ĐẠI HỌC THÁI NGUYÊN
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**


**TRẦN THỊ XUÂN**

# NÂNG CAO HIỆU QUẢ PHÂN TÍCH PROTEIN SỬA ĐỔI SAU DỊCH MÃ TRÊN CƠ SỞ KẾT HỢP MÔ HÌNH HỌC MÁY VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

## LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

**THÁI NGUYÊN - NĂM 2025**

ĐẠI HỌC THÁI NGUYÊN
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**


**TRẦN THỊ XUÂN**

# NÂNG CAO HIỆU QUẢ PHÂN TÍCH PROTEIN SỬA ĐỔI SAU DỊCH MÃ TRÊN CƠ SỞ KẾT HỢP MÔ HÌNH HỌC MÁY VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

**Ngành: Khoa học máy tính**
**Mã số: 9.48.01.01**

## LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

**TẬP THỂ HƯỚNG DẪN KHOA HỌC:**

**1. PGS.TS. LÊ NGUYỄN QUỐC KHÁNH**
**2. TS. NGUYỄN VĂN NÚI**

**THÁI NGUYÊN - NĂM 2025**

# LỜI CAM ĐOAN

Nghiên cứu sinh (NCS) xin cam đoan các kết quả trình bày trong luận án Tiến sĩ ***"Nâng cao hiệu quả phân tích protein sửa đổi sau dịch mã trên cơ sở kết hợp mô hình học máy và xử lý ngôn ngữ tự nhiên"*** là các công trình nghiên cứu của NCS dưới sự hướng dẫn của PGS. TS. Lê Nguyễn Quốc Khánh và TS. Nguyễn Văn Núi, trừ những kiến thức tham khảo từ các tài liệu đã được tham chiếu rõ ràng.

Các kết quả nghiên cứu trong luận án là trung thực, một phần đã được công bố trên các Tạp chí, Hội thảo khoa học (danh sách các công trình được liệt kê tại cuối Luận án), phần còn lại chưa được công bố trong bất kỳ công trình nào khác.

Mọi nội dung dữ liệu được tham khảo trong luận án đều được trích dẫn đầy đủ và đúng quy định.

Thái Nguyên, ngày .... tháng .... năm 2025
Tác giả luận án

**Trần Thị Xuân**

# LỜI CẢM ƠN

Luận án tiến sĩ này là kết quả của cả một quá trình nghiên cứu lý thuyết và thực nghiệm đầy thách thức và khó khăn, đòi hỏi sự kiên trì và sự tập trung cao độ. Kết quả đạt được không chỉ là những nỗ lực cá nhân mà còn có sự hỗ trợ, giúp đỡ của tập thể người hướng dẫn, cơ sở đào tạo, cơ quan chủ quản, đồng nghiệp và gia đình.

Với lòng biết ơn sâu sắc, NCS xin gửi lời cảm ơn chân thành đến tập thể hướng dẫn khoa học: PGS.TS. Lê Nguyễn Quốc Khánh – Trường Đại học Y Đài Bắc (Đài Loan) và TS. Nguyễn Văn Núi – Trường Đại học Công nghệ Thông tin và Truyền thông, những người đã luôn tận tâm hướng dẫn, hỗ trợ và động viên NCS trong suốt quá trình thực hiện luận án.

NCS xin trân trọng gửi lời cảm ơn Trường Đại học Công nghệ Thông tin và Truyền thông, Phòng Đào tạo - Bộ phận Sau đại học, Khoa Công nghệ Thông tin đã tạo điều kiện thuận lợi cho NCS trong quá trình học tập và nghiên cứu.

NCS cũng xin gửi lời cảm ơn đến Trường Đại học Kinh tế và Quản trị Kinh doanh, Khoa Khoa học Cơ bản, Khoa Kinh doanh và Logistics, đồng nghiệp đã luôn cổ vũ, động viên và tạo điều kiện tốt nhất cho NCS hoàn thành nhiệm vụ học tập và nghiên cứu.

Đặc biệt, NCS xin chân thành cảm ơn sự hỗ trợ từ Đề tài Khoa học và Công nghệ cấp Đại học Thái Nguyên mã số ĐH2023-TN08-05 [2], và Đề tài thuộc Quỹ Phát triển Khoa học và Công nghệ Quốc gia (NAFOSTED) mã số 102.05-2023.49, đã tạo nguồn lực quan trọng để NCS triển khai các nghiên cứu chuyên sâu và hoàn thiện luận án này.

Cuối cùng, NCS xin gửi lời cảm ơn sâu sắc đến gia đình, bạn bè và các anh chị em nghiên cứu sinh trong nhóm nghiên cứu – những người đã luôn bên cạnh, cổ vũ, chia sẻ và ủng hộ NCS vượt qua những khó khăn trong suốt chặng đường học tập và nghiên cứu. Sự quan tâm và động viên của mọi người là nguồn động lực to lớn giúp NCS vững bước hoàn thành nhiệm vụ của mình.

Xin chân thành biết ơn!

**NCS. Trần Thị Xuân**

# MỤC LỤC

# DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

| Thuật ngữ | Diễn giải tiếng Anh | Diễn giải tiếng Việt |
|---|---|---|
| AAC | Amino Acid Composition | AAC là giá trị tần suất xuất hiện của mỗi axit amin trong chuỗi protein. Đây là một trong những đặc trưng cơ bản nhất được sử dụng trong phân tích protein bằng học máy [75]. |
| AAIndex | | AAIndex là một tập hợp các chỉ số sinh hóa và vật lý của axit amin (tính kị nước, tính phân cực, độ linh động), được sử dụng để biểu diễn đặc trưng của protein trong các bài toán học máy [52]. |
| ACC | Accuracy | Là một chỉ số đánh giá hiệu suất của mô hình phân loại, được tính bằng tỷ lệ giữa số lượng dự đoán đúng trên tổng số mẫu dữ liệu. |
| AUC | Area under the curve | Chỉ số được tính toán dựa trên đường cong ROC nhằm đánh giá khả năng phân loại của mô hình. |
| BE | Binary Encoding | Đặc trưng Binary encoding là một phương pháp mã hóa nhị phân để biểu diễn thông tin vị trí của các axit amin trong trình tự protein, hỗ trợ các mô hình học máy dự đoán vị trí PTM. |
| BERT | Bidirectional Encoder Representations from Transformers | Mô hình BERT. |

| Thuật ngữ | Diễn giải tiếng Anh | Diễn giải tiếng Việt |
|---|---|---|
| Bi-LSTM | Bidirectional Long Short-Term Memory | Bi-LSTM là mạng LSTM hai chiều, là một mạng nơ-ron hồi quy được sử dụng chủ yếu trong xử lý ngôn ngữ tự nhiên. |
| CKSAAP | Composition of k-Spaced Amino Acid Pairs | CKSAAP là một phương pháp trích xuất đặc trưng từ trình tự protein bằng cách đếm tần suất xuất hiện của các cặp axit amin trong chuỗi với một khoảng cách $k$ cố định. CKSAAP tạo ra một véc tơ 400 chiều. |
| CNN1D | 1-dimensional Convolutional Neural Network | Mạng nơ-ron tích chập một chiều. |
| CNNs | Convolutional Neural Networks | Mạng nơ-ron tích chập. |
| DNN | Deep Neural Network | Mạng nơ-ron sâu. |
| FN | False Negative | Số mẫu dương tính nhưng bị dự đoán nhầm là âm tính. |
| FP | False Positive | Số mẫu âm tính nhưng bị dự đoán nhầm là dương tính. |
| GPS | Group-based Prediction Score | GPS là một phương pháp đặc trưng sinh học thường được sử dụng trong dự đoán vị trí biến đổi sau dịch mã [128]. GPS không chỉ là một đặc trưng đơn thuần mà còn là một mô hình tính điểm dựa trên nguyên tắc điểm số nhóm. |
| LSTM | Long Short-Term Memory | LSTM là một loại mạng nơ-ron hồi quy được thiết kế để giải quyết vấn đề vanishing gradient của RNN truyền thống khi làm việc với dữ liệu chuỗi dài. |

| Thuật ngữ | Diễn giải tiếng Anh | Diễn giải tiếng Việt |
|---|---|---|
| MCC | Matthews Correlation Coefficient | Hệ số tương quan Matthews, là một số liệu hiệu suất cho các bộ phân loại nhị phân trong học máy. Nó đo lường mối tương quan giữa kết quả dự đoán và thực tế, xem xét đầy đủ ma trận nhầm lẫn. |
| Natural Language-Based | | Kỹ thuật xử lý ngôn ngữ tự nhiên mã hóa chuỗi protein: Là kỹ thuật mã hóa chuỗi protein theo cách tiếp cận của xử lý ngôn ngữ tự nhiên, từ đó tạo ra véc tơ đặc trưng sử dụng trong các mô hình học máy. |
| NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên là một lĩnh vực của trí tuệ nhân tạo (AI) giúp máy tính có thể hiểu, diễn giải và tạo ra ngôn ngữ con người một cách tự nhiên và hiệu quả. |
| PseAAC | Pseudo Amino Acid Composition | PseAAC là một phương pháp biểu diễn trình tự protein được đề xuất bởi Kuo-Chen Chou (2001), mở rộng AAC bằng cách đưa vào thông tin về mối quan hệ tuần tự và các đặc tính sinh hóa của axit amin [98]. |
| PSSM | Position-Specific Scoring Matrix | Biểu diễn mức độ bảo tồn của các axit amin tại mỗi vị trí trong chuỗi protein bằng cách sử dụng ma trận điểm số dựa trên nhiều chuỗi tương đồng. PSSM được tạo ra thông qua căn chỉnh dãy trình tự (MSA), được dùng trong dự đoán cấu trúc protein, tương tác protein-protein, và dự đoán vị trí PTM. |

| Thuật ngữ | Diễn giải tiếng Anh | Diễn giải tiếng Việt |
|---|---|---|
| PTM | Post-Translational Modification | Sửa đổi sau dịch mã. |
| RF | Random Forest | Thuật toán học máy rừng ngẫu nhiên. |
| RNNs | Recurrent Neural Networks | Mạng thần kinh hồi quy. |
| SVM | Support Vector Machine | Thuật toán máy vector hỗ trợ. |
| SEN | Sensitivity | Độ nhạy phản ánh khả năng mô hình phát hiện đúng các trường hợp dương tính (positive).SEN càng cao, mô hình càng ít bỏ sót mẫu dương tính |
| SPE | Specificity | Độ đặc hiệu phản ánh khả năng mô hình nhận diện đúng các trường hợp âm tính (negative). |
| TN | True Negative | Số mẫu âm tính được dự đoán đúng. |
| TP | True Positive | Số mẫu dương tính được dự đoán đúng. |
| XGBoost | Extreme Gradient Boosting | Thuật toán học máy tăng cường độ dốc cực đại. |

# DANH MỤC CÁC HÌNH VẼ

# DANH MỤC CÁC BẢNG BIỂU

# MỞ ĐẦU

## 1. Tính cấp thiết của đề tài

***Bối cảnh khoa học và thực tiễn:***

Sửa đổi sau dịch mã (Post-Translational Modification – PTM) là những biến đổi hóa học diễn ra sau khi quá trình tổng hợp protein hoàn tất. Các dạng PTM phổ biến như glycosyl hóa, phosphoryl hóa, ubiquitin hóa, acetyl hóa, lipid hóa, hay phân giải protein... có vai trò đặc biệt quan trọng trong việc điều chỉnh cấu trúc, chức năng và hoạt động sinh học của protein [4].

PTM tác động sâu rộng đến nhiều quá trình sinh học then chốt, chẳng hạn như truyền tín hiệu tế bào, điều hòa miễn dịch, và biểu hiện gen. Sự sai lệch trong quá trình PTM liên quan trực tiếp đến nhiều bệnh lý nguy hiểm như ung thư, rối loạn thần kinh và bệnh truyền nhiễm [3, 37, 91]. Do đó, việc xác định chính xác các vị trí PTM trong chuỗi protein là một nhiệm vụ có ý nghĩa quan trọng trong nghiên cứu y sinh, hỗ trợ làm sáng tỏ cơ chế phân tử, phát triển thuốc và liệu pháp điều trị mới.

Khối phổ (Mass Spectrometry – MS) được coi là phương pháp tiêu chuẩn vàng để phát hiện PTM [7,24]. Tuy nhiên, kỹ thuật này thường yêu cầu quy trình thí nghiệm phức tạp, tốn kém và mất nhiều thời gian, đồng thời khó mở rộng quy mô. Do đó, sự phát triển của các phương pháp tính toán có khả năng dự đoán vị trí PTM một cách nhanh chóng, chi phí thấp và hiệu quả là hết sức cần thiết nhằm hỗ trợ cho các nghiên cứu trong lĩnh vực y sinh.

***Sự phát triển của các phương pháp tính toán:***

Trong hơn hai thập kỷ qua, các phương pháp tính toán đã góp phần quan trọng trong dự đoán vị trí PTM, đặc biệt với ba hướng tiếp cận nổi bật: học máy truyền thống, học sâu, và xử lý ngôn ngữ tự nhiên (NLP) và mô hình ngôn ngữ protein (PLMs).

(i) Học máy truyền thống (Machine Learning): Các mô hình học máyđược sử dụng xấy dựng các mô hình dự đoán PTM như SVM, Random Forest, XGBoost hay kNN, tuy nhiên các mô hình này thường dựa trên tập đặc trưng thủ công được thiết kế từ kiến thức sinh học (ví dụ: PseAAC, CKSAAP, BE, PsePSSM) [12,27,49,62,64,103,124] . Hướng nghiên cứu này có ưu điểm nổi bật là dễ huấn luyện, triển khai nhanh, và có khả năng diễn giải tốt, đặc biệt phù hợp khi làm việc với dữ liệu nhỏ. Tuy nhiên, nhược điểm lớn là phụ thuộc nhiều vào đặc trưng thủ công vốn mang tính chủ quan và dễ bỏ sót các tín hiệu ngữ cảnh quan trọng, khiến khả năng tổng quát hóa bị hạn chế.

(ii) Học sâu (Deep Learning): Mô hình dự đoán PTM được phát triển dựa trên các

kiến trúc mạng học sâu như CNN, LSTM, Bi-LSTM hoặc các mô hình học sâu lai. Học sâu cho phép tự động trích xuất đặc trưng từ dữ liệu thô và mô hình hóa mối quan hệ phi tuyến phức tạp trong chuỗi protein [30, 45, 60, 106, 117, 125]. Các nghiên cứu gần đây cho thấy mô hình học sâu thường vượt trội hơn so với học máy truyền thống về hiệu quả dự đoán. Tuy nhiên, chúng thường đòi hỏi tập dữ liệu huấn luyện quy mô lớn và tiêu tốn nhiều tài nguyên tính toán. Trong điều kiện dữ liệu sinh học thường hạn chế và mất cân bằng, mô hình học sâu dễ gặp phải vấn đề quá khớp, làm giảm khả năng ứng dụng thực tiễn.

(iii) Xử lý ngôn ngữ tự nhiên (NLP) và mô hình ngôn ngữ protein (PLMs):

Trong hướng tiếp cận này, chuỗi protein được xem như một "ngôn ngữ sinh học", trong đó mỗi axit amin tương ứng với một token, và ngữ cảnh xung quanh token quyết định chức năng sinh học của nó. Quan niệm này mở ra khả năng ứng dụng các kỹ thuật NLP vào dự đoán PTM. Các mô hình ngôn ngữ lớn như BERT [22] và T5 [89] được sử dụng để trích xuất các embedding ngữ cảnh, sau đó các embedding này được đưa vào làm đặc trưng cho các mô hình học máy hoặc học sâu, xây dựng nên các mô hình dự đoán PTM hiệu quả [55, 79, 82, 101].

Ngoài ra, một số mô hình PTM còn khai thác các mô hình tiền huấn luyện dựa trên BERT chuyên biệt cho protein, chẳng hạn như ProteinBERT [11], điển hình là DeepPTM [101]. Tuy nhiên, một hạn chế quan trọng là chi phí tính toán rất cao, gây khó khăn trong triển khai thực tế, đặc biệt khi dữ liệu hạn chế hoặc tài nguyên tính toán bị giới hạn.

***Các thách thức và khoảng trống nghiên cứu:***

Mặc dù đã đạt được nhiều tiến bộ, các nghiên cứu dự đoán vị trí PTM hiện nay vẫn tồn tại một số thách thức sau:

- Phụ thuộc đặc trưng thủ công: Phần lớn các phương pháp học máy truyền thống vẫn dựa nhiều vào đặc trưng do con người thiết kế, mang tính chủ quan và thiếu khả năng khái quát khi áp dụng cho loài mới hoặc dạng PTM khác.

- Nguy cơ quá khớp do dữ liệu hạn chế: Trong bối cảnh dữ liệu PTM thường nhỏ và mất cân bằng, các mô hình học sâu dễ bị quá khớp, làm giảm tính tổng quát trong thực tiễn.

- Chi phí dữ liệu và tài nguyên lớn: Các mô hình học sâu và PLMs/LLMs yêu cầu tập dữ liệu khổng lồ và hạ tầng mạnh, khó áp dụng trong môi trường nghiên cứu hạn chế về tính toán.

- Chưa khai thác kỹ thuật chắt lọc tri thức (Knowledge Distillation-KD). KD đã chứng minh hiệu quả trong thị giác máy tính (Computer Vision) và xử lý ngôn ngữ tự

nhiên (NLP), cho phép xây dựng mô hình gọn nhẹ nhưng vẫn duy trì hiệu suất cao. Tuy nhiên, đến nay chưa có công trình nào áp dụng kỹ thuật học chắt lọc tri thức vào dự đoán PTM, trong khi đây là một hướng hứa hẹn phù hợp với dữ liệu hạn chế và môi trường tài nguyên giới hạn.

Xuất phát từ tầm quan trọng của việc xác định chính xác vị trí PTM trong nghiên cứu y sinh, cùng với nhu cầu phát triển các phương pháp tính toán tiên tiến và những khoảng trống nghiên cứu đã phân tích, NCS lựa chọn đề tài "Nâng cao hiệu quả phân tích protein sửa đổi sau dịch mã trên cơ sở kết hợp mô hình học máy và xử lý ngôn ngữ tự nhiên" làm luận án tiến sĩ ngành Khoa học máy tính.

## 2. Đối tượng và phạm vi nghiên cứu

*(1) Đối tượng thứ nhất là các protein sửa đổi sau dịch mã:*

Hiện tại, có hơn 600 loại PTM khác nhau đã được phát hiện và định danh. Mong muốn của NCS là có thể thực hiện nghiên cứu được với nhiều loại PTM khác nhau nhằm bổ sung, góp phần làm giàu tri thức, sự hiểu biết của con người đối với tất cả các loại PTM hiện có. Tuy nhiên, trong phạm vi luận án này, nghiên cứu tập trung vào ba loại phổ biến và có dữ liệu tương đối đầy đủ và còn khoảng trống nghiên cứu: SUMOylation, Succinylation và Ubiquitination.

Ngoài ra, qua khảo sát, cấu trúc protein bậc cao (cấu trúc bậc 2,3,4- thường được lưu trữ dưới dạng ảnh 3D) trong các ngân hàng Protein(UniProt, NCBI, Ensembl...) còn thiếu, chưa đầy đủ và rất tốn kém bộ nhớ để lưu trữ; hơn nữa hầu hết dữ liệu protein hiện nay được lưu trữ dưới dạng chuỗi FASTA (Protein bậc 1). Dạng biểu diễn này không chỉ phổ biến mà còn tiết kiệm tài nguyên và phù hợp với các kỹ thuật học máy, học sâu hiện đại và NLP. Vì vậy, luận án lựa chọn cấu trúc protein bậc 1 làm đầu vào để phát triển mô hình dự đoán vị trí PTM với hiệu năng cao cho ba loại nêu trên.

*(2) Đối tượng thứ hai là mô hình dự đoán vị trí PTM dựa trên mô hình học máy kết hợp với xử lý ngôn ngữ tự nhiên:*

Kỹ thuật phổ biến để dự đoán vị trí PTM, có độ chính xác cao hiện nay chính là kỹ thuật khối phổ và giải trình tự. Tuy nhiên, kỹ thuật MS này có chi phí rất lớn, thời gian thực hiện lâu, và đặc biệt là khó áp dụng với nhiều protein cùng lúc. Chính vì vậy, việc nghiên cứu các mô hình dự đoán PTM dựa trên mô hình học máy, kết hợp với NLP là một cách tiếp cận phù hợp bởi nó khai thác được những tiến bộ của công nghệ thông tin, các mô hình học máy và kỹ thuật NLP nhằm giúp ngắn thời gian hỗ trợ cho các nhà sinh/y học đưa ra những kết luận nhanh và chính xác, phù hợp nhu cầu và xu hướng phát triển hiện nay.

## 3. Phương pháp nghiên cứu

Để đạt được mục tiêu nghiên cứu, luận án triển khai song song hai hướng chính, đó là: (1) Nghiên cứu cơ sở lý thuyết để đề xuất mô hình mới và (2) Nghiên cứu thực nghiệm nhằm kiểm chứng hiệu quả các mô hình này.

(1) Về lý thuyết, luận án kế thừa và phát triển các phương pháp hiện đại của khoa học dữ liệu và trí tuệ nhân tạo, bao gồm: Học máy tổ hợp (Ensemble Learning) nhằm khai thác ưu thế của nhiều bộ phân loại để nâng cao độ tin cậy dự đoán. Xử lý ngôn ngữ tự nhiên để biểu diễn "ngôn ngữ protein" và trích xuất ngữ nghĩa, ngữ cảnh từ chuỗi axit amin; Các kiến trúc học sâu lai (Hybrid Deep Learning) kết hợp CNN và LSTM/Bi-LSTM nhằm tận dụng đồng thời khả năng phát hiện đặc trưng cục bộ và quan hệ tuần tự dài hạn; Học chắt lọc tri thức để xây dựng các mô hình gọn nhẹ, thích ứng với dữ liệu hạn chế;

(2) Về thực nghiệm, các mô hình được huấn luyện và kiểm định trên dữ liệu PTM thực tế, sau đó so sánh với các phương pháp tiên tiến hiện có. Kết quả đánh giá giúp khẳng định tính khả thi, hiệu quả và ý nghĩa ứng dụng của các mô hình đề xuất trong việc dự đoán vị trí sửa đổi sau dịch mã trên protein.

## 4. Các đóng góp của luận án

Luận án tập trung nghiên cứu và đề xuất phương pháp dự đoán ba loại PTM phổ biến, bao gồm: SUMOylation, Succinylation và Ubiquitination. Trên cơ sở kết hợp các phương pháp truyền thống, học máy, học sâu và kỹ thuật xử lý ngôn ngữ tự nhiên, luận án đã giải quyết được các mục tiêu đặt ra của đề tài, đề xuất được những mô hình cải tiến với hiệu suất cao. Trong quá trình thực hiện luận án NCS đã công bố 08 bài báo khoa học trên các tạp chí và hội thảo chuyên ngành trong nước và quốc tế. Trong đó, 06 công bố có nội dung gắn trực tiếp với tên đề tài, phản ánh rõ ràng các kết quả nghiên cứu chính của luận án, được trình bày đầy đủ và chi tiết trong các chương nội dung. Bên cạnh đó, 02 công bố khác có nội dung liên quan và hỗ trợ, góp phần làm phong phú thêm cho luận án, mở rộng phạm vi nghiên cứu, đồng thời khẳng định khả năng ứng dụng, tính tổng quát và hướng phát triển của kết quả nghiên cứu, các công bố bổ trợ này không đi sâu vào nội dung chính của luận án nhưng có ý nghĩa bổ trợ về phương pháp, kỹ thuật và lĩnh vực ứng dụng.

Danh mục chi tiết các công bố đã được liệt kê trong phần Danh mục công trình khoa học kèm theo luận án.

**Luận án có ba đóng góp chính sau:**

**(1) Cơ sở lý luận và tổng quan hệ thống:** Luận án đã hệ thống hóa, phân tích và so sánh các phương pháp từ truyền thống, học máy tổ hợp, học sâu lai, kỹ thuật NLP

trong bài toán dự đoán PTM, qua đó xây dựng nền tảng khoa học vững chắc cho các nghiên cứu tiếp theo.

**(2) Khai thác NLP cho dữ liệu protein:** Luận án đã chứng minh khả năng ứng dụng và hiệu quả của các kỹ thuật NLP trong việc biểu diễn ngữ cảnh của chuỗi protein, giúp vượt qua hạn chế của đặc trưng thủ công và nâng cao độ chính xác trong dự đoán.

**(3) Đề xuất và phát triển mô hình mới:** Luận án đã đề xuất bốn mô hình PTM với hiệu suất cao, trong đó có các mô hình lai kết hợp học sâu với NLP và đặc biệt là mô hình áp dụng học chắt lọc tri thức cho Ubiquitination, phù hợp với bối cảnh dữ liệu hạn chế và môi trường tính toán hạn chế. Cụ thể, bốn đề xuất chính gồm:

- Đề xuất mô hình dự đoán vị trí PTM (SUMOylation) dựa trên học máy tổ hợp và các đặc trưng lai ghép.

- Đề xuất hai mô hình dự đoán vị trí PTM (SUMOylation và Succinylation) dựa trên kỹ thuật học sâu lai ghép và kỹ thuật xử lý ngôn ngữ tự nhiên.

- Đề xuất mô hình dự đoán PTM (Ubiquitination) dựa trên học chắt lọc tri thức và kỹ thuật xử lý ngôn ngữ tự nhiên.

## 5. Bố cục của luận án

Luận án bao gồm các phần: Mở đầu, 4 chương nội dung chính, kết luận và hướng phát triển, danh mục các công trình khoa học đã công bố và danh mục tài liệu tham khảo. Nội dung chính của 4 chương được tóm tắt như sau:

**Chương 1. Tổng quan dự đoán vị trí sửa đổi sau dịch mã trong chuỗi protein và các kiến thức nền tảng**

Trong chương này, NCS trình bày các kiến thức nền tảng về protein và protein sửa đổi sau dịch mã (PTM), vai trò của việc xây dựng mô hình dự đoán vị trí PTM hiệu suất cao. Phần tiếp theo của chương trình bày về bài toán dự đoán vị trí PTM, các bước xây dựng mô hình dự đoán vị trí PTM, các phương pháp mã hoá đặc trưng protein hiện nay, tổng quan tình hình nghiên cứu dự đoán vị trí PTM, một số hạn chế và đề xuất hướng nghiên cứu.

**Chương 2. Mô hình học máy tổ hợp dự đoán vị trí sửa đổi sau dịch mã trong chuỗi protein**

Trong chương này, NCS trình bày về phương pháp dự đoán vị trí sửa đổi sau dịch mã trên protein dựa trên kỹ thuật học máy tổ hợp. Phương pháp đề xuất sử dụng ba mô hình thành phần gồm Random Forest (RF), Extreme Gradient Boosting (XGBoost) và Support Véc tơ Machine (SVM), được huấn luyện độc lập và kết hợp kết quả dự đoán bằng chiến lược trung bình có trọng số (Weighted Average Voting). Cách tiếp cận này

giúp khai thác thế mạnh riêng biệt của từng mô hình học máy, từ đó cải thiện độ chính xác và tính ổn định trong dự đoán. Các kết quả thực nghiệm cho thấy phương pháp tổ hợp mang lại hiệu suất vượt trội so với các mô hình đơn lẻ.

**Chương 3. Mô hình học sâu lai kết hợp xử lý ngôn ngữ tự nhiên dự đoán vị trí sửa đổi sau dịch mã trong chuỗi protein**

Trong chương này, NCS giới thiệu mô hình học sâu lai tích hợp với kỹ thuật NLP nhằm nâng cao khả năng dự đoán vị trí PTM. Mô hình được xây dựng dựa trên sự kết hợp giữa mạng CNN1D để khai thác đặc trưng cục bộ và mạng LSTM/Bi-LSTM để học ngữ cảnh tuần tự trong chuỗi protein. Kỹ thuật NLP được áp dụng để biểu diễn chuỗi axit amin dưới dạng véc tơ, giúp mô hình học được các thông tin ẩn trong chuỗi sinh học một cách hiệu quả hơn. Thực nghiệm trên các tập dữ liệu PTM cho thấy mô hình lai cho kết quả chính xác và ổn định hơn so với các mô hình đơn cấu trúc, đồng thời khẳng định được khả năng tổng quát hóa của phương pháp.

**Chương 4. Mô hình học chắt lọc tri thức kết hợp xử lý ngôn ngữ tự nhiên dự đoán vị trí sửa đổi sau dịch mã trong chuỗi protein**

Trong chương này, NCS trình bày về đề xuất một phương pháp dự đoán vị trí PTM dựa trên kiến trúc học chắt lọc tri thức kết hợp NLP. Phương pháp sử dụng mô hình Giáo viên-Học viên, trong đó mô hình Giáo viên có cấu trúc lớn hơn và được huấn luyện trên tập dữ liệu đa loài để rút trích tri thức, sau đó truyền lại cho mô hình Học viên nhỏ gọn hơn, được huấn luyện trên tập dữ liệu chuyên biệt cho một loài. Việc kết hợp kỹ thuật NLP giúp mã hóa chuỗi protein thành không gian véc tơ, hỗ trợ quá trình học hiệu quả từ dữ liệu thô. Thực nghiệm chứng minh rằng mô hình Học viên không những giảm đáng kể số lượng tham số mà vẫn duy trì được hiệu suất cao, thậm chí vượt trội so với các mô hình không sử dụng học chắt lọc tri thức trong cùng điều kiện huấn luyện.

Phần cuối cùng của luận án NCS trình bày các kết luận chính và đề xuất hướng nghiên cứu tiếp theo.

# CHƯƠNG 1. TỔNG QUAN DỰ ĐOÁN VỊ TRÍ SỬA ĐỔI SAU DỊCH MÃ TRONG CHUỖI PROTEIN VÀ CÁC KIẾN THỨC NỀN TẢNG

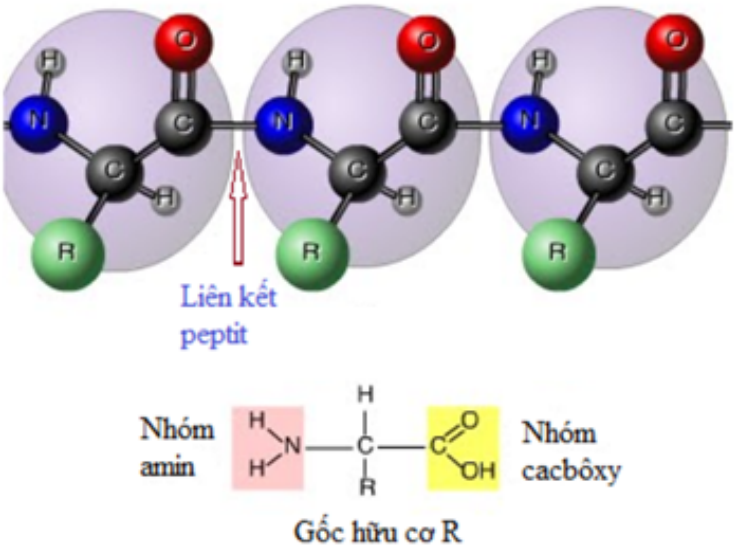
*Trong chương 1, NCS trình bày một số kiến thức nền tảng về protein và protein sửa đổi sau dịch mã (Post-translational modification - PTM), vai trò của việc xây dựng mô hình dự đoán vị trí PTM hiệu suất cao. Phần tiếp theo của chương trình bày về bài toán dự đoán vị trí PTM, các bước xây dựng mô hình dự đoán vị trí PTM, tổng quan tình hình nghiên cứu trong bối cảnh AI (SOTA), các khoảng trống nghiên cứu, thách thức của mô hình dự đoán vị trí PTM, hướng nghiên cứu trong luận án. Phần cuối của chương trình bày về môi trường sử dụng để thực nghiệm, phương pháp đánh giá mô hình đề xuất. Một phần kết quả nghiên cứu được đăng trong bài báo tổng quan các mô hình học máy trên tạp chí Expert Opinion on Drug Metabolism  Toxicology, SCIE Q1, IF=3.9 (CT1) và hội thảo iFUZZY, Kagawa, Japan (CT2).*

## 1.1 Giới thiệu chung

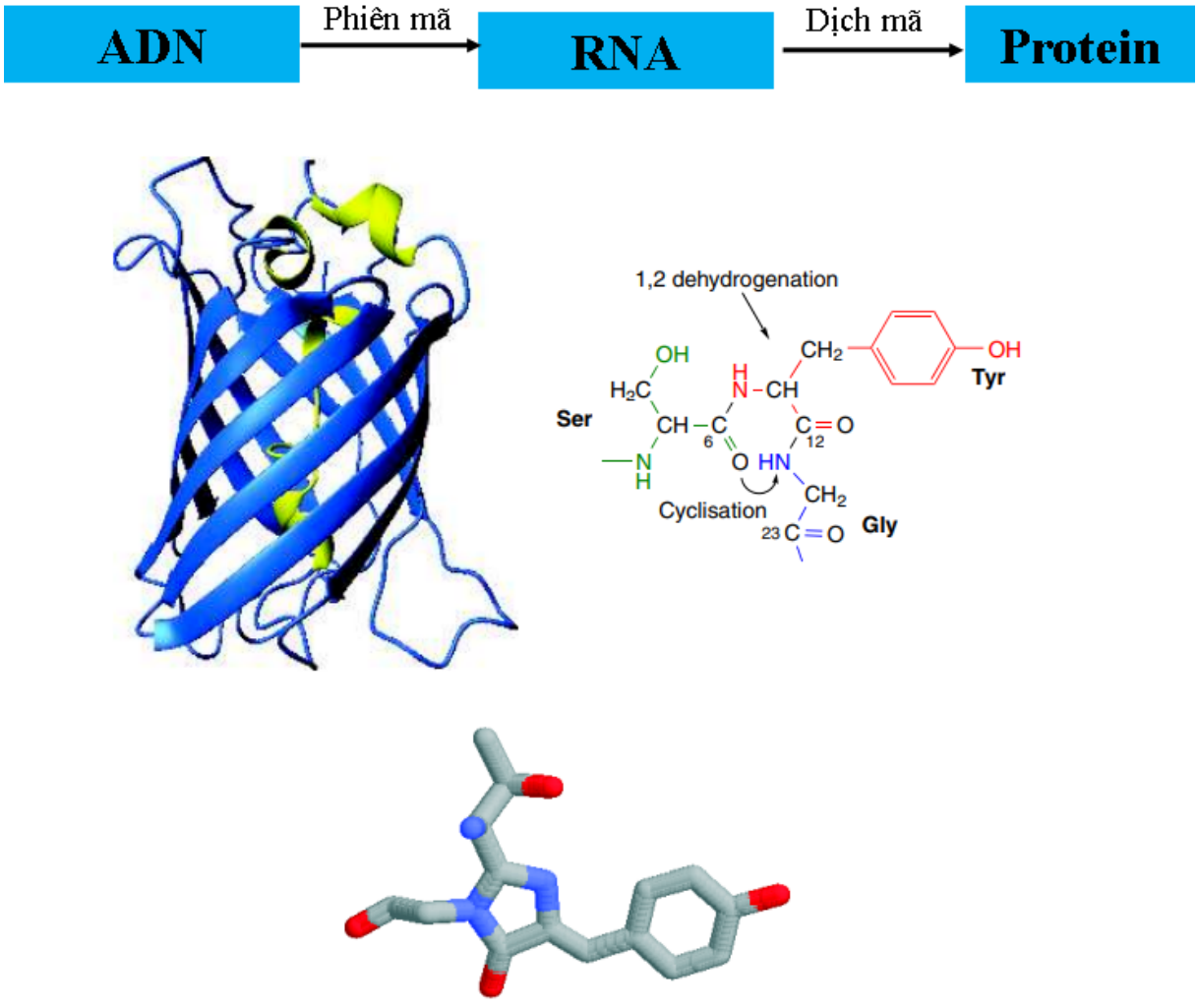
### *1.1.1 Protein*

Protein, hay còn gọi là chất đạm, là những đại phân tử sinh học được cấu thành từ chuỗi các axit amin liên kết với nhau thông qua liên kết peptide. Sự khác biệt giữa các protein chủ yếu nằm ở trình tự sắp xếp của các axit amin, vốn được mã hóa bởi trình tự nucleotide trong các gene tương ứng. Sau khi được tổng hợp, chuỗi polypeptide trải qua quá trình cuộn gập (protein folding) để hình thành cấu trúc ba chiều đặc thù, quyết định đến chức năng sinh học của protein đó. Protein có nhiều chức năng sinh học khác nhau, từ sao chép DNA, hình thành cấu trúc bộ xương tế bào, vận chuyển oxy xung quanh cơ thể của các sinh vật đa bào đến chuyển đổi một phân tử này thành phân tử khác [120].

Những tiến bộ trong di truyền học phân tử cho thấy nhiều bệnh bắt nguồn từ các khiếm khuyết cụ thể của protein [120], do đó nghiên cứu về protein có ý nghĩa quan trọng. Có 20 axit amin tham gia cấu tạo nên chuỗi protein [120]. Đặc biệt, mỗi loại protein được phân biệt không chỉ bởi số lượng mà còn bởi thành phần và trình tự sắp xếp của các axit amin trong chuỗi polypeptide. Bảng1.1 trình bày hệ thống các ký hiệu mã hóa protein được sử dụng trong các cơ sở dữ liệu protein chuyên biệt, nhằm phục vụ cho nhiều mục đích nghiên cứu khác nhau như phân tích chức năng sinh học, nghiên cứu cơ chế hoạt động của protein, cũng như hỗ trợ trong các nghiên cứu về sửa đổi sau dịch mã và thiết kế thuốc.




**Hình 1.1 Cấu tạo của axit amin (gồm một nhóm amin (-$NH_2$), một nhóm carboxyl (-COOH), và gốc hữu cơ (R)) và sự liên kết các axit amin bởi liên kết peptit [110].**




**Hình 1.2 Quá trình hình thành nên chuỗi protein ( [120])**

Quá trình tổng hợp protein diễn ra qua hai giai đoạn chính, bao gồm phiên mã và dịch mã (Hình 1.2). Trước hết, trong giai đoạn phiên mã (Transcription), ADN trong nhân tế bào được chuyển đổi thành RNA. Quá trình này giúp mã di truyền được sao chép và vận chuyển ra ngoài tế bào chất. Tiếp theo, trong giai đoạn dịch mã (Translation), RNA được ribosome đọc theo từng bộ ba mã di truyền (codon), từ đó lắp ráp các axit amin theo đúng trình tự để hình thành chuỗi polypeptide. Kết quả là một protein ban đầu (pre-protein) được tạo thành, sau đó sẽ trải qua các bước chỉnh sửa và hoàn thiện để thực hiện chức năng sinh học của nó. Protein có thể tồn tại ở bốn dạng cấu trúc [120], Cấu trúc bậc một là trình tự các axit amin trong chuỗi polypeptit, quyết định tính đặc thù và cấu trúc không gian của protein. Chuỗi này có thể gấp nếp thành cấu trúc bậc hai như vòng xoắn alpha hoặc phiến gấp beta nhờ liên kết hydro. Các cấu trúc bậc hai tiếp tục cuộn gấp thành cấu trúc bậc ba với hình dạng ba chiều ổn định. Một số protein gồm nhiều chuỗi polypeptit liên kết với nhau tạo thành cấu trúc bậc bốn (Hình 1.2). Protein cấu trúc bậc 1 được lưu trữ trong các ngân hàng protein như Hình 1.3 dưới đây.


```
>sp|O00222|GRM8_HUMAN Metabotropic glutamate receptor 8 OS=Homo sapiens OX=9606 GN=GRM8 PE=1 SV=2
MVCEGKRSASCPCFFLLTAKFYWILTMMQRTHSQEYAHSIRVDGDIILGGLFPVHAKGER
GVPCGELKKEKGIHRLEAMLYAIDQINKDPDLLSNITLGVRILDTCSRDTYALEQSLTFV
QALIEKDASDVKCANGDPPIFTKPDKISGVIGAAASSVSIMVANILRLFKIPQISYASTA
PELSDNTRYDFFSRVVPPDSYQAQAMVDIVTALGWNYVSTLASEGNYGESGVEAFTQISR
EIGGVCIAQSQKIPREPRPGEFEKIIKRLLETPNARAVIMFANEDDIRRILEAAKKLNQS
GHFLWIGSDSWGSKIAPVYQQEEIAEGAVTILPKRASIDGFDRYFRSRTLANNRRNVWFA
EFWEENFGCKLGSHGKRNSHIKKCTGLERIARDSSYEQEGKVQFVIDAVYSMAYALHNMH
KDLCPGYIGLCPRMSTIDGKELLGYIRAVNFNGSAGTPVTFNENGDAPGRYDIFQYQITN
KSTEYKVIGHWTNQLHLKVEDMQWAHREHTHPASVCSLPCKPGERKKTVKGVPCCWHCER
CEGYNYQVDELSCELCPLDQRPNMNRTGCQLIPIIKLEWHSPWAVVPVFVAILGIIATTF
VIVTFVRYNDTPIVRASGRELSYVLLTGIFLCYSITFLMIAAPDTIICSFRRVFLGLGMC
FSYAALLTKTNRIHRIFEQGKKSVTAPKFISPASQLVITFSLISVQLLGVFVWFVVDPPH
IIIDYGEQRTLDPEKARGVLKCDISDLSLICSLGYSILLMVTCTVYAIKTRGVPETFNEA
KPIGFTMYTTCIIWLAFIPIFFGTAQSAEKMYIQTTTLTVSMSLSASVSLGMLYMPKVYI
IIFHPEQNVQKRKRSFKAVVTAATMQSKLIQKGNDRPNGEVKSELCESLETNTSSTKTTY
ISYSNHSI
```

**Hình 1.3 Protein bậc 1 GRM8_HUMAN trong database UniProt**

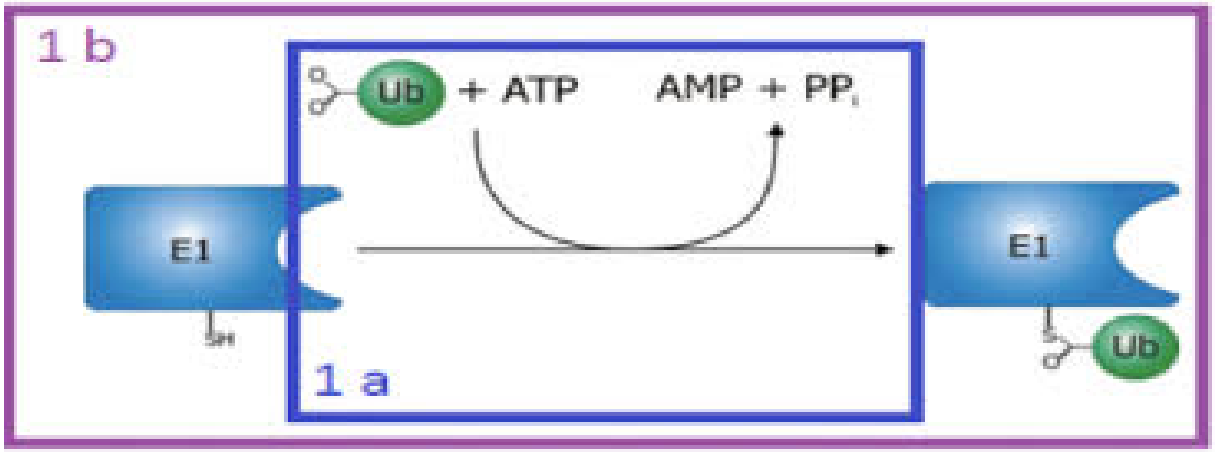
**Bảng 1.1 Bảng 20 axit amin cơ bản cấu tạo nên các chuỗi protein**

| TT | Tên | Ký hiệu 1 ký tự | Ký hiệu 3 ký tự | Trọng lượng (Dalton) |
|---|---|---|---|---|
| 1 | Glycine | G | Gly | 75.07 |
| 2 | Alanine | A | Ala | 89.09 |
| 3 | Valine | V | Val | 117.14 |
| 4 | Leucine | L | Leu | 131.17 |
| 5 | Isoleucine | I | Ile | 131.17 |
| 6 | Methionine | M | Met | 149.21 |
| 7 | Proline | P | Pro | 115.13 |
| 8 | Phenylalanine | F | Phe | 165.19 |
| 9 | Tryptophan | W | Trp | 204.22 |
| 10 | Serine | S | Ser | 105.90 |
| 11 | Threonine | T | Thr | 119.12 |
| 12 | Asparagine | N | Asn | 132.12 |
| 13 | Glutamine | Q | Gln | 146.15 |
| 14 | Tyrosine | Y | Tyr | 181.19 |
| 15 | Cysteine | C | Cys | 121.16 |
| 16 | Lysine | K | Lys | 146.20 |
| 17 | Arginine | R | Arg | 174.21 |
| 18 | Histidine | H | His | 155.16 |
| 19 | Aspartic acid | D | Asp | 133.10 |
| 20 | Glutamic acid | E | Glu | 147.12 |

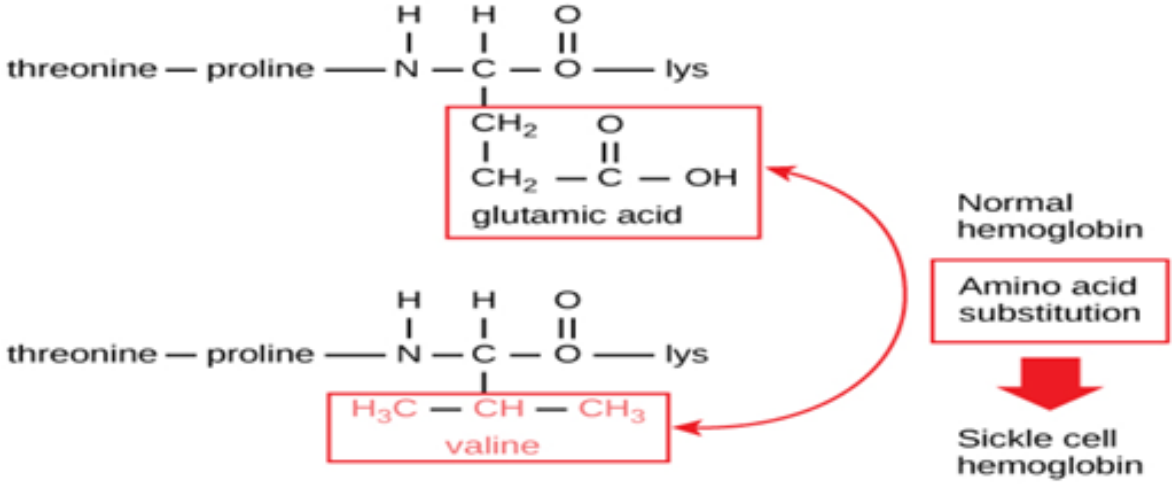
### *1.1.2 Protein sửa đổi sau dịch mã*

Sau quá trình dịch mã, các protein mới tổng hợp không hoàn toàn ở trạng thái hoạt động mà thường trải qua các biến đổi hóa học bổ sung gọi là sửa đổi sau dịch mã (Post-

Translational Modifications – PTMs). Đây là quá trình trong đó các nhóm chức hóa học (như phosphate, ubiquitin, acetyl, methyl, succinyl...) được gắn hoặc loại bỏ khỏi chuỗi protein, thông qua hệ thống enzyme chuyên biệt. Hình 1.4 là ví dụ quá trình hình thành PTM Ubiquitination.




