) của các chất phân tích được xác định bằng thuật toán PCR và PLSR

STT	Mô hình PCR			Mô hình PLSR		
	Tetracycline	Penicillin	Cephalexin	Tetracycline	Penicillin	Cephalexin
1	1,13	10,35	16,77	1,59	10,69	15,21
2	1,60	2,13	18,18	1,18	0,73	**20,15**

3	11,02	0,46	8,00	8,25	0,41	7,43
4	4,74	1,37	4,95	2,56	0,23	0,18
5	2,06	6,24	4,40	1,53	9,42	3,16
6	0,76	2,26	6,23	2,31	2,45	4,61
7	2,10	7,28	16,95	0,93	5,41	15,96
8	13,78	0,70	17,27	14,08	1,01	13,00
9	4,50	0,79	18,78	5,73	2,23	**23,50**
10	0,92	0,51	10,62	2,06	3,05	7,97

Nhìn chung, cả hai mô hình PCR và PLSR có khả năng dự đoán tốt với tập dữ liệu kiểm tra. Các giá trị phần trăm sai số ở các mẫu phần lớn đều nhỏ hơn 20%. Trước sự biến thiên của nồng độ nhóm Cephalexin, tất cả các mẫu được dự đoán bởi mô hình PCR đều có sai số nằm trong khoảng chấp nhận được, còn mô hình PLSR chỉ có 2/10 mẫu có giá trị sai số lớn hơn 20% nhưng không đáng kê. Chứng tỏ rằng mô hình PCR và PLSR đều đã khắc phục được nhược điểm của mô hình cây quyết định và mô hình rừng ngẫu nhiên khi định lượng đồng thời ba thành phần Tetracyclin, Penicillin và Cephalexin, đặc biệt mô hình PCR có khả năng ứng dụng thực tế bởi tính ổn định và kết quả định lượng tốt.

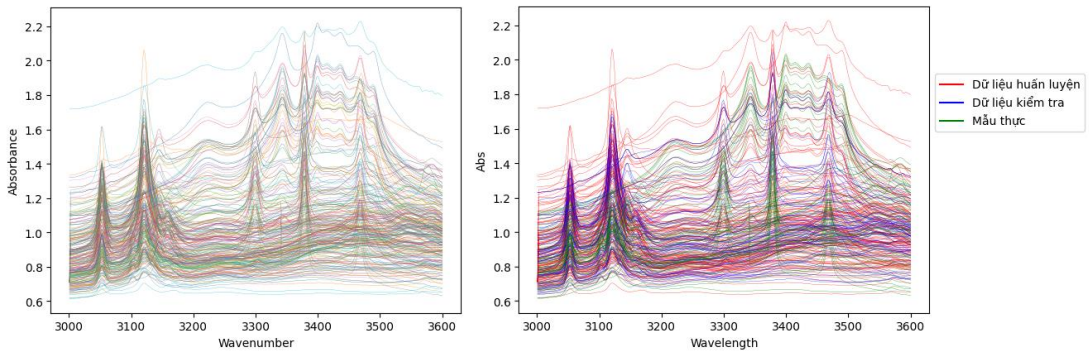
3.1.1.3 Kết luận

Nhìn chung, các thuật toán học máy đều có thể học được sự tương quan giữa phổ UV và hàm lượng của Tetracycline, Penicillin G và Cephalexin, việc định lượng đồng thời tốt thể hiện ở khả năng hồi quy một các linh loạt. Trong đó thuật toán PCR có sai số tương đối của thành phần tất cả các mẫu kiểm tra đều dưới 20%, chứng tỏ khả năng ứng dụng cao của các thuật toán học máy trong thực tế để phân tích nhanh và đồng thời hàm lượng Tetracycline, Penicillin G và Cephalexin trong mẫu thuốc bằng phổ UV.

3.1.2 Phân tích đồng thời một số nhóm thuốc kháng sinh bằng phổ IR sử dụng học sâu

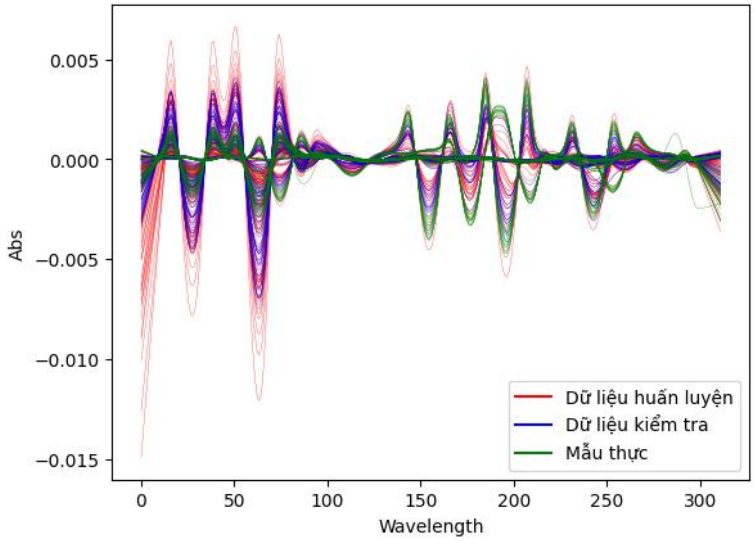
Tiến hành sử dụng câu lệnh để đọc dữ liệu phổ từ file dữ liệu, thu được ma trận dữ liệu phổ huấn luyện có dạng 142x312, ma trận dữ liệu phổ kiểm tra có dạng 65x312, ma trận dữ liệu % hàm lượng của tập huấn luyện có dạng 142x12, ma trận dữ liệu % hàm lượng của tập kiểm tra có dạng 65x12. Dữ liệu mẫu thực được đọc từ bảng tính samples của cùng file dữ liệu và được tách ra thành ma trận phổ mẫu thực có dạng 54x312. Dữ liệu về khối lượng viên, khối lượng cân và hàm lượng công bố của nhà sản xuất của các mẫu thực được đọc và lưu thành các biến riêng lẻ trong phần mềm để phục vụ cho việc tính toán hàm lượng chất/viên sau khi lựa chọn mô hình phù hợp.

Sau khi tiến hành đọc dữ liệu, phổ của cả tập huấn luyện, kiểm tra và mẫu thực được hiển thị trên hình 3.5:



Hình 3.5 Phổ IR của toàn bộ tập dữ liệu (trái), phổ IR của tập huấn luyện, kiểm tra và mẫu thực (phải)

Phổ IR của các mẫu đo có xu hướng bị nhiễu khá lớn trong vùng từ 3200-3600 cm^{-1}, một số peak đặc trưng vẫn xuất hiện ở một vài mẫu, nhưng việc sử dụng toàn bộ phổ dữ liệu để phục vụ cho mục đích hồi quy đa biến, đặc biệt là với số đặc trưng đầu ra lớn có thể dẫn đến nguy cơ khiến cho các mô hình học máy và học sâu khó đạt đến sự hội tụ, sai số lớn trong việc định lượng đồng thời. Để khắc phục tình trạng nhiễu phổ, giúp cho các mô hình học tập dữ liệu nhanh hơn và tốt hơn tiến hành đạo hàm bậc 2 và làm mượt phổ bằng thuật toán Savitzky-Golay, phổ sau khi đạo hàm và làm mượt với 25 bước và hàm bậc 3 được biểu diễn ở hình 3.6:



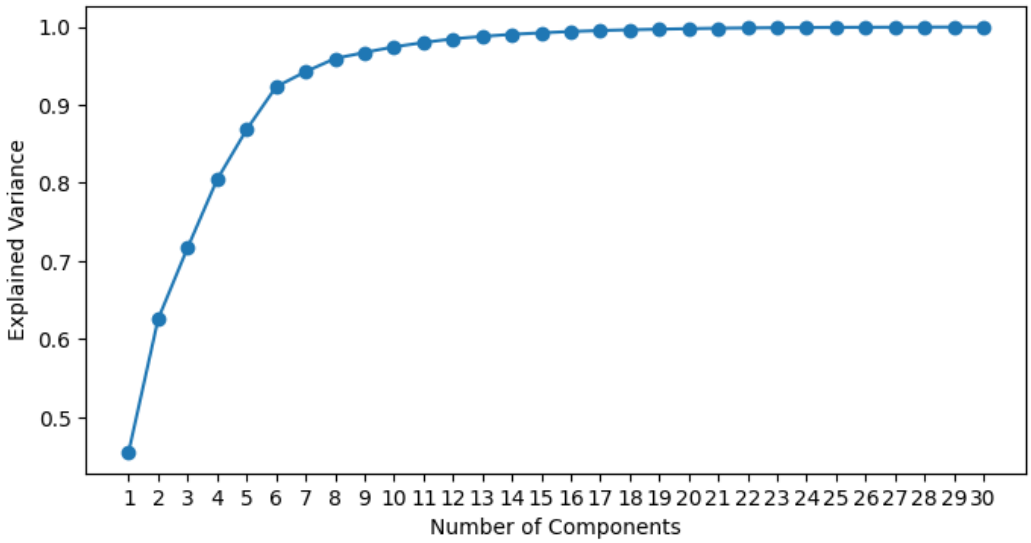
Hình 3.6 Phổ tập dữ liệu huấn luyện, kiểm tra và mẫu thực sau khi đạo hàm bậc 2 và làm mượt với thuật toán Savitzky-Golay

Sau khi đạo hàm bậc 2 và làm mượt phổ sử dụng Savitzky-Golay, phổ tín hiệu đã trở nên mượt hơn, loại bỏ được các giá trị nhiễu. Để các mô hình học một cách dễ dàng nhất, tiến hành chuẩn hóa dữ liệu phổ sử dụng thuật toán StandardScaler().

3.1.2.1 Phương pháp định lượng đồng thời sử dụng các thuật toán học máy

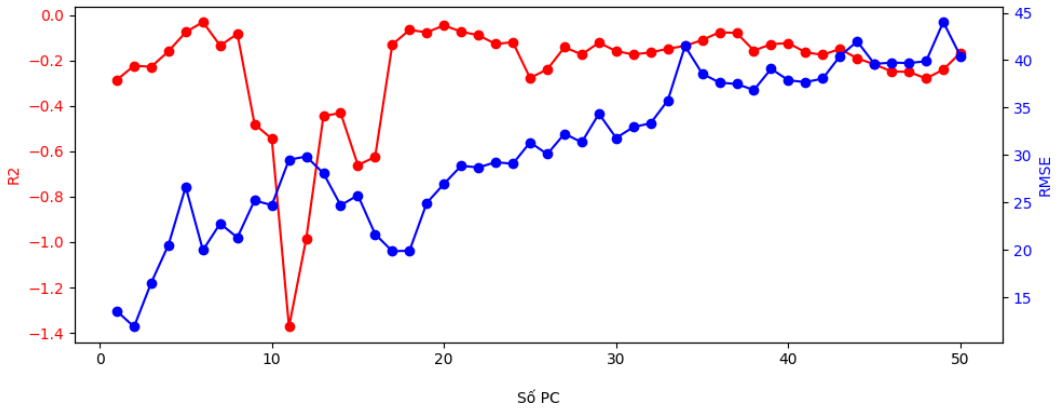
Thuật toán PCR và PLS

Khảo sát giá trị tổng % phương sai giải thích của dữ liệu phổ tập huấn luyện sử dụng thuật toán PCA, kết quả thu được ở hình 3.7:

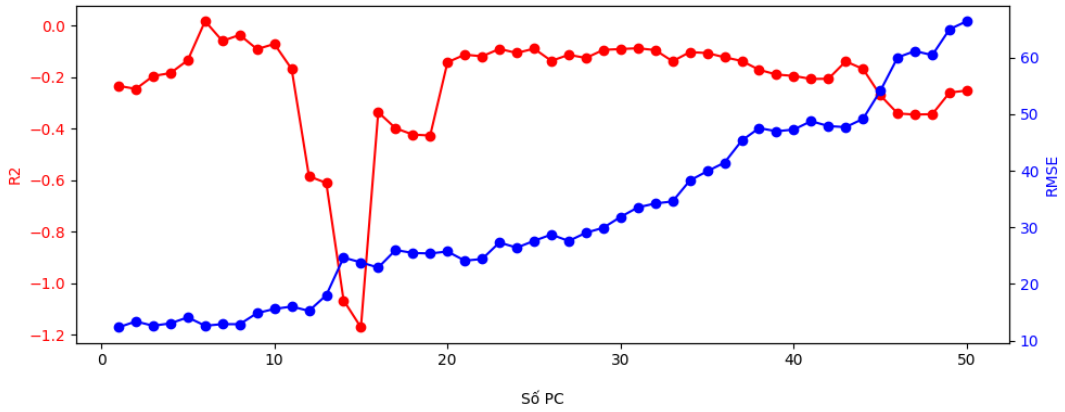


Hình 3.7 Giá trị tổng % phương sai giải thích theo từng cấu tử chính của tập huấn luyện