**Hình 1.4 Quá trình hình thành PTM Ubiqutination bằng việc gắn Ubiquitin vào protein mục tiêu bởi tác động của các enzim E1 [88].**




**Hình 1.5 Sửa đổi sau dịch mã trong chuỗi -hemoglobin (147 axit amin), trong protein này axit amin glutamic (vị trí thứ 7) bị thay bởi axit amin valine gây ra bệnh thiếu máu hồng cầu hình liềm [54].**

Sửa đổi sau dịch mã của protein là một cách quan trọng để điều chỉnh cấu trúc, chức năng protein, sự độ ổn định, và tương tác của protein trong tế bào, điều hòa tín hiệu tế bào, chu kỳ tế bào, đáp ứng miễn dịch, và các quá trình sinh học quan trọng khác. Hiện nay có khoảng hơn 600 PTM khác nhau [107], và có khoảng 50% $\sim$ 90% protein trong tế bào người có các loại biến đổi sau dịch mã khác nhau. Các nghiên cứu đã chỉ

ra rằng nhiều loại biến đổi protein sau dịch mã có liên quan đến việc điều chỉnh vi môi trường khối u, đặc biệt là sự tăng sinh, trao đổi chất của các tế bào miễn dịch và tế bào khối u. Những thay đổi trong quá trình sửa đổi sau dịch mã của nhiều gen tiền ung thư ức chế khối u hoặc các chất điều hòa ung thư quan trọng khác có thể ảnh hưởng trực tiếp đến sự xuất hiện, phát triển, điều trị và tiên lượng của ung thư . Một ví dụ về sửa đổi sau dịch mã [105] có thể gây ra bệnh thiếu máu hồng cầu hình lưỡi niềm Hình 1.5.

Bảng 1.2 trình bày một số loại PTM phổ biến cùng các axit amin liên quan. Mỗi PTM chỉ xảy ra tại một hoặc một số axit amin nhất định, do đó việc dự đoán vị trí PTM mang tính đặc thù. Từ đặc điểm này, bài toán dự đoán PTM được chia thành hai dạng: phân lớp nhị phân và phân lớp đa phân. Các loại PTM tiêu biểu như phosphorylation, acetylation, ubiquitination, succinylation, methylation, malonylation, SUMOylation và hydroxylation được tổng hợp trong bảng dưới đây.

**Bảng 1.2 Một số loại PTM phổ biến và axit amin liên quan**

| **Loại PTM** | **Axit amin liên quan** | **Loại dự đoán PTM** |
|---|---|---|
| Phosphorylation (Phosphoryl hóa) | Serine (S), Threonine (T), Tyrosine (Y) | Đa phân |
| Acetylation (Acetyl hoá) | Lysine (K) | Nhị phân |
| Ubiquitination (Ubiquitin hoá) | Lysine (K) | Nhị phân |
| Succinylation (Succinyl hóa) | Lysine (K) | Nhị phân |
| Methylation (Methyl hoá) | Lysine (K), Arginine (R) | Đa phân |
| Malonylation (Malonyl hoá) | Lysine (K) | Nhị phân |
| SUMOylation (SUMO hoá) | Lysine (K) | Nhị phân |
| Hydroxylation (Hydroxyl hoá) | Proline (P), Lysine (K) | Đa phân |

### *1.1.3 Vai trò của bài toán dự đoán vị trí PTM và các phương pháp chính dự đoán vị trí PTM hiện nay*

PTM đóng vai trò quan trọng trong việc điều chỉnh hoạt động của hầu hết các protein nhân chuẩn. Chúng có trách nhiệm cảm biến và truyền tín hiệu để điều hòa các chức năng tế bào và các quá trình tín hiệu sinh học khác nhau. Tuy nhiên, việc phân tích PTM đặt ra nhiều thách thức lớn, nhưng đồng thời cũng mang lại những hiểu biết không thể thiếu về chức năng sinh học.

Các bất thường của PTM có liên quan chặt chẽ đến các bệnh lý bao gồm ung thư, trong khi một số enzyme điều hòa liên quan đến PTM là mục tiêu của thuốc [67]. Do đó, việc xác định các vị trí PTM trong protein rất quan trọng đối với cả nghiên cứu cơ bản và thiết kế thuốc.

Hiện nay, có hai phương pháp chính để xác định vị trí PTM trong protein:

Phương pháp thực nghiệm: Chủ yếu dựa vào khối phổ và các kỹ thuật sinh học phân tử khác [25, 100]. Phép đo phổ khối (MS) có độ phân giải cao cho phép xác định chính xác khối lượng phân tử của protein hoặc peptide và khối lượng phân tử của protein sẽ thay đổi tương ứng sau khi được protein sửa đổi sau dịch mã, dựa trên đặc điểm này, kỹ thuật MS được áp dụng để xác định vị trí PTM. Mặc dù có độ chính xác cao, phương pháp này đòi hỏi chi phí lớn, thời gian thực hiện lâu và công sức đáng kể.

Phương pháp dự đoán vị trí PTM bằng học máy [29, 69, 90, 119]: Xuất phát từ sự phát triển mạnh mẽ của công nghệ tính toán và dữ liệu lớn, phương pháp này có ưu điểm vượt trội về tốc độ, chi phí và khả năng tổng quát hóa. Các mô hình học máy có thể học từ dữ liệu huấn luyện để dự đoán các vị trí PTM tiềm năng với hiệu suất cao, mở ra hướng tiếp cận hiệu quả hơn so với các phương pháp thực nghiệm truyền thống.

Với những lợi thế này, dự đoán vị trí PTM bằng học máy đang trở thành một hướng nghiên cứu quan trọng, hỗ trợ đắc lực cho các nhà khoa học trong việc khám phá các cơ chế PTM và phát triển các phương pháp điều trị bệnh dựa trên PTM.

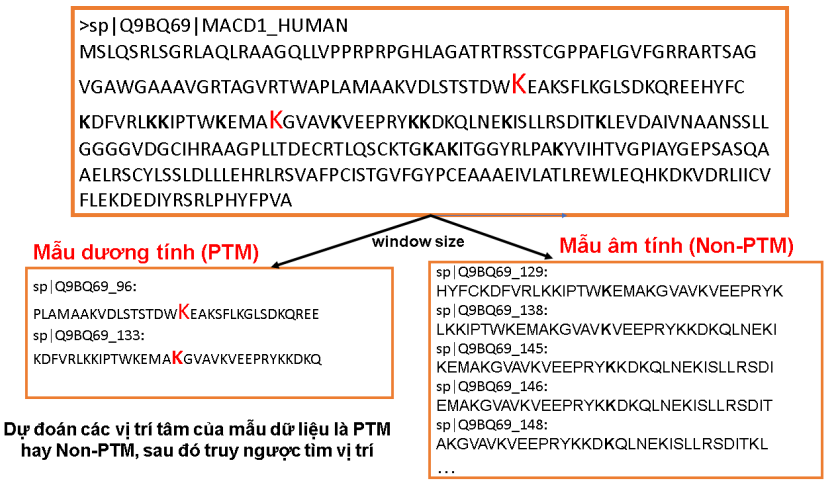
## 1.2 Bài toán dự đoán vị trí PTM dựa trên học máy

**Đầu vào:** Chuỗi protein bậc 1, kèm theo các vị trí nghi ngờ có khả năng bị sửa đổi sau dịch mã.

**Đầu ra:** Nhãn dự đoán cho từng vị trí nghi ngờ, xác định xem đó là vị trí PTM hay Non_PTM (có thể kèm theo xác suất dự đoán).

**Mục tiêu:** Đề xuất một mô hình học máy hiệu suất cao, có khả năng dự đoán chính xác các vị trí PTM trên chuỗi protein.

Quá trình biến đổi sau dịch mã là một cơ chế sinh học quan trọng, xảy ra tại một hoặc một số axit amin cụ thể trong chuỗi protein. Trong thực tế, chỉ một phần nhỏ các axit amin trên chuỗi protein chịu ảnh hưởng của việc sửa đổi sau dịch mã dưới tác động của enzim như phosphoryl hóa, ubiquitin hóa, succinyl hóa,... Mặt khác, độ dài của các chuỗi protein là khác nhau có chuỗi dài vài nghìn axit amin, các vị trí sửa đổi trên chuỗi protein thường không được biết trước, do đó cần áp dụng kỹ thuật cửa sổ trượt (sliding window) để tạo ra các phân đoạn peptide có độ dài cố định, trong đó tâm cửa sổ tương ứng với vị trí nghi ngờ bị sửa đổi [82, 112] (Hình 1.6).




**Hình 1.6 Chuyển từ bài toán tìm vị trí nghi ngờ sửa đổi sau dịch mã, vị trí nghi ngờ đó nằm ở thứ tự bao nhiêu trong chuỗi về bài toán phân loại nhị phân**

Mỗi phân đoạn được xem như một mẫu đầu vào cho mô hình học máy. Nếu axit amin tại tâm cửa sổ là vị trí PTM đã biết, mẫu đó được gán nhãn dương tính (1); ngược lại, nếu không bị sửa đổi, mẫu được gán nhãn âm tính (0). Bài toán vì vậy được quy về một bài toán phân loại nhị phân, với đầu vào là các phân đoạn đã gán nhãn, và đầu ra là xác suất của axit amin ở trung tâm của cửa sổ là PTM hay Non-PTM. Dựa trên kết quả đầu ra của mô hình, có thể ánh xạ ngược lại các tâm cửa sổ về vị trí tương ứng trên chuỗi protein gốc, từ đó xác định các vị trí có khả năng bị sửa đổi sau dịch mã. Cách tiếp cận này cho phép xây dựng các công cụ dự đoán vị trí PTM với độ chính xác cao, góp phần hỗ trợ các nghiên cứu y sinh và thiết kế thuốc.

Từ đó, bài toán dự đoán vị trí PTM được chuyển thành một bài toán phân lớp nhị phân (Hình 1.7), trong đó:

**Đầu vào:** Là một tập các đoạn peptide $X = \{x_1, x_2, \ldots, x_n\}$, trong đó mỗi $x_i$ là một đoạn peptide cố định có độ dài là w (w là kích thước cửa sổ trượt), được trích xuất từ các chuỗi protein ban đầu bằng phương pháp cửa sổ trượt. Vị trí trung tâm của mỗi đoạn peptide được giả định là vị trí nghi ngờ có thể xảy ra sự kiện sửa đổi sau dịch mã (PTM).

**Đầu ra:** Vị trí trung tâm của mỗi đoạn peptide là PTM hay Non-PTM, có thể phát biểu đầu ra như một tập các nhãn tương ứng $Y = \{y_1, y_2, \ldots, y_n\}$, trong đó:

- $y_i = 1$ nếu đoạn $x_i$ có vị trí trung tâm là PTM,
- $y_i = 0$ nếu đoạn $x_i$ có vị trí trung tâm là Non-PTM.

**Mục tiêu:** Xây dựng một hàm phân lớp nhị phân:

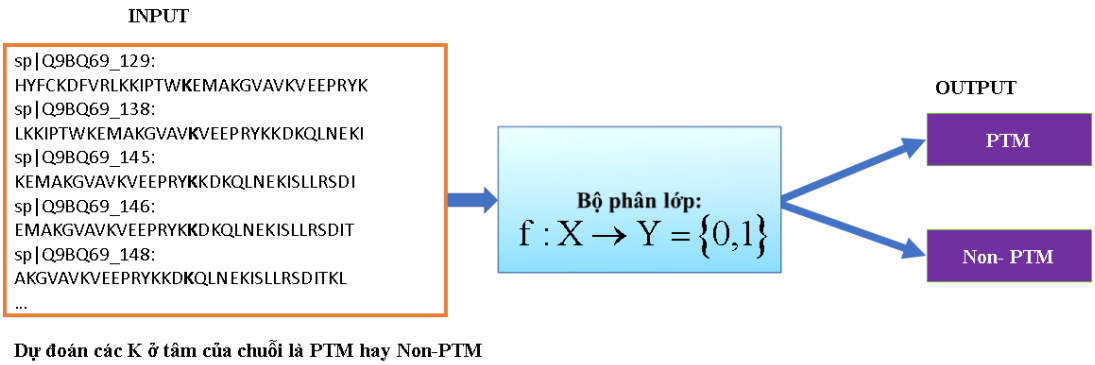
$$f : \mathscr{X} \to \mathscr{Y} \tag{1.1}$$

trong đó:

- $\mathscr{X}$ là không gian các đoạn peptide đầu vào;
- $\mathscr{Y} = \{0, 1\}$ là tập nhãn đầu ra;

sao cho:

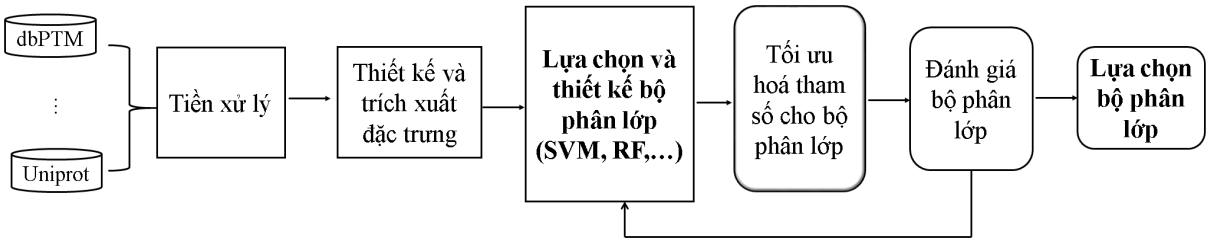
$$f(x_i) \approx y_i \quad \forall i = 1, 2, \ldots, n \tag{1.2}$$



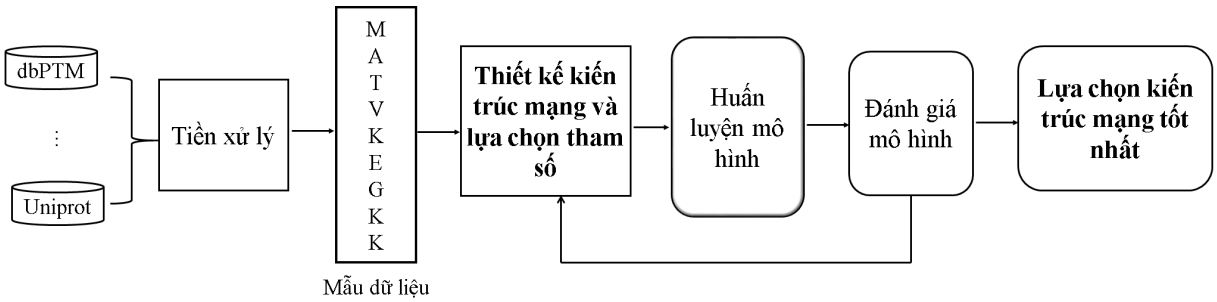

**Hình 1.7 Mô tả bài toán dự đoán vị trí PTM**

## 1.3 Xây dựng mô hình dự đoán vị trí PTM

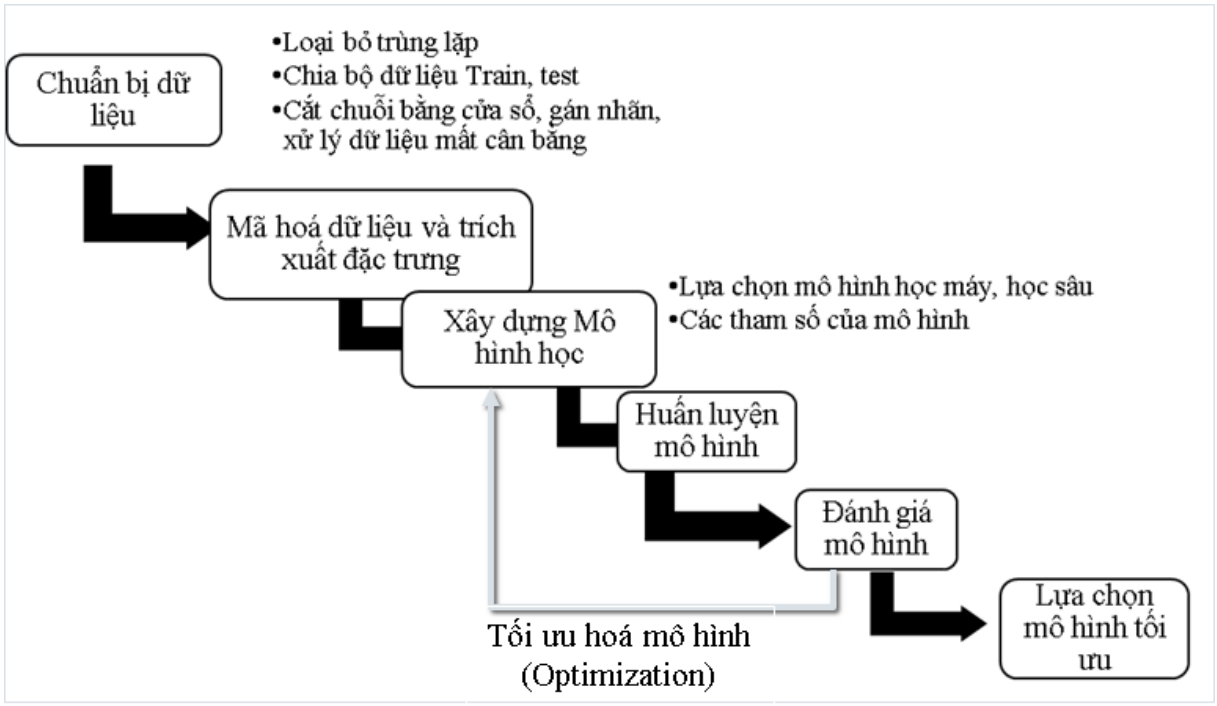
Hình 1.8, Hình 1.9 là sơ đồ tổng quan xây dựng dự đoán vị trí PTM dựa trên học máy, học sâu [29]. Như vậy bài toán dự đoán vị trí PTM có thể chia thành các pha: Thu thập dữ liệu và tiền xử lý dữ liệu; Trích xuất đặc trưng và mã hoá; Xây dựng mô hình; Huấn luyện mô hình; Đánh giá mô hình; Lựa chọn mô hình. Chi tiết như Hình 1.10 dưới đây.




**Hình 1.8 Sơ đồ tổng quan dự đoán vị trí PTM dựa trên học máy [29]**




**Hình 1.9 Sơ đồ tổng quan dự đoán vị trí PTM dựa trên học sâu [29]**




**Hình 1.10 Các bước xây dựng và huấn luyện mô hình dự đoán PTM**

### *1.3.1 Thu thập và tiền xử lý dữ liệu*

Trong phạm vi nghiên cứu của luận án này, NCS chủ yếu sử dụng các bộ dữ liệu đã được công bố từ các nghiên cứu trước đó. Việc sử dụng dữ liệu có sẵn nhằm đảm bảo tính khách quan và công bằng khi so sánh hiệu suất giữa các mô hình, đồng thời giúp kết quả nghiên cứu có thể đối sánh trực tiếp với các phương pháp hiện có trong cùng lĩnh vực. Bên cạnh đó, việc sử dụng bộ dữ liệu chuẩn cũng giúp giảm thiểu sai lệch do quá trình thu thập dữ liệu mới và đảm bảo tính tái lập cho các nghiên cứu tiếp theo. Quy trình tiền xử lý dữ liệu vẫn được trình bày đầy đủ trong luận án để minh bạch hóa quy trình nghiên cứu, đồng thời tạo tiền đề cho việc mở rộng hoặc phát triển các nghiên cứu tương lai khi cần thiết.

Do tính phức tạp và đặc thù của các PTM khác nhau, không có cơ sở dữ liệu

(CSDL) nào có thể cung cấp tài nguyên toàn diện cho nghiên cứu PTM. Trong lĩnh vực nghiên cứu này, có một số CSDL chuẩn được sử dụng rộng rãi như: Uniprot, dbPTM, PhosphoSitePlus, NCBI,... và dữ liệu từ các bài báo đăng trên các tạp chí uy tín. Hình 1.11 chi viết tóm tắt tên các CSDL, địa chỉ truy cập, và loại PTM có thể khai thác để phục vụ cho các nghiên cứu dự đoán vị trí PTM.

| Cơ sở dữ liệu | Link Web | Loại PTM |
|---|---|---|
| Uniprot | www.uniprot.org | Multiple |
| PhosphoSitePlus | www.phosphosite.org | Multiple |
| dbPTM | dbptm.mbc.nctu.edu.tw | Multiple |
| PLMD | Cplm.biocuckoo.org | Multiple |
| PTMfunc | ptmfunc.com | Multiple |
| Phospho.ELM | Phospho.elm.eu.org | Phosphorylation |
| PhosphoPep | www.phosphopep.org | Phosphorylation |
| PhosPhAt | Phosphat.mpimp-golm.mpg.de | Phosphorylation |
| SCUD | Scud.kaist.ac.kr | Ubiquitination |
| dbOGAP | Cbsb.lombardi.geogetown.edu/OGAP.html | Glycosylation |
| iPTMnet | Research.bioinfomatics.udel.edu/iptmnet/ | Multiple |
| NCBI | www.ncbi.nlm.nih.gov/guide/proteins | Multiple |
| EPSD | https://epsd.biocuckoo.cn/ | Phosphorylation |

**Hình 1.11 Một số CSDL về PTM chuẩn**

**Tiền xử lý dữ liệu gồm các bước sau:** *Thứ nhất, xử lý dữ liệu thô*

- Chuẩn hóa trình tự protein, kiểm tra lại tính đúng đắn của dữ liệu.

- Loại bỏ dữ liệu trùng lặp các chuỗi protein ở các nguồn thu thập, giữ lại chuỗi không trùng lặp.

| ID | Position | Gene | Type | Sequence |
|---|---|---|---|---|
| O00139 | 499 | KIF2A | Succinylation | MATANFGKIQIGIYVEIKRSDGRIHQAMVTSLNEDNESVTVEWIENGDTKGKEIDLESIFSLNPDLVPDEEIEPSPETPPI |
| O00148 | 155 | DDX39A | Succinylation | MAEQDVENDLLDYDEEEEPQAPQESTPAPPKKDIKGSYVSIHSSGFRDFLLKPELLRAIVDCGFEHPSEVQHECIPQAIL |
| O00148 | 187 | DDX39A | Succinylation | MAEQDVENDLLDYDEEEEPQAPQESTPAPPKKDIKGSYVSIHSSGFRDFLLKPELLRAIVDCGFEHPSEVQHECIPQAIL |
| O00148 | 333 | DDX39A | Succinylation | MAEQDVENDLLDYDEEEEPQAPQESTPAPPKKDIKGSYVSIHSSGFRDFLLKPELLRAIVDCGFEHPSEVQHECIPQAIL |
| O00148 | 52 | DDX39A | Succinylation | MAEQDVENDLLDYDEEEEPQAPQESTPAPPKKDIKGSYVSIHSSGFRDFLLKPELLRAIVDCGFEHPSEVQHECIPQAIL |
| O00159 | 667 | MYO1C | Succinylation | MALQVELVPTGEIIRVVHPHRPCKLALGSDGVRVTMESALTARDRVGVQDFVLLENFTSEAAFIENLRRRFRENLIYTYI |
| O00159 | 704 | MYO1C | Succinylation | MALQVELVPTGEIIRVVHPHRPCKLALGSDGVRVTMESALTARDRVGVQDFVLLENFTSEAAFIENLRRRFRENLIYTYI |
| O00159 | 885 | MYO1C | Succinylation | MALQVELVPTGEIIRVVHPHRPCKLALGSDGVRVTMESALTARDRVGVQDFVLLENFTSEAAFIENLRRRFRENLIYTYI |
| O00186 | 214 | STXBP3 | Succinylation | MAPPVAERGLKSVVWQKIKATVFDDCKKEGEWKIMLLDEFTTKLLASCCKMTDLLEEGITVVENIYKNREPVRQMKA |
| O00186 | 590 | STXBP3 | Succinylation | MAPPVAERGLKSVVWQKIKATVFDDCKKEGEWKIMLLDEFTTKLLASCCKMTDLLEEGITVVENIYKNREPVRQMKA |
| O00203 | 1088 | AP3B1 | Succinylation | MSSNSFPYNEQSGGGEATELGQEATSTISPSGAFGLFSSDLKKNEDLKQMLESNKDSAKLDAMKRIVGMIAKGKNAS |

**Hình 1.12 Bộ dữ liệu ở bước 2 ở định dạng .csv**

Thứ hai, Tạo mẫu dữ liệu

Đề tài luận án tập trung vào việc xác định vị trí bị sửa đổi sau dịch mã xảy ra tại một axit amin cụ thể nào đó trên chuỗi (giả sử là axit amin K). Do đó, việc khoanh vùng vị trí bị sửa đổi sau dịch mã là cần thiết. Vì vậy, NCS sử dụng một cửa sổ trượt (window-size = 2×n +1) để cắt chuỗi protein đầu vào thành các đoạn con (peptides) có độ dài 2×n +1 (với trung tâm là vị trí của axit amin K đã được kiểm chứng bằng các thực nghiệm y/sinh học là bị sửa đổi sau dịch mã). Các đoạn con này (peptides) sau đó sẽ được sử dụng như là các mẫu dữ liệu dương tính (positive samples).

Đối với các mẫu dữ liệu âm tính (negative sample): Về bản chất, tất cả các peptides có độ dài 2×n +1 (với trung tâm là vị trí của axit amin không phải K, hoặc axit amin là K nhưng chưa được kiểm chứng thực nghiệm là bị sửa đổi sau dịch mã) đều là mẫu dữ liệu âm tính. Tuy nhiên, việc lấy dữ liệu âm tính như vậy sẽ dẫn đến hiện tượng mất cân bằng quá lớn vì thực tế số lượng các axit amin K so với các axit amin còn lại là quá bé. Vì vậy, trong Bioinformatics, các nhà nghiên cứu thường chỉ sử dụng mẫu âm tính là các (peptides) có độ dài 2×n +1 (với trung tâm là vị trí của axit amin K nhưng chưa được kiểm chứng bằng các thực nghiệm y/sinh học là bị sửa đổi sau dịch mã).

Quá trình tạo mẫu dữ liệu huấn luyện trong dự đoán vị trí PTM có thể được minh họa qua ví dụ sau:

Giả sử với chuỗi protein sp|Q9BQ69, các nghiên cứu đã xác định rằng các vị trí 96 và 133 là các vị trí có sửa đổi sau dịch mã (PTM sites – K màu đỏ trong Hình 1.6). Khi sử dụng phương pháp cửa sổ trượt với độ dài 33, các vị trí lysine (K) tại 96 và 133 sẽ được đặt làm trung tâm để tạo ra các mẫu dữ liệu dương tính đại diện cho các vị trí có PTM. Ngược lại, các vị trí lysine (K) khác trong chuỗi protein, nhưng không liên quan đến sửa đổi PTM, sẽ được chọn làm trung tâm để tạo ra các mẫu dữ liệu âm tính đại diện cho các vị trí Non-PTM.

Ngoài ra, nếu một vị trí lysine nằm gần đầu hoặc cuối chuỗi protein, cửa sổ trượt có thể vượt ra ngoài giới hạn của chuỗi. Để xử lý trường hợp này, các ký hiệu axit amin

giả X sẽ được sử dụng để bổ sung vào hai đầu, đảm bảo rằng tất cả các đoạn đều có độ dài đúng bằng cửa sổ trượt.

Cuối cùng: Loại bỏ các chuỗi tương đồng bằng CD-HIT. Nhằm loại các trình tự protein có mức độ tương đồng cao giúp giảm sự trùng lặp trong dữ liệu. Điều này, giúp mô hình học được các đặc trưng đa dạng thay vì dựa vào các mẫu tương tự, tránh hiện tượng quá khớp trong quá trình huấn luyện. Công cụ sử dụng: CD-HIT [46] (Cluster Database at High Identity with Tolerance) Đây là một công cụ mạnh mẽ để phân cụm các trình tự protein hoặc DNA dựa trên mức độ tương đồng.

| ID | Peptide_WS | Label | label |
|---|---|---|---|
| Q92575_UBXN4_287 | IALDRAERAARFAKTKEEVEAAKAAALLAKQ | 1 | positive |
| P62829_RPL23_43 | CADNTGAKNLYIISVKGIKGRLNRLPAAGVG | 1 | positive |
| P46777_RPL5_241 | YIKNSVTPDMMEEMYKKAHAAIRENPVYEKK | 1 | positive |
| O60879_DIAPH2_25 | GAGGGSEEPGGGRSNKRSAGNRAANEEETKN | 0 | negative |
| P00338_LDHA_284 | RVHPVSTMIKGLYGIKDDVFLSVPCILGQNG | 0 | negative |
| Q99798_ACO2_651 | VTQEFGPVPDTARYYKKHGIRWVVIGDENYG | 0 | negative |
| Q13045_FLII_974 | AEGKEGEEATAEAEEKQPEEDFQCIVYFWQG | 0 | negative |
| O15091_PRORP_202 | FHMQTSEVIDVFEIMKARYKTLEPRGYSLLI | 0 | negative |

**Hình 1.13 Bộ dữ liệu sử dụng trong huấn luyện mô hình (Peptide_WS, nhãn Label)**

### *1.3.2 Phương pháp mã hoá và trích chọn đặc trưng*

Trong học máy trích chọn đặc trưng và mã hoá đóng vai trò quan trọng trong việc cải thiện hiệu suất mô hình. Với dữ liệu protein có một số phương pháp mã hoá sau:

#### *1.3.2.1 Phương pháp trích chọn đặc trưng dựa trên chuỗi*

Phương pháp trích chọn đặc trưng dựa trên chuỗi (Sequence-Based Features) thường được trích xuất thủ công nhờ một số Tool được cung cấp sẵn như iFeature [18] hay iLearn [19]. Một số đặc trưng trích xuất theo phương pháp này như:

**Đặc trưng CKSAAP:**

Đặc trưng CKSAAP phản ánh các mối tương tác ngắn hạn giữa các cặp axit amin và đã được sử dụng rộng rãi trong lĩnh vực tin sinh học.

Phương pháp này xem xét tần suất xuất hiện của các cặp axit amin cách nhau k (k= 0, 1, 2, 3, 4, 5) vị trí trong chuỗi protein. Ví dụ, khi k = 0, ta xét đến các cặp axit amin liền kề nhau như: AA, AC, AD, ..., YY. Với mỗi giá trị k, véc tơ đặc trưng sẽ được tính toán theo công thức sau:

$$\left(\frac{N_{AA}}{N_{\text{total}}}, \frac{N_{AC}}{N_{\text{total}}}, \frac{N_{AD}}{N_{\text{total}}}, \ldots, \frac{N_{YY}}{N_{\text{total}}}\right) \tag{1.1}$$

Trong đó:

- $N_{\text{total}} = L - k - 1$, với L là độ dài đoạn protein.

- $N_{AC}$ là số lần xuất hiện của cặp axit amin A và C cách nhau k vị trí trong đoạn protein.

Đặc trưng CKSAAP giúp mô hình học được thông tin về mối liên kết ngắn hạn giữa các axit amin trong chuỗi, từ đó cải thiện hiệu quả dự đoán các đặc điểm sinh học quan trọng như vị trí sửa đổi sau dịch mã.

**Đặc trưng AAindex:**

Đặc trưng AAindex được xây dựng dựa trên các thuộc tính sinh hóa và lý hóa cơ bản của axit amin, được trích xuất từ cơ sở dữ liệu AAindex – nơi tổng hợp các chỉ số mô tả đặc điểm sinh học của từng loại axit amin.

Trong nghiên cứu này, mỗi đoạn protein sẽ được mã hóa bằng các giá trị số đại diện cho các đặc tính vật lý và hóa học của từng axit amin tại mỗi vị trí. Cụ thể, mỗi axit amin trong đoạn được biểu diễn bằng 12 giá trị, tương ứng với 12 đặc tính sau: Điện tích ròng (Net charge), tần suất chuẩn hóa hình thành alpha-helix, xu hướng tạo alpha-helix, tỷ lệ thành phần axit amin trong protein nội bào, tỷ lệ thành phần axit amin trong màng protein xuyên màng nhiều vùng, thể tích của phân tử nước kết tinh, giá trị thông tin liên quan đến mức độ tiếp xúc với dung môi, năng lượng chuyển tiếp trong môi trường hữu cơ/nước, tỷ lệ thành phần axit amin trong protein màng, Entropy hình thành, xu hướng cấu trúc dạng beta-strand, năng lượng phân bố tối ưu tương đối

**Đặc trưng BINARY:** Mỗi axit amin được biểu diễn bằng một véc tơ 20 chiều. Sơ đồ mã hóa này để mã hóa cho các chuỗi peptit có độ dài bằng nhau. Ví dụ sự mã hoá của axit amin với đặc trưng BINARY:

A (10000000000000000000); C (01000000000000000000)

**Đặc trưng AAC:** Biểu diễn tần số xuất hiện của các axit amin trong chuỗi peptit. Các tần số của tất cả 20 axit amin ACDEFGHIKLMNPQRSTVWY được tính như sau:

$$f(t) = \frac{N(t)}{N}, \quad t \in \{A, C, D, \ldots, Y\} \tag{1.2}$$

Trong đó $N(t)$ là tần suất xuất hiện của axit amin $t$, $N$ là độ dài của chuỗi peptit hoặc protein.

**Đặc trưng CTDC:** Các amino acid được chia thành ba phân nhóm: phân cực, trung tính và kị nước. Bộ mô tả thành phần gồm ba giá trị: thành phần tổng thể (phần trăm của dư lượng phân cực, trung tính và kị nước của protein). CTDC (Composition of

Tripeptide Distribution of Charge) có thể được tính như sau:

$$CTDC = \frac{N_{R_i}}{L}, \quad R_i \in \{\text{Tích điện dương, Trung tính, Tích điện âm}\} \tag{1.3}$$

Trong đó:

**Tích điện dương** = {K, R},

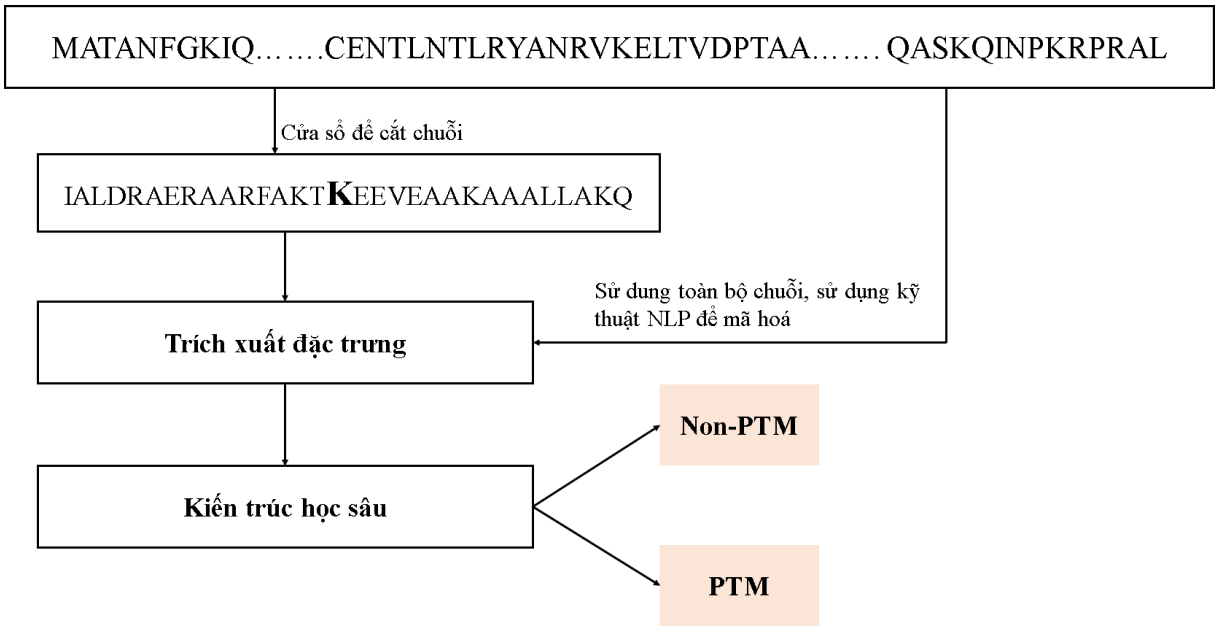
**Trung tính** = {N, A C, G, Q, H, L, I, F, M, P, T, S, W, Y, V},

**Tích điện âm** = {D, E}.

*1.3.2.2 Phương pháp mã hoá và trích chọn đặc trưng dựa trên kỹ thuật xử lý ngôn ngữ tự nhiên*

NLP là một hướng nghiên cứu trong trí tuệ nhân tạo, cho phép máy tính phân tích và hiểu ngôn ngữ của con người ở cả dạng văn bản và lời nói.

Văn bản trong bài toán dự đoán vị trí PTM là các chuỗi protein dạng biểu diễn bậc 1. Các mô hình NLP phổ biến trong dự đoán vị trí PTM [17, 58] như One-hot, Word2Vec, FastText, BERT, và ELMo chủ yếu được sử dụng để trích xuất đặc trưng, tạo ra véc tơ đặc trưng (Hình 1.14). Tuy nhiên, cách tiếp cận này vẫn còn hạn chế do chỉ dừng lại ở bước biểu diễn dữ liệu đặc trưng, chưa khai thác được khả năng tự học từ dữ liệu thô của các mô hình học sâu và phát hiện các đặc trưng bậc cao.

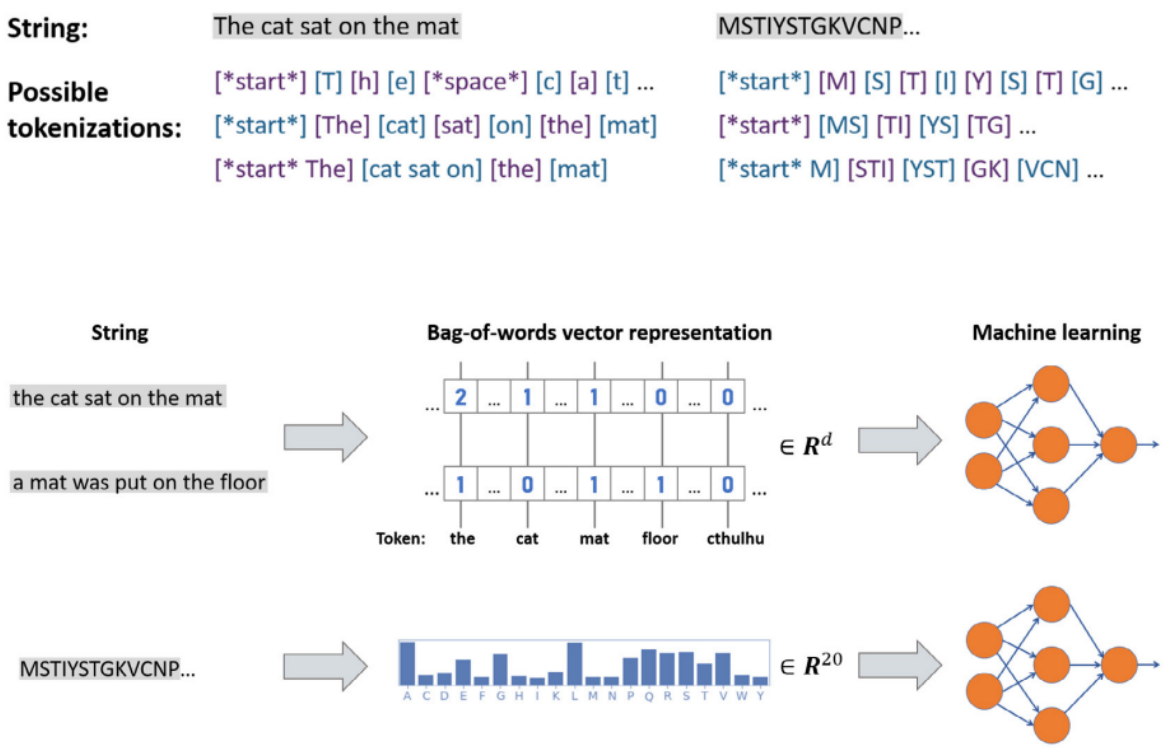



**Hình 1.14 Trích xuất đặc trưng protein tạo ra véc tơ số biểu diễn protein theo kỹ thuật NLP [84]**

Giống như ngôn ngữ tự nhiên của con người, chuỗi protein [77] được biểu diễn

dưới dạng chuỗi ký tự (Hình 1.15). Bảng chữ cái của protein bao gồm 20 axit amin phổ biến (loại trừ các axit amin không thông dụng và hiếm). Tương tự như ngôn ngữ tự nhiên, các protein tiến hóa tự nhiên thường được tạo thành từ các mô đun với những biến thể nhỏ, có thể được sắp xếp và kết hợp theo một cấu trúc phân cấp. Theo phép tương tự này, các mô típ (motifs) và miền (domains) protein phổ biến, vốn là các đơn vị chức năng cơ bản của protein, có thể được xem như các từ, cụm từ và câu trong ngôn ngữ con người.

Một đặc điểm quan trọng khác mà protein và ngôn ngữ tự nhiên có chung là tính hoàn chỉnh thông tin. Mặc dù protein không chỉ đơn thuần là một chuỗi axit amin, nó còn có các cấu trúc không gian ba chiều (cấu trúc bậc 2, bậc 3 và bậc 4) với cấu trúc và chức năng xác định, nhưng tất cả những đặc điểm này đều được quy định bởi trình tự axit amin. Mặc dù cấu trúc và chức năng của protein có thể thay đổi theo ngữ cảnh (trạng thái tế bào, các phân tử khác và các sửa đổi sau dịch mã), nhưng chúng vẫn được xác định bởi trình tự axit amin. Như vậy, theo quan điểm lý thuyết thông tin, toàn bộ thông tin về protein (chẳng hạn như cấu trúc của nó) đều nằm trong trình tự axit amin.




**Hình 1.15 Phương pháp biểu diễn NLP cho ngôn ngữ protein.Văn bản và protein sử dụng bảng chữ cái và được xử lý bằng các kỹ thuật NLP để nghiên cứu các thuộc tính cục bộ và toàn cục, bước tiền xử lý phổ biến trong NLP là mã hóa chúng thành các mã thông báo riêng biệt, là các đơn vị thông tin, biểu diễn túi từ đôi khi được sử dụng để đếm các mã thông báo duy nhất trong văn bản, biến đổi văn bản đầu vào thành một véc tơ có kích thước cố định. Sau đó, các biểu diễn véc tơ này có thể được phân tích thông qua bất kỳ thuật toán học máy nào [84].**

Với những điểm tương đồng này, việc áp dụng kỹ thuật NLP cho chuỗi protein là một hướng tiếp cận hợp lý. Mặc dù thuật ngữ NLP thường đề cập đến ngôn ngữ tự nhiên của con người, nhưng các phương pháp tính toán tương tự cũng có thể được sử dụng để nghiên cứu các ngôn ngữ phi tự nhiên như mã lập trình. Trong những thập kỷ qua, nhiều thuật toán thống kê và học máy từ lĩnh vực NLP đã được chuyển giao sang lĩnh vực tin sinh học. Tuy nhiên, phép so sánh giữa protein và ngôn ngữ con người có những giới hạn nhất định, chúng ta có thể đọc và hiểu ngôn ngữ con người, trong khi protein, mặc dù mang thông tin sinh học quan trọng, không phải là một hệ thống ngôn ngữ theo nghĩa hiểu biết trực tiếp của con người. Hầu hết các ngôn ngữ của con người đều bao gồm dấu câu và từ ngắt là thống nhất, với các cấu trúc có thể tách biệt rõ ràng như từ, câu và đoạn văn. Với protein, không phải lúc nào cũng biết liệu một chuỗi axit amin có phải là một phần của một đơn vị chức năng hay không. Không có sự tương tự rõ ràng giữa các khối xây dựng của ngôn ngữ và các khối xây dựng của protein. Ví dụ, việc coi các miền protein tương đương với các từ thường gây hiểu lầm. Hơn nữa, các đơn vị chức năng

protein thường chồng chéo lên nhau. Do đó, trong khi các ngôn ngữ tự nhiên có vốn từ vựng được xác định rõ ràng (với  triệu từ trong tiếng Anh), thì protein lại không có vốn từ vựng rõ ràng.

Khi tiếp cận bài toán xử lý chuỗi protein bằng kỹ thuật NLP, việc lựa chọn phương pháp mã hóa là một bước then chốt, quyết định khả năng mô hình học được các đặc trưng quan trọng. Các phương pháp mã hóa này có thể được phân loại dựa trên cách chúng nắm bắt thông tin và ngữ cảnh của chuỗi.

Loại đầu tiên là Embedding độc lập như One-hot và TF-IDF. Đây là những kỹ thuật đơn giản, dễ triển khai và không đòi hỏi nhiều dữ liệu. Chúng mã hóa mỗi axit amin một cách độc lập, không xét đến mối quan hệ với các axit amin lân cận. Tuy nhiên, chính sự đơn giản này lại là hạn chế lớn nhất, vì chúng không thể nắm bắt được ngữ nghĩa sinh học hay bối cảnh phức tạp của các axit amin trong chuỗi, dẫn đến việc bỏ sót thông tin quan trọng. Thêm vào đó, cách biểu diễn này thường tạo ra các vector đặc trưng có kích thước rất lớn, gây khó khăn cho việc xử lý.

Để khắc phục hạn chế trên, các phương pháp Embedding theo ngữ cảnh như Word2Vec, GloVe và FastText ra đời. Các mô hình này học cách biểu diễn một axit amin dựa trên các axit amin xung quanh nó, giúp vector đặc trưng mang theo thông tin ngữ nghĩa cục bộ. Điều này cho phép mô hình hiểu được vai trò của một axit amin trong một đoạn chuỗi cụ thể, từ đó cải thiện đáng kể hiệu suất so với các phương pháp độc lập. Mặc dù vậy, chúng vẫn chưa đủ mạnh mẽ để nắm bắt được toàn bộ ngữ cảnh dài hạn, tức là mối quan hệ giữa các axit amin ở xa nhau trong chuỗi protein.

Thế hệ tiếp theo của các phương pháp mã hóa là Mô hình ngôn ngữ lớn (LLMs) như BERT và GPT-2. Dựa trên kiến trúc Transformer, những mô hình này vượt trội trong việc học các biểu diễn sâu và toàn diện. Với cơ chế tự chú ý (Self-Attention), chúng có thể đồng thời xem xét tất cả các vị trí trong chuỗi protein, giúp nắm bắt được các mối quan hệ ngữ cảnh dài hạn phức tạp. Nhờ đó, chúng tạo ra các vector đặc trưng chất lượng cao, biểu diễn được các đặc trưng bậc cao và cải thiện đáng kể hiệu suất của các bài toán dự đoán. Tuy nhiên, ưu điểm này đi kèm với một nhược điểm lớn là chúng đòi hỏi lượng dữ liệu khổng lồ cho việc tiền huấn luyện và yêu cầu tài nguyên tính toán rất lớn, gây khó khăn trong việc triển khai.

Mỗi phương pháp đều có những ưu điểm và hạn chế riêng, việc lựa chọn phương pháp phù hợp phụ thuộc vào mục tiêu nghiên cứu và tài nguyên sẵn có. Phân tích chi tiết hơn về các phương pháp này được thể hiện trong Bảng 1.3.

**Bảng 1.3 So sánh các phương pháp mã hóa trong NLP**

| Phương pháp | Ưu điểm | Hạn chế |
|---|---|---|
| Embedding độc lập (One-hot, TF-IDF) | Đơn giản, dễ triển khai, không cần nhiều dữ liệu | Không giữ được ngữ nghĩa, kích thước véc tơ lớn |
| Embedding theo ngữ cảnh (Word2Vec, GloVe, FastText) | Biểu diễn ngữ nghĩa tốt hơn, tận dụng bối cảnh cục bộ | Không nắm bắt toàn bộ ngữ cảnh dài hạn, cần nhiều dữ liệu |
| Mô hình ngôn ngữ lớn (BERT, GPT-2) | Học biểu diễn sâu hơn, giữ được thông tin dài hạn | Cần tài nguyên tính toán lớn |

### *1.3.3 Xây dựng mô hình*

Luận án được thực hiện theo hướng thực nghiệm, tập trung xây dựng, thử nghiệm và đánh giá các mô hình học máy, học sâu để xác định kiến trúc tối ưu cho bài toán dự đoán vị trí PTM. Các mô hình sau được lựa chọn dựa trên khả năng xử lý dữ liệu chuỗi và hiệu quả dự đoán:

- Mô hình học máy truyền thống: SVM, XGBoost, RF.

- Mô hình học sâu: CNN1D, LSTM, Bi-LSTM và các biến thể kết hợp theo hướng mô hình học sâu lai, học chắt lọc tri thức.

### *1.3.4 Lựa chọn các tham số trong quá trình huấn luyện mô hình dự đoán*

Huấn luyện mô hình học sâu là phần cốt lõi nhằm nâng cao hiệu quả học và khả năng tổng quát hóa. Để đạt hiệu suất tối ưu, cần điều chỉnh một số siêu tham số quan trọng [73], bao gồm:

(1) Tốc độ học (learning rate):

Trong học máy và thống kê, tốc độ học là một tham số điều chỉnh trong thuật toán tối ưu hoá xác định kích thước bước tại mỗi lần lặp trong khi di chuyển về phía giá trị tối thiểu của hàm mất mát. Tốc độ học là một trong những tham số quan trọng nhất, quyết định mức độ điều chỉnh trọng số trong mỗi bước lan truyền ngược (backpropagation).

Khi thiết lập tốc độ học, có sự đánh đổi giữa tốc độ hội tụ và vượt ngưỡng. Trong khi hướng đi xuống thường được xác định từ độ dốc của hàm mất mát, tốc độ học xác định bước đi lớn như thế nào theo hướng đó. Tốc độ học quá cao sẽ khiến quá trình học vượt qua giá trị tối thiểu nhưng tốc độ học quá thấp sẽ mất quá nhiều thời gian để hội tụ hoặc bị kẹt ở giá trị tối thiểu cục bộ không mong muốn.

(2) Kích thước lô (batch size):

Kích thước lô là việc xác định số lượng mẫu được sử dụng để tính toán gradient và cập nhật trọng số trong mỗi vòng lặp huấn luyện. Kích thước lô nhỏ giúp mô hình học nhanh hơn nhưng dễ bị nhiễu, trong khi kích thước lô lớn giúp gradient ổn định hơn nhưng yêu cầu tài nguyên tính toán lớn hơn.

(3) Số lớp (number of layers) và số nơ-ron (number of neurons): Các kiến trúc mạng sâu với nhiều lớp hoặc nhiều nơ-ron có thể học được các đặc trưng phức tạp hơn, tuy nhiên nếu thiết kế không hợp lý dễ dẫn đến hiện tượng overfitting. Việc lựa chọn cấu trúc phù hợp cần dựa trên đặc thù của bài toán và đặc điểm dữ liệu.

(4) Hàm kích hoạt (activation function): Lựa chọn hàm kích hoạt như ReLU, sigmoid, softmax,... ảnh hưởng trực tiếp đến khả năng học phi tuyến và tốc độ hội tụ của mô hình.

(5) Số epoch: Là số lần toàn bộ tập dữ liệu huấn luyện được đưa qua mạng. Việc xác định số epoch phù hợp giúp mô hình hội tụ đúng mức, tránh hiện tượng underfitting hoặc overfitting.

(6) Kỹ thuật như dừng sớm (early stopping): giám sát một tiêu chí (thường là độ chính xác hoặc hàm mất mát trên tập kiểm tra) và dừng huấn luyện nếu tiêu chí này không được cải thiện sau một số epoch liên tiếp nhất định.

(7) Việc bỏ một số nút của mạng (dropout) cũng thường được tích hợp để cải thiện khả năng khái quát hóa và tránh quá khớp dữ liệu huấn luyện.

(8). Tối ưu hoá Adam

Adam viết tắt của Adaptive Moment Estimation, là một thuật toán tối ưu hóa phổ biến được sử dụng trong học máy và đặc biệt là trong học sâu. Adam kết hợp các ý tưởng chính từ hai kỹ thuật tối ưu mạnh mẽ khác là momentum và RMSprop. Thuật toán này được gọi là "thích ứng" vì nó điều chỉnh tỷ lệ học cho mỗi tham số. Tối ưu hoá Adam giống như một trợ lý thông minh trong việc huấn luyện mạng nơ ron. Nó giúp điều chỉnh các thiết lập (tham số) của mạng để mạng làm việc hiệu quả hơn, như nhận dạng hình ảnh hoặc hiểu văn bản.

### *1.3.5 Đánh giá mô hình*

Hiệu quả dự đoán vị trí PTM được NCS đánh giá bằng hai phương pháp: *(i)* kiểm thử chéo k-fold để ước lượng hiệu suất tổng thể và tính ổn định của mô hình, và *(ii)* kiểm thử độc lập để đánh giá khả năng khái quát trên dữ liệu chưa từng thấy.

Trong các bài toán phân lớp nhị phân, đặc biệt khi phân bố giữa hai lớp không bị mất cân bằng nghiêm trọng, việc lựa chọn các chỉ số đánh giá phù hợp là yếu tố

quan trọng nhằm phản ánh chính xác hiệu suất của mô hình. Các chỉ số thường được sử dụng bao gồm độ chính xác (Accuracy - ACC), độ nhạy (Sensitivity - SEN), độ đặc hiệu (Specificity - SPE), hệ số tương quan Matthews (Matthews Correlation Coefficient - MCC), và diện tích dưới đường cong ROC (Area Under Curve - AUC). Những chỉ số này cung cấp góc nhìn toàn diện về khả năng phân biệt, mức độ chính xác, và sự nhất quán trong dự đoán của mô hình. Để đánh giá toàn diện hiệu quả của mô hình, AUC cần được xem xét đồng thời với các chỉ số khác như ACC, SEN, SPE, MCC.

$$\begin{aligned}
\text{SEN} &= \frac{TP}{TP+FN} \\
\text{SPE} &= \frac{TN}{TN+FP} \\
\text{ACC} &= \frac{TP+TN}{TP+FN+TN+FP} \\
\text{MCC} &= \frac{(TP\times TN)-(FN\times FP)}{\sqrt{(TP+FN)(TN+FP)(TP+FP)(TN+FN)}} \\
\text{AUC} &= \frac{TPR-FPR+1}{2}
\end{aligned} \tag{1.4}$$

$$\begin{aligned}
TPR &= \frac{TP}{TP+FN}; \quad FPR = \frac{FP}{FP+TN} \\
TNR &= \frac{TN}{TN+FP}; \quad FNR = \frac{FN}{FN+TP}
\end{aligned} \tag{1.5}$$

**Trong đó:**

- SEN sử dụng để tính toán khả năng mô hình phát hiện đúng các trường hợp dương tính.

-SPE phản ánh khả năng mô hình nhận diện chính xác các trường hợp âm tính. Sự kết hợp giữa SEN và SPE giúp đánh giá mức độ cân bằng trong việc phát hiện cả hai lớp, đặc biệt hữu ích trong các bối cảnh sinh học nơi cả hai loại lỗi đều có thể ảnh hưởng nghiêm trọng đến kết luận nghiên cứu.

- ACC là tỷ lệ giữa số lượng mẫu được phân loại đúng và tổng số mẫu trong tập kiểm tra. Trong trường hợp dữ liệu cân bằng, ACC có thể phản ánh tương đối chính xác hiệu suất tổng thể của mô hình. Tuy nhiên, chỉ số này không cung cấp thông tin về các loại lỗi khác nhau (như dự đoán dương tính sai hay âm tính sai), do đó cần được sử dụng kết hợp với các chỉ số khác để đảm bảo đánh giá khách quan.

- MCC là một thước đo thống kê mạnh, phản ánh mối tương quan giữa nhãn thực tế và nhãn dự đoán. MCC tích hợp đồng thời cả bốn thành phần của ma trận nhầm lẫn (TP, TN, FP, FN), mang lại một chỉ số duy nhất nhưng đầy đủ về hiệu quả của mô hình.

Với giá trị nằm trong khoảng từ 0 đến 1, MCC = 1 biểu thị mô hình hoàn hảo. MCC được xem là một trong những chỉ số đánh giá đáng tin cậy nhất cho phân lớp nhị phân, đặc biệt trong các nghiên cứu học thuật.

- TP (True Positive - Dương tính thật): Là số trường hợp mà mô hình dự đoán là dương tính và thực tế cũng đúng là dương tính.

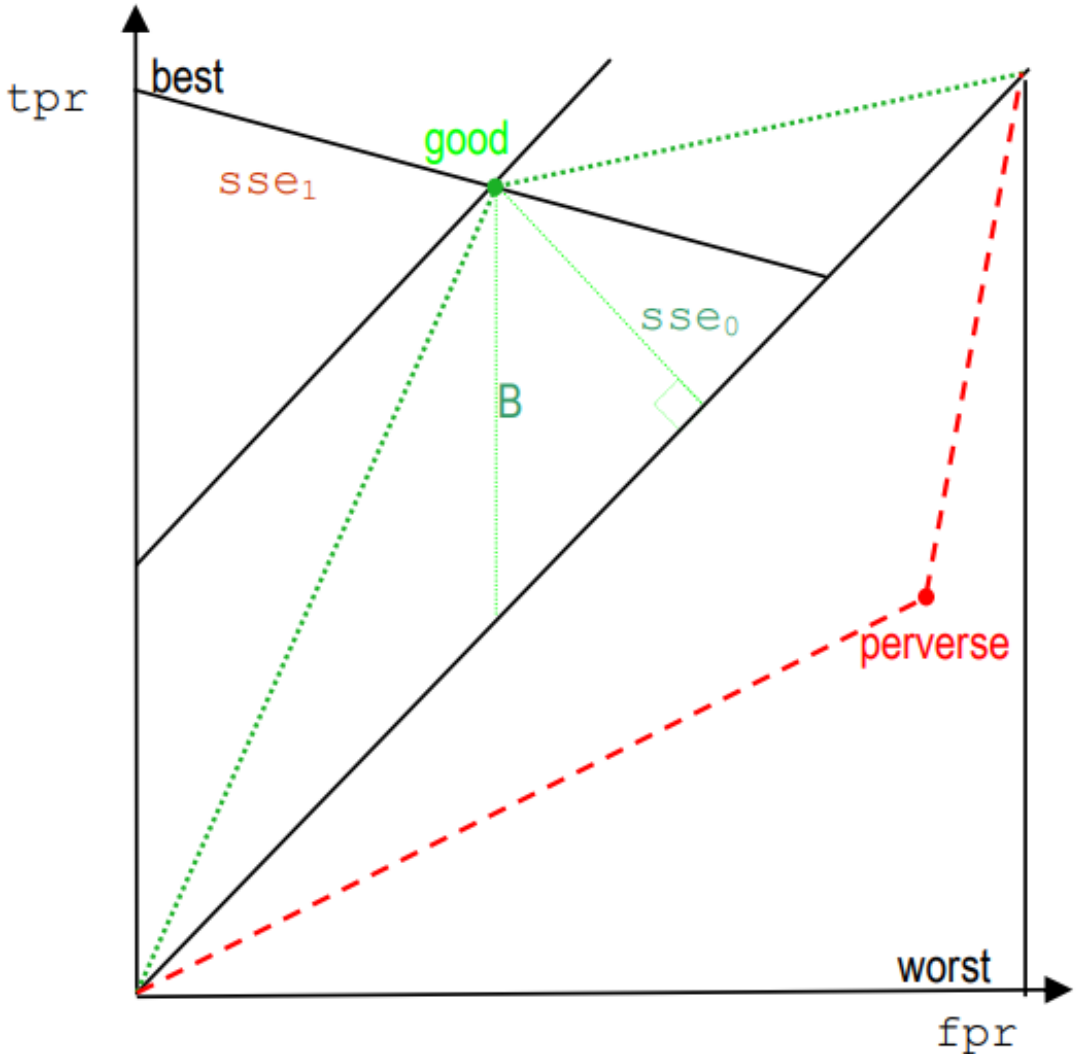
- FP (False Positive - Dương tính giả): Là số trường hợp mà mô hình dự đoán là dương tính, nhưng thực tế lại là âm tính.

- TN (True Negative – Âm tính thật): Là số trường hợp mà mô hình dự đoán là âm tính và thực tế cũng đúng là âm tính.

- FN (False Negative – Âm tính giả): Là số trường hợp mà mô hình dự đoán là âm tính, nhưng thực tế lại là dương tính.

- TPR, FPR, TNR, FNR: lần lượt đo độ nhạy, tỷ lệ nhầm dương tính, độ đặc hiệu và tỷ lệ bỏ sót dương tính.

- AUC-ROC: diện tích dưới đường cong ROC, thể hiện xác suất mô hình gán điểm cao hơn cho mẫu dương tính so với mẫu âm tính ngẫu nhiên. AUC = 1 biểu thị mô hình hoàn hảo, AUC = 0.5 tương đương ngẫu nhiên.




**Hình 1.16 Minh họa trực quan cho phân tích đường cong ROC. Đường chéo chính giữa biểu đồ đại diện cho hệ thống dự đoán ngẫu nhiên. Các điểm nằm phía trên đường chéo này thể hiện mô hình có hiệu suất dự đoán tốt hơn ngẫu nhiên, trong khi các điểm phía dưới thể hiện hiệu suất tệ hơn cả ngẫu nhiên. Với một mô hình phân loại tốt (đường nét chấm màu xanh lá), AUC chính là diện tích phía dưới đường cong tạo bởi mô hình này và trục hoành chạy từ 0 đến 1. Mô hình tệ (đường nét đứt màu đỏ) nhãn đầu ra đã bị gán sai [85].**

### *1.3.6 Lựa chọn mô hình*

Quá trình lựa chọn mô hình tốt trong dự đoán vị trí PTM đòi hỏi một cách tiếp cận có hệ thống nhằm đảm bảo độ chính xác cao, tính tổng quát hoá tốt và hiệu suất tính toán hợp lý. Trong số rất nhiều mô hình đã thực nghiệm, mô hình tốt được lựa chọn dựa trên các tiêu chí sau:

Hiệu suất trên tập dữ liệu huấn luyện và dữ liệu kiểm tra độc lập ACC, SEN, SPE, MCC là các chỉ số quan trọng để đánh giá chất lượng mô hình.

Giá trị AUC-ROC cũng được xem xét để đánh giá khả năng phân biệt giữa các mẫu dương tính và âm tính.

**Khả năng tổng quát hóa**

Mô hình không chỉ đạt kết quả cao trên tập huấn luyện (kiểm thử chéo) mà còn phải thể hiện tốt trên tập kiểm tra độc lập, tránh hiện tượng quá khớp.

**Hiệu suất tính toán**

Mô hình cần có thời gian huấn luyện và suy luận hợp lý, không yêu cầu tài nguyên tính toán quá lớn so với độ cải thiện hiệu suất mang lại.

Đối với các mô hình quá phức tạp, nếu độ chính xác chỉ cải thiện không đáng kể so với các mô hình đơn giản hơn, thì mô hình đơn giản hơn sẽ được ưu tiên.

**Khả năng mở rộng và ứng dụng**

Mô hình được lựa chọn cần có khả năng triển khai trong thực tế, có thể mở rộng cho nhiều loại PTM khác nhau hoặc tích hợp vào các công cụ sinh học hiện có.

Dựa trên các tiêu chí trên, mô hình tối ưu được xác định dựa trên sự cân bằng giữa độ chính xác, khả năng tổng quát hóa, tính giải thích được và hiệu suất tính toán.

### *1.3.7 Các yêu cầu hệ thống và môi trường cài đặt*

Hệ thống và môi trường cần thiết để chạy mô hình bao gồm:

**Ngôn ngữ lập trình:** Python

**Môi trường thực thi:**

- Google Colab

- Máy chủ GPU hỗ trợ các dòng card như T4, L4, A100

**Các thư viện cần thiết:**

- Numpy để xử lý dữ liệu số

- Pandas để thao tác với dữ liệu dạng bảng

- Tensorflow để xây dựng và huấn luyện mô hình học sâu

- Sklearn (scikit-learn) để tiền xử lý dữ liệu và đánh giá mô hình

- Matplotlib để trực quan hóa dữ liệu và kết quả mô hình

**Dữ liệu đầu vào:** File dữ liệu ở định dạng .csv

## 1.4 Thách thức của các mô hình dự đoán vị trí PTM

Mặc dù các mô hình học máy và học sâu đã đạt được nhiều tiến bộ trong việc dự đoán vị trí PTM, tuy nhiên, vẫn tồn tại một số thách thức lớn ảnh hưởng đến độ chính xác, khả năng tổng quát và khả năng ứng dụng thực tiễn của các mô hình. Những thách thức này bao gồm:

**Dữ liệu không cân bằng và hạn chế về số lượng:**

Đối với hầu hết các loại PTM, số lượng các vị trí được xác nhận thực nghiệm (mẫu dữ liệu dương tính) rất hạn chế so với số lượng vị trí không được gắn nhãn (dữ liệu âm tính). Tỷ lệ mất cân bằng nghiêm trọng này gây khó khăn trong huấn luyện mô hình, làm tăng nguy cơ thiên lệch (bias) và giảm khả năng phát hiện chính xác các vị trí PTM thực sự.

Ngoài ra, một số PTM ít phổ biến hơn vẫn chưa có đủ dữ liệu chất lượng cao để huấn luyện mô hình học sâu phức tạp, khiến việc xây dựng mô hình tổng quát gặp nhiều trở ngại.

**Đặc trưng không đồng nhất và khó xác định:**

Các đặc trưng sinh học của vị trí PTM thường không rõ ràng và phân tán. Việc lựa chọn hoặc thiết kế đặc trưng thủ công như AAindex, BLOSUM, CKSAAP,... đòi hỏi nhiều kiến thức chuyên ngành và có thể không khai thác đầy đủ các mối quan hệ phức tạp trong protein.

Việc sử dụng quá nhiều loại đặc trưng từ nhiều nguồn khác nhau có thể làm tăng độ phức tạp và nguy cơ dư khớp, đồng thời giảm tính mở rộng của mô hình.

**Khó tích hợp thông tin ngữ cảnh và phụ thuộc dài hạn:**

Các PTM không chỉ phụ thuộc vào vài axit amin lân cận mà còn có thể bị ảnh hưởng bởi các yếu tố ngữ cảnh dài hạn trong chuỗi protein, cấu trúc bậc cao hoặc tín hiệu sinh học khác. Các mô hình đơn giản dựa trên cửa sổ trượt hoặc chỉ dùng CNN thường không nắm bắt được đầy đủ các phụ thuộc phức tạp này.

**Thiếu khả năng mở rộng và thích ứng với dữ liệu mới:**

Các mô hình được huấn luyện trên một loài hoặc tập dữ liệu cụ thể thường không tổng quát tốt sang các loài khác, hoặc các tập dữ liệu có phân bố khác biệt.

Ngoài ra, hầu hết các mô hình hiện nay vẫn chưa tận dụng hiệu quả các mô hình ngôn ngữ lớn chuyên biệt cho sinh học để thích ứng với dữ liệu mới theo cách học chuyển giao.

Hiệu suất tính toán và khả năng triển khai thực tiễn:

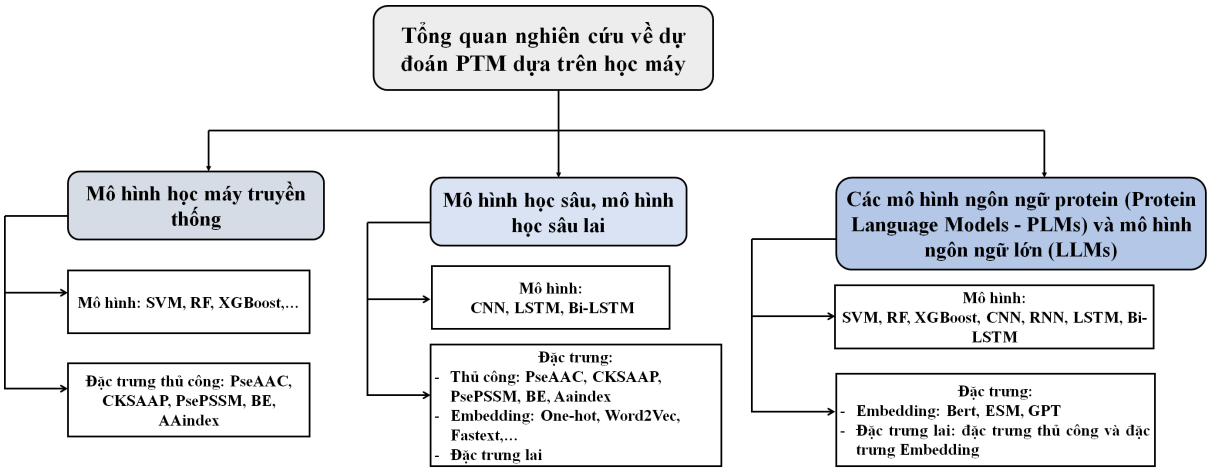
Các mô hình học sâu lớn, sử dụng nhiều đặc trưng đầu vào hoặc embedding từ

nhiều mô hình ngôn ngữ tiền huấn luyện (như mô hình Bert, GPT), thường đòi hỏi tài nguyên tính toán lớn, khó triển khai trong môi trường hạn chế hoặc các ứng dụng sinh học quy mô lớn.

Do đó, cần có hướng tiếp cận vừa đảm bảo hiệu quả dự đoán, vừa giảm tải tính toán, chẳng hạn như các kiến trúc lai nhẹ, hoặc sử dụng học chắt lọc tri thức để xây dựng mô hình nhẹ hơn nhưng vẫn duy trì hiệu năng.

## 1.5 Tổng quan nghiên cứu về dự đoán PTM và các phương pháp tiên tiến hiện nay

Trong hai thập kỷ gần đây, các phương pháp tính toán dự đoán vị trí sửa đổi sau dịch mã (PTM) đã phát triển mạnh mẽ và trở thành hướng nghiên cứu quan trọng song hành cùng các phương pháp thực nghiệm. Nhiều công trình khoa học đã khai thác các kỹ thuật từ học máy truyền thống, học sâu, cho đến xử lý ngôn ngữ tự nhiên và mô hình ngôn ngữ protein. Tuy nhiên, mỗi hướng tiếp cận đều có ưu điểm và hạn chế riêng, đặt nền tảng cho việc tìm kiếm các giải pháp tối ưu hơn. Hình 1.17 là sơ đồ bức tranh tổng thể về ba hướng nghiên cứu chính trong dự đoán PTM hiện nay.




**Hình 1.17 Sơ đồ tổng quan các hướng tiếp cận trong dự đoán vị trí PTM**

(i) **Các mô hình học máy truyền thống:** Trong dự đoán vị trí PTM, các mô hình học máy truyền thống đã đóng vai trò nền tảng nhờ khả năng xử lý tốt các bộ dữ liệu nhỏ, dễ huấn luyện và cho kết quả dễ diễn giải. Các thuật toán phổ biến như SVM, RF, k-NN, XGBoost hay LightGBM đều tận dụng triệt để các đặc trưng chuỗi protein được thiết kế thủ công, bao gồm PseAAC, CKSAAP, BE, PsePSSM hay AAIndex. Mỗi loại đặc trưng cung cấp một góc nhìn riêng: PseAAC tổng hợp thông tin về thành phần amino acid và tính chất vật lý-hóa học, CKSAAP và BE ghi nhận mối liên hệ giữa các cặp amino acid hoặc cấu trúc chuỗi, trong khi PsePSSM và AAIndex phản ánh thông tin tiến hóa và sinh học, từ đó nâng cao khả năng phân biệt các vị trí PTM.

Một số mô hình tiêu biểu minh họa cách các thuật toán và đặc trưng được kết hợp hiệu quả (Bảng 1.4). Mô hình *pSumo-CD* sử dụng PseAAC và thuật toán Covariance Discriminant, đạt hiệu quả cao trong dự đoán SUMOylation nhờ khả năng trích xuất thông tin có ý nghĩa sinh học từ đặc trưng thủ công. *iAcet-Sumo* kết hợp one-hot encoding với SVM, chứng minh rằng ngay cả các biểu diễn đơn giản cũng có thể đem lại hiệu quả đáng kể khi dữ liệu không quá lớn. Trong khi đó, các công cụ như [27, 62] khai thác đặc trưng lai CKSAAP, EAAC, 188D hoặc BE_PAA_EBGW kết hợp với PsePSSM, giúp mở rộng khả năng biểu diễn chuỗi và cải thiện hiệu suất dự đoán. Các mô hình học máy tổ hợp như SUMOgo (RF) hay [64] (AdaBoost) nhấn mạnh sức mạnh của phương pháp tập hợp, vừa tăng độ chính xác vừa nâng cao tính ổn định của dự đoán. Mới đây, O-GlyThr [103] (2023) sử dụng 7 đặc trưng thủ công kết hợp RF để dự đoán O-linked threonine glycosylation, minh chứng cho việc lựa chọn bộ đặc trưng phù hợp có thể tối ưu hóa hiệu suất dự đoán cho từng loại PTM cụ thể.

Nhìn chung, các mô hình học máy truyền thống tỏ ra rất hữu ích khi làm việc với các bộ dữ liệu hạn chế, đặc biệt là nhờ khả năng giải thích kết quả và triển khai nhanh. Tuy nhiên, nhược điểm lớn của chúng là phụ thuộc nhiều vào đặc trưng thủ công, khiến khả năng tổng quát hóa khi áp dụng cho các loại PTM hoặc loài mới còn hạn chế. Ngoài ra, khi đối mặt với chuỗi protein dài hoặc các mối quan hệ PTM phức tạp, hiệu quả dự đoán của các mô hình này có xu hướng đạt giới hạn, tạo tiền đề cho việc sử dụng học sâu hoặc các mô hình ngôn ngữ protein nhằm trích xuất đặc trưng tự động và khai thác thông tin ngữ cảnh dài hơn.

**Bảng 1.4 Mô hình học máy trong dự đoán các PTM**

| TT | Công cụ | Loại PTM | Thuật toán | Đặc trưng | Năm |
|---|---|---|---|---|---|
| 1 | pSumo-CD [49] | SUMOylation | Covariance Discriminant | PseAAC | 2016 |
| 2 | [64] | SUMOylation | AdaBoost | – | 2017 |
| 3 | iAcet-Sumo [124] | SUMOylation | SVM | One-Hot | 2018 |
| 4 | SUMOgo [12] | SUMOylation | RF | – | 2018 |
| 5 | [27] | Glutarylation | AdaBoost | CKSAAP, 188D, EAAC | 2020 |
| 6 | [62] | Crotonylation | LightGBM | BE, PAA, EBGW, KNN, PsePSSM | 2020 |
| 7 | [61] | Ubiquitination | XGBoost | PseAAC, CKSAAP, ANBPB, AAindex, EBGW, BLOSUM62, PsePSSM | 2021 |
| 8 | O-GlyThr [103] | O-Linked Thr. Glycosylation | RF | 7 handcrafted features | 2023 |
| 9 | HOTGpred [81] | O-Linked Thr. Glycosylation | XGBoost | 25 features (14 PLM embeddings + 11 conventional descriptors) | 2024 |

(ii) **Các mô hình học sâu (Deep Learning), học sâu lai:** Trong những năm gần đây, học sâu đã trở thành hướng tiếp cận chủ đạo trong dự đoán các vị trí PTM, nhờ khả năng tự động trích xuất đặc trưng và mô hình hóa các mối quan hệ phi tuyến phức tạp trong chuỗi protein (Bảng 1.5). Phần lớn các mô hình hiện nay sử dụng kiến trúc mạng tích chập một chiều (CNN1D), đôi khi kết hợp với các mạng hồi tiếp như LSTM hoặc Bi-LSTM nhằm khai thác thông tin theo thứ tự chuỗi, từ đó nâng cao khả năng nhận

dạng các vị trí sửa đổi sinh học. Các mô hình tiêu biểu như *DeepUbi* [30] (2019) và

**Bảng 1.5 Mô hình học sâu trong dự đoán sửa đổi sau dịch mã (PTM)**

| TT | Công cụ | Loại PTM | Thuật toán | Đặc trưng | Năm |
|---|---|---|---|---|---|
| 1 | DeepUbi [30] | Ubiquitination | CNN1D | One-hot | 2019 |
| 2 | DeepSuccinylSite [106] | Succinylation | CNN1D | One-hot | 2020 |
| 3 | Arabidosis [117] | Ubiquitination | CNN1D | AAindex | 2021 |
| 4 | LSTMCNNsucc [45] | Succinylation | CNN1D_Bi-LSTM | Embedding | 2021 |
| 5 | HubipPred [118] | human Ubiquitination | CNN1D_RNN | Binary and physicochemical properties | 2021 |
| 6 | DeepSucc [125] | Succinylation | CNN1D_LSTM | CKSAAP, ACF, BLOSUM62, one-hot | 2022 |
| 7 | Deep-KsuccSite [60] | Succinylation | CNN1D_LSTM | Cascading characteristics | 2022 |
| 8 | ResSUMO [129] | SUMOylation | CNN1D | BLOSUM62 | 2022 |
| 9 | PSSM-SUMO [53] | SUMOylation | DNN | PSSM | 2024 |

*DeepSuccinylSite* [106] (2020) sử dụng CNN1D kết hợp với mã hóa one-hot đơn giản để đại diện chuỗi protein, cho thấy rằng ngay cả các biểu diễn cơ bản cũng có thể mang lại hiệu quả dự đoán vượt trội nhờ khả năng học tự động các mẫu đặc trưng từ dữ liệu. Trong khi đó, *Arabidosis Ubiquitination* [117] (2021) kết hợp CNN1D với đặc trưng AAIndex, minh họa cách tích hợp các thông tin sinh học thiết kế thủ công với học sâu để cải thiện khả năng nhận dạng vị trí PTM.

Một số các nghiên cứu khác cố gắng cải thiện hiệu suất của các mô hình học sâu bằng các mô hình học sâu lai. Sự kết hợp giữa CNN và Bi-LSTM, vừa khai thác khả năng học đặc trưng cục bộ từ CNN, vừa nắm bắt ngữ cảnh dài từ Bi-LSTM. Các mô hình học sâu lai tiêu biểu như *DeepSucc* [125] (2022) và *Deep-KsuccSite* [60] (2022) sử dụng mạng CNN1D kết hợp với mạng LTSM/Bi-LSTM và các đặc trưng sinh học được trích

xuất thủ công như CKSAAP, BLOSUM62 và Cascading characteristics để cải thiện dự đoán PTM.

Nhìn chung, các mô hình học sâu và học sâu lai thể hiện ưu thế vượt trội so với học máy truyền thống nhờ khả năng tự động trích xuất đặc trưng, tận dụng ngữ cảnh dài trong chuỗi protein và xử lý các quan hệ phi tuyến phức tạp. Tuy nhiên, nhược điểm là chúng đòi hỏi tập dữ liệu lớn và tài nguyên tính toán đáng kể, đồng thời vẫn có nguy cơ quá khớp nếu dữ liệu huấn luyện hạn chế.

**(iii) Các mô hình ngôn ngữ protein (Protein Language Models - PLMs) và mô hình ngôn ngữ lớn (LLMs):** Chuỗi protein có thể được xem như một dạng "ngôn ngữ sinh học", trong đó mỗi axit amin đóng vai trò là một đơn vị cơ bản và ngữ cảnh xung quanh quyết định chức năng sinh học. Quan niệm này cho phép tận dụng các kỹ thuật xử lý ngôn ngữ tự nhiên để khai thác thông tin tiềm ẩn trong chuỗi protein. Các mô hình ngôn ngữ tiền huấn luyện như BERT [22] hay T5 [89] được sử dụng để tạo các embedding ngữ cảnh, từ đó làm đặc trưng cho các mô hình học máy hoặc học sâu trong dự đoán PTM [55, 79, 82, 101]. Thay vì dựa hoàn toàn vào các encoding như one-hot hay PseAAC, PLMs khai thác embedding ngữ cảnh, cho phép mô hình nhận biết mối quan hệ phi tuyến phức tạp giữa các vị trí axit amin trong chuỗi protein.

Một số nghiên cứu điển hình minh chứng hiệu quả của hướng tiếp cận này bao gồm *LMSuccSite* [83] (2022), *PTMGPT2* [99], *HOTGpred* [81] (2024), *DeepPTM* [101] (2024) và *LMPTMSite* [86] (2024) (Bảng 1.6). Những công trình này cho thấy rằng việc tích hợp embedding từ PLMs với các kiến trúc học sâu không chỉ nâng cao độ chính xác dự đoán mà còn cải thiện khả năng tổng quát hóa trên các tập dữ liệu mới. Cụ thể, embedding từ PLMs giúp mô hình nhận diện các motif sinh học phức tạp mà trước đây khó mô tả bằng các đặc trưng thủ công, đồng thời giảm nhu cầu thiết kế đặc trưng phức tạp và tốn thời gian.

Tuy nhiên, nhược điểm lớn của PLMs và LLMs là yêu cầu tài nguyên tính toán khổng lồ. Việc huấn luyện từ đầu hoặc tinh chỉnh trên các tập dữ liệu nhỏ trở nên khó khả thi, đồng thời đòi hỏi phần cứng mạnh và thời gian tính toán đáng kể. Điều này tạo ra một thách thức đối với các nghiên cứu trong môi trường hạn chế tài nguyên hoặc khi dữ liệu PTM không đủ lớn, khiến các phương pháp này cần được kết hợp khéo léo với học sâu truyền thống hoặc kỹ thuật học chắt lọc tri thức (Knowledge Distillation) để tối ưu hóa hiệu quả và khả năng triển khai thực tiễn.

**Khoảng trống nghiên cứu:**

Mặc dù các phương pháp học máy, học sâu và mô hình ngôn ngữ protein đã đạt được nhiều tiến bộ quan trọng trong dự đoán vị trí PTM, nhưng hiện tại vẫn tồn tại một số hạn chế cần cải thiện. Các khoảng trống nghiên cứu có thể tóm lược như sau:

**Bảng 1.6 Các mô LLM và PLMs trong dự đoán vị trí PTM**

| TT | Công cụ | Loại PTM | Thuật toán | Đặc trưng / Embedding | Năm |
|---|---|---|---|---|---|
| 1 | LMSuccSite [83] | Succinylation | BiLSTM + PLM embeddings | ProtBERT/ESM embeddings + sequence descriptors | 2022 |
| 11 | PTMGPT2 [99] | Multiple PTMs | GPT-2 based transformer | Pre-trained language model embeddings | 2023 |
| 2 | DeepPTM [101] | Multiple PTMs | Hybrid DL (CNN + Transformer) | PLM embeddings + handcrafted features | 2024 |
| 3 | LMPTMSite [86] | Multiple PTMs | Transformer + PLM fine-tuning | Embeddings từ ProtT5, ESM2 (contextualized PLM representation) | 2024 (tháng 11) |

- Phụ thuộc đặc trưng thủ công: hạn chế tính tổng quát khi áp dụng trên PTM hoặc loài mới.

- Yêu cầu dữ liệu và tài nguyên lớn: học sâu và PLMs cần tập dữ liệu khổng lồ và hạ tầng mạnh. Nguy cơ quá khớp: dữ liệu PTM thường nhỏ, mất cân bằng, dễ dẫn đến mô hình kém tổng quát.

- Chưa khai thác kỹ thuật học chắt lọc tri thức (Knowledge Distillation) trong dự đoán PTM: mặc dù kỹ thuật này đã được ứng dụng thành công trong các lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên, nhưng đến nay chưa có công trình nào áp dụng cho dự đoán PTM. Đây là một hướng nghiên cứu tiềm năng, đặc biệt phù hợp trong bối cảnh dữ liệu hạn chế và môi trường tài nguyên giới hạn.

## 1.6 Hướng nghiên cứu trong luận án

Xuất phát từ những khoảng trống trong nghiên cứu dự đoán PTM, nhu cầu phát triển các mô hình tự động học đặc trưng từ dữ liệu thô (end-to-end) nhằm giảm thiểu sự phụ thuộc vào các đặc trưng trích xuất thủ công, cùng với tiềm năng chưa được khai thác nhiều của kỹ thuật học sâu lai và học chắt lọc tri thức, luận án được thực hiện với 3 nhánh nghiên cứu chính, được nâng cấp và cải thiện theo thời gian, cụ thể như sau:

Thứ nhất, trong bối cảnh dữ liệu huấn luyện hạn chế, để nâng cao khả năng khai thác thông tin từ cả đặc trưng sinh học thủ công và biểu diễn theo ngữ cảnh từ NLP, nghiên cứu đã đề xuất một mô hình học máy tổ hợp, sử dụng đặc trưng lai ghép giữa hai nhóm: (i) đặc trưng thủ công (AAindex, BLOSUM62, CKSAAP) và (ii) đặc trưng học được từ mô hình NLP (Word2Vec). Sự kết hợp này giúp tận dụng đồng thời kiến thức sinh học chuyên biệt và khả năng học mẫu ngữ cảnh từ chuỗi protein. Ngoài ra, thiết kế lai ghép còn đóng vai trò như một bước nền quan trọng để đánh giá định lượng mức độ đóng góp của từng nhóm đặc trưng trong nhiệm vụ dự đoán vị trí PTM.

Thứ hai, nhằm giảm thiểu sự phụ thuộc vào đặc trưng thủ công, nghiên cứu phát triển một mô hình học sâu lai kết hợp giữa CNN1D và LSTM/Bi-LSTM, tích hợp kỹ thuật NLP để tự động học biểu diễn đặc trưng từ mẫu dữ liệu thô. Mô hình được huấn luyện theo cơ chế đầu-cuối (end-to-end), trong đó toàn bộ quy trình từ biểu diễn, trích chọn đặc trưng đến phân loại đều được tối ưu hóa đồng thời trong một mạng duy nhất. Cách tiếp cận này giúp mô hình học trực tiếp từ dữ liệu thô, nâng cao tính tổng quát và khả năng tự thích ứng.

Thứ ba, nhận thấy mô hình học sâu lai và tổ hợp tuy hiệu quả nhưng tiêu tốn tài nguyên tính toán đáng kể, nghiên cứu đề xuất ứng dụng kỹ thuật học chắt lọc tri thức kết hợp với kỹ thuật NLP để xây dựng mô hình dự đoán PTM hiệu quả hơn về mặt chi phí. Cụ thể, một mô hình mạnh (giáo viên) sẽ truyền tri thức cho một mô hình nhẹ hơn (học viên), qua đó duy trì hiệu năng dự đoán trong khi giảm đáng kể độ phức tạp và thời gian huấn luyện. Đây là hướng tiếp cận mới, chưa được áp dụng trong lĩnh vực dự đoán PTM, có tiềm năng lớn cả về giá trị học thuật lẫn tính ứng dụng thực tiễn.

## 1.7 Kết luận chương 1

Trong chương 1, NCS đã trình bày các kiến thức nền tảng về protein, đặc biệt protein sửa đổi sau dịch mã, vai trò việc xác định vị trí PTM trên chuỗi protein. Phát biểu bài toán dự đoán vị trí PTM. Quy trình xây dựng mô hình dự đoán, phương pháp mã hoá đặc trưng hiện nay, phương pháp đánh giá mô hình dự đoán, yêu cầu hệ thống thư viện và môi trường cài đặt. Tổng quan tình hình nghiên cứu trong bối cảnh sự phát triển của AI và SOTA, khoảng trống nghiên cứu, lựa chọn hướng nghiên cứu của luận án. Một phần nội dung trong chương này đã được NCS công bố trên tạp chí và hội thảo sau:

[CT1] Le N.Q.K, Tran T.X, Nguyen P.A. Nguyen V.N, et al. (2023), Recent progress in machine learning approaches for predicting carcinogenicity in drug development. Expert Opinion on Drug Metabolism & Toxicology. p 621-628, DOI: `https://doi.org/10.1080/17425255.2024.2356162`.**(SCIE Q1, IF: 3.9)**

[CT2] Le N.Q.K, Nguyen V.N, Nguyen T.T, Tran T.X, at al. (2024), Enhancing Protein Sequence Classification with a Fuzzy Neural Network: A Study in Anticancer Peptide Identification, International Conference on Fuzzy Theory and Its Applications (iFUZZY), Kagawa, Japan. pp. 1-6, DOI: `https://doi.org/10.1109/iFUZZY63051.2024.10662887`.

# CHƯƠNG 2. MÔ HÌNH HỌC MÁY TỔ HỢP DỰ ĐOÁN VỊ TRÍ SỬA ĐỔI SAU DỊCH MÃ TRONG CHUỖI PROTEIN

*Trong giai đoạn đầu của nghiên cứu, để giải quyết vấn đề dữ liệu hạn chế và để hiểu một số loại đặc trưng trong dự đoán PTM, NCS nghiên cứu các mô hình học máy và các đặc trưng (bao gồm cả thủ công và đặc trưng từ mô hình NLP Word2vec), tiếp theo NCS đề xuất mô hình học máy tổ hợp dựa trên đặc trưng lai để dự đoán PTM. Mặc dù việc kết hợp nhiều đặc trưng làm tăng chiều và độ phức tạp, tuy nhiên bước này có vai trò như một bước nền để đánh giá khả năng đóng góp của loại đặc trưng trong mô hình dự đoán vị trí PTM. Kết quả nghiên cứu được công bố tại Hội thảo quốc tế CITA2023 (Indexed: Scopus Q4) - (CT3).*

## 2.1 Đặt vấn đề

Dự đoán vị trí PTM là một bài toán quan trọng trong sinh học phân tử, có ý nghĩa lớn trong việc hiểu rõ hơn về các cơ chế sinh học và phát triển các phương pháp điều trị bệnh. Tuy nhiên, bài toán này gặp nhiều thách thức do tính phức tạp của dữ liệu sinh học, sự đa dạng của các loại PTM và sự hạn chế về dữ liệu.