Do tác động của việc đạo hàm bậc 2, giá trị tổng % phương sai giải thích thường sẽ tăng chậm hơn so với việc chạy PCA trên dữ liệu gốc, làm cho ở 7 PC đầu giá trị tổng % phương sai giải thích vẫn dưới 95%, phải từ từ 8PC trở đi tổng % phương sai giải thích mới lớn hơn 95% và số PC càng cao thì tổng % phương sai giải thích càng tiệm cận 100%. Vì vậy phải lựa chọn số cấu tử chính từ 8PC trở lên. Tuy nhiên, để xác định số PC phù hợp cho thuật toán PCR cần phải tiến hành đánh giá chéo với mô hình PCR sử dụng phương pháp KFold. Với lượng dữ liệu ở tập huấn luyện khá lớn, có thể sử dụng phương pháp đánh giá chéo trên tập dữ liệu huấn luyện, bằng cách tách một phần nhỏ của tập dữ liệu huấn luyện ra gọi là tập dữ liệu kiểm chứng (validation set). Mô hình sẽ giảm chiều về từng PC và hồi quy trên lượng dữ liệu kiểm chứng này bằng cách học trên tập dữ liệu còn lại của tập huấn luyện. Kết quả các giá trị R^2 và RMSE đánh giá chéo trên từng PC của mô hình PCR và mô hình PLSR được thể hiện ở hình 3.8 và hình 3.9:



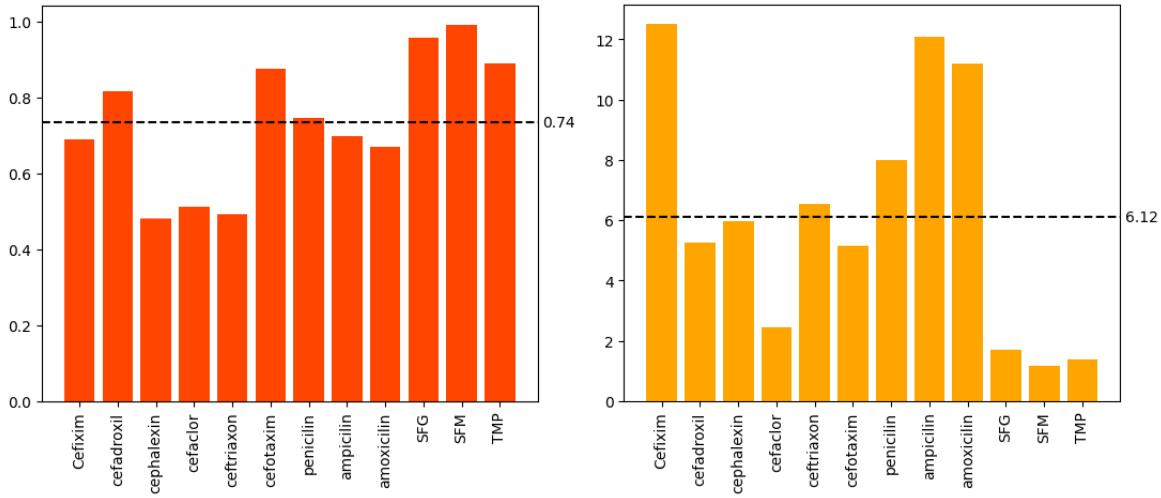
Hình 3.8 Ảnh hưởng của số cấu tử chính (PC) lên giá trị R^2 và RMSE của mô hình PCR



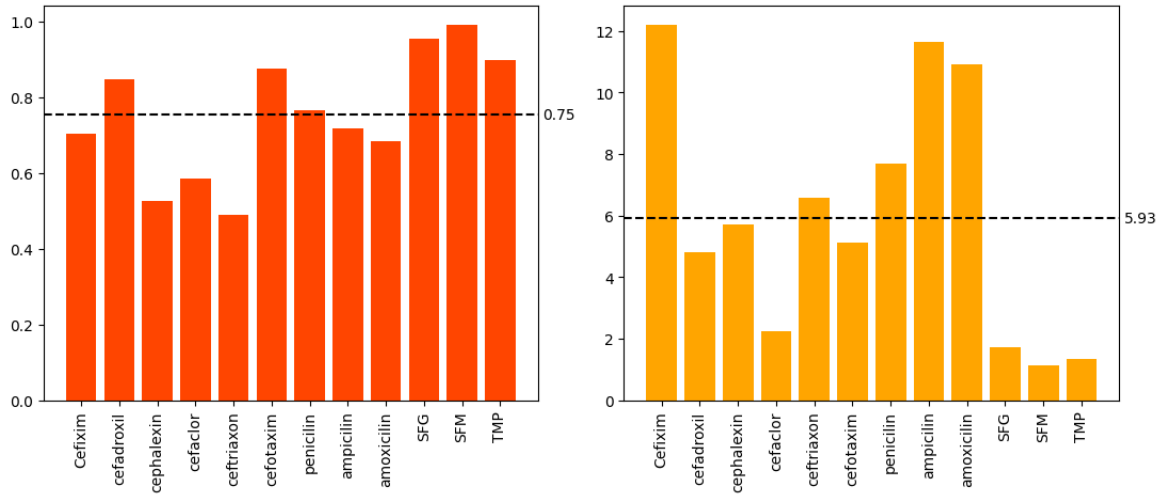
Hình 3.9 Ảnh hưởng của số cấu tử chính (PC) lên giá trị R^2 và RMSE của mô hình PLSR

Ở cả hai mô hình, có một xu hướng chung được thể hiện là khi số PC càng tăng thì giá trị RMSE đánh giá chéo của các mô hình cũng tăng theo, dù giá trị R^2 có xu hướng ổn định hơn. Điều đó chứng tỏ rằng với càng nhiều PC thì khả năng dự đoán của mô hình càng kém, sai số xảy ra lớn. Vì vậy để đạt được độ hiệu quả tốt nhất lựa chọn 8PC cho cả hai thuật toán PCR và PLS.

Xây dựng các mô hình PCR và PLSR với 8 cấu tử chính, tiến hành cho mô hình học toàn bộ tập dữ liệu huấn luyện và định lượng trên tập dữ liệu kiểm tra, kết quả được hiển thị trên hình 3.10 và hình 3.11:



Hình 3.10 Kết quả R^2 (trái) và RMSE (phải) của thuật toán PCR trên tập dữ liệu kiểm tra, đường nét đứt thể hiện giá trị trung bình



Hình 3.11 Kết quả R^2 (trái) và RMSE (phải) của thuật toán PLSR trên tập dữ liệu kiểm tra, đường nét đứt thể hiện giá trị trung bình

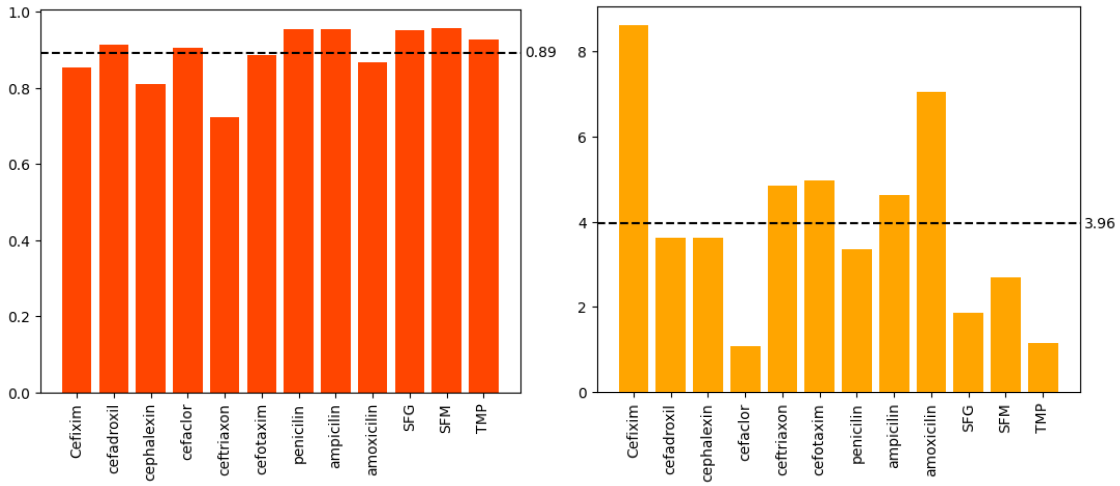
Giá trị R^2 trung bình của mô hình PCR là 0,74, với mô hình PLSR là 0,75 phân bố khá đều với các nhóm thành phần dự đoán. Tuy nhiên, sự phân bố giá trị RMSE của hai mô hình ở 12 đặc trưng đầu ra lại không đồng đều. Cả hai mô hình đều có giá trị RMSE thấp ở SFG, SFM, TMP (nhóm sulfamid). Dù giá trị RMSE trung bình ở tập PLSR là 5,93, thấp hơn của mô hình PCR nhưng sự khác biệt đó là không đáng kể thì sai số thể hiện sự bất đồng đều của hai mô hình này khi định lượng đồng thời 12 loại hoạt chất.

Thuật toán PCA-RandomForest

Xây dựng một mô hình liên kết PCA-RandomForest sử dụng đối tượng Pipeline của thư viện scikit-learn kết hợp thuật toán GridSearchCV để khảo sát các tham số của mô hình liên kết PCA-RandomForest như số thành phần chính, số mô hình con và chiều sâu tối đa. Sau khi tiến hành chạy thuật toán khảo sát thu được các tham số tối ưu là 8 thành phần chính cho PCA, 10 mô hình con có độ sâu tối đa là 9 cho thuật toán RandomForest

Xây dựng mô hình PCA-Randomforest dựa trên các tham số tối ưu đã tìm ra và cho mô hình học toàn bộ tập dữ liệu huấn luyện. Kết quả R^2 và RMSE của mô hình PCA-

RandomForest xây dựng bởi các tham số tối ưu và học trên tập dữ liệu kiểm tra được hiển thị trên hình 3.12:



Hình 3.12 Kết quả R^2 (trái) và RMSE (phải) của thuật toán PCA-RandomForest trên tập dữ liệu kiểm tra, đường nét đứt thể hiện giá trị trung bình

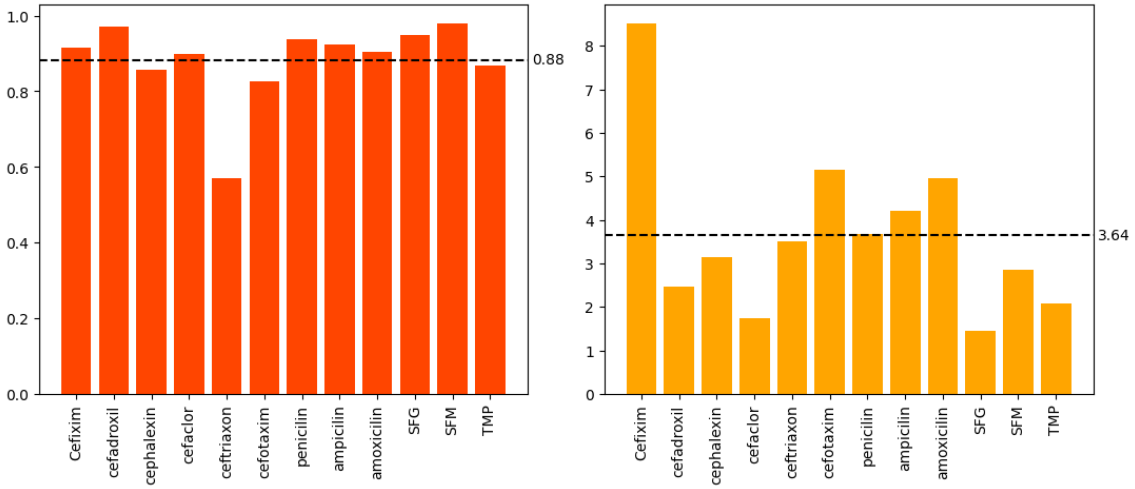
Mô hình đem lại giá trị R^2 khá cao là 0.89 cùng với giá trị RMSE trung bình là 3,96, giá trị sai số tại các thành phần cefadroxil, cephalexin, cefaclor, cefotaxime đều nhỏ hơn so với thuật toán PCR và PLSR, chứng tỏ khả năng dự đoán tốt hơn của mô hình PCA-Randomforest khi định lượng đồng thời 12 chất. Tuy nhiên sai số khá cao ở nhóm amoxicillin và nhóm Cefixim, chứng tỏ rằng hai nhóm chất này vẫn là một thách thức đối với mô hình hồi quy.

Thuật toán PCA-ANN

Cùng với thuật toán GridSearchCV, xây dựng một mô hình liên kết PCA-ANN sử dụng đối tượng Pipeline của thư viện scikit-learn, các tham số trong mô hình này đã được khảo sát để đạt được khả năng dự đoán tối ưu bao gồm số thành phần chính là 20, mô hình ANN có 2 lớp ẩn với 24 node mỗi lớp, hàm kích hoạt ReLU và cho phép điểm dữ liệu đi qua mỗi lần học, sử dụng thuật toán tối ưu hóa Adam kết hợp với chiến thuật dừng sớm để tránh hiện tượng quá khớp.