Một trong những hướng tiếp cận tiềm năng để nâng cao hiệu suất dự đoán vị trí PTM với học máy truyền thống đó là sử dụng kỹ thuật học máy tổ hợp, giúp kết hợp nhiều mô hình thành phần nhằm cải thiện độ chính xác và tính ổn định của mô hình dự đoán. Các nghiên cứu trước đây đã chứng minh rằng học máy tổ hợp có thể nâng cao hiệu suất dự đoán trong nhiều lĩnh vực, bao gồm nhận diện mẫu sinh học, phân loại bệnh và dự đoán tương tác protein-protein [47, 93].

Trong lĩnh vực y sinh, học máy tổ hợp đã được ứng dụng thành công vào các bài toán như chẩn đoán bệnh tim, ung thư, và các bệnh thần kinh, cho thấy khả năng tổng hợp thông tin từ nhiều mô hình (Decision Tree, RF, Naïve Bayes (NB), SVM) để cải thiện độ chính xác và giảm thiểu sai số [28, 51, 109]. Điều này cho thấy phương pháp học máy tổ hợp có thể mang lại lợi ích đáng kể trong bài toán dự đoán vị trí PTM, nơi yêu cầu sự chính xác cao trong việc xác định vị trí sửa đổi sau dịch mã của protein.

Bên cạnh việc lựa chọn mô hình, hiệu suất của mô hình dự đoán vị trí PTM còn phụ thuộc vào đặc trưng đầu vào. Các đặc trưng phổ biến hiện nay có thể chia thành hai nhóm chính: (1) đặc trưng dựa trên trình tự, phản ánh các thuộc tính sinh lý–hóa học và cấu trúc cục bộ của axit amin; và (2) đặc trưng dựa trên kỹ thuật xử lý ngôn ngữ tự nhiên, cho phép khai thác các mối quan hệ ngữ nghĩa tiềm ẩn trong chuỗi protein. Tuy nhiên, việc sử dụng một nhóm đặc trưng duy nhất có thể khiến mô hình bỏ sót các chiều

thông tin bổ trợ cần thiết cho việc phân biệt tín hiệu PTM.

Từ những lý do trên, trong nghiên cứu này NCS đề xuất một phương pháp học máy tổ hợp, trong đó kết hợp nhiều mô hình học máy cổ điển (RF, XGBoost và SVM), đồng thời khai thác đặc trưng lai ghép giữa đặc trưng thủ công và đặc trưng từ mô hình ngôn ngữ tự nhiên. Phương pháp này không chỉ tận dụng lợi thế của học máy tổ hợp trong việc nâng cao hiệu quả dự đoán, mà còn cho phép đánh giá mức độ đóng góp của từng loại đặc trưng và từng mô hình thành phần, làm nền tảng cho các bước nghiên cứu ở các chương sau.

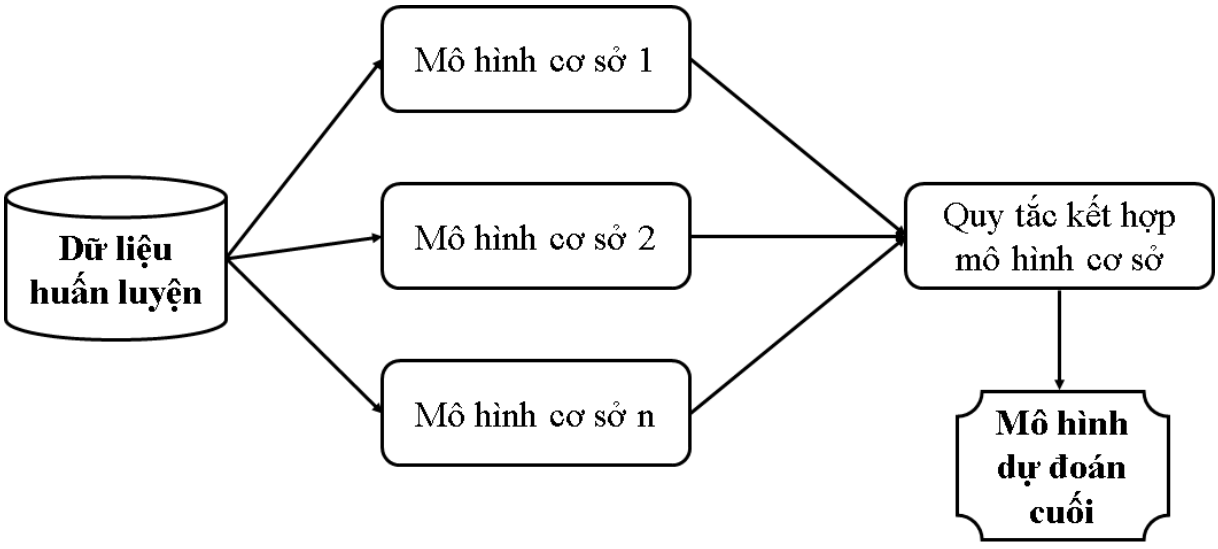
## 2.2 Kỹ thuật học máy tổ hợp

học máy tổ hợp (Ensemble learning) [23] là một kỹ thuật được sử dụng để kết hợp hai hoặc nhiều thuật toán học máy nhằm đạt được hiệu suất vượt trội so với khi sử dụng từng thuật toán riêng lẻ. Thay vì chỉ dựa vào một mô hình duy nhất, các dự đoán từ từng mô hình thành phần được kết hợp lại bằng một quy tắc tổ hợp để tạo ra một dự đoán duy nhất có độ chính xác cao hơn.

Có thể chia kỹ thuật học máy tổ hợp thành 2 nhóm [70]:

Nhóm 1: Tổ hợp học song song

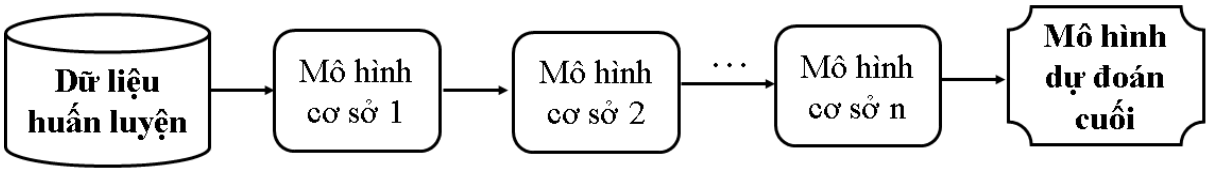
Tổ hợp song song huấn luyện các mô hình cơ sở một cách độc lập và kết hợp dự đoán của chúng thông qua một bộ kết hợp. Một phương pháp tổ hợp song song phổ biến là bagging, cùng với phần mở rộng của nó, thuật toán RF. Các thuật toán tổ hợp song song sử dụng việc tạo ra các mô hình cơ sở theo cách song song nhằm khuyến khích sự đa dạng giữa các mô hình trong tổ hợp.




**Hình 2.1 Kiến trúc học máy tổ hợp song song**

Nhóm 2: Tổ hợp học tuần tự

Tổ hợp tuần tự không huấn luyện các mô hình cơ sở một cách độc lập mà được huấn luyện theo từng bước lặp. Ở mỗi vòng lặp, mô hình mới học cách sửa lỗi của mô hình trước đó nhằm cải thiện hiệu suất tổng thể.




**Hình 2.2 Kiến trúc học máy tổ hợp tuần tự**

**Lựa chọn mô hình học máy con trong học máy tổ hợp** [70]:

Trong học máy tổ hợp, việc lựa chọn một tập hợp con phù hợp từ các mô hình cơ sở là một vấn đề quan trọng, vì không phải tất cả các mô hình đều có hiệu suất tốt như nhau. Thay vì kết hợp cả các mô hình mạnh và yếu, việc chỉ chọn ra các mô hình có hiệu suất cao sẽ giúp cải thiện độ chính xác và khả năng tổng quát hóa của bộ phân loại tổ hợp.

Một số phương pháp học máy tổ hợp tiêu biểu bao gồm:

Bagging: Sử dụng bootstrap sampling để tạo ra nhiều tập dữ liệu huấn luyện con. Các mô hình cơ sở được huấn luyện song song và kết quả được tổng hợp bằng bỏ phiếu (phân loại) hoặc trung bình (hồi quy).

Boosting: Xây dựng các mô hình tuần tự, trong đó mỗi mô hình tập trung vào những mẫu mà mô hình trước đó dự đoán sai. Các thuật toán nổi bật theo hướng này là Gradient Boosting Machines (GBM) và XGBoost.

Voting: Biểu quyết (Voting) là một kỹ thuật tổng hợp phổ biến trong phân loại, kết hợp đầu ra của nhiều bộ phân loại để đưa ra dự đoán cuối cùng [32, 93]. Có ba biến thể chính:

+ Bỏ phiếu đa số (Max Voting): Nhãn được chọn là nhãn có số phiếu nhiều nhất từ các mô hình.

+ Bỏ phiếu trung bình cộng (Average): Kết quả dự đoán cuối cùng là trung bình cộng các dự đoán của các mô hình.

+ Bỏ phiếu trung bình cộng có trọng số (Weighted average Voting - WAV): Mỗi mô hình sẽ được gắn một tỉ trọng (trọng số quan trọng của mô hình). Kết quả cuối cùng là trung bình cộng của dự đoán với trọng số của các mô hình.

Các chiến lược trên cho thấy sự khác biệt trong cách kết hợp mô hình: Bagging

thiên về giảm phương sai, Boosting tập trung giảm sai lệch, còn Voting cung cấp một cách tiếp cận đơn giản nhưng hiệu quả để khai thác sức mạnh tập thể của nhiều bộ phân loại.

Lựa chọn mô hình học máy con là một kỹ thuật được sử dụng để xây dựng bộ phân loại tổ hợp từ một tập hợp các mô hình cơ sở. Đây là một chủ đề quan trọng trong học máy tổ hợp, vì việc chọn một tập hợp con phù hợp của các mô hình cơ sở có thể mang lại hiệu suất tốt hơn so với khi sử dụng toàn bộ các mô hình để tạo bộ phân loại tổ hợp.

Do các mô hình cơ sở được phát triển bằng các thuật toán học máy khác nhau hoặc trên các tập con khác nhau của dữ liệu huấn luyện, hiệu suất của chúng cũng sẽ khác nhau; một số mô hình có thể đạt hiệu suất tốt, trong khi những mô hình khác có thể hoạt động kém. Thay vì kết hợp cả mô hình tốt và kém, việc chỉ chọn một tập hợp con gồm các mô hình có hiệu suất cao có thể mang lại lợi ích, giúp cải thiện hiệu suất tổng thể của tổ hợp. Giả mã của thuật toán Bagging, Boosting và Stacking được trình bày chi tiết tại phụ lục:

## 2.3 Mô hình dự đoán SUMOylation dựa trên kỹ thuật học máy tổ hợp đề xuất

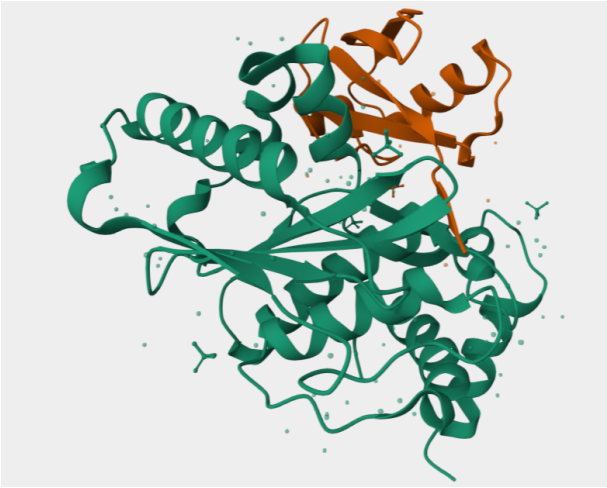
### *2.3.1 Tên viết tắt*

Trong nghiên cứu này, NCS đề xuất một mô hình học máy tổ hợp nhằm dự đoán vị trí SUMOylation, được đặt tên là **RSX_SUMO**. Tên gọi này là viết tắt của "An ensemble learning approach combining RF, SVM, and XGBoost for SUMOylation site prediction". RSX_SUMO kết hợp ba thuật toán RF, SVM and XGBoost với trọng số cho dự đoán PTM SUMOylation.Trong luận án NCS sử dụng tên gọi **RSX_SUMO** này để chỉ mô hình học máy tổ hợp để xuất cho dự đoán SUMOylation.

### *2.3.2 Dữ liệu thực nghiệm*

Trong nghiên cứu này, NCS sử dụng kỹ thuật học máy tổ hợp để xây dựng mô hình dự đoán vị trí PTM SUMOylation.

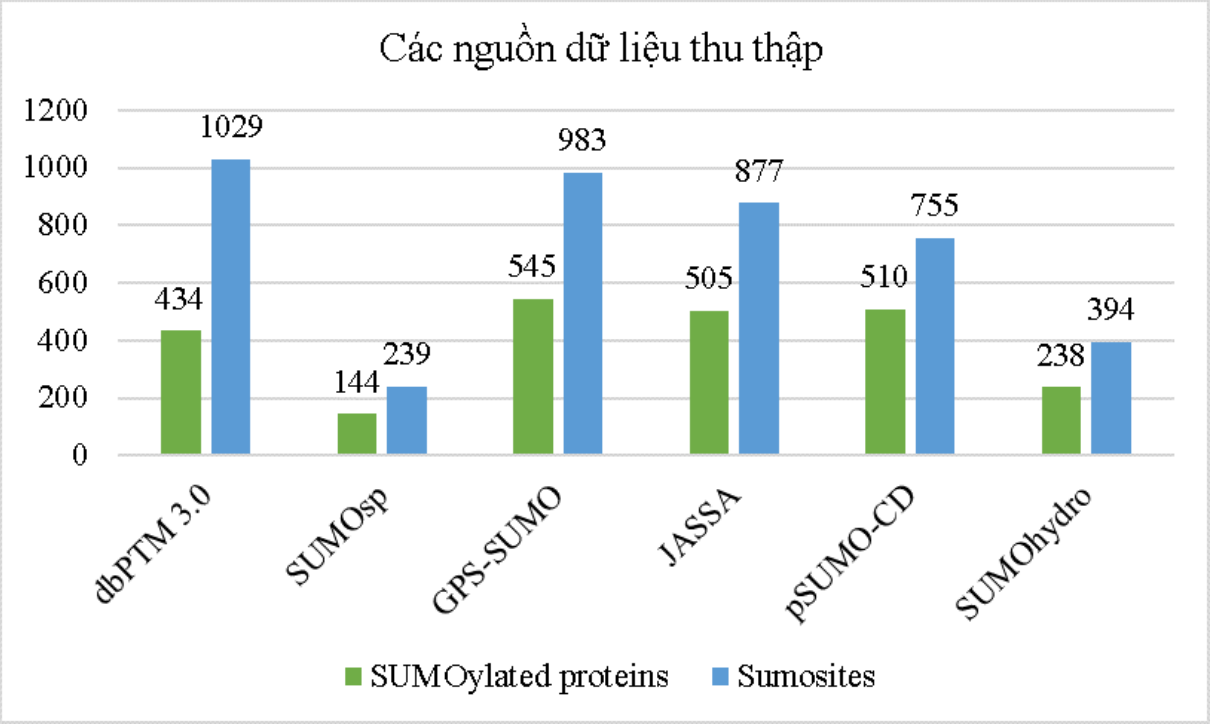
SUMOylation là một dạng biến đổi sau dịch mã, trong đó các protein thuộc họ Small Ubiquitin-like Modifier (SUMO) được gắn vào protein mục tiêu, đóng vai trò quan trọng trong nhiều quá trình sinh học như vận chuyển nội bào, phiên mã, sửa chữa DNA và truyền tín hiệu [33, 40, 74]. SUMO protein có bốn isoform chính: SUMO-1, SUMO-2, SUMO-3, và SUMO-4, với cấu trúc đặc trưng thể hiện qua các mô hình không gian của protein (Hình 2.3).



**Hình 2.3 Cấu trúc protein SUMO1 ở người (P63165_SUMO1_HUMAN) [108]**

Nghiên cứu gần đây cho thấy SUMOylation có thể tăng cường khả năng liên kết của protein, đặc biệt là với một số protein như Claspin, vốn phụ thuộc vào SUMOylation để thực hiện chức năng liên kết. Hơn nữa, nhiều bệnh lý nghiêm trọng như Alzheimer và Parkinson có liên quan chặt chẽ đến quá trình SUMOylation [68, 87, 96].

Dữ liệu về các vị trí SUMOylation xác thực thực nghiệm được thu thập từ nhiều cơ sở dữ liệu nguồn mở và các nghiên cứu đã công bố, tiêu biểu như dbPTM3.0 [65], SUMOsp [92], GPS-SUMO [127], JASSA [9], pSumo-CD [49], seeSUMO [104], và SUMOhydro [15]. Số lượng dữ liệu thu thập được trình bày trong Hình 2.4 và Bảng 2.1 . Sau khi thực hiện một số bước kỹ thuật nhằm loại bỏ các protein trùng lặp hoặc dư thừa, thu được bộ dữ liệu không trùng lặp cuối cùng gồm 1160 protein duy nhất. Thực hiện chọn ngẫu nhiên 160 protein từ bộ dữ liệu không trùng lặp để làm bộ dữ liệu kiểm thử độc lập. Phần dữ liệu còn lại 1000 protein được sử dụng làm bộ dữ liệu huấn luyện.




**Hình 2.4 Nguồn dữ liệu SUMOylation thu thập**

**Bảng 2.1 Thống kê dữ liệu SUMOylation thu thập**

| | **SUMOylated proteins** | **SUMO sites** |
|---|---|---|
| Tổng dữ liệu thu thập | 2623 | 4654 |
| Dữ liệu đã loại bỏ trùng giữa các bộ dữ liệu | 1160 | 2109 |
| Dữ liệu huấn luyện | 1000 | 1820 |
| Dữ liệu kiểm tra | 160 | 289 |

Để tạo ra mẫu dữ liệu dương tính (SUMO), NCS sử dụng cửa sổ có kích thước là 2n + 1 để trích xuất các đoạn chuỗi có trung tâm là vị trí lysine (K) đã được xác minh thực nghiệm là có SUMOylation, đồng thời chứa n axit amin lân cận ở cả hai phía. Với một số lượng protein SUMOylated đã được xác minh thực nghiệm, các đoạn peptid có chiều dài cửa sổ 2n + 1 axit amin và có trung tâm là lysine nhưng không được chú thích là có SUMOylation được xem là mẫu âm tính (non-SUMO). Do đó, bộ dữ liệu huấn luyện ban đầu bao gồm 1820 mẫu dương tính và 37222 mẫu âm tính. Tuy nhiên, do một số mẫu âm tính có thể giống hệt mẫu dương tính nên hiệu suất dự đoán của mô hình có thể bị đánh giá quá cao trong cả bộ dữ liệu huấn luyện và kiểm thử. Để khắc phục điều này, NCS đã sử dụng chương trình CD-HIT để loại bỏ dữ liệu với độ giống 40%. Cuối cùng bộ dữ liệu về dự đoán vị trí PTM SUMOylation được sử dụng trong nghiên cứu

thực nghiệm mô hình học máy tổ hợp như trong Bảng 2.2 dưới đây.

**Bảng 2.2 Bộ dữ liệu SUMOylation sử dụng trong nghiên cứu**

| | SL mẫu dương tính | SL mẫu âm tính |
|---|---|---|
| Tập dữ liệu huấn luyện | 745 | 1490 |
| Tập dữ liệu kiểm tra | 117 | 234 |

### *2.3.3 Phương pháp mã hoá và trích chọn đặc trưng*

Để xây dựng các mô hình dự đoán cho việc xác định các vị trí SUMOylation, NCS trích chọn các đặc trưng dựa trên chuỗi: AAIndex, CKSAAP, BLOSUM62, các đặc trưng này được trích chọn bởi tool iFeature; đặc trưng dựa trên kỹ thuật NLP được trích chọn bởi mô hình Word2Vec (Skip-gram). Chi tiết các đặc trưng và kích thước của véc tơ đặc trưng được hiển thị trong Bảng 2.3.

**Bảng 2.3 Véc tơ đặc trưng sử dụng trong nghiên cứu**

| Đặc trưng đơn | Kích thước | Đặc trưng lai ghép | Kích thước |
|---|---|---|---|
| AAIndex | 6903 | AAIndex_CKSAAP | 7053 |
| CKSAAP | 150 | AAIndex_BLOSUM62 | 7163 |
| BLOSUM62 | 260 | AAIndex_Word2Vec | 7003 |
| Word2Vec | 100 | CKSAAP_BLOSUM62 | 410 |
| | | CKSAAP_Word2Vec | 250 |
| | | BLOSUM62_Word2Vec | 360 |
| | | AAIndex_CKSAAP_BLOSUM62 | 7313 |
| | | AAIndex_CKSAAP_Word2Vec | 7153 |
| | | CKSAAP_BLOSUM62_Word2Vec | 510 |
| | | AAIndex_CKSAAP_BLOSUM62_Word2Vec | 7413 |

### *2.3.4 Kiến trúc mô hình dự đoán PTM đề xuất dựa trên kỹ thuật học máy tổ hợp*

Mục tiêu chính của nghiên cứu là cải thiện hiệu suất dự đoán vị trí PTM SUMOylation thông qua việc kết hợp nhiều mô hình học máy với các nguyên lý khác nhau. Để thực hiện, ba mô hình được lựa chọn gồm RF [42], XGBoost [14] và SVM [21], đại diện

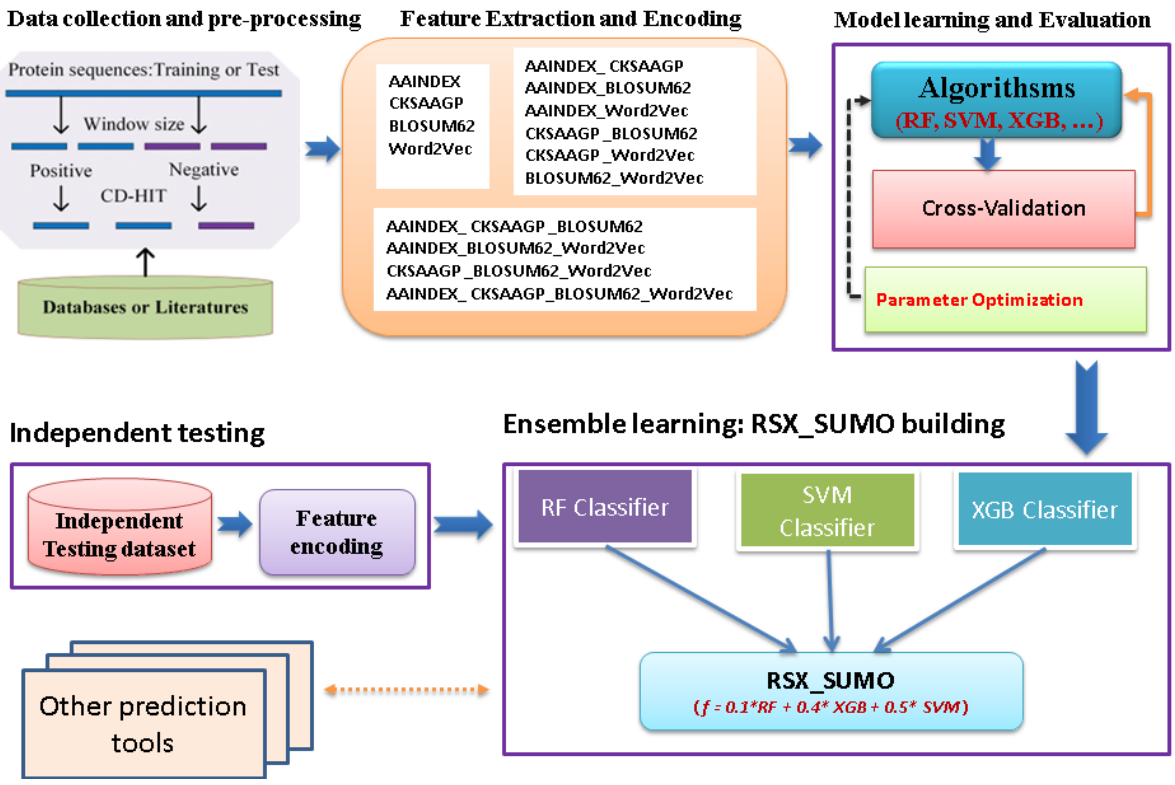
cho ba hướng tiếp cận học máy khác biệt. Kết quả dự đoán cuối cùng được tổng hợp bằng cơ chế kết hợp có trọng số, với công thức:

$$f = w_1 \cdot RF + w_2 \cdot XGB + w_3 \cdot SVM$$

trong đó các trọng số $w_1, w_2, w_3$ được xác định thông qua quá trình tối ưu trên tập validation (Algorithm 2.2). Nhờ vậy, mỗi mô hình đóng góp theo thế mạnh riêng, đồng thời hạn chế nhược điểm của nhau.

Ba mô hình được chọn dựa trên nguyên tắc đa dạng hoá nhằm tăng khả năng khái quát hoá trên dữ liệu sinh học phức tạp. Cụ thể, RF là thuật toán Bagging với nhiều cây quyết định huấn luyện song song trên các tập con dữ liệu, giúp giảm phương sai và tăng tính ổn định. XGBoost đại diện cho nhóm Boosting, học tuần tự để khắc phục sai số của các mô hình trước, đồng thời tích hợp các cơ chế regularization nhằm hạn chế overfitting và tăng khả năng tổng quát hoá. SVM, tuy không phải là phương pháp tổ hợp, nhưng bổ sung tính đa dạng bằng khả năng phân tách mạnh trên không gian đặc trưng cao, đặc biệt nhờ kỹ thuật kernel cho phép phân loại cả dữ liệu phi tuyến.

Sự kết hợp này tạo ra sự bổ trợ lẫn nhau giữa ba hướng tiếp cận (song song, tuần tự, tối ưu biên), giúp mô hình tổ hợp không chỉ nâng cao hiệu suất dự đoán vị trí PTM mà còn khắc phục hạn chế của từng thuật toán đơn lẻ. Kiến trúc tổng thể của mô hình được trình bày trong Hình 2.5.




**Hình 2.5 Kiến trúc mô hình học máy tổ hợp dự đoán PTM đề xuất**

Hình 2.5 minh họa quy trình xây dựng mô hình dự đoán vị trí PTM SUMOylation dựa trên kỹ thuật học máy tổ hợp. Trước hết, dữ liệu protein được tiền xử lý và trích xuất đặc trưng, sau đó được chia thành các tập huấn luyện, kiểm tra và validation. Trên tập huấn luyện, ba mô hình học máy cơ sở gồm RF, XGBoost và SVM được xây dựng độc lập theo nguyên lý học khác nhau. Tiếp đó, đầu ra dự đoán của từng mô hình được tổng hợp thông qua cơ chế cộng có trọng số, trong đó các trọng số tối ưu được xác định nhờ quá trình tìm kiếm trên tập validation. Kết quả cuối cùng là một mô hình tổ hợp, vừa tận dụng được ưu điểm của từng thuật toán, vừa giảm thiểu hạn chế riêng lẻ, qua đó cải thiện hiệu suất dự đoán SUMOylation trên dữ liệu sinh học phức tạp. Các bước thực hiện của mô hình đề xuất, bao gồm giả mã của hai thuật toán: thuật toán xác định trọng số (Algorithm 2.2) và thuật toán học máy tổ hợp (Algorithm 2.1):

Bước 1: Đánh giá hiệu suất mô hình trên các đặc trưng đơn, đặc trưng lai ghép trên các mô hình học máy cơ sở, để tìm đặc trưng cho hiệu suất tốt nhất. Đặc trưng tốt nhất này sẽ là đầu vào cho mô hình học tập tổng hợp.

Bước 2: Định nghĩa mô hình tổ hợp với 3 mô hình học máy cơ sở (Algorithm 2.1).

Bước 3: Sử dụng (Algorithm 2.2) để tìm các trọng số cho mô hình tổ hợp.

Bước 4: Huấn luyện mô hình tổ hợp đề xuất trên bộ trọng số tốt nhất đã tìm được ở bước 4 bởi (Algorithm 2.1).

Bước 5: Sử dụng mô hình ở Bước 4 để dự đoán mẫu dữ liệu mới.

### *2.3.5 Chiến lược và tham số huấn luyện mô hình*

Các tham số chi tiết của ba mô hình cơ sở (XGBoost, SVM và RF) được trình bày trong Bảng 2.4.Quá trình huấn luyện mô hình RSX_SUMO được thực hiện trên môi trường Google Colab với sự hỗ trợ của GPU, giúp rút ngắn đáng kể thời gian tính toán. Trong quá trình huấn luyện, các mô hình cơ sở được huấn luyện theo bộ tham số đã định nghĩa và đánh giá hiệu suất trên tập validation. Kết quả dự đoán của từng mô hình sau đó được tổng hợp thông qua cơ chế cộng có trọng số, trong đó bộ trọng số tối ưu được xác định bằng giải thuật riêng (Algorithm 2.5). Mỗi mô hình cơ sở vì vậy đóng góp theo đúng thế mạnh của mình, đồng thời hạn chế được những nhược điểm vốn có.

Chiến lược này cho phép mô hình tổ hợp đạt được hiệu suất dự đoán ổn định và đáng tin cậy hơn so với việc sử dụng từng mô hình đơn lẻ, đồng thời khai thác hiệu quả sự đa dạng trong nguyên lý học tập của XGBoost (Boosting), RF (Bagging) và SVM (tối ưu biên).

**Algorithm 2.1** học máy tổ hợp có trọng số với 3 mô hình học máy cơ sở

**Đầu vào:** Tập dữ liệu ban đầu $\mathscr{D}$; tập huấn luyện $X_{\text{train}} \subseteq \mathscr{D}$; tập kiểm tra $X_{\text{test}} \subseteq \mathscr{D}$; mô hình tổ hợp $\mathscr{M} = \{\text{RF}, \text{XGB}, \text{SVM}\}$; véc tơ trọng số $\mathbf{w} = (w_1, w_2, w_3)$

**Đầu ra:** Kết quả phân lớp Result $= \{\hat{y}_j\}$ cho mọi $x_j \in X_{\text{test}}$

$k \leftarrow 3$
Result $\leftarrow \emptyset$
**Huấn luyện từng mô hình cơ sở**
**for** $i = 1$ to $k$ **do**
    Huấn luyện mô hình $M_i$ trên $X_{\text{train}}$
**end for**
**Dự đoán cho mỗi mẫu trong tập kiểm tra**
**for all** $x \in X_{\text{test}}$ **do**
    **for** $i = 1$ to $k$ **do**
        Dự đoán xác suất từ mô hình $M_i$
        $p_i(x) \leftarrow M_i(x)$
    **end for**
    $p(x) \leftarrow \sum_{i=1}^{k} w_i \cdot p_i(x)$
    **if** $p(x) \geq 0.5$ **then**
        $\hat{y} \leftarrow 1$
    **else**
        $\hat{y} \leftarrow 0$
    **end if**
    Result $\leftarrow$ Result $\cup \{\hat{y}\}$
**end for**
**return** Result

**Algorithm 2.2** Tìm trọng số tối ưu $w_1, w_2, w_3$ cho mô hình tổ hợp RF, XGB và SVM

**Đầu vào:** Tập dữ liệu ban đầu $\mathscr{D}$; tập huấn luyện $X_{\text{train}}$; tập tinh chỉnh $X_{\text{val}}$; mô hình tổ hợp $\mathscr{M} = \{\text{RF}, \text{XGB}, \text{SVM}\}$; bước tăng $\delta = 0.1$

**Đầu ra:** Trọng số tối ưu $(w_1^*, w_2^*, w_3^*)$ sao cho hàm mất mát MSE là nhỏ nhất.

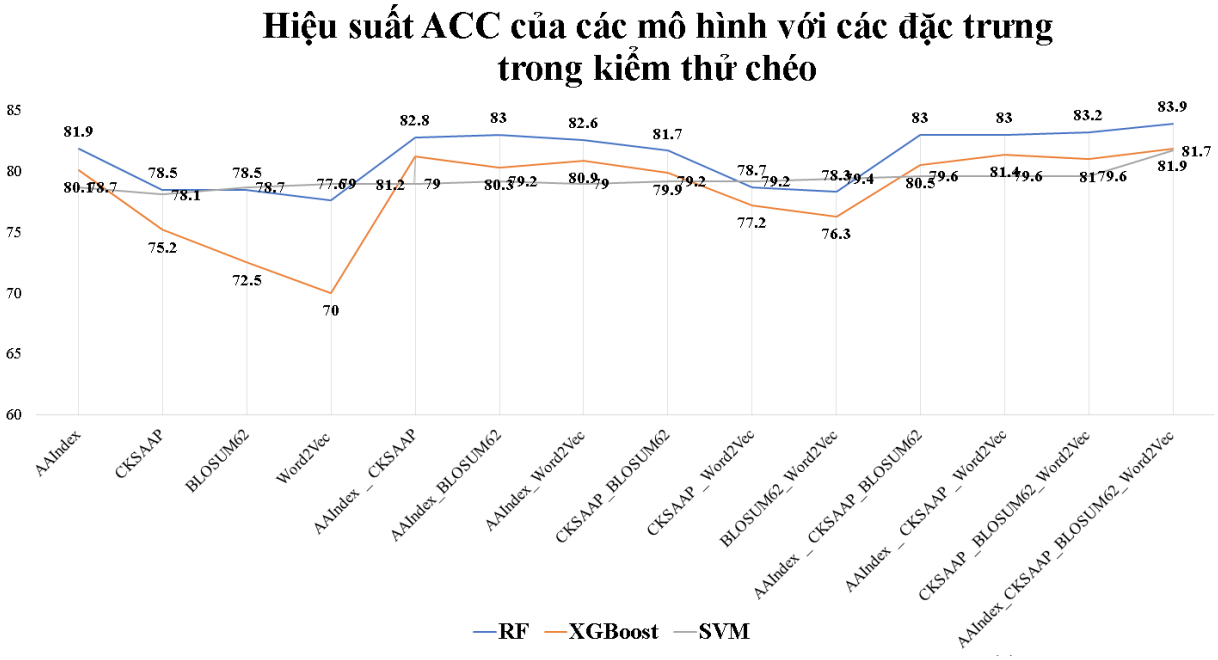
best_score $\leftarrow +\infty$
best_weights $\leftarrow \emptyset$
**for** $w_1 = 0$ to 1 step $\delta$ **do**
    **for** $w_2 = 0$ to 1 step $\delta$ **do**
        $w_3 \leftarrow 1 - (w_1 + w_2)$
        **if** $0 \leq w_3 \leq 1$ **then**
            Khởi tạo mô hình tổ hợp
            $M = \text{Ensemble\_model}(\text{RF} : w_1, \text{XGB} : w_2, \text{SVM} : w_3)$
            Huấn luyện $M$ trên $X_{\text{train}}$
            $\hat{y} \leftarrow M(X_{\text{val}})$
            score $\leftarrow \text{MSE}(y_{\text{val}}, \hat{y})$
            **if** score $<$ best_score **then**
                best_score $\leftarrow$ score
                best_weights $\leftarrow (w_1, w_2, w_3)$
            **end if**
        **end if**
    **end for**
**end for**
**return** best_weights

**Bảng 2.4 Các tham số của XGBoost, SVM và RF**

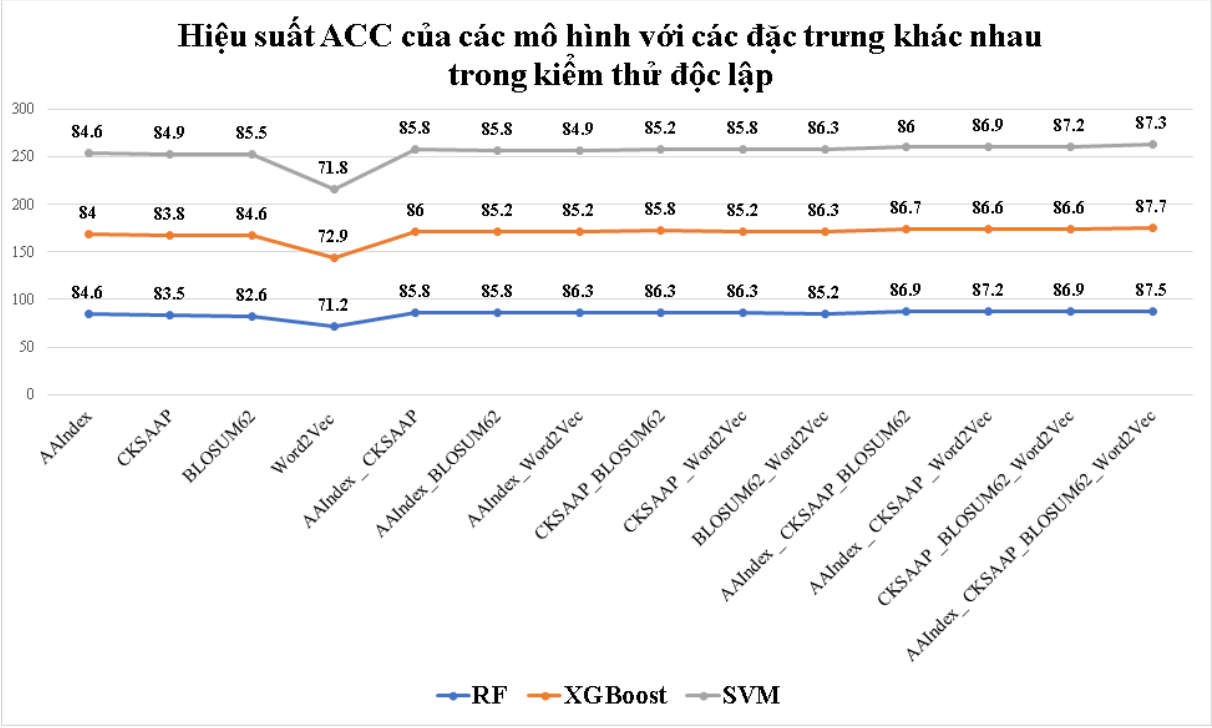
| Tham số cho mô hình XGBoost | Tham số cho mô hình SVM và RF |
|---|---|
| colsample_bytree: 0.9 | **Mô hình SVM** |
| learning_rate: 0.01 | C: 1.0 |
| max_depth: 7 | gamma: 0.01 |
| min_child_weight: 30 | kernel: 'linear' |
| nthread: -1 | probability: True |
| objective: 'binary:logistic' | **Mô hình RF** |
| reg_lambda: 1.0 | max_features: 8 |
| use_label_encoder: False | n_estimators: 100 |
| eval_metric: 'logloss' | |

### *2.3.6 Kết quả và thảo luận*

Kết quả kiểm thử chéo (Hình 2.6) và kiểm thử độc lập (Hình 2.7) cho thấy việc kết hợp các đặc trưng AAIndex, CKSAAP, BLOSUM62 và Word2Vec giúp cải thiện hiệu suất dự đoán so với từng đặc trưng đơn lẻ. Trong kiểm thử chéo, các mô hình RF, XGBoost và SVM đều ghi nhận mức tăng ACC từ 1–3% khi sử dụng đặc trưng lai, phản ánh sự bổ sung thông tin giữa các loại đặc trưng sinh học và ngữ nghĩa. Tuy mức cải thiện không quá lớn và đi kèm chi phí tính toán cao hơn, đặc trưng lai giúp cải thiện hiệu suất dự đoán thế đáng kể trong bối cảnh dữ liệu hạn chế. Đặc biệt, trong kiểm thử độc lập, tổ hợp đặc trưng này đạt ACC cao nhất với RF (0.875), XGBoost (0.877) và SVM (0.873), cao hơn 3% so với đặc trưng đơn. Do đó, đặc trưng lai AAIndex_CKSAAP_BLOSUM62_Word2Vec được lựa chọn làm đầu vào chính cho mô hình tổ hợp RSX_SUMO nhằm tối ưu hóa hiệu suất dự đoán.



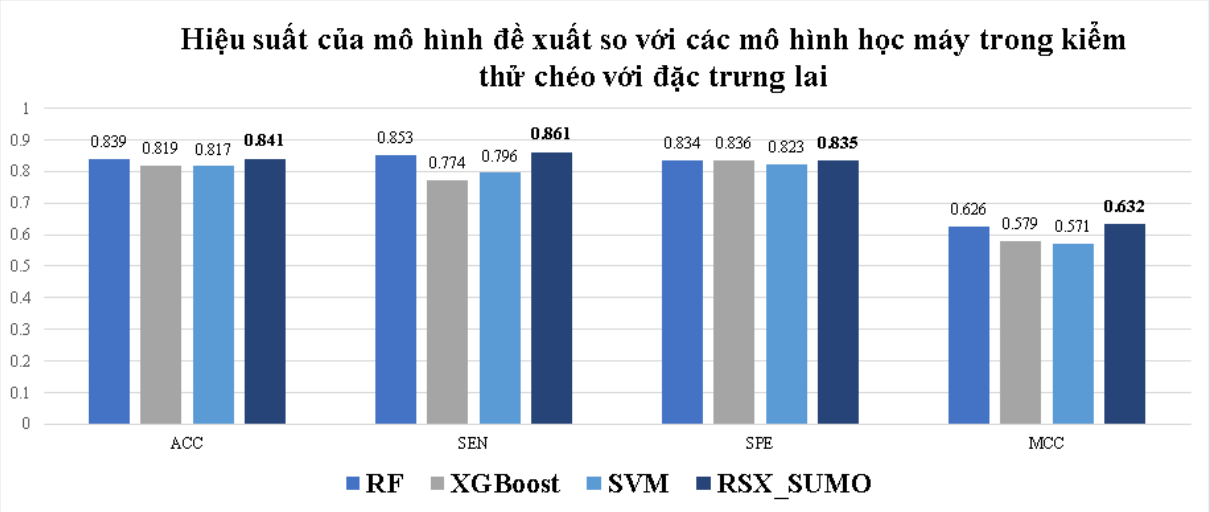

**Hình 2.6 Hiệu suất ACC của các thuật toán học máy trên các đặc trưng nghiên cứu trong kiểm thử chéo**



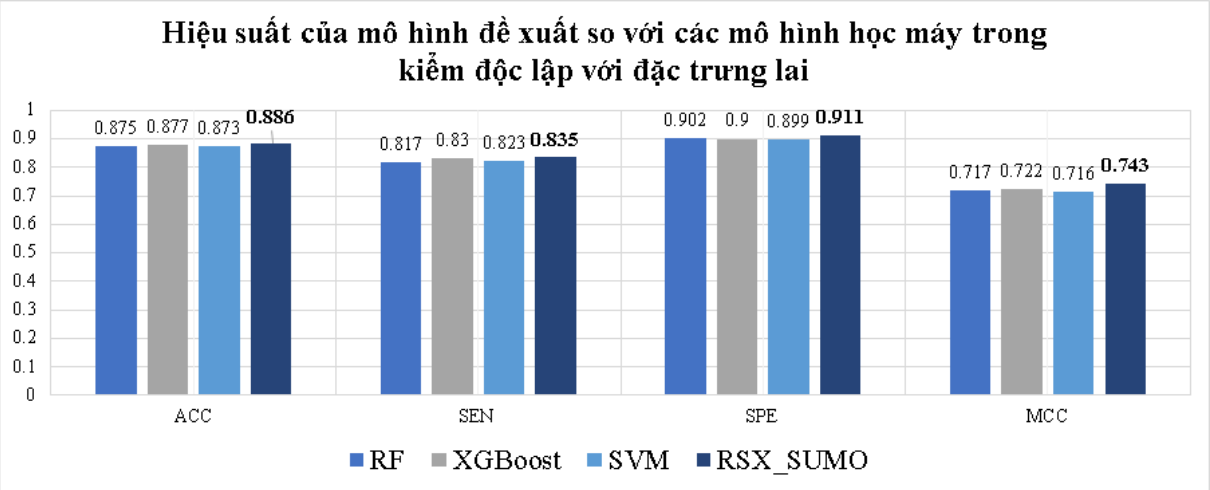

**Hình 2.7 Hiệu suất ACC của các thuật toán trên các đặc trưng nghiên cứu trong kiểm thử độc lập**

Việc kết hợp các đặc trưng dựa trên chuỗi (AAIndex, CKSAAP, BLOSUM62) và đặc trưng ngữ nghĩa từ kỹ thuật NLP (Word2Vec) đã giúp cải thiện hiệu suất dự đoán

đáng kể. Trong khi đặc trưng chuỗi giúp mô hình khai thác thông tin vật lý – hóa học của axit amin, đặc trưng Word2Vec lại bổ sung khả năng nắm bắt mối quan hệ ngữ nghĩa ẩn trong chuỗi protein. Sự kết hợp này tận dụng ưu thế của cả hai nhóm đặc trưng, giúp mô hình hiểu rõ hơn bản chất dữ liệu và tăng cường khả năng nhận diện chính xác vị trí SUMOylation.




**Hình 2.8 Hiệu suất của mô hình đề xuất và các mô hình cơ sở trong kiểm thử chéo với đặc trưng lai được chọn**




**Hình 2.9 Hiệu suất của mô hình đề xuất và các mô hình cơ sở trong kiểm thử độc lập với đặc trưng lai được chọn**

Đánh giá hiệu quả của các mô hình với đặc trưng lai được chọn kết quả ở Hình 2.8 và Hình 2.9 cho thấy mô hình tổ hợp RSX_SUMO vượt trội hơn so với các mô hình cơ sở (RF, XGBoost, SVM) trong cả kiểm thử chéo và kiểm thử độc lập. Cụ thể, RSX_SUMO đạt ACC (0.841) cao hơn XGBoost và SVM 3%, và cao hơn RF không đáng kể trong

kiểm thử chéo. Trong kiểm thử độc lập, RSX_SUMO đạt ACC 0.886, SEN 0.835, SPE 0.911 và MCC 0.743, đều cao hơn các mô hình cơ sở.

## 2.4 So sánh với các công cụ dự đoán khác

Trong nghiên cứu này, việc so sánh được thực hiện với hai công cụ dự đoán SUMOylation phổ biến và được xem là state-of-the-art tại thời điểm nghiên cứu, bao gồm (GPS-SUMO2.0 [127] (truy cập thời điểm năm 2022), seeSUMO2.0 [97]. Đây là những công cụ đã được cộng đồng quốc tế công nhận và sử dụng rộng rãi, có độ tin cậy cao và thường được dùng làm đối chứng trong các nghiên cứu dự đoán PTM. Việc lựa chọn hai công cụ này thay vì các công cụ khác nhằm đảm bảo tính khách quan, đồng thời phản ánh đúng trình độ phát triển của lĩnh vực tại thời điểm triển khai nghiên cứu.

Kết quả so sánh hiệu suất giữa RSX_SUMO và các công cụ dự đoán hiện có được trình bày trong Bảng 2.5.

**Bảng 2.5 So sánh hiệu suất giữa các công cụ dự đoán SUMOylation**

| **Công cụ** | **Ngưỡng** | **ACC** | **SEN** | **SPE** |
|---|---|---|---|---|
| GPS-SUMO2.0 | Low | 0.877 | 0.884 | 0.875 |
| | Medium | 0.794 | 0.694 | 0.838 |
| | High | 0.877 | 0.884 | 0.875 |
| seeSUMO2.0 | Low | 0.855 | 0.828 | 0.865 |
| | Medium | 0.769 | 0.644 | 0.829 |
| | High | 0.836 | 0.790 | 0.853 |
| **RSX_SUMO (Đề xuất)** | | **0.886** | **0.835** | **0.911** |

Kết quả trong Bảng 2.5 cho thấy mô hình đề xuất RSX_SUMO đạt độ chính xác (ACC) 0.886 và độ đặc hiệu (SPE) 0.911, vượt trội so với cả hai công cụ GPS-SUMO2.0 và seeSUMO2.0 trên cùng bộ dữ liệu kiểm thử. Đặc biệt, chỉ số SPE cao cho thấy mô hình đề xuất có khả năng phân biệt tốt giữa các vị trí SUMO và Non-Sumo, giảm thiểu các dự đoán dương tính giả. Điều này chứng minh rằng việc kết hợp học máy tổ hợp với đặc trưng lai đã mang lại hiệu quả rõ rệt, nâng cao khả năng tổng quát hóa của mô hình so với các phương pháp hiện hành.

Như vậy, RSX_SUMO không chỉ kế thừa ưu điểm của các phương pháp học máy thành phần mà còn khắc phục được một số hạn chế vốn có, từ đó trở thành một giải pháp tiềm năng trong dự đoán vị trí SUMOylation.

## 2.5 Kết luận chương 2

Trong chương này, NCS đã đề xuất mô hình RSX_SUMO dự đoán vị trí SUMOyaltion. Mô hình RSX_SUMO xây dựng dựa trên kỹ thuật học máy tổ hợp bằng cách kết hợp ba thuật toán học máy: RF, XGBoost và SVM. Việc kết hợp này giúp khai thác ưu điểm của từng thuật toán, nâng cao hiệu suất phân loại và đảm bảo tính ổn định của mô hình. Tuy nhiên mô hình học máy tổ hợp với nhiều mô hình cơ sở và các đặc trưng lai tuy có hiệu suất cao hơn một chút so với mô hình cơ sở nhưng tốn rất nhiều tài nguyên tính toán. Đây cũng là tiền đề để NCS nghiên cứu đề xuất các phương pháp dự đoán ở các chương tiếp theo. Nội dung của chương NCS được công bố trên hội thảo và sau:

[CT3] ran T.X, Nguyen V.N, and Le N.Q.K. (2023) Incorporating Natural Language- Based and Sequence-Based Features to Predict Protein SUMOylation Sites. Con- ference on Information Technology and its Applications. DOI: https://doi.org/10.1007/978-3-031-36886-8$_7$.$(Indexed : ScopusQ4)$.

# CHƯƠNG 3. MÔ HÌNH HỌC SÂU LAI KẾT HỢP XỬ LÝ NGÔN NGỮ TỰ NHIÊN DỰ ĐOÁN VỊ TRÍ SỬA ĐỔI SAU DỊCH MÃ TRONG CHUỖI PROTEIN

*Tiếp tục nghiên cứu, trong chương này NCS tập trung vào việc khắc phục các hạn chế của mô hình học máy trong bài toán dự đoán vị trí SUMOylation. Ở thời điểm nghiên cứu trong Chương 2, nguồn dữ liệu SUMOylation còn hạn chế, ảnh hưởng không nhỏ đến hiệu quả huấn luyện và khả năng tổng quát hóa của các mô hình. Nhằm cải thiện điều này, trong Chương 3, NCS đã tiến hành cập nhật và mở rộng tập dữ liệu SUMOylation, thu thập bổ sung dữ liệu, giúp cải thiện chất lượng dữ liệu đầu vào cho mô hình học sâu.*

*Song song với việc mở rộng dữ liệu, NCS đề xuất một hướng tiếp cận mới dựa trên mô hình học sâu lai, kết hợp giữa CNN1D và LSTM/Bi-LSTM, đồng thời tích hợp kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để biểu diễn trình tự protein một cách hiệu quả. CNN1D cho phép trích xuất đặc trưng cục bộ liên quan đến tín hiệu PTM, trong khi LSTM/Bi-LSTM có khả năng học các phụ thuộc dài trong chuỗi, vốn rất cần thiết cho các đặc trưng ngữ cảnh sinh học. Cách tiếp cận theo hướng NLP đề xuất còn giúp giảm sự phụ thuộc vào kỹ thuật trích chọn thủ công, thay vào đó cho phép mô hình học đặc trưng tự động từ chuỗi đầu vào.*

*Không dừng lại ở SUMOylation, chương này cũng mở rộng phạm vi nghiên cứu sang một loại PTM khác là Succinylation. Việc phát triển mô hình cho cả hai loại PTM này nhằm vừa kiểm chứng khả năng tổng quát hóa của mô hình đề xuất, vừa góp phần làm giàu hiểu biết của cộng đồng khoa học về các dạng sửa đổi sau dịch mã trong hệ gen và proteome sinh vật.*

*Kết quả thực nghiệm cho thấy các mô hình học sâu đề xuất không chỉ cải thiện độ chính xác và khả năng tổng quát so với phương pháp học máy ở chương trước, mà còn tối ưu hơn về chi phí tính toán và lưu trữ. Một phần kết quả nghiên cứu đã được công bố trên các tạp chí khoa học uy tín như Tạp chí Tin học Điều khiển (CT4) và Computer and Medicine (SCIE Q1, IF 7.0) (CT5), Hội thảo CITA2024 (CT6) và hội thảo ICTA2024 (CT7).*

## 3.1 Mô hình học sâu lai

Khái niệm mô hình học sâu lai (Hybrid Deep learning) không xuất phát từ một bài báo khoa học cụ thể như CNN hay LSTM mà là một thuật ngữ tổng quát chỉ các mô hình kết hợp nhiều kỹ thuật học sâu khác nhau. Mô hình học sâu lai là sự kết hợp giữa các

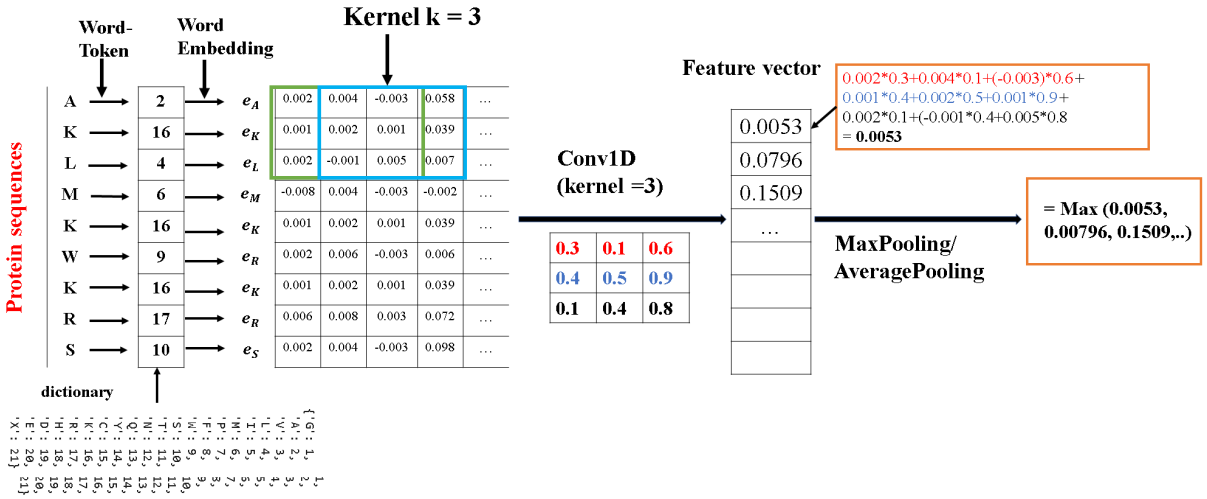
kiến trúc mạng học sâu khác nhau nhằm tận dụng ưu điểm của từng mô hình để cải thiện hiệu suất dự đoán, và kiến trúc học này đã được ứng dụng thành công trong nhiều lĩnh vực giúp cải thiện hiệu suất của các mô hình phân loại [63, 122]. Mặc dù không có một nhà khoa học cụ thể nào được ghi nhận là người đầu tiên đề xuất khái niệm này, nhưng việc kết hợp các mô hình học sâu đã trở thành một xu hướng phổ biến trong nghiên cứu trí tuệ nhân tạo [8].

Trong nghiên cứu này, NCS đề xuất một mô hình học sâu lai để dự đoán vị trí PTM, bằng cách kết hợp hai kiến trúc mạng nơ ron sâu là CNN1D và LSTM/Bi-LSTM. Sự kết hợp này nhằm tận dụng đồng thời khả năng trích xuất đặc trưng cục bộ của CNN1D và khả năng học các quan hệ phụ thuộc dài hạn trong chuỗi của LSTM/Bi-LSTM. Nhờ đó, mô hình có thể học được cả motif cục bộ và ngữ cảnh toàn cục trong chuỗi protein, hai yếu tố then chốt giúp cải thiện độ chính xác trong dự đoán vị trí PTM.

### *3.1.1 Mô hình mạng neural tích chập một chiều (CNN1D)*

Trong nghiên cứu về mô hình học sâu dự đoán vị trí PTM được trình bày trong phần tổng quan trong chương 1, CNN1D là một trong những phương pháp phổ biến ứng dụng trong lĩnh vực này. Với các lớp tích chập (convolutional layers) và lớp giảm chiều (pooling layers), CNN1D có thể phát hiện các mẫu quan trọng trong chuỗi protein.

Trong số các biến thể của CNN, CNN1D được sử dụng phổ biến để dự đoán vị trí PTM [44, 111, 113, 125] do phù hợp với dữ liệu chuỗi protein bậc 1.




**Hình 3.1 Mô hình CNN1D học mẫu dữ liệu protein (1-gram) đề xuất**

CNN1D hoạt động bằng cách sử dụng bộ lọc trượt dọc theo chuỗi, áp dụng phép tích chập lên từng đoạn nhỏ để phát hiện các mẫu theo trình tự. Điều này giúp mô hình nhận diện các đặc trưng quan trọng, chẳng hạn như các cấu trúc con trong chuỗi axit amin. Lớp pooling giúp giảm số lượng tham số và cải thiện khả năng tổng quát hoá,

trong khi các lớp fully connected thực hiện phân loại cuối cùng để xác định vị trí PTM (Hình 3.1).

**Dưới đây là mô tả quy trình học đặc trưng từ chuỗi protein của mạng CNN1D:**

Bước 1: Biểu diễn chuỗi protein dưới dạng Word-Token

- Mỗi axit amin trong chuỗi protein được ánh xạ thành một chỉ số duy nhất dựa trên một từ điển mã hóa (dictionary).

- Ví dụ, axit amin 'A' được mã hóa thành 2, 'K' thành 16, 'L' thành 4, ..v.v.

Bước 2: Embedding

- Các chỉ số Word-Token này được đưa vào một lớp Word Embedding để chuyển thành các véc tơ nhúng có kích thước cố định.

- Mỗi véc tơ Word Embedding tương ứng với một véc tơ đặc trưng của một axit amin, biểu diễn thông tin ngữ nghĩa của nó trong chuỗi protein.

Bước 3: Áp dụng Convolution 1D (Conv1D)

- Một bộ lọc (kernel) kích thước k = 3 trượt qua các véc tơ embedding theo từng cụm liên tiếp để trích xuất đặc trưng cục bộ.

- Mỗi phép tích chập giữa bộ lọc và các giá trị embedding của axit amin trong cửa sổ tạo ra một giá trị đặc trưng mới trong véc tơ đặc trưng.

Bước 4: Tính toán giá trị đặc trưng

- Các giá trị trong véc tơ đặc trưng được tính dựa trên trọng số của bộ lọc và các giá trị embedding của axit amin trong cửa sổ kích thước 3.

- Công thức tính toán là tích có hướng giữa bộ lọc và embedding trong mỗi cửa sổ.

Bước 5: Áp dụng MaxPooling/AveragePooling

- Sau khi qua lớp Convolution, véc tơ đặc trưng được đưa vào lớp Pooling để chọn ra các đặc trưng quan trọng nhất.

- MaxPooling chọn giá trị lớn nhất trong véc tơ đặc trưng để làm đầu ra của lớp pooling, giúp giảm chiều dữ liệu và giữ lại thông tin quan trọng nhất.

Hệ thống học sâu sử dụng CNN1D để trích xuất đặc trưng từ chuỗi protein bằng cách kết hợp Word Embedding, Convolution, và Pooling. Quá trình này giúp mô hình học được các đặc trưng quan trọng trong chuỗi protein, từ đó cải thiện hiệu quả dự đoán vị trí PTM.

Lợi ích của CNN1D trong học chuỗi protein:

- Việc áp dụng CNN1D trong xử lý chuỗi protein mang lại nhiều lợi ích đáng kể,

đặc biệt trong bối cảnh trích xuất đặc trưng tự động và phát hiện các mẫu sinh học có ý nghĩa. Cụ thể:

+ Khả năng phát hiện mẫu cục bộ (local patterns): Lớp tích chập trong CNN1D có khả năng tự động học và nhận diện các mẫu lặp cục bộ trong chuỗi protein, chẳng hạn như các motif chức năng hoặc vùng bảo tồn, vốn thường có liên quan trực tiếp đến hoạt tính sinh học và vị trí biến đổi sau dịch mã (PTM sites).

+ Tự động trích xuất đặc trưng: CNN1D loại bỏ nhu cầu thiết kế thủ công đặc trưng đầu vào, thay vào đó học trực tiếp các biểu diễn đặc trưng từ dữ liệu thô thông qua quá trình huấn luyện, giúp tăng tính khách quan và khả năng khái quát hóa của mô hình.

- Khả năng xử lý chuỗi dài: Việc tích hợp các lớp max pooling giúp giảm chiều dài của chuỗi đầu vào mà vẫn giữ lại các thông tin quan trọng nhất, nhờ đó mô hình có thể học hiệu quả ngay cả với những trình tự protein dài, đồng thời giảm chi phí tính toán.

+ Tổng quát hóa tốt trên dữ liệu chưa thấy: Mô hình CNN1D có khả năng học các mẫu biểu diễn giàu thông tin từ tập huấn luyện và áp dụng hiệu quả cho dữ liệu kiểm thử hoặc dữ liệu chưa từng gặp, từ đó nâng cao độ tin cậy và hiệu suất tổng thể trong các bài toán dự đoán PTM.

### *3.1.2 Mô hình LSTM, Bi-LSTM*

LSTM [43] ra đời năm 1997 là một loại mạng nơ-ron hồi quy (RNN) được thiết kế để xử lý dữ liệu chuỗi dài và lưu trữ thông tin qua nhiều bước tính toán. Khác với các mạng RNN truyền thống, LSTM sử dụng các cổng (gates) để kiểm soát thông tin được lưu trữ hoặc quên, giúp giảm thiểu vấn đề mất mát thông tin trong quá trình học.

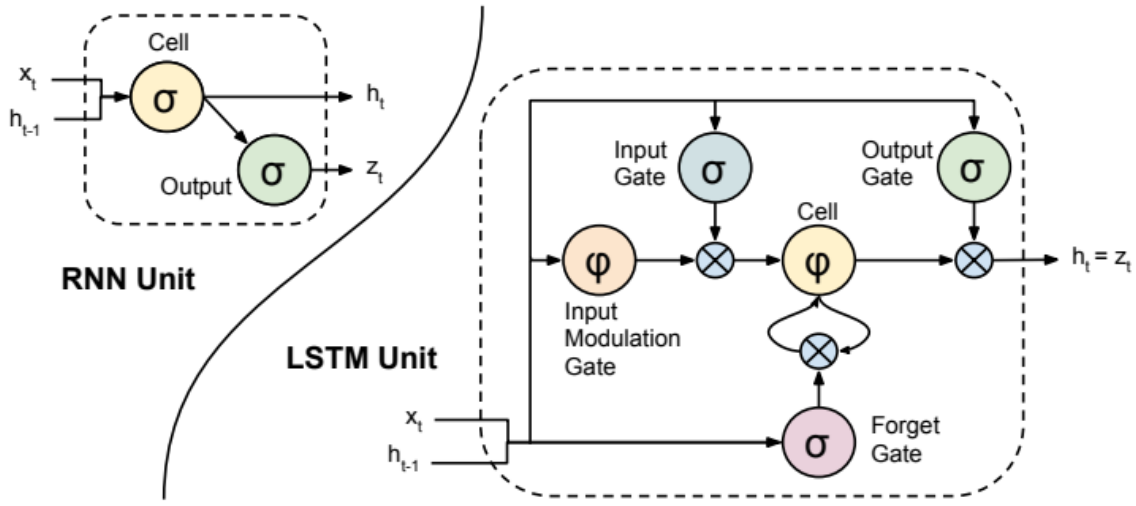
Kiến trúc của mạng LSTM bao gồm các cổng (gate) để kiểm soát việc thông tin được lưu trữ và truyền qua thời gian (Hình 3.2, Hình 3.3). Các cổng này bao gồm:

- Cổng quên (Forget gate): Quyết định thông tin nào trong cell state sẽ được quên đi hoàn toàn.

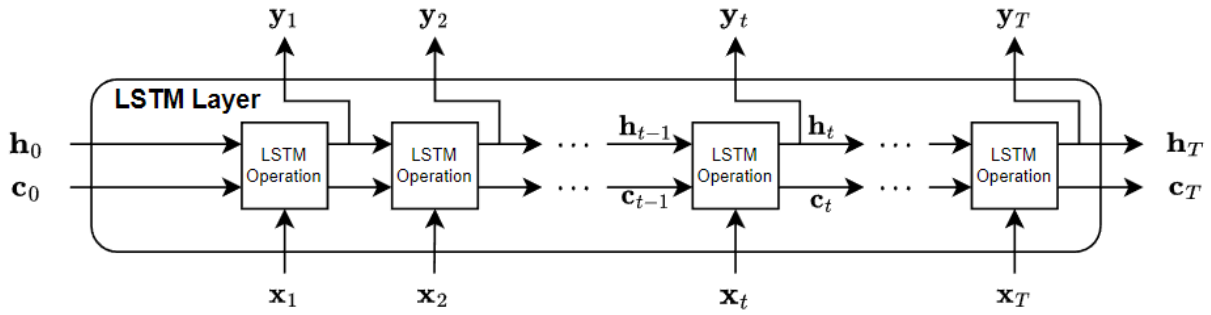
- Cổng đầu vào (Input gate): Quyết định thông tin mới nào sẽ được thêm vào cell state.

- Cổng đầu ra (Output gate): Quyết định thông tin nào từ cell state sẽ được sử dụng để tính toán đầu ra.

Điều này giúp LSTM có khả năng học và lưu trữ thông tin dài hạn một cách hiệu quả, phù hợp cho các bài toán có tính chuỗi dài và phức tạp.




**Hình 3.2 Sơ đồ cơ bản RNN cell (bên trái) và một LSTM cell (bên phải) [26]**

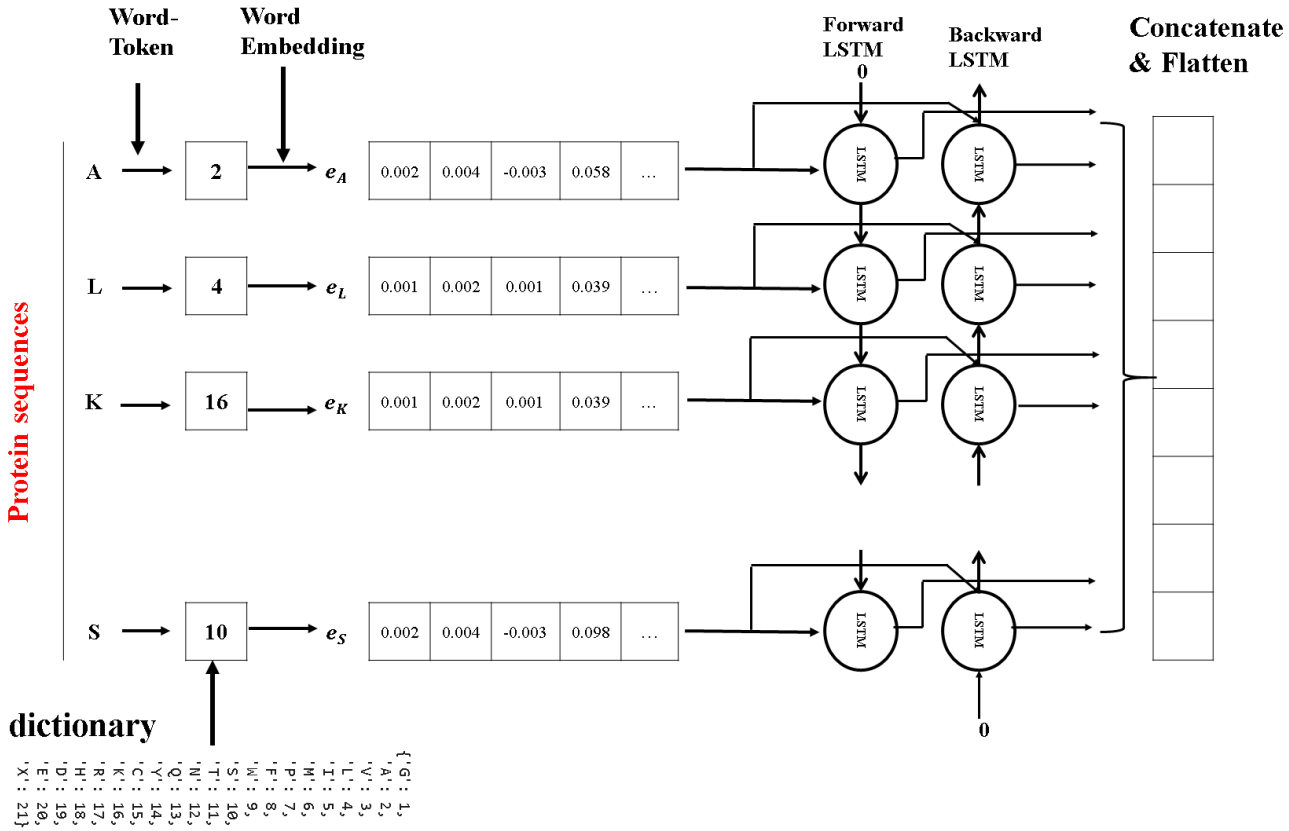



**Hình 3.3 Kiến trúc mạng LSTM [1]**

Bi-LSTM là một phiên bản mở rộng của LSTM, trong đó dữ liệu chuỗi được xử lý từ cả hai chiều: từ trước ra sau và từ sau ra trước. Điều này giúp Bi-LSTM học được các thông tin từ cả hai phía của dữ liệu.

Cấu trúc của Bi-LSTM: Bi-LSTM kết hợp hai mô hình LSTM: một mô hình đọc chuỗi từ đầu đến cuối (forward LSTM) và một mô hình đọc chuỗi từ cuối đến đầu (backward LSTM). Kết quả từ cả hai mô hình được kết hợp để cải thiện khả năng nhận diện các mẫu phức tạp trong chuỗi protein.

Ứng dụng trong dự đoán vị trí PTM: Bi-LSTM có ưu thế trong việc học các mối quan hệ phức tạp và dài hạn trong chuỗi protein, đồng thời giúp mô hình nhận diện các dấu hiệu PTM có thể xuất hiện ở các vị trí xa trong chuỗi.




**Hình 3.4 Bi-LSTM học chuỗi protein (1-gram) đề xuất**

**Dưới đây là mô tả quy trình học đặc trưng từ chuỗi protein của mạng LSTM, Bi-LSTM (Hình 3.4):**

Bước 1: Biểu diễn chuỗi protein dưới dạng Word-Token. Mỗi axit amin trong chuỗi protein được ánh xạ thành một Word-Token theo một từ điển mã hóa (dictionary.

Bước 2: Nhúng chuỗi protein bằng Word Embedding

Bước 3: LSTM xử lý chuỗi protein

- Các véc tơ embedding được đưa vào mạng LSTM/Bi-LSTM để học thông tin:

LSTM có khả năng nhớ các mối quan hệ xa trong chuỗi, ví dụ, mối quan hệ giữa các axit amin cách nhau nhiều vị trí trong chuỗi protein.

Bước 4: Ghép nối (Concatenate) và làm phẳng (Flatten)

- Đầu ra của forward và backward LSTM tại mỗi bước thời gian được kết hợp (concatenate) lại.

- Sau đó, toàn bộ chuỗi được làm phẳng (Flatten) để chuẩn bị đưa vào tầng đầu ra hoặc các tầng tiếp theo (như Dense).

Lợi ích của LSTM trong học chuỗi protein:

LSTM mang lại nhiều lợi ích trong việc học chuỗi protein nhờ khả năng ghi nhớ thông tin theo thời gian. Đầu tiên, mô hình này có thể nắm bắt quan hệ tuần tự dài, giúp ghi nhớ thông tin trong toàn bộ chuỗi, đặc biệt hữu ích khi xử lý các trình tự protein dài. Bên cạnh đó, LSTM học ngữ cảnh hiệu quả bằng cách sử dụng bộ nhớ để lưu trữ thông tin từ các bước trước đó, cho phép mô hình hiểu ý nghĩa của một axit amin trong bối cảnh toàn chuỗi. Ngoài ra, LSTM có tính linh hoạt cao, có thể áp dụng cho nhiều loại bài toán sinh học khác nhau, từ phân loại protein đến dự đoán các vị trí PTM như succinyl hóa hoặc ubiquitin hóa.

Bi-LSTM mở rộng lợi thế của LSTM bằng cách xử lý thông tin theo cả hai chiều, giúp mô hình nắm bắt ngữ cảnh toàn diện, không chỉ từ các axit amin trước đó mà còn từ các axit amin phía sau. Điều này làm cho Bi-LSTM có hiệu suất cao hơn so với LSTM đơn hướng, đặc biệt trong các bài toán yêu cầu hiểu sâu về mối quan hệ giữa các axit amin trong chuỗi protein. Hơn nữa, Bi-LSTM tương thích tốt với dữ liệu sinh học, do trình tự protein thường mang tính liên kết chặt chẽ theo cả hai chiều. Nhờ vậy, mô hình này đặc biệt phù hợp cho các bài toán phức tạp, nơi việc khai thác mối quan hệ dài hạn và đa chiều đóng vai trò quan trọng trong việc nâng cao độ chính xác dự đoán.

## 3.2 Mô hình dự đoán SUMOylation dựa trên kiến trúc học sâu lai (CNN1D_LSTM) và kỹ thuật xử lý ngôn ngữ tự nhiên đề xuất

### *3.2.1 Tên viết tắt*

Trong nghiên cứu này, NCS đề xuất một mô hình học sâu lai kết hợp giữa mạng tích chập một chiều (CNN1D) và mạng ghi nhớ dài ngắn hạn (LSTM), dựa trên kỹ thuật mã hoá từ Word2Vec để biểu diễn chuỗi protein. Để thuận tiện cho việc trích dẫn và trình bày trong các phần tiếp theo, mô hình được đặt tên là **CLW_SUMO** viết tắt của "a hybrid deep learning model combining **C**NN1D and **L**STM architectures with **W**ord2Vec for **SUMO**ylation prediction".

### *3.2.2 Dữ liệu thực nghiệm*

Mô hình học sâu lai kết hợp giữa mạng CNN1D và LSTM, gọi tắt là CLW_SUMO, được xây dựng nhằm dự đoán các vị trí SUMOylation trong protein. Trong chương 2, mô hình đã được thử nghiệm trên một tập dữ liệu SUMOylation còn hạn chế, chưa đủ quy mô để khai thác đầy đủ tiềm năng của các mô hình học sâu vốn yêu cầu lượng dữ liệu lớn để huấn luyện hiệu quả. Nhằm khắc phục hạn chế này, trong chương này, NCS đã mở rộng tập dữ liệu bằng cách thu thập thêm các vị trí SUMOylation đã được xác minh thực nghiệm từ nhiều cơ sở dữ liệu và nguồn tài liệu công bố gần đây, bao gồm: dbPTM3.0 (phiên bản 2024), JASSA [9], SUMOhydro [16], pSumo-CD [49], HseSUMO [97], GPS-SUMO [127], ResSUMO [129]. Tổng cộng, một tập dữ liệu gồm

3639 protein với 8838 vị trí SUMO đã được thu thập. Sau khi thực hiện một số bước kỹ thuật để loại bỏ các protein trùng lặp, tập dữ liệu cuối cùng không trùng lặp chứa 3000 protein duy nhất. Để chuẩn bị cho thử nghiệm độc lập, NCS chọn ngẫu nhiên 1/3 protein từ tập dữ liệu không trùng lặp để làm tập dữ liệu thử nghiệm độc lập. Sau đó, dữ liệu còn lại được coi là tập dữ liệu đào tạo.

**Bảng 3.1 Dữ liệu SUMOylation sites thu thập**

| **Nguồn thu thập** | **SUMOylated proteins** | **SUMO-sites** |
|---|---|---|
| dbPTM 3.0 (01/2024) | 1432 | 5191 |
| SUMOsp (01/2024) | 197 | 332 |
| seeSUMO | 247 | 377 |
| GPS-SUMO 2.0 (01/2024) | 510 | 912 |
| JASSA | 505 | 877 |
| pSUMO-CD | 510 | 755 |
| SUMOhydro | 238 | 394 |
| **Tổng dữ liệu thu thập** | **3639** | **8838** |
| **Dữ liệu thu thập loại bỏ trùng lặp** | **3000** | **7982** |
| **Dữ liệu huấn luyện** | **2000** | **5890** |
| **Dữ liệu kiểm tra** | **1000** | **2092** |

Các bước tạo bộ dữ liệu thực hiện với cửa sổ trượt như chương 2, và cũng tiếp tục áp dụng loại bỏ trùng lặp với công cụ CD_hit 40%. Cuối cùng bộ dữ liệu được sử dụng trong huấn luyện mô hình trong Bảng 3.2 dưới đây.

**Bảng 3.2 Dữ liệu sử dụng trong nghiên cứu**

| | **SL mẫu dương tính** | **SL mẫu âm tính** |
|---|---|---|
| Tập dữ liệu huấn luyện | 4985 | 9967 |
| Tập dữ liệu kiểm tra | 1245 | 2870 |

### *3.2.3 Phương pháp mã hoá và trích chọn đặc trưng*

Việc lựa chọn phương pháp mã hóa chuỗi protein là một bước nền tảng, ảnh hưởng trực tiếp đến hiệu quả của mô hình học sâu. Trong nghiên cứu này, NCS đề xuất sử dụng ma trận nhúng (embedding matrix) được tạo ra từ mô hình Word2Vec, một quyết định dựa trên sự cân nhắc kỹ lưỡng giữa khả năng biểu diễn ngữ nghĩa và tài nguyên tính toán.

Thứ nhất, Word2Vec thể hiện hiệu quả vượt trội trong việc nắm bắt ngữ cảnh cục bộ của chuỗi protein. Không giống như các phương pháp mã hóa độc lập như One-hot, Word2Vec học cách biểu diễn từng axit amin hoặc n-gram thành một vector dày đặc (dense vector) dựa trên các axit amin lân cận. Điều này cho phép các n-gram có ý nghĩa sinh học hoặc thường xuất hiện trong cùng một ngữ cảnh được ánh xạ gần nhau trong không gian vector. Khả năng này đặc biệt quan trọng trong bài toán dự đoán PTM, nơi các motif hoặc các chuỗi axit amin ngắn thường đóng vai trò then chốt trong việc xác định vị trí sửa đổi.

Thứ hai, việc lựa chọn Word2Vec là một giải pháp tối ưu về mặt cân bằng giữa hiệu suất và tài nguyên. Trong khi các mô hình ngôn ngữ lớn như BERT hay ESM có khả năng học các biểu diễn ngữ cảnh dài hạn và phức tạp hơn, chúng lại đòi hỏi một lượng tài nguyên tính toán khổng lồ và thời gian huấn luyện dài. Điều này có thể không phù hợp với giới hạn về tài nguyên của một nghiên cứu cụ thể. Ngược lại, Word2Vec có thể được huấn luyện nhanh chóng trên một kho dữ liệu protein lớn và không có nhãn. Ma trận nhúng thu được có thể được sử dụng làm trọng số khởi tạo (initial weights) cho lớp embedding của các mô hình học sâu sau này. Phương pháp này không chỉ tiết kiệm thời gian và tài nguyên mà còn giúp mô hình bắt đầu quá trình huấn luyện với một "kiến thức" nền tảng về ngôn ngữ protein.

Thứ ba, Word2Vec cho phép tích hợp linh hoạt với các kiến trúc mạng học sâu khác nhau. Cụ thể, ma trận nhúng tĩnh được tạo ra có thể dễ dàng làm đầu vào cho các mô hình như Mạng tích chập 1 chiều (CNN1D) và Mạng nơ-ron hồi quy (LSTM).

Để tạo ma trận embedding bằng Word2Vec, cần thực hiện hai giai đoạn chính: huấn luyện ma trận embedding và tích hợp vào mô hình học sâu.

**Giai đoạn 1. Huấn luyện ma trận Word2Vec**

Word2Vec sẽ học các vector biểu diễn cho từng "từ" (n-gram) trong kho dữ liệu protein. Các vector này được học sao cho những từ có ngữ cảnh tương tự nhau sẽ có vector biểu diễn gần nhau trong không gian vector.

Bước 1: Chuẩn bị dữ liệu huấn luyện. Tập hợp một lượng lớn chuỗi protein không có nhãn. Đây sẽ là "corpus" (tập văn bản) để huấn luyện Word2Vec.

Bước 2: Tokenization. Sử dụng kỹ thuật n-gram (ví dụ 3-gram) để tách các chuỗi

protein thành danh sách các "từ". Ví dụ, chuỗi "MKTLV" sẽ trở thành [’MKT’, ’KTL’, ’TLV’]. Bộ từ điển n-gram có thể được tóm tắt trong Bảng 3.3:

**Bảng 3.3 Kích thước từ điển n-gram và diễn giải**

| n-gram | Cỡ của từ điển | Diễn giải |
|---|---|---|
| 1-gram | 21 | Mỗi axit amin là một từ |
| 2-gram | 441 | Mỗi cặp 2 axit amin liền kề được coi là 1 từ |
| 3-gram | 9261 | Mỗi 3 axit amin liền kề được coi là 1 từ |

Bước 3: Huấn luyện mô hình Word2Vec. Sử dụng thư viện như Gensim trong Python để huấn luyện mô hình Word2Vec trên dữ liệu đã được token hóa. Với các tham số sau:

- size: Kích thước của vector embedding (ví dụ: 300). Kích thước này sẽ là chiều của vector biểu diễn cho mỗi n-gram.

- window: Kích thước của "cửa sổ ngữ cảnh". Tham số này xác định số lượng n-gram lân cận mà mô hình sẽ xem xét khi học. Trong nghiên cứu này NCS chọn window = 5.

- min_count: Ngưỡng tần suất tối thiểu của một n-gram để được đưa vào từ điển.

Bước 4: Tạo ma trận embedding. Sau khi huấn luyện, thu được một ma trận có kích thước là (số lượng từ trong từ điển) x (kích thước vector). Mỗi hàng của ma trận này là vector embedding của một n-gram tương ứng. Ma trận này sẽ được lưu lại để sử dụng cho các mô hình sau.

**Giai đoạn 2. Tích hợp ma trận Word2Vec vào mô hình học sâu**

Khi xây dựng mô hình CNN1D hoặc LSTM cho bài toán dự đoán PTM, ma trận embedding đã được huấn luyện sẽ được sử dụng để khởi tạo lớp embedding.

Bước 1: Tải ma trận embedding.

Tải ma trận Word2Vec đã huấn luyện ở bước trên.

Bước 2: Xây dựng lớp Embedding.

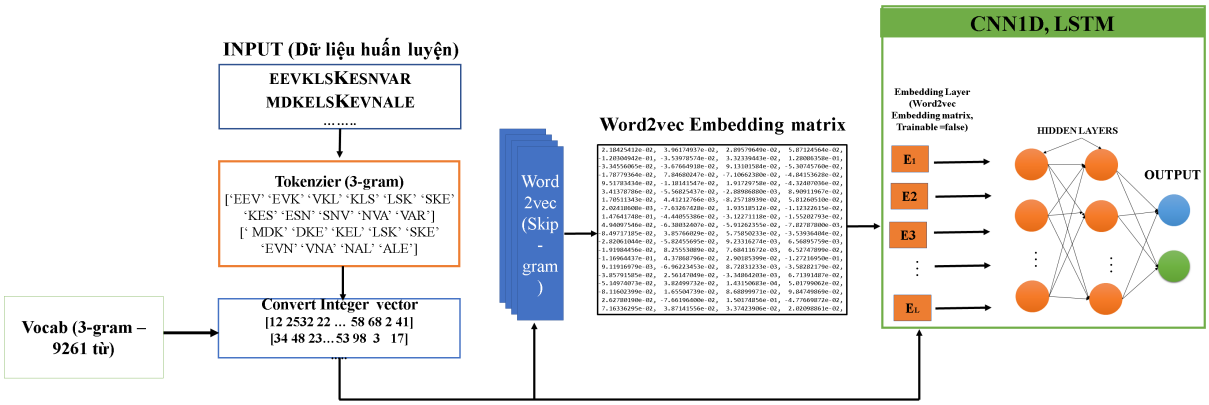
Trong mô hình học sâu lớp Embedding sẽ được khởi tạo với tham số weights là ma trận Word2Vec đã tải, đặt tham số trainable của lớp này thành False. Điều này có nghĩa là các trọng số của lớp embedding sẽ được giữ cố định trong quá trình huấn luyện mô hình, không bị cập nhật. Phương pháp này thường được gọi là transfer learning (học chuyển giao).

Bước 3: Kết nối với các lớp mạng.

Đầu ra của lớp Embedding sẽ được đưa vào các lớp tiếp theo của mô hình, chẳng hạn như lớp CNN1D hoặc LSTM (các mô hình cơ sở trong mô hình học sâu lai).

Trong kiến trúc đề xuất của NCS, CNN1D sử dụng các bộ lọc để phát hiện các mẫu cục bộ từ các vector đã được mã hóa, trong khi LSTM tận dụng khả năng của nó để nắm bắt các mối quan hệ phụ thuộc dài hạn. Sự kết hợp này tận dụng được điểm mạnh của cả hai phương pháp: khả năng mã hóa ngữ nghĩa hiệu quả của Word2Vec và khả năng học đặc trưng sâu của CNN/LSTM, tạo nên một mô hình mạnh mẽ và hiệu quả cho bài toán dự đoán PTM.

Để minh họa cho phương pháp mã hóa dữ liệu đã đề xuất, Hình 3.5 trình bày một cách tổng quan quy trình từ việc tiếp nhận dữ liệu đầu vào là chuỗi protein đến khi tạo ra các vector số biểu diễn, sẵn sàng cho các mô hình học sâu như CNN1D và LSTM xử lý




**Hình 3.5 Quy trình mã hóa dữ liệu đề xuất, bao gồm các bước: (1) tokenization chuỗi protein bằng n-gram, (2) chuyển đổi token thành chỉ số số, và (3) sử dụng ma trận nhúng Word2Vec làm đầu vào cho lớp embedding trong CNN1D và LSTM.**

### *3.2.4 Kiến trúc mô hình học sâu lai (CNN_LSTM) dự đoán vị trí SUMOylation*

Mô hình học sâu lai CLW_SUMO được thiết kế dựa trên sự kết hợp giữa mạng tích chập một chiều (CNN1D) và mạng bộ nhớ dài ngắn hạn (LSTM). Ý tưởng chính của sự lai ghép này là khai thác đồng thời hai ưu điểm bổ trợ lẫn nhau: (i) CNN1D có khả năng phát hiện các motif cục bộ và mẫu ngắn trong chuỗi axit amin – những tín hiệu thường liên quan trực tiếp đến vị trí SUMOylation; (ii) LSTM lại nổi trội trong việc nắm bắt quan hệ tuần tự và phụ thuộc dài hạn, giúp mô hình hiểu được ngữ cảnh xa giữa các vị trí trong chuỗi protein. Sự kết hợp này vì vậy vừa tận dụng được sức mạnh biểu diễn đặc trưng cục bộ, vừa duy trì khả năng học thông tin toàn cục, thay vì chỉ thiên lệch về một phía. Đây cũng là một hướng nghiên cứu phù hợp trong bối cảnh dữ liệu hạn chế, khi việc áp dụng các kiến trúc nặng hơn như Transformer/PLMs thường đòi hỏi tài nguyên

lớn và có nguy cơ quá khớp.

**Quy trình xây dựng mô hình học sâu lai CLW_SUMO:**

**Bước 1. Thu thập và tiền xử lý dữ liệu:** Dữ liệu peptide ban đầu được thu thập từ các cơ sở dữ liệu chuyên biệt. Để đảm bảo tính độc lập và khách quan của các mẫu, dữ liệu được lọc dư thừa bằng công cụ CD-HIT với ngưỡng tương đồng 40%. Điều này giúp loại bỏ các chuỗi peptide có độ tương tự cao, từ đó ngăn ngừa tình trạng quá khớp và đảm bảo mô hình có thể khái quát hóa tốt hơn trên dữ liệu mới. Sau đó, tập dữ liệu được chia thành hai phần riêng biệt: tập huấn luyện và tập kiểm thử. Các chuỗi peptide đầu vào được xử lý thêm bằng phương pháp cửa sổ trượt (windowing) để tạo ra các mẫu dữ liệu có độ dài cố định, sẵn sàng cho việc mã hóa.

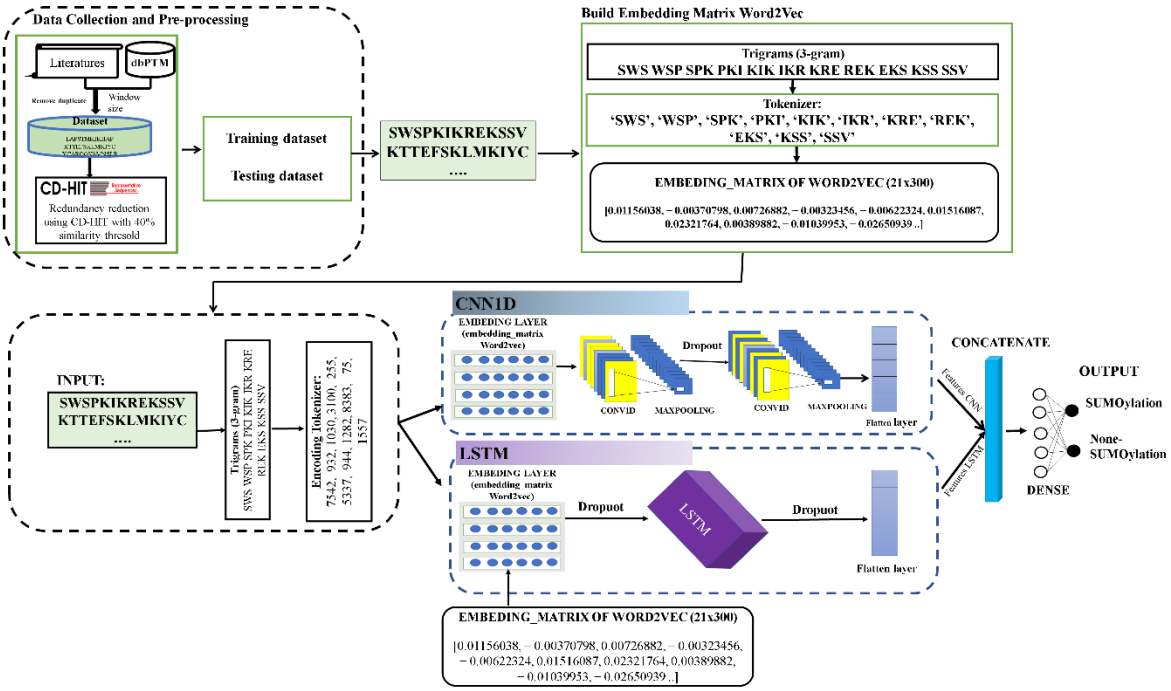
**Bước 2. Mã hóa và trích chọn đặc trưng:** Các chuỗi peptide sau khi được tiền xử lý sẽ được chuyển đổi thành định dạng số. Quy trình này sử dụng phương pháp mã hóa và trích chọn đặc trưng được đề xuất trong mục 3.2.3, với chiến lược n-gram (trigram). Các n-gram này được xem như các "từ" trong ngôn ngữ protein và được mã hóa thành các vector số bằng kỹ thuật Word2Vec. Quá trình này tạo ra một ma trận embedding, nơi mỗi vector đại diện cho một trigram và chứa thông tin ngữ nghĩa cục bộ của nó.