Với các tham số tối ưu đã tìm ra, mô hình ANN được xây dựng với hàm Relu là hàm kích hoạt sử dụng thuật toán tối ưu hóa Adam, cấu trúc mô hình bao gồm 2 lớp ẩn, mỗi lớp chứa 24 nơ ron. Dữ liệu trước được đưa vào mô hình ANN sẽ được giảm chiều về 20 cấu tử chính, điều này có thể chấp nhận được do dữ liệu huấn luyện có tới 142

mẫu, lớn hơn rất nhiều so với số cấu tử chính đã lựa chọn. Cho mô hình học trên tập huấn luyện, kết quả R^2 và RMSE trên tập kiểm tra của mô hình được biểu thị trên hình 3.13:



Hình 3.13 Kết quả R^2 (trái) và RMSE(phải) của thuật toán PCA-ANN trên tập dữ liệu kiểm tra, đường nét đứt thể hiện giá trị trung bình

So với mô hình hình liên kết PCA-RandomForest, mô hình PCA-ANN cần nhiều số thành phần chính hơn nhưng chỉ đem lại kết quả không quá khác biệt khi giá trị R^2 là 0,88 (với PCA-RandomForest là 0.89) và giá trị RMSE là 0,64 (với PCA-RandomForest là 0.96). Tỷ lệ phân bố sai số ở các thành phần dự đoán đã đồng đều hơn, điều đó chứng tỏ mô hình PCA-ANN với cấu trúc mô hình phức tạp hơn là mô hình ổn định nhất trong số các mô hình học máy xem xét sử dụng. Tuy nhiên giá trị RMSE của mô hình ở thành phần Cefixim vẫn rất cao so với các thành phần khác, đồng thời giá trị sai số trung bình là 3,64 là chưa đủ tốt bởi trong thực tế sẽ chỉ lấy ra những giá trị thành phần mà nhà sản xuất công bố rồi đối chứng với hàm lượng ghi trên bao bì, vậy nên vẫn cần phải khảo sát mô hình có sai số thấp hơn và đồng đều giữa các thành phần hơn.

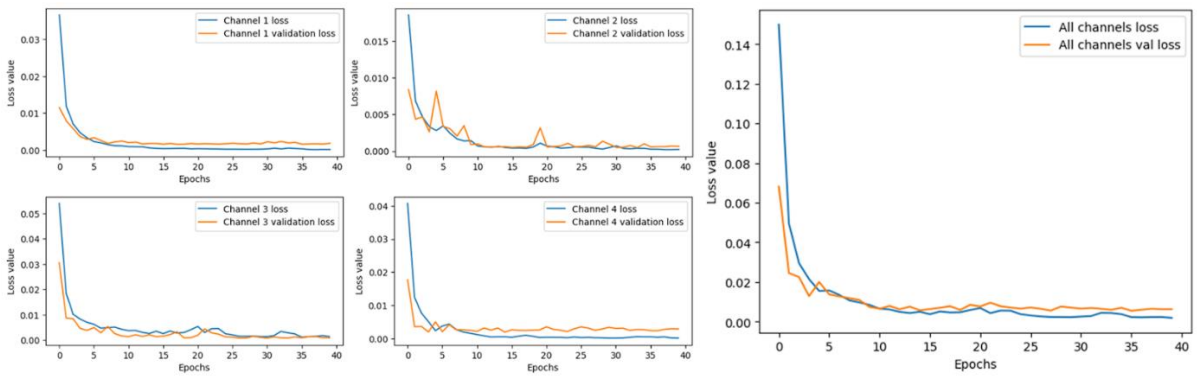
3.1.2.2 Phương pháp định lượng đồng thời sử dụng mô hình học sâu 1D-CNN

Trước khi xây dựng mô hình 1D-CNN đa kênh, để cho mô hình hội tụ nhanh hơn, dữ liệu đầu ra được chuẩn hóa bằng hàm MinMaxScaler để các giá trị đầu ra chỉ dao động trong khoảng 0 đến 1. Mô hình cấu trúc 1D-CNN yêu cầu dữ liệu đầu vào phải có dạng tensor, vậy nên các ma trận dữ liệu sẽ được thêm một chiều không gian vào để trở thành tensor (hình dạng tensor ba chiều để đưa vào mô hình là chiều sâẫu chiều dài x

chiều rộng). Sau khi tensor hóa, dữ liệu huấn luyện có kích thước là 142x132x1, dữ liệu kiểm tra có kích thước 65x312x1.

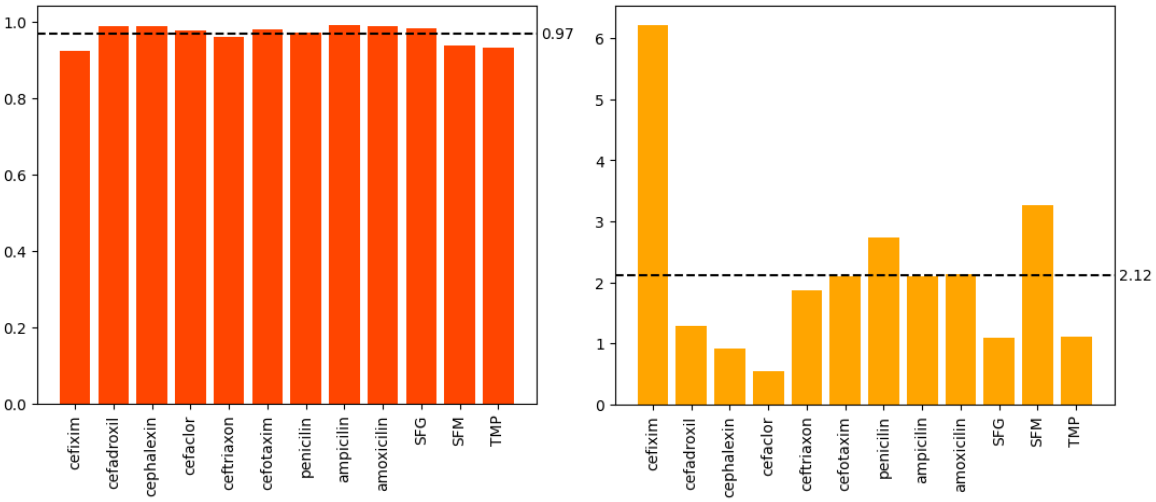
Mô hình 1D-CNN đa kênh có cấu trúc được hiển thị trong mục lục được xây dựng bằng thư viện tensorflow, các siêu tham số được khởi tạo bao gồm: 40 bước học với 12 điểm dữ liệu được học mỗi bước, sử dụng thuật toán tối ưu Adam và hàm mất mát MSE (Mean squared error). Hàm MSE đo đạc sự sai khác giữa giá trị thực tế và giá trị dự đoán trong mỗi bước học, từ đó sử dụng các giá trị mất mát này, thuật toán Adam sẽ tiến hành tối ưu hóa để khiến cho giá trị mất mát này giảm đến cực tiểu. Việc sử dụng thuật toán Adam cho các thuật toán học sâu đỡ trở nên phổ biến [29] khi càng ngày các mô hình học sâu càng mở rộng với nhiều tham số hơn, thuật toán Adam vẫn có khả năng giúp các mô hình đạt được sự hội tụ với tốc độ rất nhanh và tránh được các lỗi thường gặp như sự bùng nổ gradient hay sự tiêu biến gradient. Hiện tại, mô hình 1D CNN đa kênh được xây dựng chứa tới 1.997.580 tham số, trong khi mô hình PCA-ANN được sử dụng trong phần 3.1.2.1 chỉ chứa tới 138.240 tham số, vậy tức là mô hình 1D-CNN đa kênh phức tạp gấp 14 lần so với mô hình PCA-ANN, với một lượng tham số lớn như vậy, việc sử dụng thuật toán Adam là cần thiết. Tham số batch_size=12 tương ứng với việc sử dụng 12 mẫu dữ liệu cho mỗi lần huấn luyện, giá trị batch_size cao hơn tức là trong mỗi lần học mô hình có thể nhìn thấy nhiều dữ liệu hơn khiến mô hình học nhanh hơn tuy nhiên điều này có thể gây ra tình trạng quá khớp, đặc biệt số lượng trong tập dữ liệu huấn luyện là có giới hạn. Ngược lại, nếu batch_size nhỏ thì mô hình sẽ nhìn thấy ít dữ liệu trong mỗi lần học hơn, khiến nó phải học nhiều hơn để lặp qua được toàn bộ dữ liệu, làm cho mô hình đạt được đến sự hội tụ tốt hơn, tuy nhiên thời gian luyện mô hình sẽ rất lâu. Việc lựa chọn batch_size bằng 12 là một sự cân đối để khiến cho mô hình không bị xảy ra tình trạng quá khớp mà vẫn đảm bảo thời gian luyện mô hình.

Cho mô hình học tập dữ liệu huấn luyện với 40 bước học, kết quả giá trị mất mát của mô hình được hiển thị ở hình 3.14:



Hình 3.14 Giá trị MSE của tập dữ liệu huấn luyện (màu xanh) và của tập dữ liệu kiểm tra (cam) theo từng bước học

Việc học tập dữ liệu ở bốn kênh diễn ra thuận lợi khi ở cả bốn kênh, giá trị mất mát của tập kiểm tra không bị tăng dần theo từng bước học, từ đó có thể nhận định rằng mô hình đang không bị quá khớp. Đồ thị hàm mất mát ở cả bốn kênh có xu hướng đạt tới giá trị nhỏ nhất rồi giữ nguyên, riêng chỉ có kênh số 2 trong quá trình học tập dữ liệu đôi lúc có sự không ổn định, thể hiện ở các đỉnh gai từ bước học số 3 tới bước học số 20 tuy nhiên hiện tượng này là bình thường đối với việc lựa chọn số batch_size không lớn, tới các bước học cuối đồ thị đã ổn định trở lại. Vậy vậy có thể nhận định, mô hình 1D CNN đã đạt đến sự tối ưu. Giá trị R^2 và RMSE của mô hình trên tập dữ liệu kiểm tra được hiển thị trong hình 3.15:



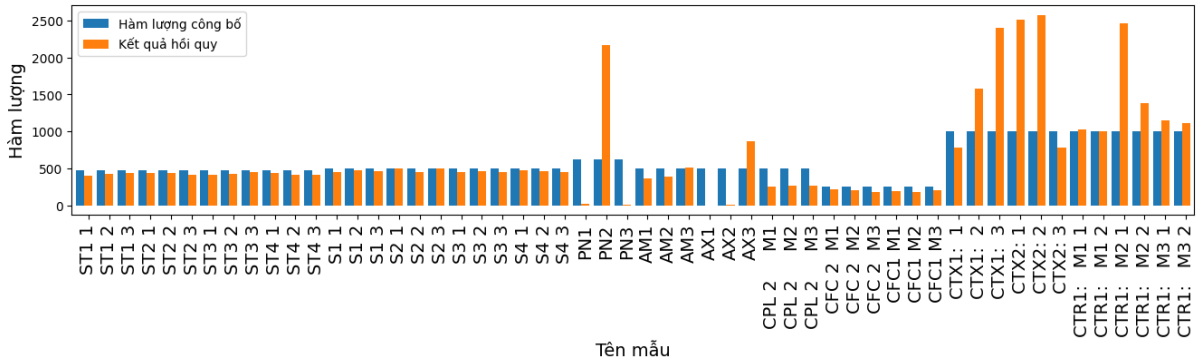
Hình 3.15 Kết quả R^2 (trái) và RMSE(phải) của mô hình 1D-CNN đa kênh trên tập dữ liệu kiểm tra, đường nét đứt thể hiện giá trị trung bình

So với các thuật toán học máy, giá trị R^2 trung bình của mô hình 1D-CNN (0,97) hiện là cao nhất, đồng thời giá trị RMSE của 1D-CNN là giá trị thấp nhất (2,12). Nhìn chung xu hướng sai số thấp phân bố ở khá nhiều các thành phần dự đoán, chứng tỏ khả năng định lượng đồng thời của mô hình học sâu 1D-CNN tốt hơn rất nhiều so với khi sử dụng các thuật toán học máy. Tuy nhiên mô hình vẫn chưa thể khắc phục được hiện tượng sai số lớn của nhóm Cefixim (dù sai số ở nhóm cefixim của mô hình 1D-CNN chỉ bằng một nửa so với mô hình PCR và PLSR), theo sau là nhóm SFM và nhóm penicillin. Để giúp cho việc học các thành phần một cách đồng đều và định lượng đồng thời với sai số thấp, mô hình 1D-CNN cần phải khảo sát cải tiến và thay đổi về cấu trúc và cơ chế huấn luyện.

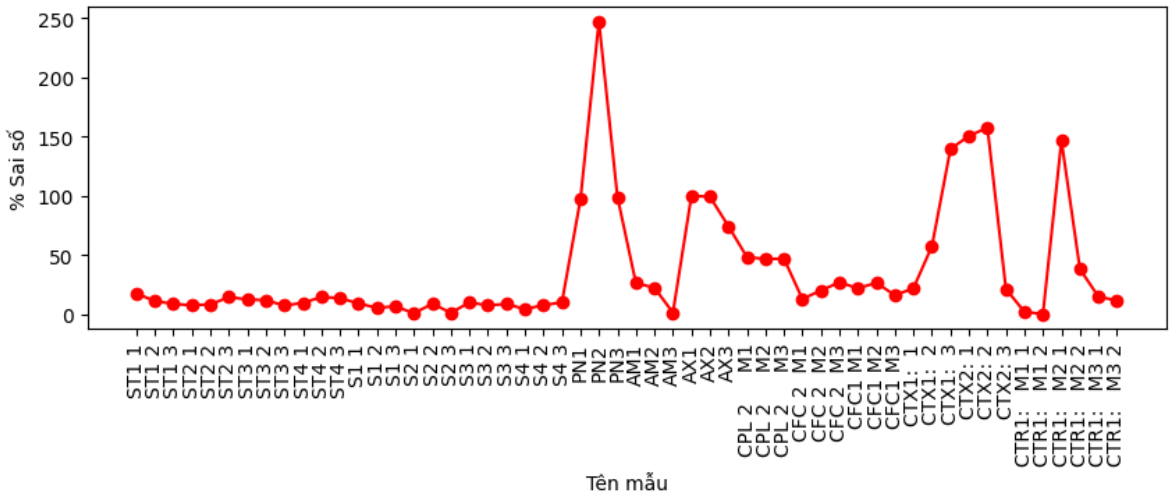
Với các lý do trên, mô hình 1D-CNN được lựa chọn để tiến hành khảo sát định lượng đồng thời các nhóm thuốc kháng sinh trong mẫu thực.

3.1.2.3 Định lượng đồng thời các nhóm thuốc kháng sinh trong mẫu thuốc bằng mô hình 1D-CNN

Phổ của 54 mẫu thực sau khi tiền xử lý được đưa vào trong mô hình 1D-CNN đã được học tập huấn luyện, kết quả hàm lượng hoạt chất/viên được tính bằng cách lấy ra nhóm chất của mẫu thực (thông tin nhóm chất được ghi cùng bảng tính với phổ mẫu thực) sau đó được tính sử dụng công thức đã được đề cập ở mục 2.1.2.3. Giá trị hàm lượng/viên mô hình xác định được biểu thị trên hình 3.16, sai số tương đối so sánh với hàm lượng công bố từ nhà sản xuất được biểu thị trên hình 3.17:



Hình 3.16 Hàm lượng hoạt chất (mg) có trong một viên thuốc được tính từ mô hình 1D-CNN và hàm lượng được công bố từ nhà sản xuất



Hình 3.17 Sai số tương đối (%) của hàm lượng chất (mg) trong một viên thuốc được tính từ mô hình 1D-CNN với hàm lượng công bố từ nhà sản xuất

Đối với các mẫu thuộc nhóm sulfamid (Từ ST 1 với S4 3) sai số hàm lượng dự đoán và hàm lượng công bố là rất thấp, không có mẫu nào vượt quá 20% sai số. Tuy nhiên ở các mẫu ngoài thuộc nhóm khác, % sai số tương đối cao, đặc biệt ở các mẫu: PN1, PN2, PN3 (nhóm penicilline); AX1, AX2, AX3 (nhóm amoxicilin) và các mẫu CTX1, CTX2, CTX3 thuộc nhóm cefotaxim. Chứng tỏ rằng mô hình chỉ hoạt động ổn định khi định lượng đồng thời các hoạt chất SFG, SFM và TMP thuộc nhóm sulfamid còn đối với các nhóm thuốc khác khả năng định lượng đồng thời của mô hình gây ra sai số rất cao, dù trên tập dữ liệu kiểm tra một số hoạt chất có giá trị RMSE thấp hơn các hoạt chất trong nhóm sulfamid. Điều này có thể do sự tương tác giữa các mẫu chuẩn với tá dược làm thay đổi tín hiệu bên trong phổ mẫu thực, sự thay đổi này tiếp tục bị khuếch đại bởi việc đạo hàm phổ khiến cho tín hiệu đầu vào, do đó chỉ với việc chất chuẩn bị lấy ít hay nhiều cũng ảnh hưởng lớn đến sai số của mô hình (các mẫu CTR1).

Việc sử dụng mô hình 1D-CNN có tiềm năng lớn trong việc hồi quy phân tích đồng thời với nhiều đặc trưng đầu ra, có khả năng phân tích và dự đoán chính xác hơn so với các mô hình học máy. Tuy nhiên, để ứng dụng mô hình 1D-CNN đa kênh để có thể tiến tới định lượng trong các mẫu thực khác nhau với nền tá dược phức tạp cần có

thêm nhiều nghiên cứu, khảo sát và nâng cấp các mô hình, đồng thời thu thập thêm dữ liệu sao cho đa dạng để giúp cho mô hình học một cách chính xác hơn.

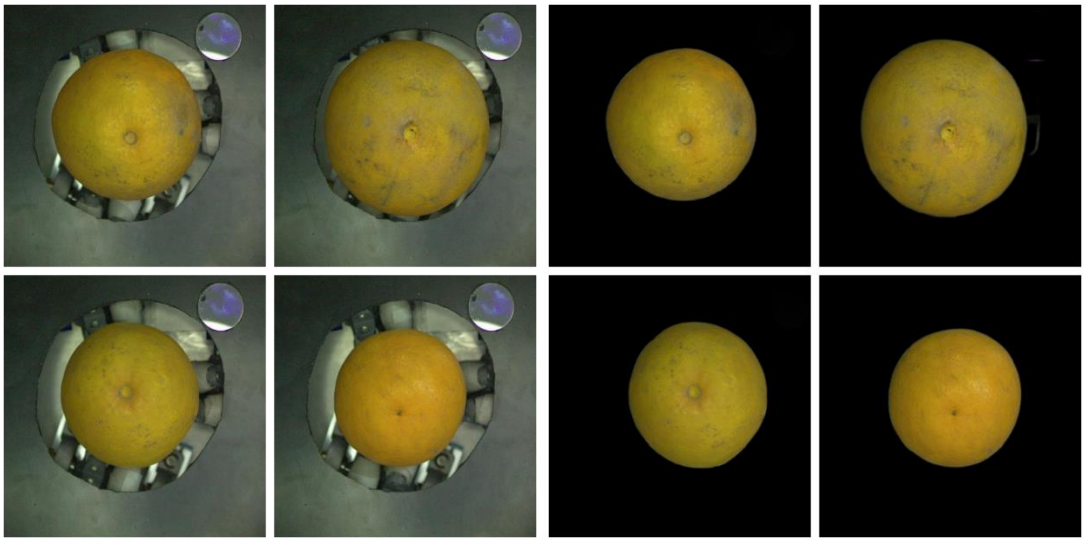
Kết luận

Mô hình học sâu đa kênh 1D-CNN có sai số thấp hơn rất nhiều so với các mô hình học máy trong nhiệm vụ phân tích đồng thời hàm lượng của 12 hoạt chất. Tuy nhiên việc định lượng đồng thời vẫn gặp khó khăn khi sai số của thành phần Cefixim vẫn cao hơn rất nhiều so với các thành phần còn lại, đồng thời khi sử dụng mô hình 1D-CNN để định lượng đồng thời các mẫu thực thì mô hình trở nên không ổn định và chỉ dự đoán tốt với các mẫu thực chứa thuốc thuộc nhóm sulfamid. Cần cải thiện cơ sở dữ liệu sao số lượng mẫu thuốc tăng lên và hàm lượng thành phần mẫu trong tập dữ liệu trở nên đa dạng hơn, đồng thời khảo sát nâng cấp cấu trúc mô hình để đưa ra các dự đoán chính xác hơn phù hợp với nhu cầu ứng dụng thực tế.

3.2 Xác định hàm lượng đường trong cam sử dụng thị giác máy tính và phân tích hình ảnh

3.2.1 Kết quả phân đoạn dữ liệu hình ảnh cam

Dữ liệu hình ảnh gốc được đọc bằng thư viện OpenCV thông qua ngôn ngữ lập trình Python dưới dạng ma trận 900x900 pixel. Tuy nhiên để tối ưu hóa dung lượng hình ảnh và tăng tốc cho quá trình huấn luyện mô hình mà vẫn đảm bảo không làm mất đi thông tin đặc trưng quan trọng, việc giảm kích thước hình ảnh xuống 256x256 pixel là một bước quan trọng. Quy trình này không chỉ giúp giảm tải bộ nhớ mà còn cải thiện hiệu suất tính toán, đảm bảo mô hình học sâu hoạt động hiệu quả hơn. Dữ liệu ảnh cam được chụp bởi hệ và sau khi tách nền được hiển thị ở hình 3.18:



Hình 3.18 Dữ liệu ảnh chụp của quả cam trên hệ (trái) và ảnh sau khi quả cam được phân đoạn và tách nền (phải)

Mô hình phân đoạn ELUNet đã chứng minh tính hiệu quả vượt trội về cả độ chính xác và tốc độ thực thi so với các mô hình phân đoạn khác. Điểm đáng chú ý là ELUNet có số lượng tham số giảm đáng kể so với mô hình tiền nhiệm của nó, U-Net, điều này góp phần làm giảm độ phức tạp tính toán và tiết kiệm tài nguyên hệ thống. Kết quả phân đoạn của mô hình ELUNet, với những cải tiến về kiến trúc và hiệu suất, được thể hiện chi tiết trong bảng 3.6, minh họa rõ ràng sự cải thiện về độ chính xác và tốc độ xử lý với hình ảnh cam.

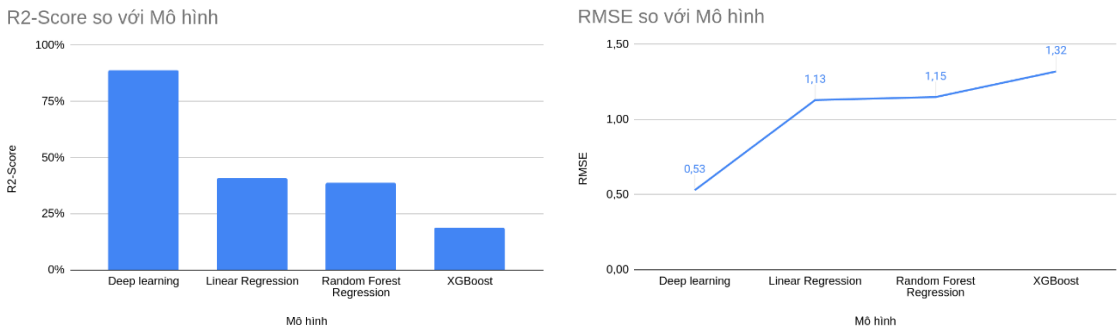
Bảng 3.6 Kết quả phân đoạn của mô hình ELUNet với hình cảnh quả cam

Accuracy	F1-Score	Precision	Recall	Thời gian chạy (s)
96,23%	95,81%	95,15%	96,81%	0,10

3.2.2 Kết quả xác định hàm lượng đường trong quả cam sử dụng kết quả hình ảnh đã phân đoạn

Dựa trên hình 3.19 biểu diễn kết quả dự đoán của các mô hình khác nhau, có thể thấy rằng mô hình học sâu đạt được R^2-Score cao nhất, xấp xỉ 90%. Điều này cho thấy việc sử dụng hình ảnh thay vì sử dụng dữ liệu đầu vào là các giá trị kênh màu RGB để dự đoán độ ngọt của quả cam thông là rất hiệu quả. Nhờ khả năng xử lý và phân tích các

đặc điểm phức tạp như vùng bị rám trong hình ảnh, mô hình học sâu có thể khai thác nhiều thông tin hơn so với các mô hình sử dụng giá trị RGB đơn giản.



Hình 3.19 Kết quả R^2 (trái) và RMSE (phải) của các mô hình dự đoán độ đường trong cam

Mô hình hồi quy tuyến tính đạt được R^2-Score khoảng 45%, mặc dù không đạt hiệu quả cao như mô hình học sâu, nhưng vẫn là kết quả chấp nhận được khi chỉ sử dụng thông tin RGB, tuy nhiên, linear regression có thể không đủ mạnh để nắm bắt hết các mối quan hệ phi tuyến tính phức tạp giữa giá trị RGB và độ ngọt.

Mô hình hồi quy rừng ngẫu nhiên có R^2-Score cao hơn mô hình XGBoost - một mô hình thường đạt hiệu suất rất cao trong các bài toán khác nhau, nhưng vẫn thấp hơn đáng kể so với học sâu. Mô hình có khả năng xử lý các mối quan hệ phức tạp hơn so với linear regression, nhưng dường như vẫn chưa đủ để đạt hiệu quả tối ưu trong việc dự đoán độ ngọt từ giá trị RGB.