**Bước 3. Kiến trúc lai CNN-LSTM:** Các vector embedding đầu ra được đưa vào kiến trúc học sâu lai, được chia thành hai nhánh xử lý song song:

- Nhánh CNN1D: Nhánh này gồm các lớp tích chập một chiều (1D) và các lớp max pooling. CNN1D có nhiệm vụ quét qua các vector embedding để tự động phát hiện các motif cục bộ và các mẫu ngắn quan trọng trong chuỗi axit amin, là những đặc trưng thường liên quan trực tiếp đến vị trí SUMOylation.

- Nhánh LSTM: Nhánh này sử dụng mạng bộ nhớ dài ngắn hạn (LSTM) để xử lý dữ liệu tuần tự. LSTM có khả năng nắm bắt các mối quan hệ phụ thuộc dài hạn và ngữ cảnh toàn cục của chuỗi protein, điều mà CNN1D khó thực hiện.

**Bước 4. Hợp nhất và phân loại:** Đặc trưng được trích xuất từ hai nhánh CNN1D và LSTM được kết hợp lại bằng cách ghép nối (concatenate) để tạo ra một vector đặc trưng tổng hợp. Vector này sau đó được truyền qua một hoặc nhiều lớp kết nối đầy đủ (fully connected layers) và cuối cùng là lớp đầu ra với hàm kích hoạt sigmoid để phân loại peptide là vị trí SUMOylation hoặc không phải SUMOylation.




**Hình 3.6 Mô hình học sâu lai dự đoán SUMOylation (CLW_SUMO) đề xuất**

Dưới đây là thuật toán giả mã của mô hình CLW_SUMO đề xuất (Algorithm 3.3 (phần 1), Algorithm 3.3 (phần 2)). Bảng 3.4 trình bày chi tiết các lớp (layer) và số tham số của mô hình CLW_SUMO.

**Algorithm 3.3** Thuật toán CLW_SUMO: Mã hóa chuỗi protein bằng mô hình Word2Vec (3-gram) (Phần 1)

**Đầu vào:** $\mathscr{D}_{\text{raw}}$: Tập dữ liệu ban đầu; $\mathscr{D}_{\text{train}}$: Tập huấn luyện; $\mathscr{D}_{\text{val}}$: Tập xác thực; $\mathscr{D}_{\text{test}}$: Tập kiểm tra độc lập

**Đầu ra:** $\theta^*$: tham số của mô hình; $\widehat{\mathbf{Y}}_{\text{test}}$: Nhãn dự đoán cho $\mathscr{D}_{\text{test}}$

**Bước 1: Tạo từ điển 3-gram**:
AA_list $\leftarrow$ {`G, A, V, ..., X`} ▷ 20 axit amin + ký hiệu giả 'X'
AA_dict $\leftarrow \emptyset$, num $\leftarrow 1$
**for all** $i \in$ AA_list **do**
  **for all** $j \in$ AA_list **do**
    **for all** $k \in$ AA_list **do**
      trigram $\leftarrow i \| j \| k$
      AA_dict[trigram] $\leftarrow$ num
      num $\leftarrow$ num $+ 1$
    **end for**
  **end for**
**end for**
**Bước 2: Tiền xử lý dữ liệu**:
word_index $\leftarrow$ AA_vocab()
$\mathscr{X}_{\text{train\_ngram}} \leftarrow$ tokenize_3gram($X^{\text{raw}}_{\text{train}}$)
$\mathscr{D}_{\text{train}} \leftarrow$ map_to_index($\mathscr{X}_{\text{train\_ngram}}$, word_index)
**Bước 3: Huấn luyện Word2Vec và tạo ma trận nhúng**:
Embedding_dim $\leftarrow 300$
vocab_size $\leftarrow |$word_index$|$
w2v_model $\leftarrow$ Word2Vec($\mathscr{X}_{\text{train\_ngram}}$, Embedding_dim)
Embedding_matrix $\leftarrow$ zeros(vocab_size, Embedding_dim)
**for all** word, $i \in$ word_index **do**
  **if** word $\in$ w2v_model.wv **then**
    Embedding_matrix[$i$] $\leftarrow$ w2v_model.wv[word]
  **end if**
**end for**
$\mathbf{W}_{\text{w2v}} \leftarrow$ Embedding_matrix

**Algorithm 3.3** Thuật toán CLW_SUMO (tiếp theo): Huấn luyện mô hình và dự đoán

**Bước 4: Huấn luyện mô hình**:
**for** $e = 1$ **to** $N_{\text{epoch}}$ **do**
  **for all** mini-batch $(\mathbf{X}, \mathbf{y}) \subset \mathscr{D}_{\text{train}}$ **do**
    **Nhánh CNN1D**:
$$\mathbf{X}_{\text{CNN1D}} \leftarrow \text{Flatten}\big(\text{AvgPool}\big(\text{GRU}\big(\text{MaxPool}_2\big(\text{Conv1D}_2\big(\text{Dropout}\big(\text{MaxPool}_1\big(\text{Conv1D}_1\big(\text{Embedding}(\mathbf{X}, \mathbf{W}_{\text{w2v}})\big)\big)\big)\big)\big)\big)\big)\big)$$
    **Nhánh LSTM**:
      $\mathbf{X}_{\text{LSTM}} \leftarrow \text{Flatten}(\text{LSTM}(\text{Dropout}(\text{Embedding}(\mathbf{X}, \mathbf{W}_{\text{w2v}}))))$
    **Kết hợp đặc trưng của CNN1D và LSTM**:
      $\mathbf{X}' \leftarrow \text{Concatnate}(\mathbf{X}_{\text{CNN1D}}, \mathbf{X}_{\text{LSTM}})$
  **Phân lớp**:
    $\mathbf{F} \leftarrow \text{ReLU}(\text{Dense}_{64}(\mathbf{X}')) \in \mathbb{R}^{B \times 64}$
    $\widehat{\mathbf{y}} \leftarrow \sigma(\text{Dense}_1(\mathbf{F})) \in [0,1]^B$ *Trong đó $\sigma$ : *sigmoid*
  **Tính hàm mất mát và cập nhật tham số**:
    $\mathscr{L} \leftarrow \text{BCE}(\mathbf{y}, \widehat{\mathbf{y}})$ *Trong đó BCE là hàm mất mát Binary-CrossEntropy
    $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathscr{L}$
  **end for**
  **if** Không cải thiện trên $\mathscr{D}_{\text{val}}$ **then**
    **Dừng sớm (Early Stopping)**
  **end if**
**end for**
**Return** $\theta^*$
**Bước 5: Dự đoán trên tập kiểm tra**:
$\mathscr{D}_{\text{test}} \leftarrow \phi(\texttt{Tokenize}(\mathscr{D}_{\text{raw}}^{\text{test}}))$
$\widehat{\mathscr{Y}_{\text{test}}} \leftarrow \sigma(f(\mathscr{D}_{\text{test}}; \theta^*))$

### *3.2.5 Chiến lược và tham số huấn luyện mô hình*

Mô hình CLW_SUMO được huấn luyện trên Google Colab với GPU. NCS sử dụng thuật toán Adam để tối ưu hóa, với learning rate ban đầu là 0.001, batch size là 128 và huấn luyện trong 100 epochs. Hàm mất mát Binary Cross-Entropy được chọn vì đây là bài toán phân loại nhị phân. Để ngăn chặn hiện tượng overfitting, NCS áp dụng kỹ thuật early stopping, giúp dừng quá trình huấn luyện sớm khi mô hình không còn cải thiện.. Bảng 3.4 là chi tiết các tham số của mô hình CLW_SUMO.

**Bảng 3.4 Kiến trúc và tham số của mô hình CLW_SUMO**

| Lớp (Layer) | Kích thước đầu ra | Số tham số | Mô tả chức năng |
|---|---|---|---|
| InputLayer | (None, 11) | 0 | Nhận chuỗi axit amin đầu vào |
| **Nhánh CNN** | | | |
| Embedding | (None, 11, 300) | 2,778,600 | Biểu diễn Word2Vec cho nhánh CNN |
| Conv1D_1 | (None, 9, 64) | 57,664 | Trích xuất đặc trưng cục bộ bằng tích chập 1D |
| MaxPooling1D_1 | (None, 4, 64) | 0 | Giảm chiều, giữ đặc trưng quan trọng |
| Dropout | (None, 4, 64) | 0 | Giảm overfitting bằng cách ngẫu nhiên tắt nơ-ron |
| Conv1D_2 | (None, 2, 64) | 12,352 | Tích chập tầng sâu hơn để học đặc trưng nâng cao |
| MaxPooling1D_2 | (None, 1, 64) | 0 | Giảm chiều trong nhánh CNN |
| **Nhánh LSTM** | | | |
| Embedding | (None, 11, 300) | 2,778,600 | Biểu diễn Word2Vec cho nhánh LSTM |
| LSTM | (None, 128) | 219,648 | Học phụ thuộc dài hạn trong chuỗi |
| Dropout | (None, 11, 300) | 0 | Điều chuẩn trong quá trình huấn luyện LSTM |
| **Hợp nhất và đầu ra** | | | |
| Flatten | (None, 64) | 0 | Làm phẳng đặc trưng từ nhánh CNN |
| Flatten_1 | (None, 128) | 0 | Làm phẳng đặc trưng từ nhánh LSTM |
| Concatenate | (None, 192) | 0 | Ghép nối đặc trưng từ cả hai nhánh |
| Dense | (None, 64) | 12,352 | Lớp fully connected, học quan hệ phi tuyến |
| Dense | (None, 1) | 65 | Lớp đầu ra với sigmoid, dự đoán nhị phân |

**Tổng số tham số:** 5,859,281 (22.35 MB)
**Tham số huấn luyện:** 302,081 (1.15 MB)
**Tham số cố định:** 5,557,200 (21.20 MB)

Dựa trên Bảng 3.4, mô hình CLW_SUMO có tổng cộng 5,859,281 tham số, trong đó chỉ có 302,081 tham số sử dụng trong quá trình huấn luyện, trong khi 5,557,200 tham số còn lại được giữ cố định. Phần lớn tổng số tham số cố định ( 95%) thuộc về hai lớp Embedding. Điều này hoàn toàn phù hợp với phương pháp mã hóa đã đề xuất, khi ma trận Word2Vec đã được huấn luyện trước được sử dụng để khởi tạo lớp này với tham số trainable=False. Chiến lược này giúp mô hình tận dụng được kiến thức ngữ nghĩa từ một lượng lớn dữ liệu protein không nhãn mà không cần phải huấn luyện lại toàn bộ ma trận nhúng. Điều này giúp giảm đáng kể chi phí tính toán và thời gian huấn luyện.

Ngược lại, số lượng tham số huấn luyện được tập trung ở các lớp CNN1D, LSTM và các lớp Dense cuối cùng. Tổng số 302,081 tham số này là đủ để mô hình học các đặc trưng phức tạp từ dữ liệu đã được mã hóa. Cụ thể, các lớp này có nhiệm vụ học cách trích xuất các motif cục bộ (qua CNN1D) và các mối quan hệ tuần tự dài hạn (qua LSTM), sau đó kết hợp và phân loại các đặc trưng này thông qua các lớp Dense. Sự phân bổ tham số hợp lý này cho phép mô hình tinh chỉnh các trọng số cần thiết để giải quyết bài toán dự đoán PTM, đồng thời giữ cho mô hình có kích thước vừa phải để tránh hiện tượng quá khớp trên tập dữ liệu hạn chế.

### *3.2.6 Kết quả và thảo luận*

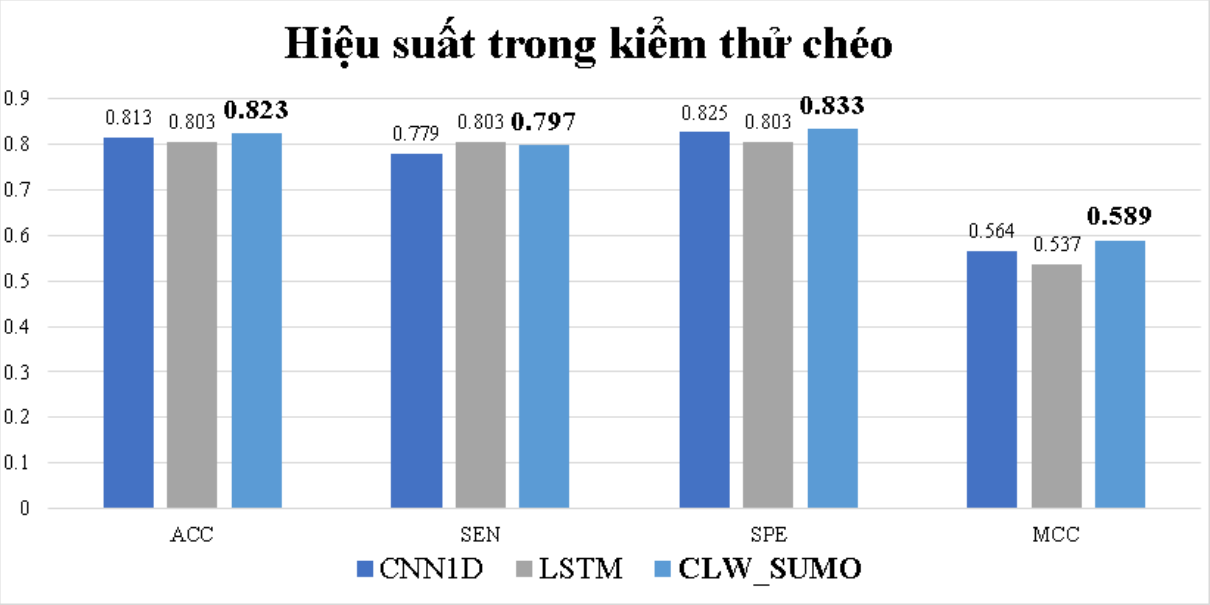
Kết quả thực nghiệm cho thấy mô hình học sâu lai CLW_SUMO đạt hiệu suất vượt trội so với các mô hình CNN1D và LSTM trong cả kiểm thử chéo và kiểm thử độc lập. Cụ thể, trong kiểm thử chéo (Hình 3.7), độ chính xác (ACC) của CLW_SUMO đạt 0.823, cao hơn 1%-2% so với các mô hình CNN1D (0.813) và LSTM (0.803). Đặc biệt, hệ số tương quan Matthews (MCC) – một chỉ số quan trọng phản ánh sự cân bằng giữa các độ nhậy và độ đặc hiệu của CLW_SUMO đạt 0.589, cao hơn đáng kể so với CNN1D (0.564) và LSTM (0.537). Trong kiểm thử độc lập (Hình 3.8), hiệu suất của CLW_SUMO tiếp tục khẳng định tính ổn định và khả năng khái quát hóa khi đạt ACC (0.900), cao hơn 1%-2% so với CNN1D (0.889) và LSTM (0.872). Đồng thời, MCC = 0.773, vượt trội hơn CNN1D (0.747) và LSTM (0.706), chứng minh khả năng dự đoán đáng tin cậy của mô hình trên dữ liệu chưa từng gặp.

Sự vượt trội của mô hình này đến từ cơ chế học bổ sung giữa CNN1D và LSTM trong kiến trúc lai. CNN1D đặc biệt mạnh trong việc phát hiện các motif cục bộ – những mẫu axit amin ngắn. Trong khi đó, LSTM lại có khả năng ghi nhớ và mô hình hóa các phụ thuộc dài hạn trong chuỗi, giúp nắm bắt các mối liên hệ xa giữa các vị trí axit amin. Sự kết hợp này cho phép mô hình vừa học được các đặc trưng cục bộ, vừa tận dụng được thông tin ngữ cảnh toàn cục, từ đó hình thành các đặc trưng phân biệt giàu ngữ nghĩa hơn cho bài toán phân loại.

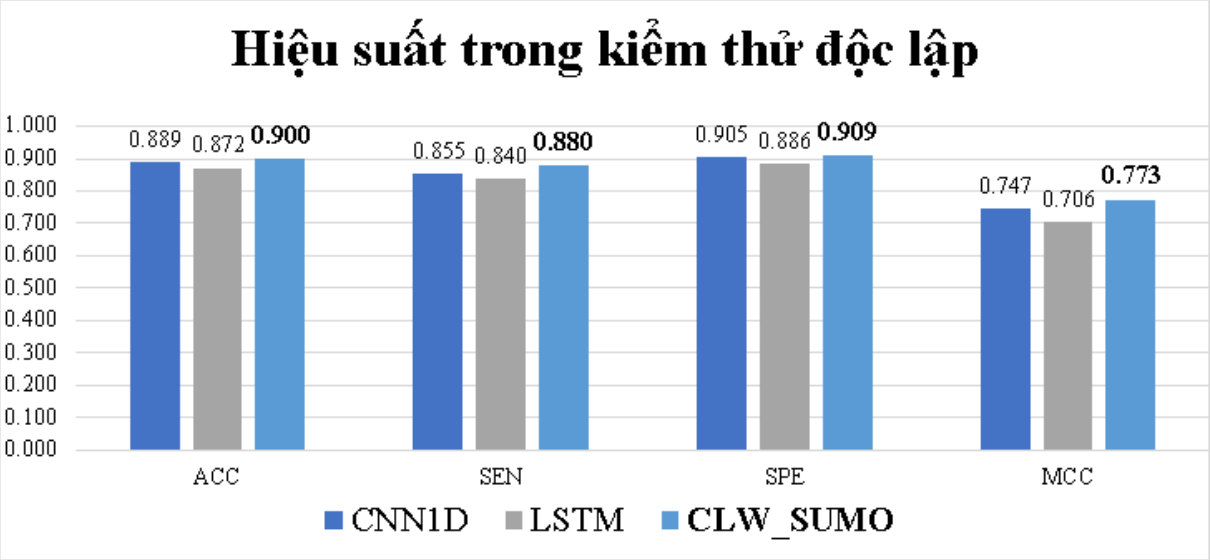
Bên cạnh kiến trúc lai, kỹ thuật mã hóa dữ liệu Word2Vec giúp chuyển đổi các axit

amin từ dạng ký tự rời rạc thành các véc tơ liên tục trong không gian đặc trưng có nghĩa thống kê. Nhờ đó, các mối quan hệ tiềm ẩn giữa các axit amin trong không gian trình tự được thể hiện rõ ràng hơn, giúp mô hình học được những biểu diễn phức tạp mà các kỹ thuật mã hóa đơn giản khác (one-hot encoding,TF-IDF) không thể nắm bắt được.

Đặc biệt, mã hóa Word2Vec trong kiến trúc học sâu lai đề xuất giúp mô hình học tự động đặc trưng từ dữ liệu thô, cho phép huấn luyện end-to-end, loại bỏ sự phụ thuộc vào các phương pháp trích chọn đặc trưng thủ công vốn mang tính cảm tính, thiếu tính khái quát và tốn tài nguyên lưu trữ. Điều này khắc phục hạn chế của các mô hình dự đoán SUMOylation khác [9, 10, 15, 16, 53, 97, 104, 129], sử dụng nhiều đặc trưng sinh học thủ công, đồng thời nâng cao hiệu quả và tính thực tiễn khi triển khai trên dữ liệu mới.



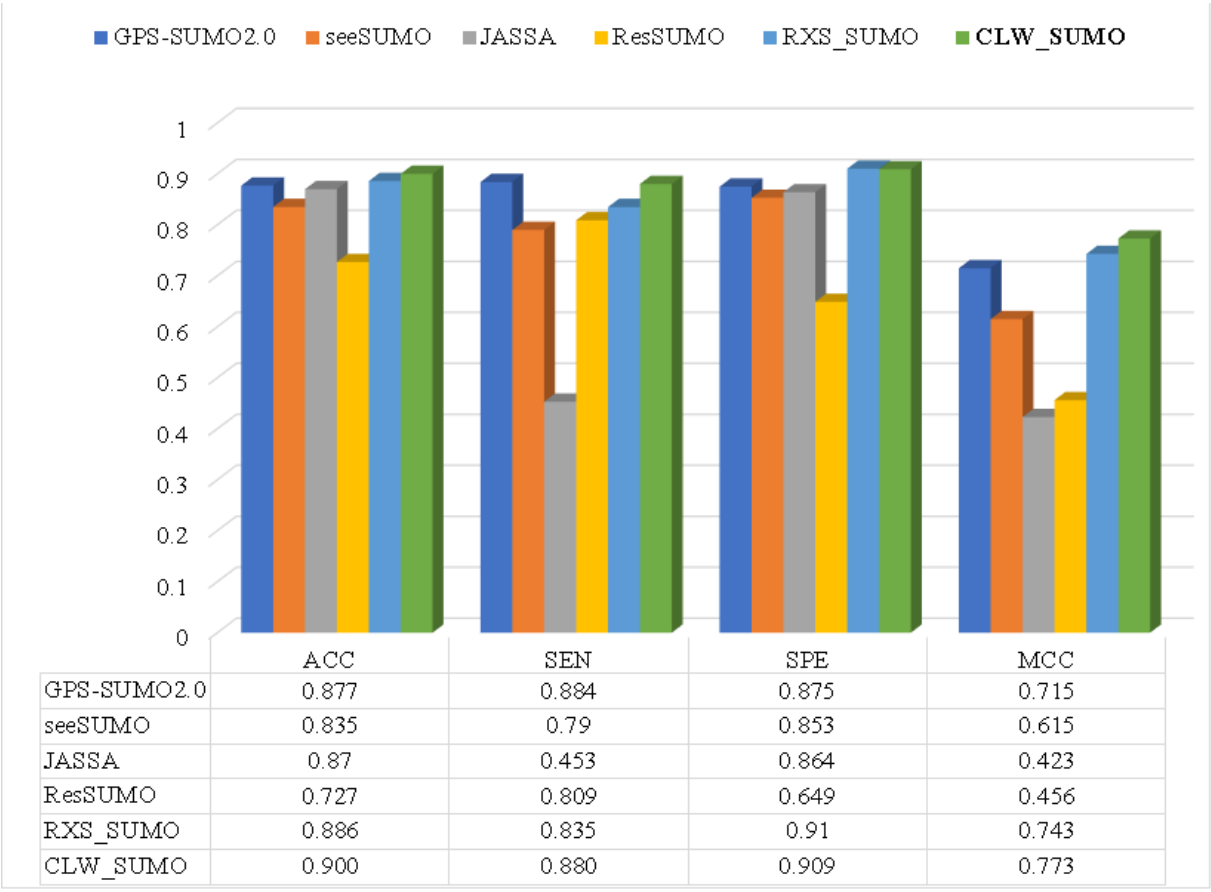

**Hình 3.7 Hiệu suất của mô hình trong kiểm thử chéo**




**Hình 3.8 Hiệu suất của mô hình trong kiểm thử độc lập**

### *3.2.7 So sánh với công cụ dự đoán khác*

Trong nghiên cứu này, việc so sánh được thực hiện với năm công cụ dự đoán SUMOylation đã được công bố và sử dụng rộng rãi, bao gồm GPS-SUMO2.0 [36], seeSUMO2.0 [97], JASSA, RXS_SUMO và ResSUMO [129]. Đây đều là những công cụ tiêu biểu đại diện cho nhiều hướng tiếp cận khác nhau: từ các phương pháp học máy truyền thống (JASSA), cho đến các mô hình học sâu hiện đại và tích hợp đặc trưng (GPS-SUMO2.0, ResSUMO).Việc lựa chọn các công cụ này đảm bảo tính khách quan trong đánh giá, đồng thời phản ánh đúng bức tranh nghiên cứu và xu thế phát triển của lĩnh vực tại thời điểm triển khai luận án.




| | ACC | SEN | SPE | MCC |
|---|---|---|---|---|
| GPS-SUMO2.0 | 0.877 | 0.884 | 0.875 | 0.715 |
| seeSUMO | 0.835 | 0.79 | 0.853 | 0.615 |
| JASSA | 0.87 | 0.453 | 0.864 | 0.423 |
| ResSUMO | 0.727 | 0.809 | 0.649 | 0.456 |
| RXS_SUMO | 0.886 | 0.835 | 0.91 | 0.743 |
| CLW_SUMO | 0.900 | 0.880 | 0.909 | 0.773 |

**Hình 3.9 So sánh hiệu suất mô hình CLW_SUMO với các công cụ dự đoán SUMOylaiton khác**

Kết quả hiển thị trong Hình 3.9, mô hình đề xuất được so sánh với năm công cụ dự đoán (GPS-SUMO2.0 [36], seeSUMO2.0 [97], RXS_SUMO, JASSA, ResSUMO [129]) và kết quả thực nghiệm cho thấy mô hình CLW_SUMO có hiệu suất cao hơn. Điều này chứng minh hiệu quả của mô hình học sâu lai kết hợp kỹ thuật mã hóa ngôn ngữ tự nhiên trong việc trích xuất và học các đặc trưng tiềm ẩn của SUMOylation.

## 3.3 Mô hình dự đoán Succinylation dựa trên kiến trúc học sâu lai (CNN1D_Bi-LSTM) và kỹ thuật xử lý ngôn ngữ tự nhiên đề xuất

### *3.3.1 Tên viết tắt*

Trong nghiên cứu này, NCS đề xuất một mô hình học sâu lai nhằm dự đoán biến đổi sau dịch mã (PTM) loại Succinylation. Mô hình được thiết kế dựa trên sự kết hợp giữa mạng tích chập một chiều (CNN1D), mạng bộ nhớ dài ngắn hạn hai chiều (Bi-LSTM), cùng với kỹ thuật embedding động để biểu diễn chuỗi axit amin. Mô hình được đặt tên là **CBILSuccSite** viết tắt từ "A hybrid deep learning model combining CNN1D and BiLSTM, enhanced with dynamic embedding for succinylation site prediction". Tên gọi này vừa phản ánh rõ thành phần cấu trúc chính của mô hình, vừa nhấn mạnh mục tiêu

ứng dụng là dự đoán vị trí Succinylation. Trong toàn bộ luận án, NCS sẽ sử dụng ký hiệu CBILSuccSite để chỉ mô hình dự đoán Succinylation đề xuất.

### *3.3.2 Dữ liệu thực nghiệm*

Để làm giầu tri thức của con người về nhiều loại PTM, trong nghiên cứu này, NCS đề xuất mô hình dự đoán vị trí Succinylation.

Succinyl hóa lần đầu tiên được quan sát thấy ở Escherichia coli vào năm 2004 và sau đó ở sinh vật nhân chuẩn, như một PTM phổ biến ở sinh vật nhân sơ và sinh vật nhân chuẩn. Succinylation là một dạng biến đổi sau dịch mã quan trọng, có liên quan đến nhiều bệnh lý như bệnh gan, tim mạch, phổi và rối loạn thần kinh [6, 123]. Quá trình này bao gồm việc gắn một nhóm succinyl (-CO-CH-CH-CO-) vào vị trí lysine của protein, với sự hỗ trợ của succinyl coenzyme A (succinyl-CoA) như một xúc tác phản ứng.

Lựa chọn bộ dữ liệu Succinylation trên một tập dữ liệu chuẩn phù hợp là một thách thức lớn, bởi vì hiện tại tồn tại quá nhiều bộ dữ liệu khác nhau từ các bài báo, để so sánh hiệu suất của các mô hình khó khăn. Do đó, trước khi đề xuất mô hình, NCS đã tiến hành khảo sát các công cụ học sâu hiện có dự đoán Succinylation, từ đó xác định một bộ dữ liệu tiêu chuẩn phù hợp. Bảng 3.5 tổng hợp các tập dữ liệu được sử dụng trong các nghiên cứu gần đây về dự đoán vị trí succinyl hoá, bao gồm thông tin về năm công bố, mức độ tương đồng của dữ liệu huấn luyện, kích thước tập dữ liệu và trạng thái hoạt động của công cụ này.

**Bảng 3.5 Bộ dữ liệu trong các nghiên cứu gần đây về dự đoán Succinylation**

| Công cụ | Tên bộ dữ liệu | SL protein | SL mẫu dương tính | SL mẫu âm tính |
|---|---|---|---|---|
| SuccinSite2.0 [38] | Dữ liệu huấn luyện | 2192 | 4750 | 9500 |
| | Dữ liệu kiểm tra | 124 | 254 | 2977 |
| GPSuc [39] | Dữ liệu huấn luyện | 2192 | 4750 | 9500 |
| | Dữ liệu kiểm tra | 124 | 254 | 2977 |
| HybridSucc [76] | Dữ liệu huấn luyện | 7415 | 21770 | 165071 |
| | Dữ liệu kiểm tra | 1415 | - | - |
| DeepSuccinylSite [106] | Dữ liệu huấn luyện (undersampling) | 2192 | 4755 | 4755 |
| | Dữ liệu kiểm tra | 124 | 254 | 254 |
| LMSuccSite [82] | Dữ liệu huấn luyện (undersampling) | 2192 | 4755 | 4755 |
| | Dữ liệu kiểm tra | 124 | 254 | 2977 |

Dựa trên các phân tích từ Bảng 3.5, NCS quyết định sử dụng tập dữ liệu từ hai nghiên cứu gần đây nhất, DeepSuccinylSite và LMSuccSite, nhằm đảm bảo tính nhất

quán trong so sánh và đánh giá mô hình. Cụ thể, tập dữ liệu huấn luyện bao gồm 4755 mẫu positive và 4755 mẫu negative. Cũng từ Bảng 3.5, NCS nhận thấy các công cụ dự đoán gần đây mỗi công cụ lại sử dụng các bộ test khác nhau. Do đó, NCS sử dụng hai bộ kiểm thử độc lập (Dữ liệu kiểm thử 1 và Dữ liệu kiểm thử 2) để so sánh hiệu suất của mô hình đề xuất với các công cụ trên và để kiểm tra mô hình đề xuất với hai bộ dữ liệu kiểm tra khác nhau một bộ dữ liệu cân bằng và một bộ dữ liệu có rất nhiều mẫu âm tính. Chi tiết bộ dữ liệu sử dụng trong nghiên cứu được trình bày trong Bảng 3.6.

**Bảng 3.6 Tập dữ liệu huấn luyện và kiểm tra sử dụng trong nghiên cứu**

| Tập dữ liệu | SL protein | SL mẫu dương tính | SL mẫu âm tính |
|---|---|---|---|
| Tập dữ liệu huấn luyện | 2192 | 4750 | 4750 |
| Tập dữ liệu kiểm thử 1 | 124 | 254 | 254 |
| Tập dữ liệu kiểm thử 2 | 124 | 254 | 2977 |

### *3.3.3 Phương pháp mã hóa và trích chọn đặc trưng (Embedding động)*

Trong nghiên cứu này, NCS đề xuất một phương pháp mã hóa dữ liệu khác biệt so với việc sử dụng ma trận nhúng Word2Vec đã được huấn luyện trước trong mô hình CLW_SUMO. Phương pháp này, được gọi là **Embedding động**, tích hợp trực tiếp lớp nhúng (*embedding layer*) vào kiến trúc mô hình học sâu (xem Hình 3.10). Thay vì sử dụng một ma trận embedding cố định, lớp này sẽ khởi tạo các vector biểu diễn cho từng axit amin một cách ngẫu nhiên và sau đó cập nhật chúng liên tục trong suốt quá trình huấn luyện. Cơ chế này cho phép mô hình tự động học được biểu diễn tối ưu nhất của các axit amin dựa trên ngữ cảnh và đặc trưng của bài toán cụ thể.

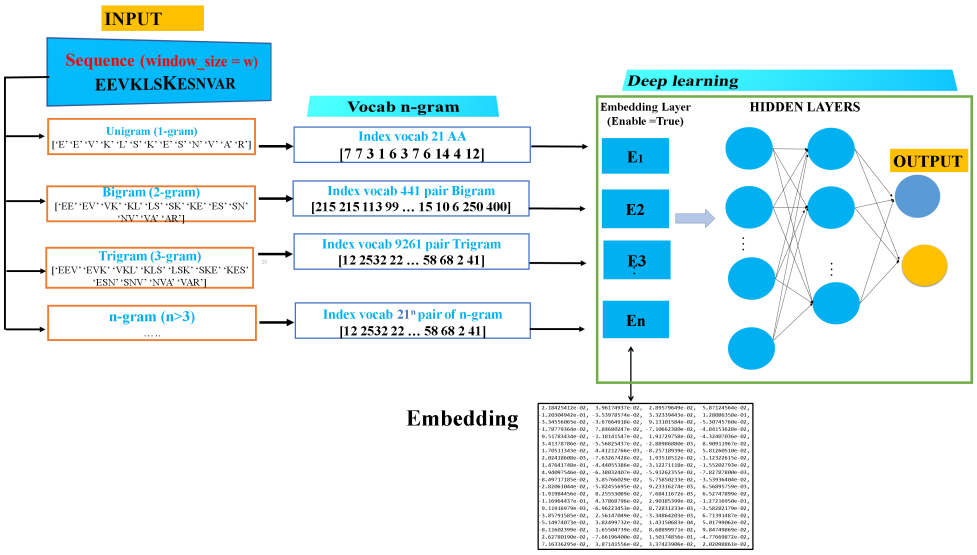
Quá trình mã hóa dữ liệu được thực hiện qua ba bước chính:

1. **Tokenizer chuỗi protein**: Tương tự như các phương pháp xử lý ngôn ngữ tự nhiên, các chuỗi protein (đoạn peptide) sẽ được phân tách thành các đoạn nhỏ hơn, gọi là các *token*. NCS sử dụng phương pháp **n-gram** để tách chuỗi protein thành một chuỗi các token $(x_1, x_2, \ldots, x_L)$, trong đó mỗi token có thể là một axit amin hoặc một chuỗi con của các axit amin.

2. **Chuyển đổi thành chỉ số số**: Mỗi token được ánh xạ tới một chỉ số số nguyên duy nhất dựa trên một bộ từ điển được xây dựng trước. Bước này chuyển đổi chuỗi các token $(x_1, x_2, \ldots, x_L)$ thành một vector số nguyên tương ứng $(b_{x_1}, b_{x_2}, \ldots, b_{x_L})$.

3. **Mã hóa Embedding động**: Vector số nguyên thu được ở bước 2 được đưa vào một

lớp *Embedding* trong mạng học sâu. Lớp này sẽ ánh xạ mỗi chỉ số số nguyên $b_{x_i}$ thành một vector nhúng thực $E_i \in \mathbb{R}^d$ (với $d$ là chiều không gian cố định của vector nhúng, trong nghiên cứu này NCS chọn d = 300). Điều khác biệt cốt lõi là các vector nhúng này không cố định. Trong quá trình huấn luyện, thông qua thuật toán lan truyền ngược, các giá trị của chúng được cập nhật và tinh chỉnh liên tục. Điều này cho phép mô hình học được biểu diễn tối ưu nhất cho từng token trong ngữ cảnh của chuỗi protein, phản ánh chính xác các đặc trưng cần thiết cho bài toán dự đoán *Succinylation*.




**Hình 3.10 Quy trình mã hóa dữ liệu protein bằng kỹ thuật embedding động**

### *3.3.4 Kiến trúc mô hình học sâu lai (CNN1D_Bi-LSTM) dự đoán vị trí Succinylation*

So với mô hình CLW_SUMO (mục 3.2), vốn sử dụng embedding tĩnh kết hợp CNN1D và LSTM đơn hướng, mô hình **CBILSuccSite** có hai cải tiến quan trọng: (i) sử dụng *embedding động* thay cho embedding tĩnh, cho phép biểu diễn chuỗi axit amin linh hoạt và giàu ngữ cảnh hơn; (ii) thay thế LSTM bằng Bi-LSTM để khai thác thông tin phụ thuộc theo cả hai chiều của chuỗi, từ đó nâng cao khả năng nhận diện chính xác vị trí Succinylation. Đây là sự khác biệt cốt lõi, thể hiện bước phát triển từ kiến trúc ở mục 3.2 sang mô hình ở mục 3.3.

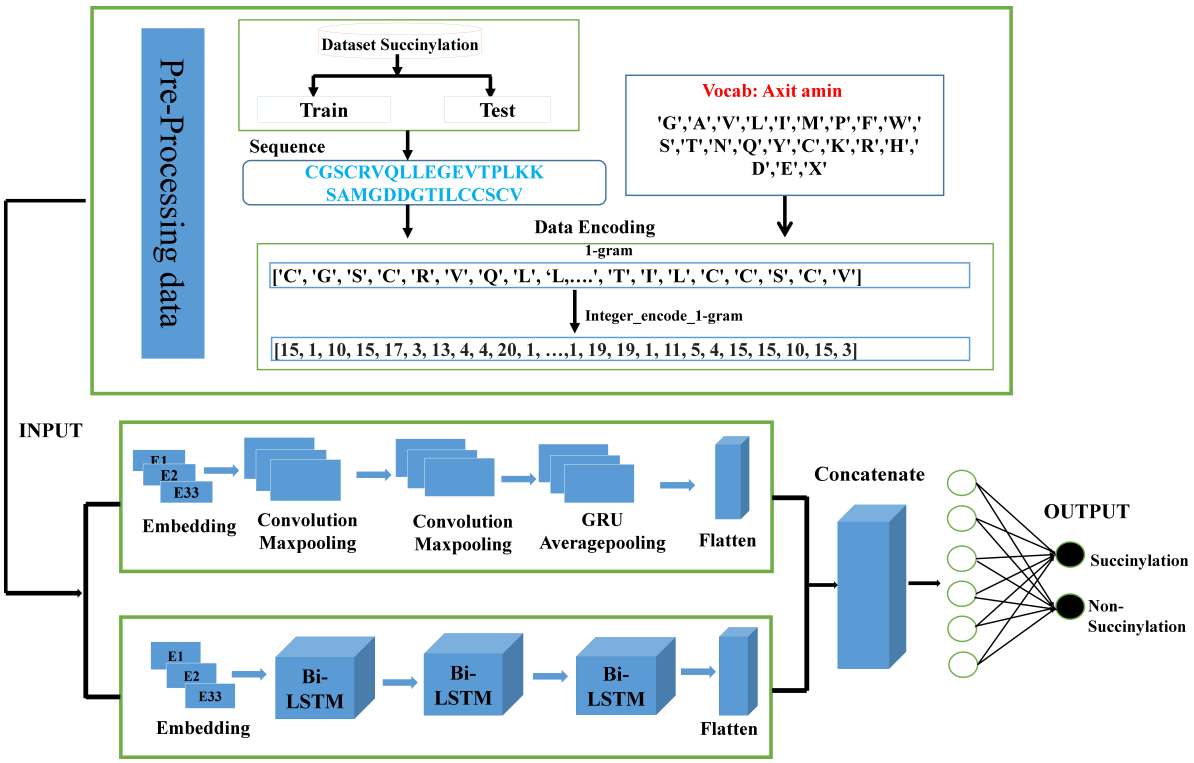
**Quy trình xây dựng mô hình đề xuất (Hình 3.11) gồm ba bước chính:**

(1) *Tiền xử lý và mã hóa chuỗi protein*: Chuỗi protein đầu vào được xử lý bằng phương pháp 1-gram để bảo toàn thứ tự axit amin. Các ký hiệu axit amin sau đó được ánh xạ sang số nguyên dựa trên từ điển n-gram. Chuỗi số nguyên này tiếp tục được đưa vào lớp embedding động để chuyển thành véc tơ 300 chiều, phục vụ cho việc học đặc trưng tự động.

Khác với embedding tĩnh, lớp embedding động trong CBILSuccSite là một thành phần trực tiếp của mạng học sâu (CNN1D và Bi-LSTM). Các véc tơ nhúng này không cố định mà được cập nhật và tối ưu liên tục trong quá trình huấn luyện thông qua thuật toán lan truyền ngược của mạng học sâu CNN1D và Bi-LSTM. Nhờ đó, mô hình có khả năng học được biểu diễn tối ưu cho từng axit amin trong chính ngữ cảnh sinh học của chuỗi protein, thay vì phải dựa hoàn toàn vào véc tơ được xây dựng sẵn từ trước.

(2) *Cấu trúc học sâu lai*: Hai nhánh xử lý song song được thiết kế: (i) CNN1D nhằm học đặc trưng cục bộ (local features) tương ứng với các motif sinh học; (ii) Bi-LSTM để học thông tin phụ thuộc dài hạn theo cả hai chiều của chuỗi. Các nhánh này kết thúc bằng lớp Flatten để đưa về véc tơ một chiều.

(3) *Kết hợp đặc trưng và phân loại*: Các đặc trưng từ CNN1D và Bi-LSTM được ghép nối qua lớp Concatenate, sau đó đưa vào lớp Dense (128 nút) để học thêm các quan hệ phi tuyến. Lớp đầu ra với hàm sigmoid thực hiện phân loại nhị phân, xác định vị trí có hoặc không bị succinyl hóa.




**Hình 3.11 Kiến trúc mô hình dự đoán vị trí PTM đề xuất (CBILSuccSite)**

Giả mã cho mô hình đề xuất được trình bày trong Algorithm 3.4 (phần 1) và Algorithm 3.4 (phần 2) dưới đây.

**Algorithm 3.4** Thuật toán học sâu lai dự đoán Succinylation (CBILSuccSite) (Phần 1)

**Đầu vào:** $\mathscr{X}_{\text{raw}}^{\text{train}}, \mathscr{Y}^{\text{train}}$ ▷ Tập huấn luyện gốc
$\mathscr{X}_{\text{raw}}^{\text{val}}$ ▷ Tập xác thực
$\mathscr{X}_{\text{raw}}^{\text{test}}$ ▷ Tập kiểm tra độc lập
**Đầu ra:** $\widehat{\mathscr{Y}}^{\text{test}}, \theta^*$ ▷ Dự đoán nhãn tập kiểm tra và tham số mô hình tối ưu

**Các bước thực hiện:**
**Bước 1. Tạo từ điển 1-gram:**
$\mathscr{A} = \{a_1, a_2, \ldots, a_{21}\}$ với $a_{21} =$ 'X'
Xây dựng ánh xạ: $\phi : \mathscr{A} \to \mathbb{N}$, $\phi(a_i) = i$
**Bước 2. Tiền xử lý dữ liệu:**
**for all** $x_i \in \mathscr{X}_{\text{raw}}^{\text{train}}$ **do**
    $\tilde{x}_i \leftarrow \texttt{Tokenize}(x_i)$
    $\bar{x}_i \leftarrow \phi(\tilde{x}_i) \in \mathbb{N}^L$
**end for**
$\mathscr{X}^{\text{train}} \leftarrow \{\bar{x}_i\}_{i=1}^{N}$

**Algorithm 3.4** Thuật toán học sâu lai dự đoán Succinylaton (CBILSuccSite) (Phần 2)

**Khởi tạo:** Trọng số $\theta$, tốc độ học $\eta$, số epoch $N_{\text{epoch}}$
**Bước 3. Huấn luyện mô hình:**
**for** $e = 1$ **to** $N_{\text{epoch}}$ **do**
    **for all** mini-batch $(\mathbf{X}, \mathbf{y}) \subset (\mathscr{X}^{\text{train}}, \mathscr{Y}^{\text{train}})$ **do**
        **Nhánh CNN1D:**
        $$\mathbf{X}_{\text{CNN1D}} \leftarrow \text{Flatten}\Big(\text{AvgPool}\big(\text{GRU}\big(\text{MaxPool}_2\big(\text{Conv1D}_2\big(\text{Dropout}\big(\text{MaxPool}_1\big(\text{Conv1D}_1(\text{Embedding}(\mathbf{X}))\big)\big)\big)\big)\big)\big)\Big)$$
        **Nhánh Bi-LSTM:**
        $$\mathbf{X}_{\text{Bi-LSTM}} \leftarrow \text{Flatten}\big(\text{Bi-LSTM}\big(\text{Dropout}(\text{Embedding}(\mathbf{X}))\big)\big)$$
        **Kết hợp đặc trưng:**
        $\mathbf{X}' \leftarrow \text{Concatenate}(\mathbf{X}_{\text{CNN1D}}, \mathbf{X}_{\text{Bi-LSTM}})$
        **Phân lớp:**
        $\mathbf{Z} \leftarrow \text{ReLU}(\text{Dense}_{64}(\mathbf{X}'))$
        $\widehat{\mathbf{y}} \leftarrow \sigma(\text{Dense}_1(\mathbf{Z}))$
        **Tính hàm mất mát và cập nhật tham số:**
        $\mathscr{L} \leftarrow \text{BCE}(\mathbf{y}, \widehat{\mathbf{y}})$
        $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathscr{L}$
    **end for**
    **if** Không cải thiện trên tập $\mathscr{D}_{\text{val}}$ **then**
        **Dừng sớm (Early Stopping)**
        **Break**
    **end if**

### *3.3.5 Chiến lược và tham số huấn luyện mô hình*

Mô hình **CBILSuccSite** được huấn luyện trong môi trường Google Colab với sự hỗ trợ của GPU nhằm tăng tốc độ tính toán. Trong quá trình huấn luyện, hàm mất mát được sử dụng là *Binary Cross Entropy*, phù hợp với bài toán phân loại nhị phân. Thuật toán tối ưu được lựa chọn là *Adam Optimizer* với tốc độ học (*learning rate*) là 0.001, giúp cân bằng giữa khả năng hội tụ nhanh và tránh rơi vào cực tiểu cục bộ. Dữ liệu huấn luyện được chia thành các lô nhỏ (*batch size*) gồm 32 mẫu, đảm bảo tính ổn định trong quá trình cập nhật trọng số. Toàn bộ mô hình được huấn luyện trong 100 *epochs*, đủ để mô hình học được đặc trưng phức tạp từ dữ liệu mà vẫn hạn chế hiện tượng quá khớp.