Bên cạnh kết quả R^2-Score như hình 3.19, giá trị RMSE cũng đã cũng khẳng định sự hiệu quả của mô hình mô hình học sâu với giá trị RMSE thấp nhất là 0.53. Ngược lại, các mô hình khác như hồi quy tuyến tính, hồi quy rừng ngẫu nhiên và XGBoost có giá trị RMSE cao hơn. RMSE cao hơn biểu thị độ sai số lớn hơn trong dự đoán, cho thấy các mô hình này kém hiệu quả hơn so với deep learning.

Kết hợp hai kết quả trên, có thể kết luận rằng mô hình học sâu không chỉ dự đoán chính xác hơn (R^2-Score cao hơn) mà còn có độ sai số thấp hơn (RMSE thấp hơn) so với các mô hình sử dụng giá trị RGB. Điều này nhấn mạnh rằng việc sử dụng dữ liệu hình ảnh kết hợp mô hình học sâu là phương pháp tốt nhất cho bài toán dự đoán độ ngọt của quả cam. Các mô hình sử dụng giá trị RGB mặc dù có hiệu quả nhất định, nhưng không thể đạt được độ chính xác và hiệu quả như mô hình deep learning. Sự khác biệt này bởi

khả năng của mô hình học sâu trong việc xử lý và phân tích các đặc điểm phức tạp trong hình ảnh, khai thác nhiều thông tin hơn so với chỉ sử dụng các giá trị RGB đơn giản.

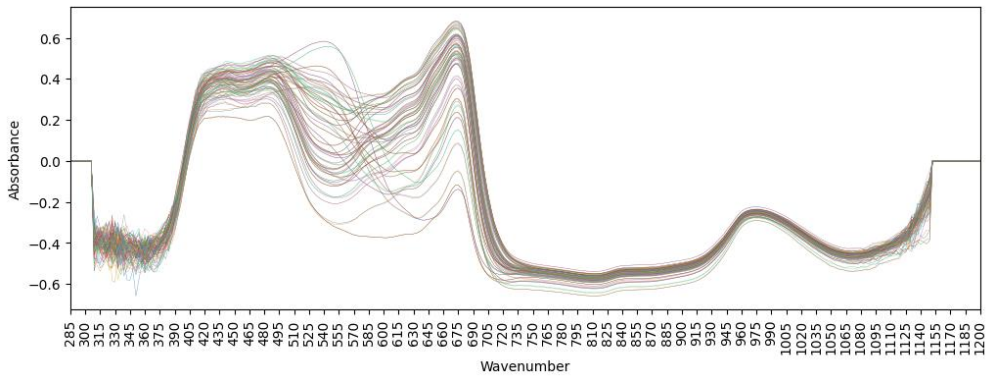
Mặt khác, khi xét về cấu trúc của mô hình MobileNet V2, việc tính toán không cần thiết đã được lọc bỏ, số lượng tham số của mô hình có thể được giảm bớt lên tới một triệu tham số, tuy nhiên vẫn giữ được khả năng trích xuất đặc trưng mạnh mẽ từ hình ảnh, điều này cho phép học tập tốt đối với dữ liệu hình ảnh quả cam trong thực tế khi xuất hiện các vết nám. Ngoài ra các tham số của mô hình đã được xác định thông qua việc đào tạo từ tập dữ liệu lớn trước đó nhằm tăng khả năng học tập và dự đoán.

Kết luận

Việc xác định hàm lượng đường trong cam bằng các thuật toán học máy có tốc độ nhanh, tiện lợi và không tốn nhiều tài nguyên, tuy nhiên độ chính xác dự đoán thường thấp hơn đáng kể so với khi sử dụng thuật toán học sâu. Thuật toán học sâu dựa trên cấu trúc MobileNet V2 với các tham số tích chập đã được huấn luyện trước giúp cho mô hình có cấu hình nhẹ, phù hợp để tích hợp vào các thiết bị xử lý có ít tài nguyên nhưng vẫn đảm bảo được độ chính xác cao và hiệu suất hồi quy tốt. Công việc chuẩn hóa quy trình chụp ảnh, tiền xử lý ảnh, lựa chọn đối tượng cam và nâng cấp mô hình cần được xem xét và nghiên cứu thêm để cải thiện hiệu suất dự đoán, từng bước tiến tới ứng dụng trong quy mô thực tiễn.

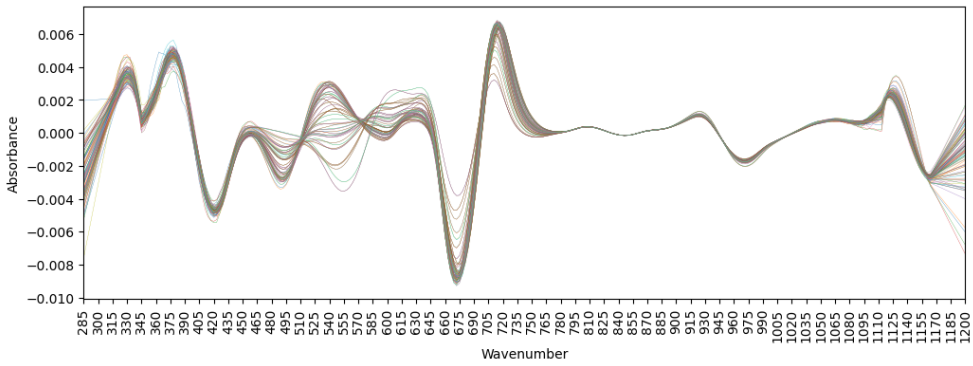
3.3 Phân loại các đặc điểm của xoài dựa trên phổ Vis-NIR với thuật toán học sâu

Đọc dữ liệu dạng bảng từ file dữ liệu, tách các dữ liệu thông tin về đầu ra bao gồm nguồn gốc, độ chín, mùa và nhiệt độ ra và lưu vào một biến riêng để phục vụ cho việc xây dựng mô hình và kiểm tra mô hình. Dữ liệu phổ được tách ra thành một ma trận có dạng 11691x306 và được biểu thị ở hình 3.20:



Hình 3.20 Phổ Vis-NIR của 100 mẫu xoài đầu tiên trong tập dữ liệu

Hiện tượng nhiễu xuất hiện ở các khoảng 350-390 cm^{-1} và 1065-1155 cm^{-1}, các peak trong vùng 390-630 cm^{-1} bị chồng chéo lên nhau, đồng thời do sử dụng thiết bị cầm tay để lấy tín hiệu của mẫu nên các ảnh hưởng của thiết bị cũng thể hiện trên phổ dữ liệu. Các yếu tố này chính là các nguy cơ dẫn đến việc gây khó khăn cho khả năng phân loại mô hình, khiến mô hình khó tối ưu và làm giảm tốc độ hội tụ của mô hình. Để khắc phục điều này, tiến hành đạo hàm bậc hai và làm mượt phổ sử dụng thuật toán Savitzky-Golay thu được phổ trên hình 3.21:



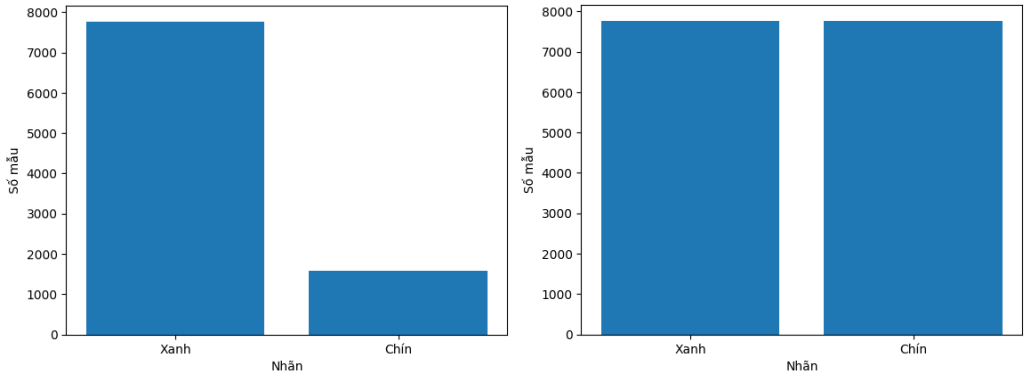
Hình 3.21 Phổ sau khi đạo hàm và làm mượt của 100 mẫu xoài đầu tiên trong tập dữ liệu

Sau khi đạo hàm và làm mượt, các hiện tượng nhiễu phổ đã biến mất, đồng thời các đặc trưng của mẫu sẽ được làm rõ hơn bằng cách đạo hàm bậc 2, điều này sẽ khiến cho mô hình dễ nhận dạng các mẫu theo nhãn hơn. Đồng thời các ảnh hưởng của thiết bị và môi trường đã được loại bỏ để khiến việc phân loại của mô hình trở nên ổn định hơn. Sau đó tiếp tục chuẩn hóa dữ liệu phổ bằng StandardScaler để tối đa hóa hiệu suất của các mô hình.

Tiến hành chia tập dữ liệu thành tập huấn luyện và tập kiểm tra, với tập kiểm tra chiếm 20% số mẫu của tập ban đầu, tương đương với 9352 mẫu luyện và 2339 mẫu kiểm tra, sử dụng câu lệnh train_test_split của thư viện scikit-learn với tham số stratify=y để phân bố các lớp trong tập kiểm tra sẽ gần giống với phân bố các lớp trong tập dữ liệu ban đầu, từ đó có thể đánh giá mô hình một cách khách quan nhất.

3.3.1 Phân loại độ chín của xoài

Kiểm tra sự phân bố nhãn xanh và chín, đồ thị phân bố của các nhẫn trên tập dữ liệu huấn luyện được biểu thị trong hình 3.22:



Hình 3.22 Phân bố số lượng xoài xanh và chín trong tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải)

Số lượng mẫu xoài xanh trong tập huấn luyện là 7768 mẫu trong khi lượng xoài chín là 1584 mẫu, tức là số lượng mẫu lớn hơn số lượng mẫu xoài chín khoảng 5 lần, vì vậy dữ liệu đang bị bị mất cân bằng một cách trầm trọng. Nếu sử dụng nguyên tập dữ liệu để xây dựng mô hình phân loại thì mô hình sẽ có xu hướng thiên vị cho lớp xoài xanh, vì vậy thuật toán SMOTE được sử dụng để tạo ra các dữ liệu xoài chín mới, khiến cho dữ liệu cân bằng và đầy đủ hơn, từ đó khiến mô hình khách quan và đáng tin cậy hơn.

3.3.1.1 Kết quả phân loại bằng các thuật toán học máy

Tiến hành xây dựng các mô hình PCA-SVC, PLS-DA, PCA-DecisionTree, PCA-RandomForest và PCA-ANN với các tham số tối ưu được khảo sát bằng thuật toán GridSearchCV với số fold = 5 (chi tiết thông số được ghi trong phụ lục đính kèm) thu được kết quả tại bảng 3.7 như sau:

Bảng 3.7 Độ chính xác phân loại độ chín của xoài sử dụng các thuật toán học máy

Thuật toán	PCA SVC	PLS DA	PCA Decision Tree	PCA RandomForest	PCA ANN

None SMOTE	0,9790	0,9811	0,9589	0,9764	0,9841
SMOTE	0,9799	0,9841	0,9606	0,9769	0,9897

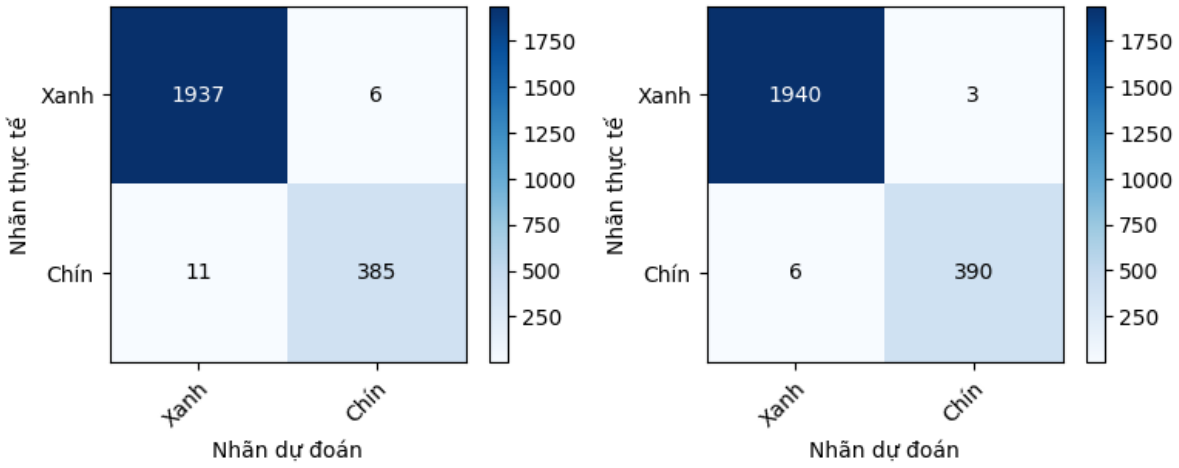
Kết quả cho thấy nhìn chung việc sử thuật kỹ thuật SMOTE để làm gia tăng số mẫu xoài chín khiến cho các mô hình có độ chính xác cao hơn, dù cách biệt là không đáng kể. Mô hình PCA-ANN có độ chính xác cao nhất, có thể lý giải với khả năng học tập dữ liệu một cách phức tạp của mô hình ANN đã được tối ưu tham số cấu trúc, đồng thời với số lượng dữ liệu lên đến hàng nghìn mẫu khiến mô hình ANN hình thành tri thức phân loại một cách khái quát hơn. Một mô hình nữa cũng có khả năng phân loại khá tốt là PLS DA với độ chính xác khi áp dụng SMOTE lên tới 0,9841, lý giải với khả năng phân loại tốt như vậy là do với mục đích phân loại xoài xanh hay chín, tức là với một bài toán phân loại nhị phân thì yêu cầu về khả năng tách dữ liệu trong không gian thường đơn giản hơn so với những bài toán phân loại đa lớp, ví vậy trong trường hợp này thuật toán PLS DA có hiệu suất khá tốt. Nhìn chung các thuật toán học máy đều phân loại ở độ chính xác trên mức 0,95, đây là một minh chứng cho khả năng hiệu quả của các phương pháp tiền xử lý dữ liệu phổ kết hợp với việc sử dụng SMOTE để cân bằng dữ liệu. Tuy nhiên để tối đa hóa hiệu suất phân loại cấu trúc mô hình cần trở nên phức tạp hơn vì vậy phải xem xét việc sử dụng mô hình học sâu cho phân loại độ chín của xoài.

3.3.1.2 Kết quả phân loại bằng thuật toán học sâu

Một điểm khác nhau ở đây so với việc sử dụng các thuật toán học máy chính là việc không tiến hành giảm chiều dữ liệu trước khi đưa vào mô hình mà mô hình sẽ có khả năng quét qua toàn bộ tập dữ liệu và trích xuất các đặc trưng cần thiết cho việc phân loại bằng các lớp tích chập. Điều này làm nổi bật lên tính ứng dụng của các thuật toán học sâu trong thực tế, khi cơ sở dữ liệu huấn luyện được cập nhật liên tục, vì vậy lúc đó việc tối ưu hóa các cấu tử chính để giảm chiều sẽ luôn luôn thay đổi và làm cho các mô hình trở nên không ổn định. Nếu một mô hình học sâu có thể luôn luôn nhận đầu vào là toàn bộ phổ dữ liệu, khi có dữ liệu mới chỉ cần truyền trực tiếp dữ liệu mới và nhãn của nó cho mô hình học dựa trên khả năng phân loại đã có sẵn của mô hình trước nó mà không cần phải huấn luyện lại trên toàn bộ tập dữ liệu, điều này làm giảm thiểu chi phí

tài nguyên duy trì mô hình, thời gian huấn luyện và đồng thời tăng tính ổn định của mô hình.

Tiến hành xây dựng mô hình 1D-CNN, độ chính xác của mô hình với dữ liệu chưa áp dụng SMOTE là 0,9927 và dữ liệu đã áp dụng SMOTE là 0,9961. Kết quả phân loại chi tiết được thể hiện hình 3.23:

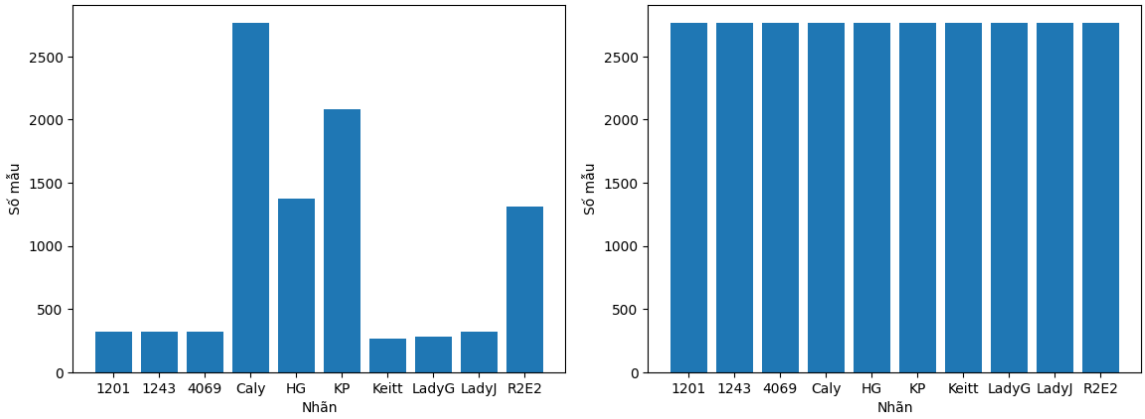


Hình 3.23 Kết quả phân loại theo độ chín dựa trên tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải)

Khả năng phân loại của mô hình 1D-CNN của cả hai trường hợp đều lớn hơn 0,99, điều này chứng tỏ tiềm năng của các lớp tích chập (trước đó đã được sử dụng phổ biến trong phân loại ảnh và thị giác máy tính) trong việc phân loại đối tượng dựa trên phổ của chúng, vượt qua khả năng phân loại của các mô hình học máy và đạt đến độ chính xác gần đến mức tuyệt đối. Việc áp dụng kĩ thuật SMOTE giúp giảm thiểu tới một nửa số lượng các mẫu bị phân loại sai (từ 17 mẫu xuống còn 9 mẫu), sau khi sử dụng SMOTE chỉ còn 6 mẫu xoài loại chín bị phân loại nhầm thành xanh, và chỉ 3 mẫu xoài xanh bị phân loại nhầm thành chính. Do sự chênh lệch giữa số lượng giữa hai loại xoài trong tập kiểm tra nên tỷ lệ xoài xanh/chín bị phân loại sai là hoàn toàn có thể chấp nhận được.

3.3.2 Phân loại giống của xoài

Kiểm tra sự phân bố giống của xoài, đồ thị phân bố của các nhãn trên tập dữ liệu huấn luyện được biểu thị trong hình 3.24:



Hình 3.24 Phân bố số lượng xoài theo giống trong tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải)

Trong tập dữ liệu gốc, chiếm ưu thế về số lượng là giống Caly, HG, KP và R2E2 với số lượng lần lượt là 2768, 2082, 1374 và 1308 mẫu. Các giống xoài còn lại đều có số lượng chưa tới 400 mẫu, điều này sẽ gây một khó khăn rất lớn cho mô hình vì bài toán lúc này đã trở thành bài toán phân loại đa lớp với 10 lớp. Số lớp phân loại càng lớn thì càng yêu cầu độ phức tạp cao của mô hình, đồng thời yêu cầu dữ liệu huấn luyện phải đủ đa dạng và mỗi lớp phải có nhiều mẫu. Tuy nhiên dữ liệu hiện tại đang bị mất cân bằng một cách nặng nề, khi số lượng mẫu bị phân bố không đồng đều và độ chênh lệch số mẫu giữa các lớp là quá cao (chênh lệch lớn nhất là gấp 10 lần), việc này sẽ khiến cho độ chính xác của các mô hình phân loại bị giảm đi. Để khắc phục vấn đề này cần sử dụng thuật toán SMOTE để làm cân bằng dữ liệu, tuy nhiên do sự chênh lệch lớn về số mẫu của các lớp phân loại có thể gây ra các sai sót trong quá trình tạo dữ liệu mới nên cần phải khảo sát và đánh giá tác động của thuật toán SMOTE một cách toàn diện.

3.3.2.1 Kết quả phân loại bằng các thuật toán học máy

Tiến hành xây dựng các mô hình học máy với các tham số tối ưu được khảo sát bằng thuật toán GridSearchCV với số fold = 5 (chi tiết thông số được ghi trong phụ lục đi kèm) thu được kết quả tại bảng 3.8 như sau:

Bảng 3.8 Độ chính xác phân loại giống của xoài sử dụng các thuật toán học máy

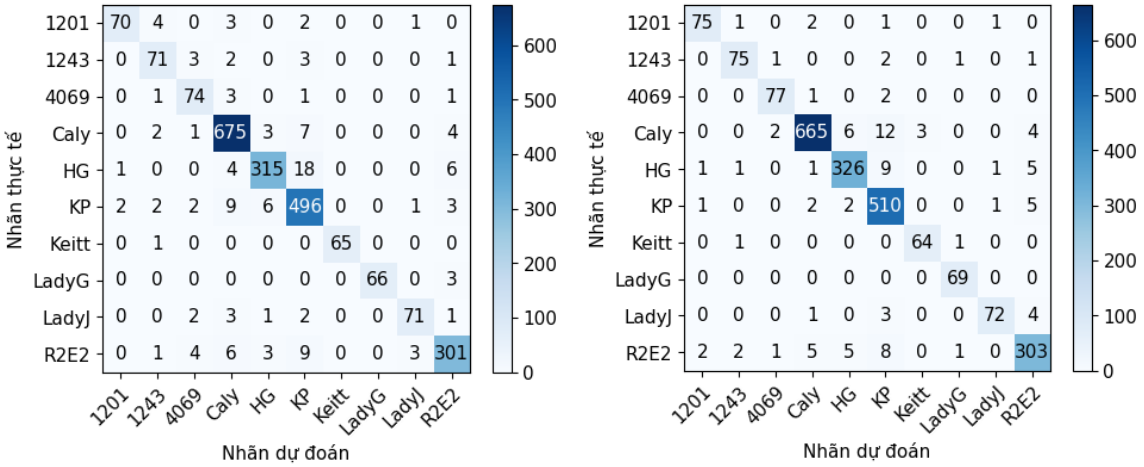
Thuật toán	PCA SVC	PLS DA	PCA Decision Tree	PCA RandomForest	PCA ANN

None SMOTE	0,725	0,032	0,672	0,847	0,826
SMOTE	0,729	0,034	0,693	0,853	0,851

Hiệu suất phân loại của các mô hình học máy giảm rõ rệt trong trường hợp này, không mô hình nào có đột chính xác từ 0,9 trở lên. Mô hình PLS-DA có độ chính xác thấp đến bất ngờ khi chỉ đạt tới 0,032 và 0,034, điều này có thể giải thích do khi tăng số lượng lớp phân loại lên sẽ làm cho các điểm dữ liệu trong không gian nằm ở các vị trí tương đối phức tạp, đồng nghĩa với việc mô hình PLS-DA gặp càng nhiều khó khăn hơn khi tìm ra siêu phẳng tối ưu. Các mô hình đáng tin cậy nhất vẫn là mô hình PCA-RandomForest với độ chính xác cao nhất với giá trị là 0,847 và 0,853, theo sau đó là mô hình PCA-ANN với độ chính xác lần lượt là 0,826 và 0,851. Sự có mặt của thuật toán SMOTE nhìn chung vẫn giúp cho các mô hình học máy tăng độ chính xác, từ đó có thể khẳng định rằng kỹ thuật SMOTE vẫn có ích trong trường hợp này.

3.3.2.2 Kết quả phân loại bằng thuật toán học sâu

Tiến hành xây dựng mô hình 1D-CNN, độ chính xác của mô hình với dữ liệu chưa áp dụng SMOTE là 0,9422 và dữ liệu đã áp dụng SMOTE là 0,9559. Kết quả phân loại chi tiết được thể hiện hình 3.25:

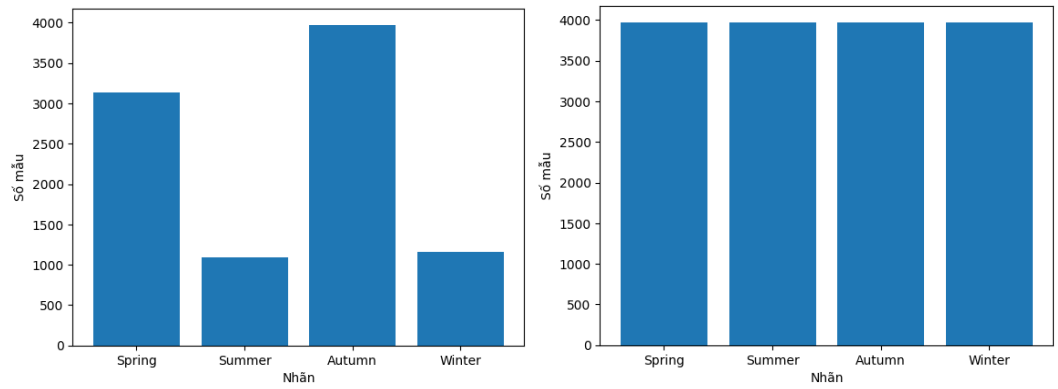


Hình 3.25 Kết quả phân loại theo giống dựa trên tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải) của mô hình 1D-CNN

Độ chính xác của mô hình khá cao với giá trị 0,9422 khi chưa áp dụng thuật toán SMOTE, thể hiện một sự chênh lệch rất lớn giữa mô hình học sâu với các mô hình học máy khi phân loại nhiều lớp. Các mẫu phân loại sai phân bố khá nhiều ở các giống xoài Caly, HG, KP và R2E2. Do đây là các giống xoài chiếm ưu thế về số lượng trong tập dữ liệu kiểm tra nên số lượng mẫu sai là có thể chấp nhận. Việc sử dụng kĩ thuật SMOTE khiến cho độ chính xác của mô hình tăng lên tới 0,0137, có tác động đáng kể để khả năng phân loại của mô hình. Nhìn chung việc làm gia tăng dữ liệu đã giúp cho mô hình giảm bớt số lượng các mẫu bị phân loại sai, tuy nhiên ở một số vị trí đặc biệt như các mẫu Caly bị phân loại nhầm thành KP, trước khi dùng SMOTE có 7 mẫu sai nhưng sau khi dùng SMOTE số mẫu sai đã tăng lên 12 mẫu. Hiện tượng này xảy ra có thể do sự đánh nhầm nhãn trong quá trình tạo dữ liệu mới, để khắc phục vấn đề này cần khảo sát thêm để đưa ra các chiến thuật SMOTE một cách hiệu quả.