**Bảng 3.7 Cấu trúc mô hình CBILSuccSite và số lượng tham số huấn luyện**

| **Lớp (Layer)** | **Kích thước đầu ra** | **Số tham số** | **Mô tả** |
|---|---|---|---|
| InputLayer | (None, 33) | 0 | Đầu vào chuỗi axit amin |
| **Nhánh CNN-GRU (Branch 1)** | | | |
| Embedding_CNN1D | (None, 33, 300) | 6,600 | Nhúng (Embedding) đầu vào cho CNN1D |
| Conv1D | (None, 31, 64) | 57,664 | Lớp Tích chập 1D thứ nhất (Kernel size: 3) |
| MaxPooling1D | (None, 15, 64) | 0 | Giảm chiều và trích xuất đặc trưng nổi bật nhất |
| Dropout | (None, 15, 64) | 0 | Ngăn ngừa quá khớp (overfitting) |
| Conv1D | (None, 13, 64) | 12,352 | Lớp Tích chập 1D thứ hai (Kernel size: 3) |
| MaxPooling1D | (None, 6, 64) | 0 | Giảm chiều, chuẩn bị cho lớp GRU |
| Bidirectional GRU | (None, 6, 32) | 7,872 | Học phụ thuộc hai chiều giữa các motif được trích xuất (16 đơn vị ẩn) |
| GlobalAveragePooling1D | (None, 32) | 0 | Gộp trung bình toàn cục, tóm tắt đặc trưng chuỗi |
| Flatten | (None, 32) | 0 | Trải phẳng đầu ra của nhánh 1 |
| **Nhánh Bi-LSTM (Branch 2)** | | | |
| Embedding_Bi-LSTM | (None, 33, 300) | 6,600 | Nhúng (Embedding) đầu vào cho Bi-LSTM |
| Dropout | (None, 33, 300) | 0 | Ngăn ngừa quá khớp (theo Code Summary) |
| Bi-LSTM | (None, 33, 64) | 85,248 | Học phụ thuộc hai chiều (trái-phải) trong chuỗi (32 đơn vị ẩn) |
| Flatten (từ Bi-LSTM) | (None, 2112) | 0 | Trải phẳng đầu ra của nhánh 2 (33×64) |
| **Tổng hợp đặc trưng và Phân loại** | | | |
| Concatenate | (None, 2144) | 0 | Nối các vector đặc trưng của hai nhánh |
| Dense | (None, 128) | 274,560 | Lớp ẩn fully connected (tổng hợp phân loại) |
| Dropout | (None, 128) | 0 | Ngăn ngừa quá khớp |
| Dense (Output) | (None, 1) | 129 | Lớp đầu ra (Sigmoid) phân loại nhị phân |

**Tổng số tham số:** 451,025 (1.72 MB)
**Tham số huấn luyện được:** 451,025 (1.72 MB)
**Tham số không huấn luyện:** 0

Dựa trên thông tin từ Bảng 3.7 có thể thấy mô hình CBILSuccSite là một kiến trúc

học sâu lai với 451,025 tham số huấn luyện, được thiết kế để cân bằng giữa khả năng học phức tạp và tính hiệu quả tính toán. Sự phân bổ tham số của mô hình thể hiện chiến lược rõ ràng là tập trung nguồn lực vào các lớp xử lý và phân loại cuối cùng, thay vì các lớp nhúng (Embedding) vốn chỉ chiếm khoảng 2.9% tổng tham số. Cụ thể, kiến trúc được xây dựng với hai nhánh đặc trưng song song:

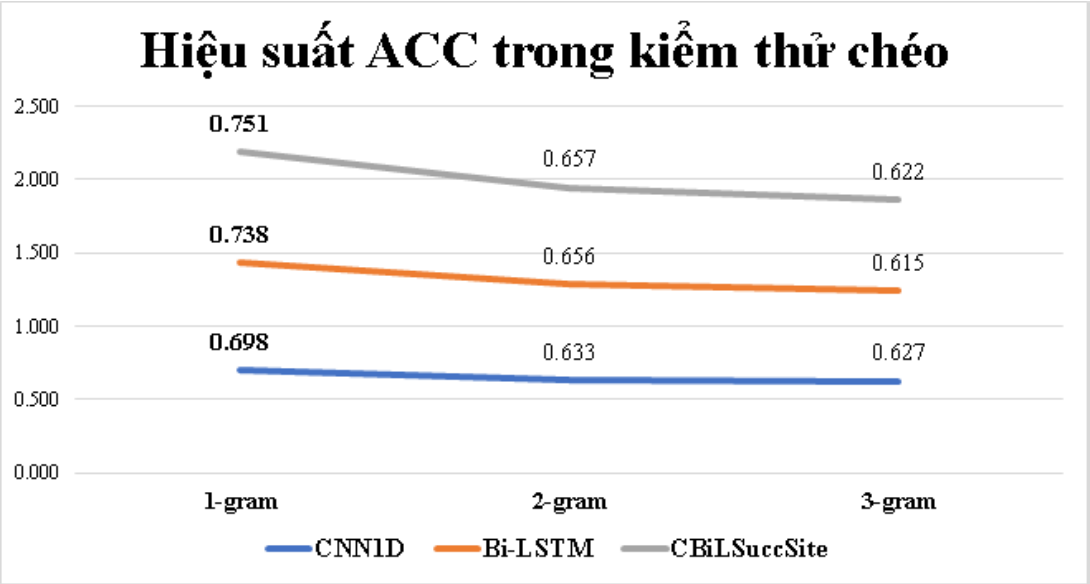
Thứ nhất là nhánh CNN-GRU, sử dụng CNN1D để trích xuất các motif cục bộ, sau đó áp dụng Bidirectional(7,872 tham số) để học mối quan hệ tuần tự giữa các motif này. GlobalAveragePooling1D được sử dụng để tóm tắt các đặc trưng đã học thành một vector cô đọng (None, 32) trước khi làm phẳng. Thứ hai là nhánh Bi-LSTM, đóng vai trò là khối học đặc trưng tuần tự chính, chiếm 85,248 tham số. Nhánh này nắm bắt ngữ cảnh hai chiều trên toàn bộ chuỗi đầu vào (33 axit amin), tạo ra vector đặc trưng (None, 2112) sau khi làm phẳng.

Sau khi các vector đặc trưng được ghép nối (Concatenate) thành 2144 chiều, phần lớn tham số của mô hình được đổ vào lớp Dense (Fully Connected) ẩn. Với 274,560 tham số, chiếm 60.8% tổng số, lớp này chịu trách nhiệm "tổng hợp tri thức từ cả hai nhánh" và học ánh xạ phi tuyến tính để đưa ra quyết định phân lớp. Số lượng tham số vừa phải này, kết hợp với các lớp Dropout chiến lược, giúp mô hình duy trì khả năng học mạnh mẽ nhưng đồng thời giảm thiểu đáng kể nguy cơ quá khớp trên tập dữ liệu dự đoán PTM thường hạn chế.

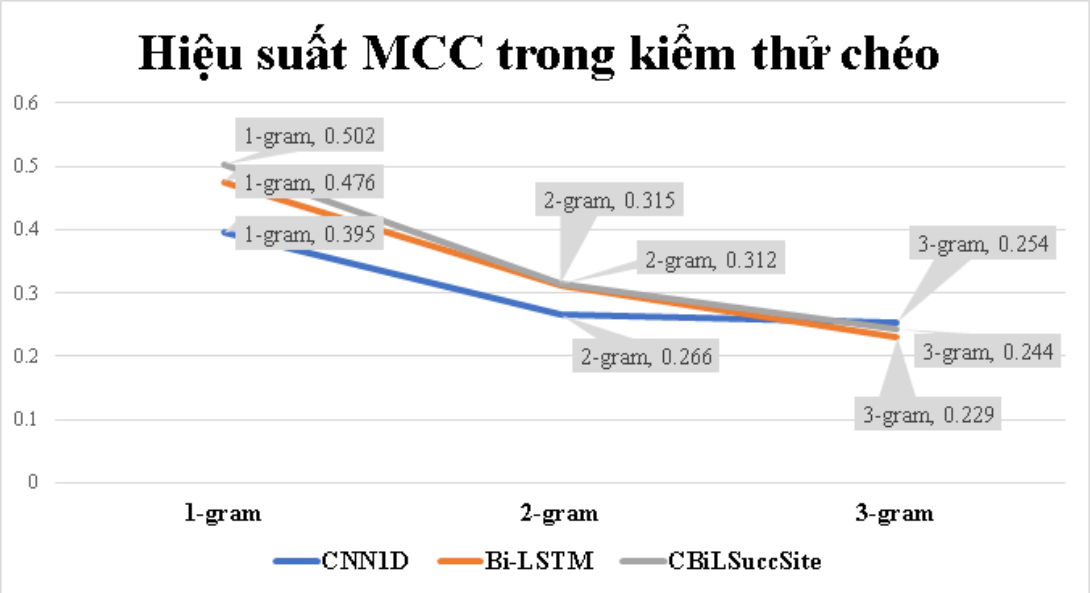
### *3.3.6 Kết quả và thảo luận*

Kết quả đánh giá chéo của các mô hình được trình bày chi tiết trong Hình 3.12, Hình 3.13 và Hình 3.14. Kết quả thực nghiệm cho thấy hiệu suất của các mô hình giảm dần khi sử dụng n-gram lớn hơn. Cụ thể, với phương pháp mã hoá 1-gram, các mô hình đạt giá trị cao hơn ở hầu hết các chỉ số so với 2-gram và 3-gram. Chẳng hạn, Bi-LSTM đạt SEN (0.77), ACC (0.738), và AUC (0.812) với 1-gram, nhưng giảm xuống SEN (0.68), ACC (0.656), và AUC (0.711) với 2-gram, và tiếp tục giảm ở 3-gram. Điều này cũng quan sát được trên CNN1D và CBiLSuccSite.

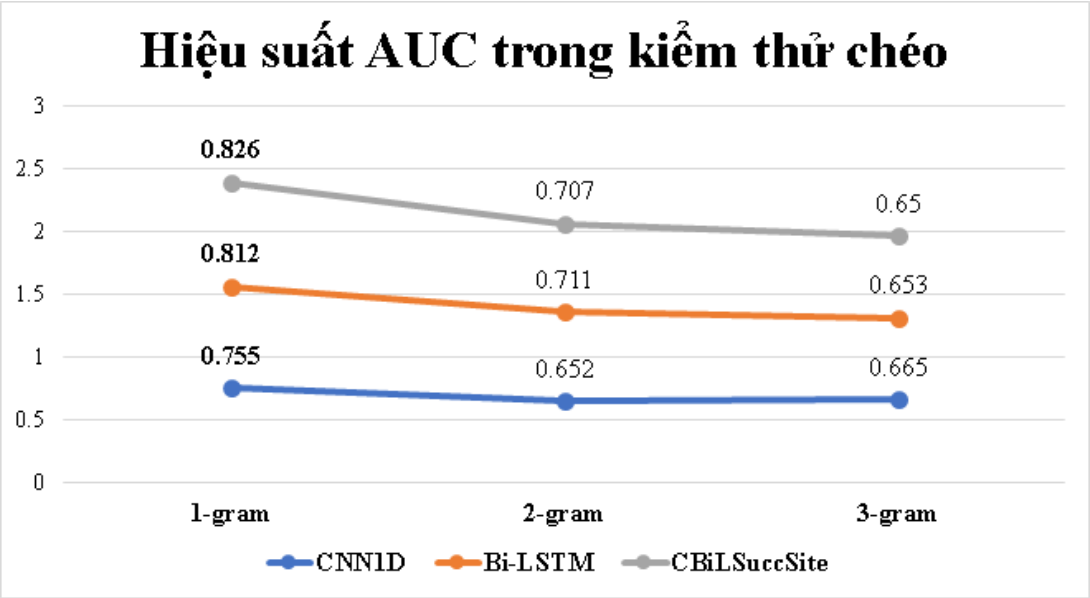
Mô hình học sâu lai CBiLSuccSite có hiệu suất cao hơn so với CNN1D và Bi-LSTM, đặc biệt khi sử dụng 1-gram, với ACC (0.75), MCC (0.502), và AUC (0.826), cao nhất trong tất cả các mô hình được thử nghiệm. Do đó 1-gram được chọn là phương pháp tách từ cho dự đoán vị trí PTM dựa trên embedding động.




**Hình 3.12 Hiệu suất ACC kiểm thử chéo 10 mặt**

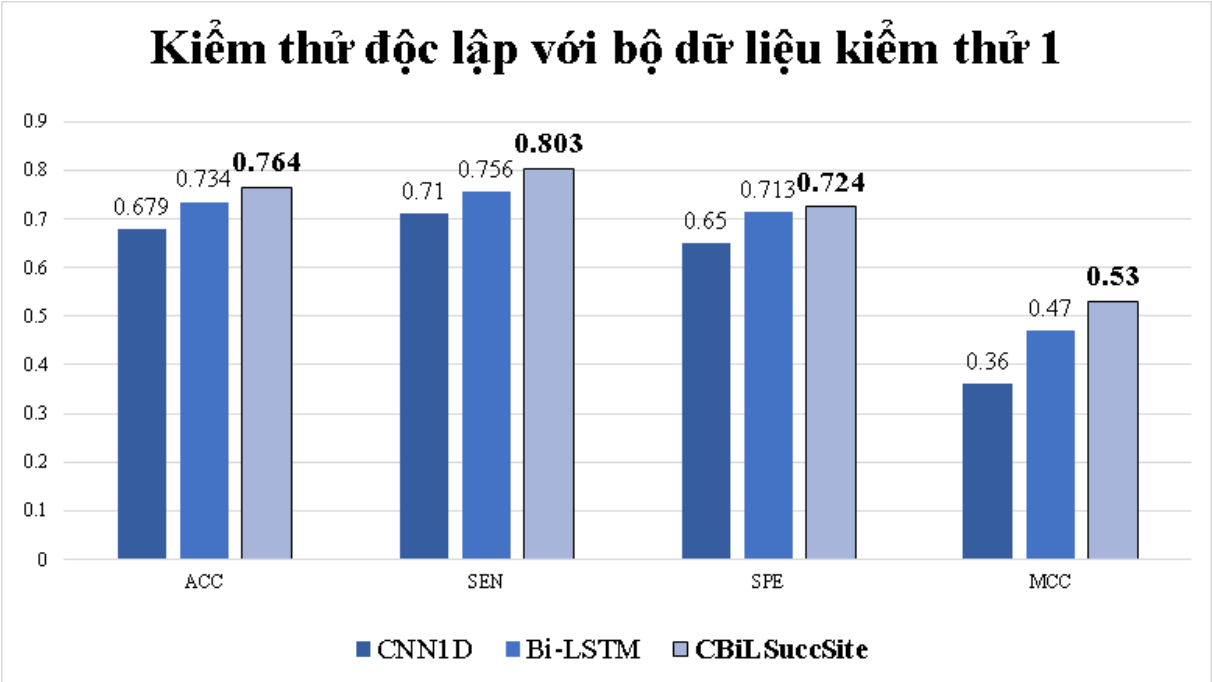



**Hình 3.13 Hiệu suất MCC kiểm thử chéo 10 mặt**

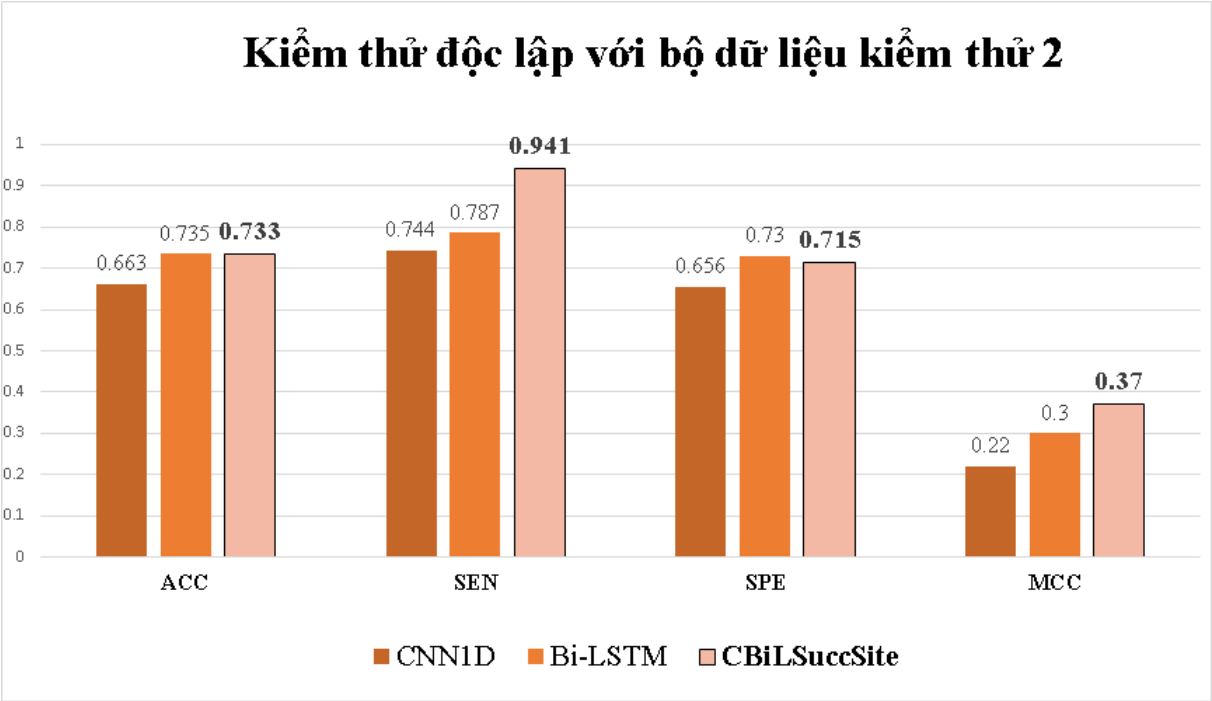



**Hình 3.14 Hiệu suất AUC kiểm thử chéo 10 mặt**

Để kiểm tra hiệu suất của các mô hình với dữ liệu thực tế, phương pháp kiểm thử độc lập được sử dụng, kết quả hiển thị trong Hình 3.15 và Hình 3.16. Kết quả cho thấy mô hình CBiLSuccSite đạt hiệu suất cao nhất trong cả hai tập kiểm tra độc lập (Tập kiểm tra 1 và Tập kiểm tra 2).




**Hình 3.15 Hiệu suất kiểm thử độc lập với bộ dữ liệu kiểm thử 1**




**Hình 3.16 Hiệu suất kiểm thử độc lập với bộ dữ liệu kiểm thử 2**

Kết quả kiểm thử độc lập (Hình 3.15 và Hình 3.16) đã minh chứng rõ ràng hiệu quả của việc kết hợp kiến trúc học sâu lai CNN1D-BiLSTM (CBiLSuccSite) với phương pháp embedding động trong bài toán dự đoán vị trí succinyl hóa.

Thứ nhất, về mặt cơ chế kiến trúc, hiệu quả cải thiện rõ rệt xuất phát từ sự bổ trợ lẫn nhau giữa CNN1D và Bi-LSTM. CNN1D đảm nhiệm vai trò trích xuất các đặc trưng cục bộ, phát hiện những motif ngắn hoặc các tín hiệu mang tính khu vực trong chuỗi protein, vốn rất quan trọng đối với các vị trí sửa đổi sau dịch mã. Trong khi đó, Bi-LSTM có khả năng mô hình hóa mối quan hệ phụ thuộc dài hạn, học được ngữ cảnh hai chiều từ chuỗi axit amin, giúp phát hiện những mối liên hệ xa hơn giữa các axit amin mà CNN1D không thể nắm bắt.

Kết quả thực nghiệm trên tập kiểm tra 1 thể hiện rõ lợi thế này: mặc dù Bi-LSTM đã cho thấy khả năng vượt trội hơn CNN1D về độ chính xác ACC tổng thể và MCC nhờ khả năng học ngữ cảnh, nhưng CBiLSuccSite tiếp tục nâng cao hiệu suất nhờ việc kết hợp song song cả cục bộ (CNN1D) và ngữ cảnh (Bi-LSTM), giúp cải thiện đồng thời ACC, SEN, SPE và đặc biệt MCC (0.53) cao hơn 0.17 so với CNN1D (MCC: 0.36). Điều này khẳng định CBiLSuccSite không chỉ đơn thuần là phép cộng cơ học giữa hai mô hình, mà là sự tích hợp mang tính bổ trợ về mặt biểu diễn đặc trưng.

Thứ hai, embedding động đóng vai trò nền tảng trong việc nâng cao hiệu quả biểu diễn của chuỗi protein. Thay vì sử dụng các phương pháp mã hóa tĩnh hoặc thủ công, embedding động giúp biến các axit amin thành các véc tơ biểu diễn giàu ngữ nghĩa,

được học trực tiếp từ chính dữ liệu huấn luyện, phản ánh linh hoạt sự tương đồng và mối quan hệ giữa các axit amin dựa trên ngữ cảnh. Khi kết hợp với CNN1D và Bi-LSTM, embedding động cho phép mô hình học sâu không chỉ đơn thuần nhận diện motif mà còn hiểu được ý nghĩa "ngữ nghĩa" của các trình tự liên kết xa nhau trong chuỗi.

Điều này giải thích vì sao, mặc dù trên tập kiểm tra 2 (với độ mất cân bằng dữ liệu cao hơn nhiều), ACC của CBiLSuccSite tương đương với Bi-LSTM, nhưng CBiLSuccSite lại đạt độ nhạy SEN cao nhất (0.941) – thể hiện khả năng mạnh mẽ trong việc nhận diện đúng các vị trí succinyl hóa tiềm năng, điều đặc biệt quan trọng trong các bài toán PTM vốn nhấn mạnh phát hiện mẫu dương tính.

### *3.3.7 So sánh mô hình đề xuất với các công cụ khác*

Trong nghiên cứu này, NCS lựa chọn so sánh mô hình **CBiLSuccSite** với các công cụ dự đoán succinyl hóa đã được công bố rộng rãi và có tính đại diện tại thời điểm tiến hành nghiên cứu (cuối năm 2023). Mặc dù trong năm 2024 bắt đầu xuất hiện thêm một số mô hình SOTA mới, nhưng khi triển khai nghiên cứu, các công cụ như GPSuc, DeepSuccinylSite, LMSuccSite, pSuc-EDBAM và MDCAN-Lys vẫn được xem là những phương pháp nổi bật, được cộng đồng sử dụng phổ biến, có mô hình công khai hoặc phần mềm triển khai. Đặc biệt, các công cụ này đều cung cấp server hoặc mã nguồn (GitHub) và sử dụng cùng bộ dữ liệu huấn luyện/kiểm thử, do đó việc lựa chọn chúng đảm bảo tính công bằng, khả năng tái lập và cho phép đối chiếu trực tiếp với các kết quả của luận án.

Cụ thể, GPSuc là một trong những công cụ sớm và thường được dùng trong thực nghiệm; DeepSuccinylSite khai thác kiến trúc CNN trong học sâu; LMSuccSite kết hợp giữa học máy và đặc trưng thủ công; pSuc-EDBAM và MDCAN-Lys đại diện cho các cải tiến học sâu gần đây. Như vậy, các công cụ này không chỉ phản ánh sự đa dạng về phương pháp (học máy truyền thống, học sâu đơn thuần, kiến trúc lai) mà còn bao quát được nhiều thế hệ phát triển của bài toán.

Bên cạnh đó, NCS cũng tiến hành đánh giá mô hình **CBiLSuccSite** trên hai bộ dữ liệu kiểm thử độc lập (Bô dữ liệu kiểm thử 1 và Bộ dữ liệu kiểm thử 2) nhằm đảm bảo tính khách quan. Kết quả (Bảng 3.8) cho thấy **CBiLSuccSite** đạt hiệu suất vượt trội hơn so với tất cả các công cụ so sánh trên cả hai bộ dữ liệu. Đặc biệt, trên Bộ dữ liệu kiểm thử 2, mô hình đạt độ nhạy (SEN) cao nhất (0.941), chứng minh khả năng phát hiện chính xác nhiều vị trí succinyl hóa thực sự.

Những kết quả này khẳng định rằng chiến lược kết hợp *embedding động* và kiến trúc học sâu lai CNN1D–BiLSTM đã mang lại lợi thế rõ rệt, cải thiện đáng kể độ chính xác trong dự đoán PTM so với các phương pháp truyền thống hoặc học sâu đơn lẻ.

**Bảng 3.8 So sánh mô hình đề xuất với các công cụ dự đoán succinyl hóa khác**

| Bộ dữ liệu | Tools | ACC | SEN | SPE | MCC |
|---|---|---|---|---|---|
| Dữ liệu kiểm thử 1 | GPSuc [39] | 0.670 | 0.660 | 0.680 | 0.350 |
| | DeepSuccinylSite [106] | 0.700 | 0.790 | 0.690 | 0.480 |
| | LMSuccSite [82] | 0.740 | 0.760 | 0.730 | 0.510 |
| | pSuc-EDBAM [48] | 0.699 | 0.748 | 0.650 | 0.400 |
| | MDCAN-Lys [115] | 0.707 | 0.768 | 0.646 | 0.420 |
| | **CBiLSuccSite (Đề xuất)** | **0.763** | **0.803** | **0.724** | **0.530** |
| Dữ liệu kiểm thử 2 | GPSuc [39] | 0.850 | 0.880 | 0.490 | 0.300 |
| | DeepSuccinylSite [106] | 0.700 | 0.790 | 0.690 | 0.270 |
| | LMSuccSite [82] | 0.790 | 0.790 | 0.790 | 0.360 |
| | pSuc-EDBAM [48] | 0.738 | 0.760 | 0.736 | 0.290 |
| | MDCAN-Lys [115] | 0.732 | 0.705 | 0.734 | 0.260 |
| | **CBiLSuccSite (Đề xuất)** | **0.733** | **0.941** | **0.715** | **0.370** |

## 3.4 Kết luận chương 3

Trong chương này, NCS đã tập trung giải quyết bài toán dự đoán PTM bằng cách đề xuất và phát triển hai mô hình học sâu lai tiên tiến: CLW_SUMO và CBiLSuccSite. Mô hình CLW_SUMO được thiết kế cho dự đoán vị trí SUMOylation, sử dụng kiến trúc kết hợp CNN1D và LSTM với embedding tĩnh dựa trên ma trận Word2Vec nhằm khai thác các đặc trưng ngữ nghĩa từ chuỗi protein. Trong khi đó, CBiLSuccSite hướng tới dự đoán vị trí Succinylation, áp dụng kiến trúc CNN1D kết hợp Bi-LSTM cùng với embedding động, giúp nâng cao khả năng học biểu diễn và tối ưu hóa hiệu suất.

Kết quả thực nghiệm trên nhiều tập dữ liệu độc lập đã khẳng định rằng cả hai mô hình đều vượt trội hơn các mô hình cơ sở và nhiều phương pháp state-of-the-art hiện có. Hiệu quả này bắt nguồn từ sự phối kết hợp hiệu quả của kiến trúc học sâu lai: CNN1D giúp trích xuất đặc trưng cục bộ, LSTM/Bi-LSTM nắm bắt được ngữ cảnh dài hạn, và Word2Vec embedding hoặc embedding động cung cấp biểu diễn ngữ nghĩa phong phú hơn so với các đặc trưng thủ công truyền thống. Những kết quả này là minh chứng cho tính ưu việt và tiềm năng của việc tích hợp kỹ thuật embedding từ NLP với mô hình học

sâu lai trong dự đoán các vị trí PTM.

*Các kết quả nghiên cứu chính trong chương này đã được NCS công bố trên hai tạp chí khoa học và hai kỷ yếu hội thảo quốc tế uy tín*

[CT4] Tran T.X., Le N.Q.K., and Nguyen V.N. (2024), CLW-SUMO: A hybrid deep learning model for predicting protein SUMOylation sites. Journal of Computer Science and Cybernetics. DOI: https://doi.org/10.15625/1813-9663/ 19626. (Tạp chí Tin học điều khiển 1.25đ)

[CT5] Tran T.X, Le N.Q.K, and Nguyen V.N. (2025), Integrating CNN and Bi-LSTM for protein succinylation sites prediction based on Natural Language Process- ing technique. Computers in Biology and Medicine. 186: p. 109664. DOI: https://doi.org/10.1016/j.compbiomed.2025.109664. (SCIE Q1, IF: 7.0)

[CT6] Tran T.X., Nguyen T.T., Le N.Q.K., et al. (2024). A novel deep learning approach for the prediction of Arabidopsis thaliana ubiquitination sites. Proceedings of the 13th International Conference on Information Technology and Its Applica- tions (CITA 2024), pp. 48–57. DOI: https://elib.vku.udn.vn/handle/ 123456789/4010. (Scopus Q4)

[CT7] Tran T.X, Nguyen T.T, Le N.Q.K, and Nguyen V.N. (2025), A hybrid deep learning and Natural Language Processing Model for Plant Ubiquitination Site Prediction, The 3rd International Conference on Advances in Information and 114 Communication Technology. ICTA 2024. DOI: https://doi.org/10.1007/ 978-3-031-80943-9_49. (Indexed: Scopus Q4)

# CHƯƠNG 4. MÔ HÌNH HỌC CHẮT LỌC TRI THỨC KẾT HỢP XỬ LÝ NGÔN NGỮ TỰ NHIÊN DỰ ĐOÁN VỊ TRÍ SỬA ĐỔI SAU DỊCH MÃ TRONG CHUỖI PROTEIN

*Trong chương 3, NCS đã để xuất một số mô hình học sâu lai kết hợp kỹ thuật NLP nhằm cải thiện hiệu suất dự đoán vị trí Succinylation và SUMOylation. Các kết quả đạt được đã được khẳng định thông qua các công bố trên các tạp chí uy tín trong và ngoài nước, cho thấy tiềm năng của hướng tiếp cận này trong bài toán dự đoán vị trí PTM. Tuy nhiên, với mong muốn tiếp tục mở rộng và làm giàu tri về các loại PTM khác. Trong nghiên cứu này, NCS chọn PTM Ubiquitination. Ubiquitination đóng vai trò thiết yếu trong điều hòa sự ổn định và phân hủy protein, và là mục tiêu nghiên cứu rộng rãi trong lĩnh vực sinh học phân tử và y học.*

*Bên cạnh đó, từ thực tiễn triển khai mô hình ở các chương trước, một thách thức đáng kể được đặt ra là chi phí tính toán do sử dụng các mô hình sâu lai (CNN, Bi-LSTM). Để giải quyết bài toán này, chương 4 đề xuất một mô hình mới dự đoán vị trí Ubiquitination dựa trên học chắt lọc tri thức (Knowledge Distillation) – một kỹ thuật cho phép huấn luyện mô hình gọn nhẹ ("mô hình Học viên") nhưng vẫn duy trì hiệu quả dự đoán tương đương với mô hình lớn ("mô hình Giáo viên"). Đặc biệt, mô hình được thiết kế kế thừa các ưu điểm của kỹ thuật mã hóa NLP đã chứng minh hiệu quả ở chương 3, giúp biểu diễn tốt hơn thông tin sinh học từ trình tự axit amin và tăng cường khả năng học của mô hình.*

*Hướng tiếp cận này không chỉ giúp mở rộng nghiên cứu làm giàu tri thức về một loại PTM khác, mà còn góp phần tối ưu hoá chi phí tính toán, giảm độ phức tạp mô hình, và nâng cao tính khả thi triển khai trong thực tiễn đáp ứng yêu cầu thiết yếu của các bài toán sinh tin hiện đại trong bối cảnh dữ liệu ngày càng lớn và đa dạng. Một phần kết quả nghiên cứu được đăng trên tạp chí Methods (SCIE Q1) [CT8].*

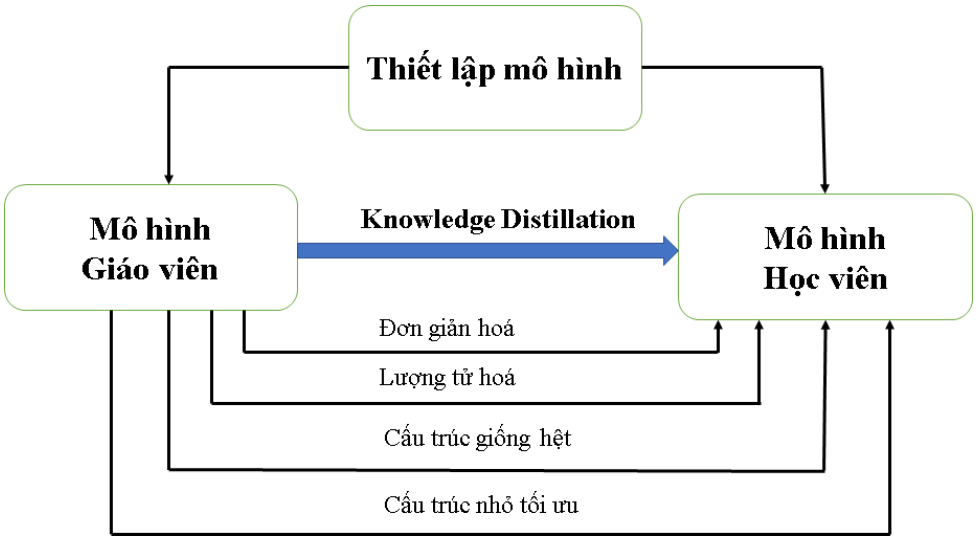
## 4.1 Học chắt lọc tri thức

Trong nhiều bối cảnh thực tế, mô hình học máy xây dựng phải đáp ứng các hạn chế về thời gian, bộ nhớ, chi phí và tài nguyên tính toán. Các mô hình có hiệu suất cao nhất cho một nhiệm vụ nhất định thường quá lớn, chậm hoặc đắt đỏ đối với hầu hết các trường hợp sử dụng thực tế. Ngược lại, rất cần các mô hình nhỏ nhanh hơn và ít đòi hỏi tính toán hơn nhưng duy trì độ chính xác và khả năng tổng quát hóa như các mô hình lớn. Để giải quyết vấn đề này, năm 2015, Hinton và cộng sự [41] đã giới thiệu phương pháp Học chắt lọc tri thức (Knowledge distillation). Thuật toán này được lấy cảm hứng từ quá trình con người tiếp thu tri thức: trong môi trường học tập, người học tiếp nhận

kiến thức từ giáo viên – những người có nền tảng chuyên môn vững chắc và kinh nghiệm dày dạn. Tương tự trong học máy, một mô hình lớn, đã được huấn luyện kỹ lưỡng sẽ đóng vai trò như giáo viên (Teacher model), trong khi một mô hình nhỏ hơn, đơn giản hơn sẽ đảm nhận vai trò học viên (Student model) tiếp thu và học lại những gì mô hình lớn đã lĩnh hội.

Các kỹ thuật học chắt lọc tri thức đã được ứng dụng thành công trong nhiều lĩnh vực khác nhau như NLP, nhận dạng giọng nói, nhận dạng hình ảnh và phát hiện đối tượng. Đặc biệt, trong những năm gần đây, học chắt lọc tri thức đóng vai trò quan trọng trong việc tối ưu hóa các mô hình ngôn ngữ lớn (LLM). Phương pháp này đã chứng minh hiệu quả trong việc chuyển giao các khả năng tiên tiến từ các mô hình lớn sang các mô hình nguồn mở nhỏ hơn, dễ tiếp cận hơn, đặc biệt là trong các lĩnh vực như y tế, nơi cần giải quyết những thách thức phức tạp [5, 35, 56, 57, 59, 126].

Quá trình học chắt lọc tri thức truyền thống thường phải đối mặt với hai thách thức: (i) Thách thức đầu tiên là việc lựa chọn các "mô hình Giáo viên" và hiệu quả chuyển giao kiến thức. Các nghiên cứu gần đây đã làm nổi bật ảnh hưởng đáng kể của việc lựa chọn "mô hình Giáo viên" đến độ chính xác của "mô hình Học viên", cho thấy "mô hình Giáo viên" có độ chính xác cao nhất chưa chắc đã là lựa chọn tối ưu để chưng cất [20, 71]. Do đó, cần phải thử nghiệm rộng rãi để xác định "mô hình Giáo viên" phù hợp nhất để chưng cất, một quá trình có thể tốn rất nhiều thời gian. (ii) Thách thức thứ hai nằm ở thực tế là các "mô hình Học viên" thường không đạt được cùng mức độ chính xác như các "mô hình Giáo viên" tương ứng, có khả năng dẫn đến sự suy giảm độ chính xác không thể chấp nhận được trong quá trình suy luận.




**Hình 4.1 Mối quan hệ giữa "mô hình Giáo viên" và "mô hình Học viên" trong học chắt lọc tri thức [35]**

Thiết lập mô hình học (Model Setup) [35] trong học chắt lọc tri thức: Đây là yếu tố ảnh hưởng đến cách thiết kế và lựa chọn kiến trúc của "mô hình Giáo viên" và "mô hình Học viên". Chưng cất tri thức (Knowledge Distillation - Mũi tên chính) là quá trình truyền tri thức từ "mô hình Giáo viên" sang "mô hình Học viên".

Các phương pháp để thiết kế mô hình Học viên: Dựa vào Hình 4.1, có bốn cách phổ biến để thiết kế "mô hình Học viên" dựa trên "mô hình Giáo viên", mỗi phương pháp có những đặc điểm riêng nhằm cân bằng giữa hiệu suất và độ phức tạp của mô hình.

Thứ nhất, cấu trúc đơn giản hóa (Simplified Structure) là phương pháp trong đó "mô hình Học viên" có ít lớp hơn và số kênh giảm bớt so với "mô hình Giáo viên". Cách tiếp cận này giúp giảm độ phức tạp của mô hình, tiết kiệm tài nguyên tính toán nhưng vẫn duy trì được khả năng học tập từ "mô hình Giáo viên" thông qua quá trình chắt lọc tri thức.

Thứ hai, cấu trúc lượng tử hóa (Quantized Structure) vẫn giữ nguyên kiến trúc của "mô hình Học viên" so với "mô hình Giáo viên" nhưng áp dụng kỹ thuật lượng tử hóa để giảm kích thước mô hình. Phương pháp này đặc biệt hữu ích khi triển khai mô hình trên các thiết bị có tài nguyên hạn chế như thiết bị di động hoặc nhúng.

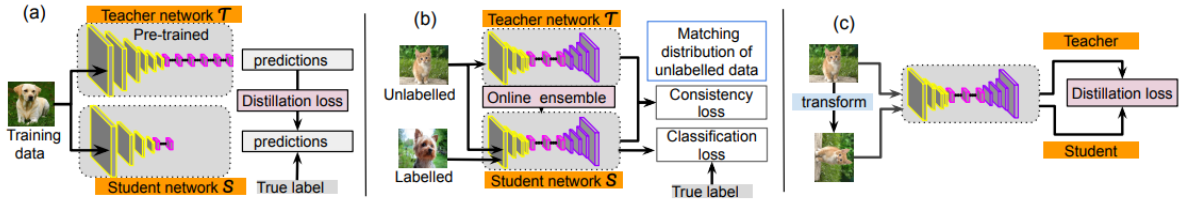
Thứ ba, cấu trúc giống hệt (Same Structure) có cách tiếp cận đơn giản nhất, khi "mô hình Học viên" giữ nguyên toàn bộ kiến trúc của "mô hình Giáo viên". Tuy nhiên, một điểm khác biệt quan trọng là "mô hình Học viên" có thể được huấn luyện với ít tài nguyên hơn bằng cách điều chỉnh kích thước batch, số lượng epoch hoặc chiến lược tối ưu hóa, giúp giảm thời gian huấn luyện mà vẫn duy trì hiệu suất tốt.

Cuối cùng, cấu trúc nhỏ tối ưu hóa hoặc cô đọng (Small Structure - Optimized/Condensed) tập trung vào việc thiết kế "mô hình Học viên" nhỏ gọn nhưng hiệu quả. Phương pháp này thường áp dụng các kỹ thuật tối ưu hóa toàn cục như kiến trúc cắt giảm tham số không cần thiết hoặc cơ chế chú ý, giúp giảm đáng kể độ phức tạp mà vẫn đảm bảo khả năng học tốt từ "mô hình Giáo viên".

**L. Wang and K.-J. Yoon [116] năm 2021 đã phân loại các phương pháp học chắt lọc tri thức hiện nay thành 5 nhóm chính, dựa trên loại tri thức được truyền từ giáo viên sang học viên :**

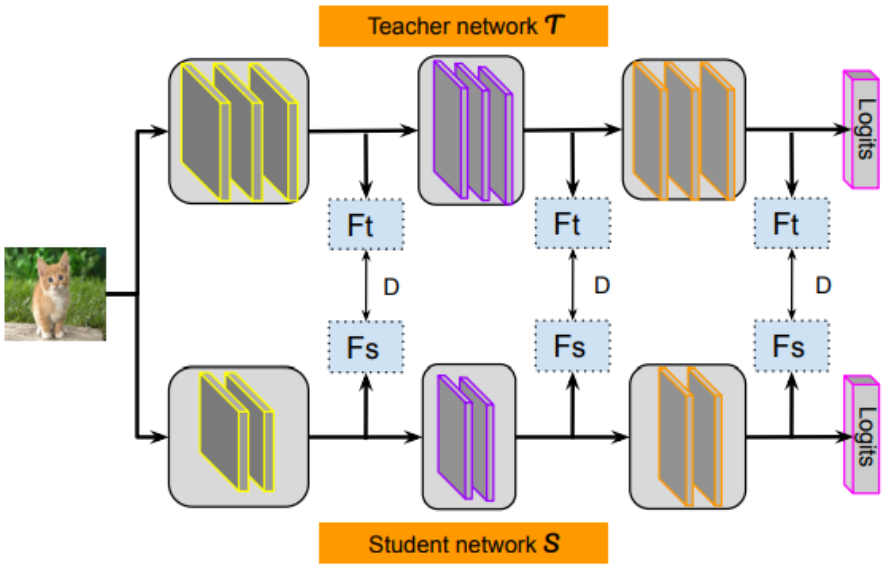
Nhóm thứ nhất, phương pháp này sử dụng trực tiếp xác suất đầu ra (softmax) của "mô hình Giáo viên" làm mục tiêu huấn luyện cho "mô hình Học viên". Đây là phương pháp phổ biến và đơn giản nhất là chắt lọc từ đầu ra (response-based distillation). Đại diện tiêu biểu cho nhóm này là phương pháp của Hinton [41], với kỹ thuật sử dụng tham số nhiệt độ (temperature) để làm mượt phân phối xác suất, từ đó giúp "mô hình Học viên" học được thêm nhiều thông tin ngữ nghĩa ẩn chứa trong các nhãn mềm. Đây là phương pháp cơ bản, hiệu quả và được ứng dụng rộng rãi trong các bài toán phân loại

(Chi tiết xem Hình 4.2).




**Hình 4.2 Minh họa các phương pháp chắt lọc tri thức (KD) với khung S-T (Student-Teacher). (a) cho mục đích nén mô hình và truyền tri thức, ví dụ: (b) học bán giám sát và (c) học tự giám sát [116].**

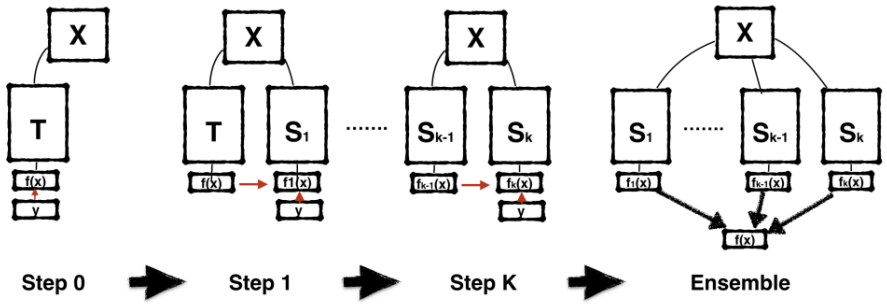
Nhóm thứ hai, là chắt lọc từ đặc trưng trung gian (feature-based distillation), trong đó "mô hình Học viên" học biểu diễn đặc trưng từ các tầng trung gian của "mô hình Giáo viên"(Hình 4.3). Một số phương pháp tiêu biểu trong nhóm này là FitNet [94] sử dụng hàm ánh xạ để điều chỉnh biểu diễn đặc trưng giữa giáo viên và học viên. Phương pháp này tỏ ra hiệu quả hơn so với response-based trong các mô hình có cấu trúc phức tạp.




**Hình 4.3 Minh họa chắt lọc từ đặc trưng trung gian [116].**

Nhóm thứ ba, chắt lọc từ mối quan hệ giữa các đặc trưng (relation-based distillation) cũng được quan tâm trong những năm gần đây. Thay vì truyền trực tiếp các đặc trưng, phương pháp này hướng tới việc bảo toàn cấu trúc hình học giữa các mẫu dữ liệu trong không gian đặc trưng của "mô hình Giáo viên". Các phương pháp tiêu biểu có thể kể đến như RKD của Park và cộng sự năm 2019 [80], trong đó học sinh mô phỏng các khoảng cách và góc giữa các đặc trưng. Phương pháp này thường phù hợp cho các bài

toán nhận diện phức tạp hoặc dữ liệu phân bố không đồng đều.