3.3.3 Phân loại mùa thu hoạch

Kiểm tra sự phân bố số mẫu theo mùa, kết quả phân bố được hiển thị ở hình 3.26:



Hình 3.26 Phân bố số lượng xoài theo mùa thu hoạch trong tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải)

Số mẫu thu hoạch ở mùa đông và mùa hè ít hơn rất nhiều so với hai mùa còn lại, tuy sự chênh lệch giữa các mùa không lớn như khi phân loại theo độ chín và theo giống, tuy nhiên sự mất cân bằng dữ liệu vẫn có thể khiến cho mô hình thiên vị cho các mẫu xoài mùa xuân và mùa thu hơn.Tiến hành sử dụng SMOTE để số mẫu ở mỗi mùa đạt 3973 mẫu.

3.3.3.1 Kết quả phân loại bằng các thuật toán học máy

Tiến hành xây dựng các mô hình học máy với các tham số tối ưu được khảo sát bằng thuật toán GridSearchCV với số fold = 5 (chi tiết thông số được ghi trong phụ lục đính kèm) thu được kết quả tại bảng 3.9 như sau:

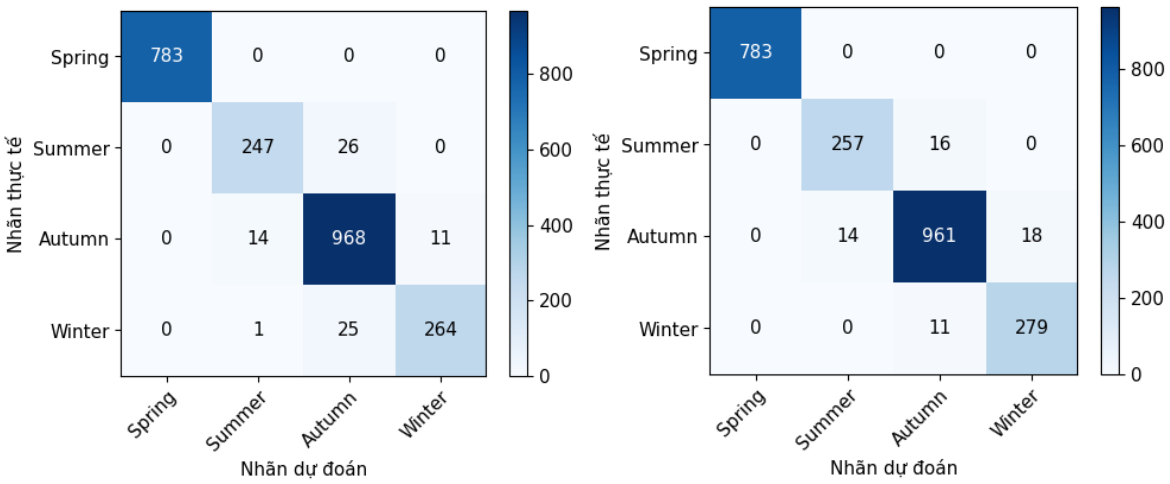
Bảng 3.9 Độ chính xác phân loại mùa thu hoạch của xoài sử dụng các thuật toán học máy

Thuật toán	PCA SVC	PLS DA	PCA Decision Tree	PCA RandomForest	PCA ANN
None SMOTE	0,8687	0,4497	0,826	0,8798	0,9085
SMOTE	0,8533	0,436	0,8135	0,908	0,9085

Việc phân loại nhiều lớp vẫn khiến cho mô hình PLS-DA đạt độ chính xác kém vì vậy có thể nhận định rằng việc ứng dụng mô hình PLS-DA cho việc phân loại đa lớp, đặc biệt là khi số lượng mẫu rất nhiều và có nhiều giá trị nhiễu. Mô hình PCA-SVC và PCA-DecisionTree có độ chính xác đều cao hơn 0,8, tuy nhiên độ chính xác lại giảm khi sử dụng SMOTE. SMOTE cũng không đóng góp vào khả năng cải thiện độ chính xác của mô hình PCA-ANN dù trong các trường hợp trên đây thường là mô hình hiện tượng tăng độ chính xác rõ rệt nhất khi áp dụng SMOTE. Việc tạo các dữ liệu mới chỉ giúp cho duy nhất mô hình PCA-RandomForest gia tăng độ chính xác. Như vậy với trường hợp này, thuật toán PCA-ANN có độ chính xác cao nhất và kỹ thuật SMOTE không đóng góp vào việc cải thiện hiệu suất phân loại của đa số thuật toán học máy.

3.3.3.2 Kết quả phân loại bằng thuật toán học sâu

Tiến hành xây dựng mô hình 1D-CNN, độ chính xác của mô hình với dữ liệu chưa áp dụng SMOTE là 0,9670 và dữ liệu đã áp dụng SMOTE là 0,9747. Kết quả phân loại chi tiết được thể hiện hình 3.27:

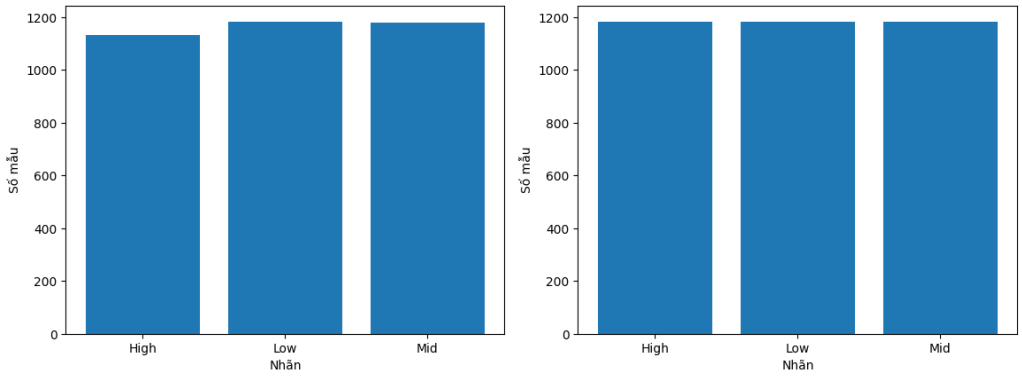


Hình 3.27 Kết quả phân loại theo mùa thu hoạch trên tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải) của mô hình 1D-CNN

Khi mô hình chạy phân loại 4 lớp, khả năng phân loại của mô hình tốt hơn khi độ chính xác của mô hình đạt tới 0,967. Giá trị này tăng lên 0,947 khi sử dụng kĩ thuật SMOTE. Trước khi áp dụng SMOTE, hai hiện tượng phân loại nhầm nhiều nhất là khi các mẫu xoài thu hoạch vào mùa hè (26 mẫu) và mùa đông (25 mẫu) bị phân loại nhầm thành mùa thu, điều này là do số lượng mẫu xoài mùa hè và lượng mẫu xoài mùa đông trong tập dữ liệu huấn luyện ít hơn số mẫu xoài mùa thu gấp 3 lần, khiến cho mô hình dễ dàng nhận thấy là có xu hướng thiên vị cho việc phân loại thành mùa xuân và mùa đông. Tuy nhiên sau khi sử dụng SMOTE, hiện tượng này đã được giảm bớt dù các mẫu dữ liệu vẫn tập trung quang các mùa hè, thu và đông. Vì vậy dù mô hình SMOTE đa phần không khiến cho mô hình phân loại tốt hơn mà còn kém hơn, nhưng kĩ thuật này vẫn giúp cho mô hình học sâu đạt được hiệu suất phân loại cao hơn, tăng độ ổn định của mô hình.

3.3.4 Phân loại nhiệt độ lấy mẫu của xoài

Dữ liệu ban đầu gồm bốn lớp nhiệt độ, bao gồm Low (15ºC), Mid (25ºC), High (35ºC), và No. Dữ liệu No tức là những mẫu xoài không được ghi lại thông tin nhiệt độ đo, tức là các mẫu xoài No có thể nằm trong ba lớp còn lại, vì vậy các mẫu xoài No được ghi nhận là nhiễu nên sẽ được xóa đi. Kiểm tra sự phân bố nhiệt độ của xoài, đồ thị phân bố của các nhãn trên tập dữ liệu huấn luyện được biểu thị trong hình 3.28:



Hình 3.28 Phân bố số lượng xoài theo nhiệt độ lẫy mẫu trong tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải)

Số lượng xoài phân bố đồng đều ở ba lớp, điều đó chứng tỏ dữ liệu không bị mất cân bằng. Tuy nhiên để ảnh đánh giá ảnh hưởng của việc sử dụng kỹ thuật SMOTE lên các học máy và học sâu, việc tạo ra dữ liệu mới vẫn được tiến hành.

3.3.4.1 Kết quả phân loại bằng các thuật toán học máy

Tiến hành xây dựng các mô hình học máy với các tham số tối ưu được khảo sát bằng thuật toán GridSearchCV với số fold = 5 (chi tiết thông số được ghi trong phụ lục đính kèm) thu được kết quả tại bảng 3.10 như sau:

Bảng 3.10 Độ chính xác phân loại nhiệt độ lấy mẫu của xoài sử dụng các thuật toán học máy

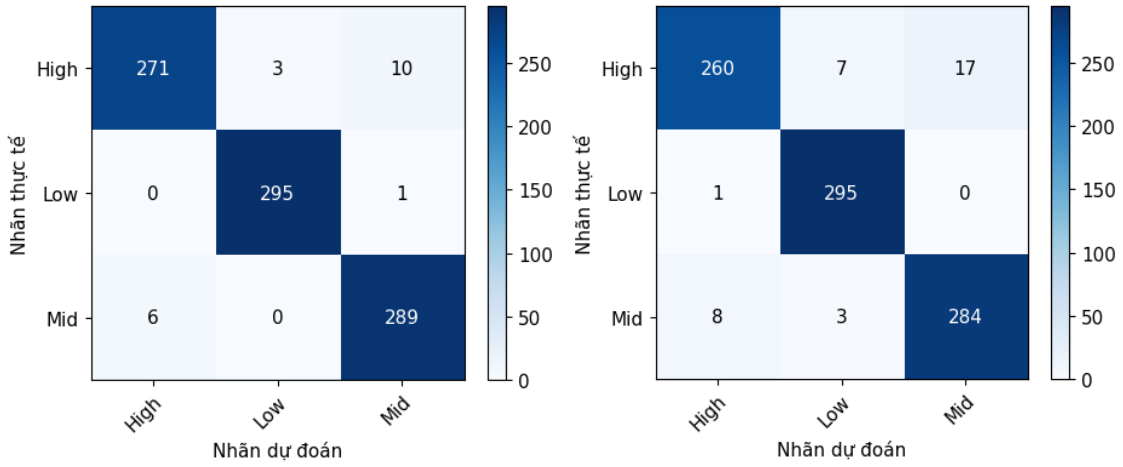
Thuật toán	PCA SVC	PLS DA	PCA Decision Tree	PCA RandomForest	PCA ANN
None SMOTE	0,8697	0,3897	0,8426	0,912	0,8948
SMOTE	0,866	0,3897	0,8217	0,9108	0,9085

Sự ảnh hưởng của SMOTE lên khả năng phân loại của đa số thuật toán học máy khiến độ chính xác của các mô hình giảm đi (có thuật toán PCA ANN gia tăng độ chính xác nhưng cách biệt không đáng kể). Điều đó chứng tỏ rằng trong trường hợp số mẫu phân phố đồng đều, việc áp dụng thuật toán SMOTE sẽ không giúp cho thuật toán cả

thiện hiệu suất phân loại, việc xem xét cải thiện hiệu suất phải phụ thuộc vào các yếu tố khác như cải thiện cấu trúc mô hình học máy, tối ưu hóa các tham số trong khoảng rộng hơn...

3.3.4.2 Kết quả phân loại bằng thuật toán học sâu

Tiến hành xây dựng mô hình 1D-CNN, độ chính xác của mô hình với dữ liệu chưa áp dụng SMOTE là 0,9771 và dữ liệu đã áp dụng SMOTE là 0,9589. Kết quả phân loại chi tiết được thể hiện hình 3.29:



Hình 3.29 Kết quả phân loại theo giống nhiệt độ lấy mẫu trên tập dữ liệu gốc (trái) và sau khi được áp dụng SMOTE (phải) của mô hình 1D-CNN

Độ chính xác của mô hình 1D-CNN thấp hơn tất cả các mô hình học máy, khi áp dụng thuật toán SMOTE độ chính xác của mô hình giảm từ 0,9771 xuống còn 0,9589, vì vậy chứng tỏ ngược lại so với các thuật toán học máy, SMOTE là một lựa chọn không tốt cho các mô hình học sâu khi tiến hành phân loại nhiệt độ đo phổ của xoài. Tuy nhiên sự cách biệt về khả năng phân loại nhiệt độ lấy mẫu của mô hình 1D-CNN so với các thuật toán học máy là vô cùng lớn, điều đó chứng tỏ tiềm năng ưu việt của mô hình 1D-CNN và các biến thể của nó trong nhiệm vụ phân loại dựa trên dữ liệu phổ. Ngoài SMOTE và sự phân bố dữ liệu, còn nhiều yếu tố ảnh hưởng khả năng phân loại của mô hình học sâu dựa trên phổ, ví dụ như sự khác biệt và đặc trưng của các điểm dữ liệu so với nhãn của chúng, sự phức tạp của cấu trúc mô hình so với nhu cầu bài toán,... Vì vậy, cần xem xét một cách đa dạng và đầy đủ tới các yếu tổ ảnh hưởng để giúp mô hình đạt hiệu suất cao hơn.

Kết luận

Với tất cả các đặc điểm phân loại đưa ra, mô hình 1D-CNN đã chứng minh là mô hình có cấu trúc tối ưu nhất với nhiệm phụ phân loại dựa trên dữ liệu phổ Vis-NIR, trong tất cả các trường hợp phân loại, mô hình 1D-CNN đều có giá trị độ chính xác cao hơn một cách rõ rệt đối với các thuật toán học máy. Điều đó chứng tỏ khả năng ứng dụng cao của mô hình 1D-CNN cho việc phân loại các đặc điểm của xoài, từ đó mở ra tiềm năng nghiên cứu ứng dụng thuật toán 1D-CNN học trên nhiều dạng dữ liệu phổ và phân loại nhiều đối tượng khác.

KẾT LUẬN

Với mục tiêu ban đầu đặt ra là ứng dụng học máy và học sâu trong nhận dạng, phân loại đối tượng và phân tích đồng thời, không xử lý mẫu, nghiên cứu đã thu được các kết quả như sau:

Xây dựng được các mô hình học máy phân tích đồng thời Tetracycline, Penicillin G và Cephalexin trong mẫu thuốc bằng phổ UV của các dung dịch. Các mô hình học máy có khả năng dự đoán tốt hàm lượng Tetracycline, Penicillin G và Cephalexin trong mẫu thuốc, trong đó mô hình cây quyết định, rừng ngẫu nhiên và PLSR vẫn còn các mẫu có sai số lớn hơn 20% so với giá trị hàm lượng trong tập dữ liệu kiểm tra bằng phương pháp HPLC. Mô hình PCR với 16 thành phần chính được tối ưu có kết quả định lượng đáng tin cậy khi không có mẫu nào có sai số lớn hơn 20% so với tập dữ liệu kiểm tra. Việc sử dụng các mô hình học máy có tốc độ cao, hiệu suất hồi quy ổn định và linh hoạt trong khoảng nồng độ nghiên cứu chứng tỏ tiềm năng ứng dụng để phân tích nhanh Tetracycline, Penicillin G và Cephalexin trong mẫu thuốc trên thị trường.

Xây dựng được mô hình học máy và học sâu phân tích đồng thời một số nhóm thuốc kháng sinh bằng phổ IR. Khi độ phức tạp của mô hình càng tăng thì khả năng phân tích đồng thời trên tập dữ liệu kiểm tra càng tốt hơn, giá trị của R2 và RMSE của mô hình PCA-ANN lần lượt là 0,88 và 3,64, thể hiện khả năng phân tích đồng thời tốt nhất trong nhóm các thuật toán học máy. Mô hình học sâu 1D-CNN chứng tỏ là sự lựa chọn phù hợp nhất cả có giá trị sai số thấp hơn so với các mô hình học máy, với giá trị R2 và RMSE lần lượt là 0,97 và 2,12. Tuy nhiên khi phân tích đồng thời trong mẫu thuốc thực, mô hình chỉ thể hiện được khả năng phân tích đồng thời tốt đối với các mẫu thuộc nhóm thuốc sulfamid, các mẫu thuộc nhóm thuốc khác có sai số lớn thể hiện sự không ổn định của mô hình. Vì vậy cần tiến hành thêm các nghiên cứu và mở rộng quy mô của dữ liệu để cải thiện và đưa ra mô hình thích hợp nhất khi phân tích mẫu thực.

Dự đoán được độ đường Brix của quả cam mà không xâm lấn quả trên cơ sở xây dựng bộ dữ liệu hình ảnh chụp từ camera vùng VIS của quả cam và mối tương quan với độ đường Brix của quả cam cùng loại. Kết quả mô hình cho thấy độ chính xác rất cao khi giá trị R2 của mô hình học sâu đã đạt khoảng 90% trong khi các mô hình học máy thông thường sử dụng giá trị RGB đơn giản thì giá trị R2 chỉ có thể đạt chưa tới 50%. Điểm mạnh của mô hình học sâu là có thể học được trên những dữ liệu có độ nhiễu cao

như vệt rám, vết châm của côn trùng nhưng vẫn cần phải phát triển thêm để giảm số lượng tham số mà vẫn giữ được khả năng trích xuất đặc trưng quan trọng một cách hiệu quả.

Xây dựng được mô hình học sâu phân loại các đặc điểm của xoài dựa trên phổ Vis-NIR tham chiếu từ nguồn mở. Tất cả các mô hình 1D-CNN được tối ưu có độ chính xác cao trên 94%, thể hiện khả phân loại tốt hơn một cách đáng kể so với các mô hình học máy truyền thống. Việc áp dụng thuật toán SMOTE làm cân bằng dữ liệu giúp cho các mô hình cải thiện hiệu quả phân loại, tăng độ chính xác đặc biệt với các trường hợp dữ liệu bị mất cân bằng một cách nghiêm trọng. Mô hình 1D-CNN khẳng định sự phù hợp với tất cả các bài toán phân loại, đặc biệt với bài toán phân loại nhị phân có độ chính xác lớn hơn 0,99, thể hiện tiềm năng lớn trong việc ứng dụng vào các mô hình dự đoán và phân loại tích hợp trong các thiết bị đo hiện trường.

TÀI LIỆU THAM KHẢO

1. Abdullah Al-Sanabani, Dheya Galal, et al. (2019), Development of non-destructive mango assessment using Handheld Spectroscopy and Machine Learning Regression, *Journal of Physics: Conference Series*, IOP Publishing, p. 012030.
2. Al-Sammarraie, Mustafa Ahmed Jalal, et al. (2022), "Predicting fruit's sweetness using artificial intelligence—case study: orange", *Applied Sciences*. 12(16), p. 8233.
3. Anderson, NT, et al. (2020), "Achieving robustness across season, location and cultivar for a NIRS model for intact mango fruit dry matter content", *Postharvest Biology Technology*. 168, p. 111202.
4. Ardila, Diego, et al. (2019), "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography", *Nature medicine*. 25(6), pp. 954-961.
5. Bentley, Ronald (2004), "The molecular structure of penicillin", *Journal of chemical education*. 81(10), p. 1462.
6. Breiman, Leo (2001), "Random forests", *Machine learning*. 45, pp. 5-32.
7. Budiastra, I Wayan and Punvadaria, Hadi K (2000), "Classification of mango by artificial neural network based on near infrared diffuse reflectance", *IFAC Proceedings Volumes*. 33(29), pp. 157-161.
8. Campbell, Murray, Hoane Jr, A Joseph, and Hsu, Feng-hsiung (2002), "Deep blue", *Artificial intelligence*. 134(1-2), pp. 57-83.
9. Colton, James A and Bower, Keith M (2002), "Some misconceptions about R2", *International Society of Six Sigma Professionals, EXTRAOrdinary Sense*. 3(2), pp. 20-22.
10. Cristianini, Nello and Shawe-Taylor, John (2000), *An introduction to support vector machines and other kernel-based learning methods*, Cambridge university press.

11. Donowitz, Gerald R and Mandell, Gerald L (1988), "Beta-lactam antibiotics", *New England Journal of Medicine*. 318(7), pp. 419-426.

12. Esposito Vinzi, Vincenzo and Russolillo, Giorgio (2013), "Partial least squares algorithms and methods", *Wiley Interdisciplinary Reviews: Computational Statistics*. 5(1), pp. 1-19.

13. Fernández, Alberto, et al. (2018), "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary", *Journal of artificial intelligence research*. 61, pp. 863-905.

14. Gallagher, Neal B (2020), "Savitzky-Golay smoothing and differentiation filter", *Eigenvector Research Incorporated*.

15. Hassannia, M, Fahimi-Kashani, N, and Hormozi-Nezhad, MR (2024), "Machine-learning assisted multicolor platform for multiplex detection of antibiotics in environmental water samples", *Talanta*. 267, p. 125153.

16. Jha, Shyam N, et al. (2013), "Authentication of mango varieties using near-infrared spectroscopy", *Agricultural Research*. 2, pp. 229-235.

17. Johnson, Justin M and Khoshgoftaar, Taghi M (2019), "Survey on deep learning with class imbalance", *Journal of Big Data*. 6(1), pp. 1-54.

18. Kiranyaz, Serkan, et al. (2021), "1D convolutional neural networks and applications: A survey", *Mechanical systems signal processing*. 151, p. 107398.

19. Maldonado-Celis, Maria Elena, et al. (2019), "Chemical composition of mango (Mangifera indica L.) fruit: Nutritional and phytochemical compounds", *Frontiers in plant science*. 10, p. 450160.

20. Marangoni-Ghoreyshi, Yasmin Garcia, et al. (2023), "Multi-resistant diarrheagenic Escherichia coli identified by FTIR and machine learning: a feasible strategy to improve the group classification", *RSC advances*. 13(36), pp. 24909-24917.

21. O'shea, Keiron and Nash, Ryan (2015), "An introduction to convolutional neural networks", *arXiv preprint arXiv:.08458*.

22. Ovung, Aben and Bhattacharyya, Jhimli (2021), "Sulfonamide drugs: Structure, antibacterial property, toxicity, and biophysical interactions", *Biophysical reviews*. 13(2), pp. 259-272.

23. Palur, Keerthisikha, Archakam, Sreenivasa Charan, and Koganti, Bharathi (2020), "Chemometric assisted UV spectrophotometric and RP-HPLC methods for simultaneous determination of paracetamol, diphenhydramine, caffeine and phenylephrine in tablet dosage form", *Spectrochimica Acta Part A: Molecular Biomolecular Spectroscopy*. 243, p. 118801.

24. Pronprasit, Rattapol and Natwichai, Juggapong (2013), "Prediction of mango fruit quality from Nir spectroscopy using an ensemble classification", *International Journal of Computer Applications*. 83(14).

25. Quinlan, J. Ross (1986), "Induction of decision trees", *Machine learning*. 1, pp. 81-106.

26. Shah, Syed Sohaib Ali, et al. (2021), "Mango maturity classification instead of maturity index estimation: A new approach towards handheld NIR spectroscopy", *Infrared Physics Technology*. 115, p. 103639.

27. Simonyan, Karen and Zisserman, Andrew (2014), "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:.08458*.

28. Soltan, Andrew AS, et al. (2021), "Real-world evaluation of AI-driven COVID-19 triage for emergency admissions: External validation & operational assessment of lab-free and high-throughput screening solutions", *medRxiv*, p. 2021.08. 24.21262376.

29. Sun, Ruo-Yu (2020), "Optimization for deep learning: An overview", *Journal of the Operations Research Society of China*. 8(2), pp. 249-294.

30. Thinh, Nguyen Truong, et al. (2019), Mango classification system based on machine vision and artificial intelligence, *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, IEEE, pp. 475-482.

31. Wang, Fei-Yue, et al. (2016), "Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond", *IEEE/CAA Journal of Automatica Sinica*. 3(2), pp. 113-120.

32. Wold, Svante, Esbensen, Kim, and Geladi, Paul (1987), "Principal component analysis", *Chemometrics intelligent laboratory systems*. 2(1-3), pp. 37-52.