**Hình 4.4 Minh họa trực quan quy trình huấn luyện của phương pháp BAN: ở bước đầu tiên, "mô hình Giáo viên" T được huấn luyện từ nhãn Y. Sau đó, ở mỗi bước tiếp theo, một mô hình mới giống hệt được khởi tạo với hạt giống ngẫu nhiên khác và được huấn luyện dưới sự hướng dẫn của thế hệ trước. Cuối cùng, hiệu quả có thể được cải thiện thêm bằng cách kết hợp nhiều thế hệ học sinh thành một tổ hợp trong tự chắt lọc tri thức) [31].**

Nhóm thứ tư, tự chắt lọc tri thức (self-distillation), trong đó mô hình tự huấn luyện chính nó mà không cần "mô hình Giáo viên" bên ngoài. Điển hình cho hướng tiếp cận này là Born-Again Networks [31], trong đó mô hình được huấn luyện lặp lại nhiều lần, với mỗi phiên bản kế tiếp học từ phiên bản trước đó (Hình 4.4). Tự chắt lọc tri thức có ưu điểm là giảm chi phí huấn luyện do không cần giáo viên riêng biệt.

Cuối cùng, các phương pháp chắt lọc tri thức đặc thù theo tác vụ (task-specific distillation) được thiết kế riêng để phù hợp với các bài toán cụ thể như xử lý ngôn ngữ tự nhiên, nhận dạng đối tượng, học tăng cường, v.v. Trong lĩnh vực NLP, các mô hình như DistilBERT [95] và TinyBERT [50] là những ví dụ nổi bật, được chắt lọc từ các mô hình Transformer lớn như BERT. Với từng loại tác vụ, các kỹ thuật học chắt lọc tri thức sẽ được điều chỉnh để phù hợp với đặc thù dữ liệu và mục tiêu bài toán, từ đó đạt hiệu quả tốt hơn so với các phương pháp học chắt lọc tri thức tổng quát.

Từ tổng quan về các phương pháp học chắt lọc tri thức, có thể nhận thấy rằng mục tiêu chính của kỹ thuật này là truyền đạt tri thức từ một mô hình phức tạp, có năng lực biểu diễn mạnh ("mô hình Giáo viên"), sang một mô hình nhỏ gọn hơn ("mô hình Học viên"), nhằm giảm thiểu chi phí tính toán nhưng vẫn duy trì hiệu suất dự đoán cao. Các phương pháp học chắt lọc tri thức đã được áp dụng thành công trong nhiều lĩnh vực như thị giác máy tính và xử lý ngôn ngữ tự nhiên, đặc biệt trong các bài toán yêu cầu triển khai mô hình nhẹ, tối ưu trên các thiết bị tính toán hạn chế.

Tuy nhiên, theo khảo sát và tổng quan tài liệu hiện nay, chưa có nghiên cứu nào ứng dụng học chắt lọc tri thức trong bài toán dự đoán vị trí PTM, đặc biệt đối với ubiq-

uitination ở thực vật. Khoảng trống này chính là cơ sở để NCS lựa chọn áp dụng phương pháp học chắt lọc tri thức trong bài toán dự đoán vị trí ubiquitination cho *Arabidopsis thaliana*. Việc kết hợp giữa kiến trúc học sâu và kỹ thuật chắt lọc tri thức được kỳ vọng sẽ góp phần nâng cao hiệu quả dự đoán, đồng thời mở ra hướng tiếp cận mới trong lĩnh vực tin sinh học, nơi các mô hình dự đoán hiện nay còn gặp nhiều hạn chế về khả năng tổng quát hoá và chi phí tính toán. Đây cũng là một đóng góp mới về mặt phương pháp luận của nghiên cứu này đối với cộng đồng nghiên cứu PTM.

## 4.2 Mô hình dự đoán Ubiquitination dựa trên học chắt lọc tri thức và kỹ thuật xử lý ngôn ngữ tự nhiên đề xuất

### *4.2.1 Tên viết tắt*

KD_ArapUBi là tên gọi của mô hình đề xuất dùng để dự đoán vị trí Ubiquitination trên loài *Arabidopsis thaliana*. KD_ArapUBi là viết tắt của "Knowledge Distillation for Arabidopsis thaliana Ubiquitination prediction". Trong toàn bộ luận án, NCS sẽ sử dụng ký hiệu KD_ArapUBi để chỉ mô hình đề xuất.

### *4.2.2 Dữ liệu thực nghiệm*

Trong nghiên cứu này, NCS đã sử dụng kỹ thuật học chắt lọc tri thức kết hợp với kỹ thuật NLP để xây dựng mô hình dự đoán vị trí PTM Ubiquitination.

Ubiquitination là một loại PTM phổ biến, được tìm thấy lần đầu vào năm 1975 bởi nhà khoa học Goldstein và cộng sự [34]. Ubiquitin hóa, liên kết cộng hóa trị của ubiquitin với nhiều protein tế bào khác nhau, là quá trình biến đổi sau dịch mã quan trọng nhất của protein điều hòa chức năng tế bào. Trong quá trình ubiquitin hóa, ubiquitin liên kết với gốc lysine (K) thông qua phản ứng enzym ba giai đoạn (E1), enzym liên kết ubiquitin (E2) và ubiquitin ligase (E3) [78, 102, 121].

Dữ liệu cho "mô hình Học viên" được thu thập từ *Arabidopsis thaliana* theo nghiên cứu của Chen và cộng sự [13], gồm 2,043 vị trí Ubiquitination từ 1,607 protein.

Dữ liệu cho "mô hình Giáo viên": Các vị trí Ubiquitination được thu thập từ các nguồn như cơ sở dữ liệu dbPTM, PLMD và nghiên cứu của Wang [114], với tổng 121,742 vị trí Ubiquitination từ 25,103 protein. Dữ liệu cho Giáo viên chọn các vị trí Ubiquitination từ các loài: *Arabidopsis thaliana*, *Oryza sativa subsp indica* và *O. sativa subsp Japonica*. Tạo mẫu dữ liệu dương tính và âm tính với cửa sổ trượt được áp dụng như các phương pháp trước, tiếp đó sử dụng CD-hit để loại bỏ trùng 30%. Cuối cùng, dữ liệu sử dụng trong nghiên cứu được tóm tắt như Bảng 4.1 dưới đây.

**Bảng 4.1 Bộ dữ liệu huấn luyện và kiểm tra sử dụng trong nghiên cứu**

| Mô hình | Bộ dữ liệu | SL Protein | SL mẫu dương tính | SL mẫu âm tính |
|---|---|---|---|---|
| "mô hình Giáo viên" | Dữ liệu huấn luyện | 25,103 | 3,373 | 3,373 |
| | Dữ liệu kiểm tra | – | 750 | 750 |
| "mô hình Học viên" | Dữ liệu huấn luyện | 1,607 | 1,532 | 1,532 |
| | Dữ liệu kiểm tra | – | 511 | 511 |

### *4.2.3 Cơ sở lựa chọn mô hình KD2 (KD_ArapUbi)*

Như đã trình bày trong mục 4.1, không có quy định bắt buộc "mô hình Giáo viên" và "mô hình Học viên" trong kiến trúc học chắt lọc tri thức phải sử dụng cùng một loại kiến trúc học sâu. Để lựa chọn được cặp "mô hình Giáo viên" - "Mô hình Học viên" tối ưu cho bài toán dự đoán vị trí PTM, NCS đã thiết kế và thực nghiệm bốn phương án kết hợp mô hình khác nhau. Việc thử nghiệm với các kiến trúc đa dạng giúp đánh giá tác động của việc kết hợp giữa các loại mạng khác nhau (CNN1D và Bi-LSTM), từ đó lựa chọn được cặp mô hình phù hợp nhất về mặt hiệu suất và khả năng khái quát hoá.

**Bốn mô hình chắt lọc tri thức được thiết kế và đánh giá gồm:**

Mô hình KD1: "mô hình Giáo viên" là mô hình CNN1D 6 lớp và "mô hình Học viên" là mô hình CNN1D 3 lớp.

Mô hình KD2 (KD_ArapUbi): "mô hình Giáo viên" là mô hình Bi-LSTM 32 đơn vị và "mô hình Học viên" là mô hình Bi-LSTM 16 đơn vị.

Mô hình KD3: "mô hình Giáo viên" là mô hình CNN1D 6 lớp và "mô hình Học viên" là mô hình Bi-LSTM 16 đơn vị.

Mô hình KD4: "mô hình Giáo viên" là mô hình Bi-LSTM 32 đơn vị và "mô hình Học viên" là mô hình CNN1D 3 lớp.

NCS xây dựng bốn phương án kiến trúc học chắt lọc tri thức (KD1–KD4) với sự kết hợp khác nhau giữa “mô hình Giáo viên” và “mô hình Học viên”. Để lựa chọn được mô hình cho bài toán dự đoán vị trí PTM, các phương án này được đánh giá trên tập dữ liệu chuẩn bằng phương pháp kiểm thử chéo 5 lần, với các tỷ lệ mẫu dương/âm khác nhau (1:1, 1:2, 1:3) và các kỹ thuật tách từ n-gram được thực nghiệm (1-gram, 2-gram, 3-gram) trước khi đưa vào Embedding của mô hình.

Kết quả kiểm thử chéo 5 lần của các mô hình được đánh giá với ba chỉ số chính là ACC, MCC và AUC được trình bày trong Bảng 4.2, Bảng 4.3 và Bảng 4.4.

**Bảng 4.2 ACC (%) của bốn mô hình KD dựa trên kiểm thử chéo 5 lần**

| Model | Ratio 1:1 | | | Ratio 1:2 | | | Ratio 1:3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1-gram | 2-gram | 3-gram | 1-gram | 2-gram | 3-gram | 1-gram | 2-gram | 3-gram |
| KD1 | 84.8 | 79.6 | 66.7 | 83.5 | 76.1 | 65.0 | 84.2 | 70.6 | 64.6 |
| KD2 | **86.3** | **80.0** | 66.1 | **85.1** | **77.7** | 65.2 | 84.1 | **77.2** | **66.1** |
| KD3 | 85.3 | 76.5 | **66.8** | 84.5 | 70.1 | **70.1** | 83.3 | **77.2** | 68.1 |
| KD4 | 83.9 | 79.3 | 63.0 | 83.8 | 74.1 | 62.4 | 82.6 | 72.0 | 62.4 |

**Bảng 4.3 MCC của bốn mô hình KD dựa trên kiểm thử chéo 5 lần**

| Model | Ratio 1:1 | | | Ratio 1:2 | | | Ratio 1:3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1-gram | 2-gram | 3-gram | 1-gram | 2-gram | 3-gram | 1-gram | 2-gram | 3-gram |
| KD1 | 0.699 | **0.617** | 0.347 | 0.660 | 0.567 | **0.371** | 0.636 | 0.494 | 0.341 |
| KD2 | **0.730** | 0.598 | 0.343 | **0.695** | 0.548 | 0.357 | **0.655** | **0.534** | **0.357** |
| KD3 | 0.712 | 0.534 | **0.359** | 0.674 | 0.384 | 0.384 | 0.636 | 0.495 | 0.351 |
| KD4 | 0.682 | 0.594 | 0.303 | 0.670 | 0.536 | 0.324 | 0.627 | 0.484 | 0.334 |

**Bảng 4.4 AUC của bốn mô hình KD dựa trên kiểm thử chéo 5 lần**

| Model | Ratio 1:1 | | | Ratio 1:2 | | | Ratio 1:3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1-gram | 2-gram | 3-gram | 1-gram | 2-gram | 3-gram | 1-gram | 2-gram | 3-gram |
| KD1 | 0.913 | 0.861 | 0.740 | 0.888 | 0.842 | 0.742 | 0.901 | 0.803 | 0.761 |
| KD2 | **0.926** | **0.872** | 0.708 | 0.922 | **0.860** | 0.738 | 0.915 | 0.851 | 0.756 |
| KD3 | 0.921 | 0.829 | **0.709** | **0.925** | 0.755 | **0.757** | **0.923** | **0.862** | **0.780** |
| KD4 | 0.916 | 0.867 | 0.695 | 0.897 | 0.820 | 0.727 | 0.892 | 0.805 | 0.732 |

Kết quả Bảng 4.2, Bảng 4.3 và Bảng 4.4cho thấy mô hình KD2 (KD_ArapUbi) đạt hiệu suất tổng thể cao và ổn định hơn các phương án còn lại, đặc biệt trên thang đo ACC

và AUC ở hầu hết các tỷ lệ dữ liệu. Trong khi KD3 có một vài trường hợp nổi bật ở 3-gram, nhưng kết quả thiếu ổn định giữa các tỷ lệ. KD1 và KD4 nhìn chung có hiệu suất thấp hơn. Xét chi tiết hơn, mỗi mô hình KD có những ưu điểm và hạn chế riêng:

- **KD1 (CNN1D–CNN1D):** Kết quả ở mức trung bình, ACC và AUC thường thấp hơn so với KD2 và KD3, đặc biệt khi sử dụng 3-gram. Mô hình này nhìn chung ít nổi bật và kém ổn định trong so sánh.
- **KD2 (Bi-LSTM–Bi-LSTM):** Là mô hình có hiệu suất cao và ổn định nhất, nổi bật ở cả ACC, MCC và AUC trên hầu hết các cấu hình dữ liệu. Đây là lựa chọn phù hợp nhất khi cần mô hình tổng quát hoá tốt và độ chính xác cao.
- **KD3 (CNN1D–Bi-LSTM):** Có một số kết quả nổi bật ở 3-gram, đặc biệt về ACC và AUC, cho thấy khả năng khai thác ngữ cảnh tốt hơn trong những tình huống nhất định. Tuy nhiên, hiệu suất không ổn định giữa các tỷ lệ dữ liệu, làm giảm tính tin cậy khi áp dụng rộng rãi.
- **KD4 (Bi-LSTM–CNN1D):** Thường có kết quả thấp nhất trong cả ba chỉ số ACC, MCC và AUC. Mô hình này thể hiện hạn chế rõ ràng trong việc duy trì hiệu suất dự đoán so với các phương án khác.

Từ phân tích trên có thể thấy, mặc dù KD3 cho thấy một số tiềm năng trong cấu hình 3-gram, và KD1/KD4 thể hiện hiệu suất khiêm tốn hơn, KD2 vẫn là phương án tốt nhất nhờ tính ổn định và độ chính xác vượt trội. Hơn nữa từ các bảng kết quả trên cho thấy tất cả kiến trúc mô hình đều cho hiệu suất cao khi sử dụng kỹ thuật 1-gram (mỗi axit amin là một từ). Vì vậy, KD2 sau đây sẽ được gọi là KD_ArapUbi với kỹ thuật tách từ 1-gram được lựa chọn làm mô hình dự đoán PTM đề xuất trong nghiên cứu này.

### *4.2.4 Kiến trúc học chắt lọc tri thức dự đoán vị tí Ubiquitination ở loài Arabidopsis thaliana (KD_ArapUbi)*

Mô hình KD_ArapUbi được thiết kế dựa trên ý tưởng học chắt lọc tri thức (Knowledge Distillation) truyền thống của Hinton [41], trong đó "mô hình Học viên" có kiến trúc nhỏ gọn hơn về số lượng nút mạng, tài nguyên huấn luyện và trên loài thuộc họ PTM Ubiquitination. Tuy nhiên, so với mô hình gốc của Hinton, KD_ArapUbi được thiết kế với hai điểm cải tiến sau:

Thứ nhất, "mô hình Giáo viên" được xây dựng trên kiến trúc Bi-LSTM với số nút mạng gấp đôi Học viên, đồng thời được huấn luyện trên tập dữ liệu đa loài bao gồm *Arabidopsis thaliana*, *Oryza sativa subsp. indica*, và *Oryza sativa subsp. japonica*. Điều này giúp mô hình Giáo viên có khả năng học được tri thức khái quát và đa dạng

hơn, từ đó truyền đạt cho Học viên vốn chỉ huấn luyện trên dữ liệu hạn chế của một loài duy nhất là *Arabidopsis thaliana*. Cách tiếp cận này khác biệt với mô hình Hinton truyền thống, vốn chỉ giả định rằng Giáo viên và "mô hình Học viên" được huấn luyện trên cùng một tập dữ liệu, làm hạn chế khả năng mở rộng tri thức.

Thứ hai, NCS đã khéo léo tích hợp kỹ thuật *embedding động* vào cả "mô hình Giáo viên" và "mô hình Học viên". Việc bổ sung embedding động tương tự như ở trong mục 3.3.3 ở chương 3 giúp biểu diễn đặc trưng linh hoạt hơn, thích ứng tốt với sự đa dạng của chuỗi protein và giảm thiểu sự mất mát thông tin khi truyền tri thức từ Giáo viên sang Học viên.

Nhờ hai điểm cải tiến này, KD_ArapUbi không chỉ kế thừa tính hiệu quả của mô hình chắt lọc tri thức truyền thống mà còn mở rộng năng lực khái quát và tăng cường khả năng biểu diễn, đặc biệt trong bối cảnh dự đoán PTM với dữ liệu hạn chế.

Hình 4.5 mô tả toàn bộ kiến trúc của mô hình KD_ArapUbi. Về phía "mô hình Giáo viên", dữ liệu huấn luyện và kiểm thử được thu thập từ ba loài *Arabidopsis thaliana*, *Oryza sativa subsp. indica* và *Oryza sativa subsp. japonica*. Chuỗi protein đầu vào được xử lý bằng cửa sổ trượt (sliding window, WS = 31) và mã hóa bằng kỹ thuật *n-gram tokenization*. Sau bước tiền xử lý, chuỗi được ánh xạ thành các chỉ số embedding động, rồi đưa vào kiến trúc Bi-LSTM nhiều tầng với số lượng nút mạng lớn, giúp "mô hình Giáo viên" học được đặc trưng ngữ cảnh giàu thông tin và tri thức khái quát hơn từ dữ liệu đa loài.

Ở phía "mô hình Học viên", dữ liệu huấn luyện chỉ giới hạn ở loài *Arabidopsis thaliana*, cũng được xử lý tương tự bằng cửa sổ trượt và tokenization. Đầu vào sau khi embedding động được đưa vào kiến trúc Bi-LSTM nhỏ gọn hơn, với số nút mạng ít hơn nhằm giảm độ phức tạp và tài nguyên tính toán. Quá trình truyền tri thức (*knowledge transfer*) diễn ra qua hai bước: *distillation* và *transfer*, trong đó Học viên không chỉ học trực tiếp từ dữ liệu thực tế mà còn từ tri thức do Giáo viên cung cấp bởi nhãn mềm. Nhờ đó, "mô hình Học viên" có thể đạt được năng lực dự đoán cao, nhưng với chi phí huấn luyện thấp hơn đáng kể.

**Quá trình học chắt lọc tri thức giữa "mô hình Giáo viên" và "mô hình Học viên" được thực hiện qua các bước sau:**

**Bước 1:** Thiết lập kiến trúc mạng cho "mô hình Giáo viên", Thiết lập kiến trúc mạng cho "mô hình Học viên".

- "Mô hình Giáo viên" có kiến trúc sâu và rộng hơn, với số lượng tham số lớn, nhằm khai thác được tri thức tiềm ẩn từ một tập dữ liệu lớn.

- "Mô hình Học viên" có kiến trúc nhẹ hơn, phù hợp cho mục tiêu triển khai trong môi trường hạn chế tài nguyên.

**Bước 2:** Huấn luyện "mô hình Giáo viên" trên tập dữ liệu đa loài

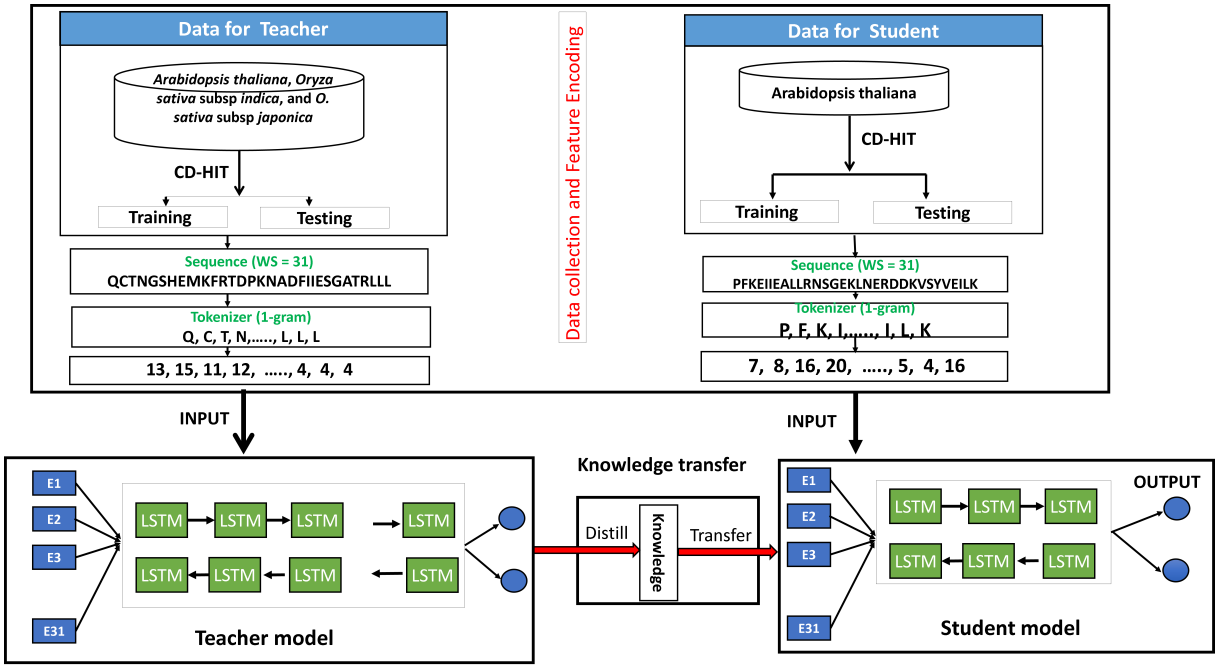
- "Mô hình Giáo viên" với kiến trúc phức tạp hơn (nhiều tham số hơn) được huấn luyện trên một tập dữ liệu lớn và đa dạng, đại diện cho không gian tri thức rộng.

- Quá trình huấn luyện này giúp Giáo viên học được các đặc trưng trừu tượng, mối quan hệ ngữ nghĩa và các quy luật tiềm ẩn trong dữ liệu, từ đó đạt được khả năng khái quát tốt và dự đoán chính xác.

**Bước 3:** Huấn luyện "mô hình Học viên" với sự hỗ trợ từ "mô hình Giáo viên"

Trong giai đoạn huấn luyện, "mô hình Giáo viên" đóng vai trò hỗ trợ bằng cách cung cấp thông tin tri thức cho "mô hình Học viên" thông qua nhãn mềm.

**Bước 4:** Đánh giá hiệu suất "mô hình Học viên" trên tập dữ liệu kiểm thử




**Hình 4.5 Kiến trúc mô hình học chắt lọc tri thức KD_ArapUbi đề xuất**

Giả mã cho mô hình học chắt lọc tri thức đề xuất dự đoán vị trí PTM được trình bày trong các Algomrithm 4.5 và Algomrithm 4.6:

**Algorithm 4.5** Huấn luyện "mô hình Giáo viên"

**Đầu vào:**

Tập dữ liệu lớn: $\mathscr{D}_{\text{large}}$

Kiến trúc "mô hình Giáo viên"

**Đầu ra:** "mô hình Giáo viên" đã huấn luyện

Khởi tạo "mô hình Giáo viên" với kiến trúc đã chọn

**for all** mini-batch $\{(x,y)\} \subset \mathscr{D}_{\text{large}}$ **do**

    Dự đoán $\hat{y} = f_{\theta_T}(x)$

    Tính hàm mất mát $\mathscr{L}_{\text{CE}} = \mathscr{L}_{\text{CE}}(y,\hat{y})$

    Cập nhật tham số $\theta_T$

**end for**

Lưu "mô hình Giáo viên" đã huấn luyện

**Algorithm 4.6** Huấn luyện "mô hình Học viên" với tri thức từ Giáo viên

**Đầu vào:**

"mô hình Giáo viên" đã huấn luyện

Tập huấn luyện $\mathscr{D}_{\text{train}}$, tập kiểm thử $\mathscr{D}_{\text{test}}$

Siêu tham số nhiệt độ $\tau \in \{0.1, 0.3, 0.5, 0.7\}$; Hệ số cân bằng $\alpha \in \{0.1, 0.3, 0.5, 0.7\}$

**Đầu ra:**

Tham số tối ưu $\theta^*$

**for all** $\tau \in \{\tau\}$ **do**

  **for all** $\alpha \in \{\alpha\}$ **do**

    Khởi tạo "mô hình Học viên" với kiến trúc đã chọn

    **for** $e = 1$ **to** $N_{\text{epoch}}$ **do**

      **for all** mini-batch $\{(x, y)\} \subset \mathscr{D}_{\text{train}}$ **do**

        Tính logits từ "mô hình Giáo viên": $Z_T(x)$

        Tính phân phối mềm của Giáo viên:

$$y_t(x) = \frac{\exp(Z_t(x)/\tau)}{\sum_j \exp(Z_t(x)(j)/\tau)}$$

        Tính logits từ "mô hình Học viên": $Z_s(x)$

        Tính phân phối mềm của Học viên:

$$y_S(x) = \frac{\exp(Z_s(x)/\tau)}{\sum_j \exp(Z_s(x)(j)/\tau)}$$

        Tính loss học chắt lọc tri thức:

$$\mathscr{L}_{\text{KD}} = \tau^2 \cdot \text{KL}(y_t, y_s)$$

        Tính loss cross-entropy với nhãn thật:

$$\mathscr{L}_{\text{CE}} = \text{CE}(y, \text{softmax}(Z_s))$$

        Tính tổng loss:

$$\mathscr{L}_{\text{total}} = (1 - \alpha) \cdot \mathscr{L}_{\text{CE}} + \alpha \cdot \mathscr{L}_{\text{KD}}$$

        Cập nhật tham số:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathscr{L}_{\text{total}}$$

      **end for**

    **end for**

    **return** $\theta^*$

### *4.2.5 Chiến lược và tham số huấn luyện mô hình*

Mô hình được huấn luyện trên môi trường Google Colab với GPU, sử dụng thuật toán tối ưu hóa Adam (learning rate = 0.0001), batch size = 16, số epoch = 100. Dropout được áp dụng nhằm hạn chế hiện tượng overfitting. Chi tiết các tham số của kiến trúc mô hình được trình bày trong Bảng 4.5.

**Bảng 4.5 So sánh kiến trúc mô hình Giáo viên và mô hình Học viên trong học chắt lọc tri thức**

| **Layer (type)** | **Giáo viên (Output Shape / Params)** | **Học viên (Output Shape / Params)** |
|---|---|---|
| Embedding | (None, 31, 300) / 6,600 | (None, 31, 300) / 6,600 |
| Bi-LSTM | (None, 31, 64) / 85,248 | (None, 31, 32) / 40,576 |
| Dropout | (None, 31, 64) / 0 | (None, 31, 32) / 0 |
| Flatten | (None, 1984) / 0 | (None, 992) / 0 |
| Dense | (None, 128) / 254,080 | (None, 128) / 127,104 |
| Activation | (None, 128) / 0 | (None, 128) / 0 |
| Dropout | (None, 128) / 0 | (None, 128) / 0 |
| Dense | (None, 2) / 258 | (None, 2) / 258 |
| Activation | (None, 2) / 0 | (None, 2) / 0 |
| **Tổng số tham số** | **346,188 (1.32 MB)** | **174,538 (681.79 KB)** |
| **Tỉ lệ rút gọn của Học viên** | **49.6% số tham số của Giáo viên** | |
| Tổng số tham số huấn luyện | 346,186 | 174,538 |
| Tổng tham số không tham gia huấn luyện | 0 | 0 |

Bảng 4.5 cho thấy: (i) Mô hình Giáo viên được thiết kế với khả năng biểu diễn mạnh hơn, thể hiện rõ qua 85,248 tham số của lớp Bi-LSTM (với 64 đơn vị) và 254,080 tham số của lớp Dense. Kích thước lớn hơn này cho phép "mô hình Giáo viên" học được một "tri thức" phong phú và phân biệt hơn.(ii) Ngược lại, "mô hình Học viên" được xây dựng bằng cách giảm một nửa số đơn vị ẩn trong lớp Bi-LSTM (còn 32 đơn vị) và giảm một nửa kích thước đầu vào cho lớp Dense(từ 1984 xuống 992). Tổng tham số giảm xuống còn 174,538 (gần 50%). Việc sử dụng cơ chế "Học chắt lọc tri thức" đảm bảo "mô hình Học viên" có thể đạt được hiệu suất tương đương mô hình Giáo viên mà chỉ cần sử dụng một nửa tài nguyên tính toán.

**Hàm mất mát:**

Trong học chắt lọc tri thức, mục tiêu của "mô hình Học viên" là vừa học khớp với nhãn cứng (ground truth), vừa học theo nhãn mềm do "mô hình Giáo viên" cung cấp.

Gọi $Z_t$ và $Z_s$ lần lượt là các logit của "mô hình Giáo viên" và "mô hình Học viên". Phân phối xác suất của mô hình Giáo viên và mô hình Học viên được định nghĩa trong Công thức 4.1 và Công thức 4.2:

$$y_t = \delta\left(\frac{Z_t}{\tau}\right) = \frac{\exp(Z_t/\tau)}{\sum_j \exp(Z_t(j)/\tau)} \tag{4.1}$$

$$y_s = \delta\left(\frac{Z_s}{\tau}\right) = \frac{\exp(Z_s/\tau)}{\sum_j \exp(Z_s(j)/\tau)} \tag{4.2}$$

Trong đó $\tau > 1$ là hệ số nhiệt độ, giúp làm mềm phân phối xác suất.

*Thành phần mất mát tri thức mềm:*

$$L_{KD} = \tau^2 \cdot \mathrm{KL}(y_s, y_t) = \tau^2 \cdot \mathrm{KL}\left(\delta\left(\tfrac{Z_s}{\tau}\right), \delta\left(\tfrac{Z_t}{\tau}\right)\right) \tag{4.3}$$

*Thành phần mất mát nhãn cứng:*

$$L_{CE} = \mathrm{CrossEntropy}(\delta(Z_s), y) \tag{4.4}$$

*Hàm mất mát tổng:*

$$L_{\text{total}} = (1-\alpha)L_{CE} + \alpha L_{KD} \tag{4.5}$$

Hay viết đầy đủ:

$$L_{\text{total}} = (1-\alpha)\cdot \mathrm{CrossEntropy}(\delta(Z_s), y) + \alpha \cdot \tau^2 \cdot \mathrm{KL}\left(\delta\left(\tfrac{Z_s}{\tau}\right), \delta\left(\tfrac{Z_t}{\tau}\right)\right) \tag{4.6}$$

Như vậy, hàm mất mát tổng hợp $L_{\text{total}}$ kết hợp cả hai nguồn tri thức: (i) tri thức từ dữ liệu gốc thông qua thành phần $L_{CE}$ giúp "mô hình Học viên" duy trì khả năng dự đoán chính xác theo nhãn cứng, và (ii) tri thức từ "mô hình Giáo viên" thông qua thành phần $L_{KD}$, cho phép "mô hình Học viên" học được các mối quan hệ tiềm ẩn trong phân phối xác suất mềm (nhãn mềm). Hệ số $\alpha$ đóng vai trò điều chỉnh mức độ ưu tiên giữa việc học theo nhãn thật và học theo phân phối từ "mô hình Giáo viên". Trong khi đó, tham số nhiệt độ $\tau$ có tác dụng làm trơn phân phối xác suất, giúp "mô hình Học viên" dễ tiếp thu thông tin ẩn về mức độ tương quan giữa các lớp. Sự phối hợp hài hòa của hai yếu tố này cho phép "mô hình Học viên" vừa đảm bảo tính chính xác, vừa nâng cao khả năng khái quát hóa khi dữ liệu huấn luyện bị hạn chế.

Hàm mất mát gồm hai thành phần: Binary Cross-Entropy và Kullback–Leibler divergence (KL) cho học chắt lọc tri thức. Hai siêu tham số $\alpha$ (trọng số cân bằng)

và $\tau$ (hệ số nhiệt) được khảo sát kỹ lưỡng bằng kiểm thử chéo 5 lần, với các giá trị $\alpha \in \{0.1, 0.3, 0.5, 0.7\}$ và $\tau \in \{5, 10, 15\}$.

**Bảng 4.6 Ảnh hưởng của $\alpha$ và $\tau$ đến học chắt lọc tri thức**

| $\alpha$ | $\tau$ | ACC | MCC | $\alpha$ | $\tau$ | ACC | MCC |
|---|---|---|---|---|---|---|---|
| 0.1 | 5 | 0.855 | 0.715 | 0.5 | 10 | 0.847 | 0.697 |
| 0.3 | 5 | 0.854 | 0.711 | 0.7 | 10 | 0.846 | 0.692 |
| 0.5 | 5 | 0.840 | 0.680 | 0.1 | 15 | 0.861 | 0.720 |
| 0.7 | 5 | 0.839 | 0.681 | 0.3 | 15 | 0.849 | 0.700 |
| **0.1** | **10** | **0.863** | **0.730** | 0.5 | 15 | 0.837 | 0.676 |
| 0.3 | 10 | 0.855 | 0.711 | 0.7 | 15 | 0.843 | 0.686 |

Bảng 4.6 cho thấy:

Khi $\alpha$ quá lớn (0.5 hoặc 0.7), "mô hình Học viên" phụ thuộc nhiều vào phân phối của"mô hình Giáo viên", dẫn đến suy giảm hiệu suất. Ngược lại, với $\alpha$ nhỏ (0.1 hoặc 0.3), kết quả ổn định và cao hơn.

Khi $\tau$ tăng từ 5 lên 10, hiệu năng cải thiện rõ rệt, đạt tốt nhất tại $\alpha = 0.1, \tau = 10$ (ACC = 0.863, MCC = 0.730). Tuy nhiên, khi $\tau$ tăng lên 15, hiệu suất có xu hướng giảm nhẹ.

Do đó, cấu hình $\alpha = 0.1, \tau = 10$ được lựa chọn là tối ưu, vừa khai thác hiệu quả kiến thức từ "mô hình Giáo viên", vừa duy trì khả năng học từ nhãn thật, giúp "mô hình Học viên" đạt hiệu suất dự đoán cao và ổn định.

### *4.2.6 Kết quả và thảo luận*

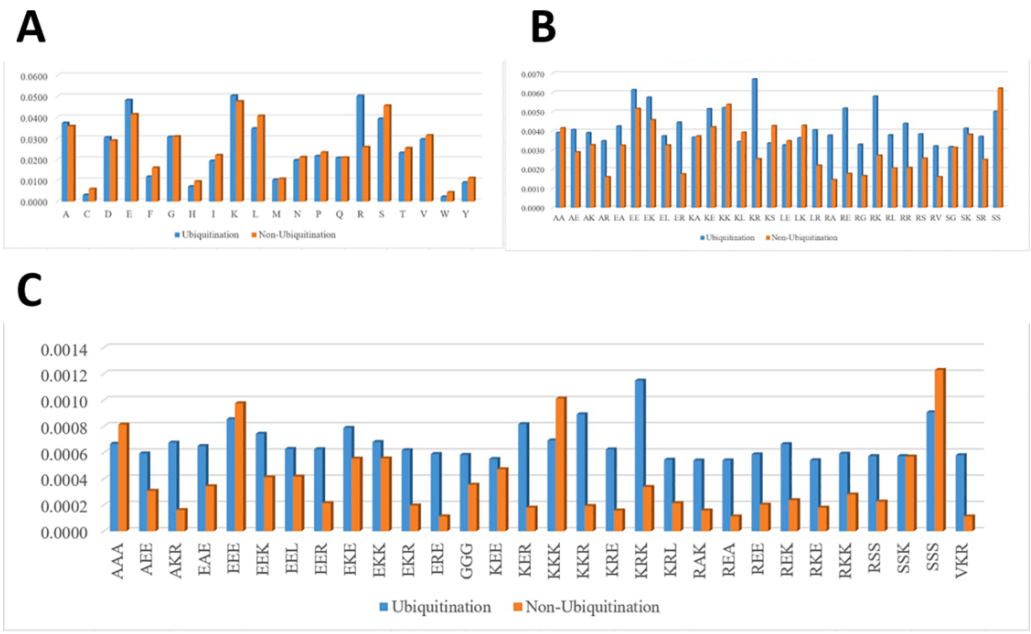
**Phân tích tần suất xuất hiện của các từ trong dữ liệu huấn luyện:**

Như đã trình bày trong phần phương pháp, nghiên cứu này sử dụng kỹ thuật mã hóa embedding để biểu diễn các chuỗi protein, coi protein như những câu và các axit amin riêng lẻ như các từ trong mô hình ngôn ngữ. Tần suất xuất hiện của các n-gram trong tập dữ liệu huấn luyện đóng vai trò quan trọng, ảnh hưởng trực tiếp đến quá trình huấn luyện embedding và từ đó tác động đến hiệu quả của mô hình dự đoán.

Phân tích tần suất 1-gram, thể hiện phân bố tần suất của từng axit amin đơn lẻ trong tập dữ liệu huấn luyện, được minh họa trong Hình 4.6 (A). Kết quả cho thấy bốn axit

amin có tần suất xuất hiện cao nhất trong tập dữ liệu dương (có gắn ubiquitin) bao gồm lysine (K), arginine (R), glutamic acid (E), và serine (S). Tiếp theo, Hình 4.6 (B) trình bày 30 cặp axit amin liên tiếp (2-gram) xuất hiện nhiều nhất trong tập dữ liệu. Trong tập dương, các cặp phổ biến nhất là KR, EE, EK và PK, trong khi đó ở tập âm (không ubiquitin hóa), các cặp SS, KK, EE và EK lại chiếm ưu thế. Hình 4.6 (C) minh họa 30 bộ ba axit amin liên tiếp (3-gram) có tần suất cao nhất. Trong tập dữ liệu dương, các bộ ba phổ biến gồm KRK, SSS, KKR và EEE, trong khi ở tập dữ liệu âm, các bộ ba xuất hiện nhiều nhất là SSS, KKK, EEE và AAA. Những kết quả này cho thấy sự khác biệt rõ rệt về mẫu trình tự đặc trưng giữa hai nhóm dữ liệu (dương tính và âm tính).

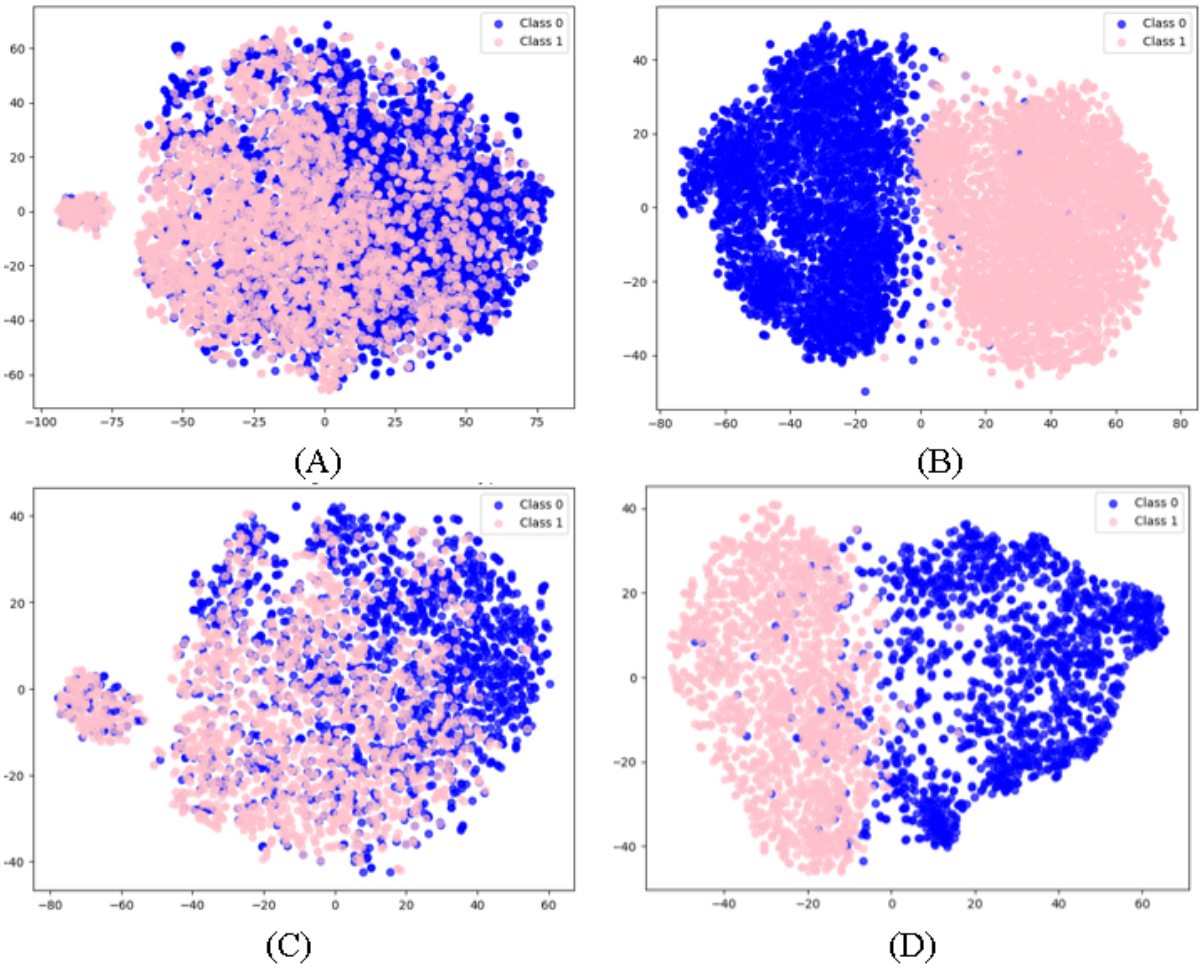
**Trực quan dữ liệu huấn luyện *Arabidopsis thaliana*:**




**Hình 4.6 tần suất xuất hiện n-gram trong bộ dữ liệu huấn luyện (A) Tần suất xuất hiện của các axit amin đơn lẻ (1-gram), (B) 30 cặp axit amin liên tiếp xuất hiện nhiều nhất (2-gram), (C) 30 bộ ba axit amin liên tiếp xuất hiện nhiều nhất (3-gram)**

**Đánh giá hiệu quả mô hình học chắt lọc tri thức đề xuất thông qua trực quan hóa t-SNE**

Nhằm phân tích sâu hơn về khả năng học đặc trưng và hiệu quả phân loại của mô hình, NCS đã tiến hành trực quan hoá bằng phương pháp t-SNE [66]. Kết quả được trình bày trong Hình 4.7 gồm bốn biểu đồ thể hiện sự phân bố của các mẫu dữ liệu trong không gian hai chiều trước khi huấn luyện mô hình và sau khi được huấn luyện bởi mô hình.




**Hình 4.7 Trực quan hoá T-sne (A) Dữ liệu của "mô hình Giáo viên" (các loài thực vật) trước khi huấn luyện, (B) Dữ liệu sau khi được huấn luyện bởi "mô hình Giáo viên", (C) Dữ liệu loài *Arabidopsis thaliana* trước khi huấn luyện, (D) Dữ liệu sau khi huấn luyện bởi mô hình học chắt lọc tri thức KD_ARAPUBI đề xuất ("mô hình Học viên" được hướng dẫn bởi "mô hình Giáo viên" đã được huấn luyện trên bộ dữ liệu đa loài). (Class 0: Mẫu âm tính, Class 1: Mẫu dương tính)**

Hình 4.7 (A) là trực quan hoá sự phân bố dữ liệu gốc của các loài thực vật - bộ dữ liệu của "mô hình Giáo viên" trước khi huấn luyện. Quan sát cho thấy các điểm dữ liệu của hai lớp (class 0 và class 1) phân bố chồng lấn nhau rất lớn. Sau quá trình huấn luyện với "mô hình Giáo viên" (Hình 4.7 (B)), không gian đặc trưng đã phân tách rõ rệt giữa hai lớp. Các điểm dữ liệu thuộc hai lớp hình thành hai cụm riêng biệt, chứng tỏ "mô hình Giáo viên" đã học được các đặc trưng có khả năng phân biệt mạnh mẽ. Điều này khẳng định hiệu quả của "mô hình Giáo viên" khi được huấn luyện trên tập dữ liệu đa loài, giúp thu nhận tri thức tổng quát về đặc điểm nhận dạng vị trí Ubiquitination.

Dữ liệu huấn luyện của "mô hình Học viên" (Hình 4.7 (C)) - dữ liệu huấn luyện hạn chế về số lượng, các điểm dữ liệu của hai lớp phân bố đan xen. Đây là các thách

thức trong việc xây dựng mô hình dự đoán ubiquitination riêng cho từng loài, đặc biệt là trong điều kiện dữ liệu hạn chế.

Tuy nhiên, khi áp dụng phương pháp học chắt lọc tri thức, "mô hình Học viên" sau khi được huấn luyện với tri thức truyền từ "mô hình Giáo viên" đã cải thiện đáng kể khả năng phân biệt hai lớp trên dữ liệu *Arabidopsis thaliana* ((Hình 4.7 (D)), hình thành hai cụm tách biệt rõ so với trước khi áp dụng học chắt lọc tri thức.

Các kết quả thực nghiệm cho thấy, khi kết hợp embedding động với phương pháp học chắt lọc tri thức, "mô hình Học viên" đã có sự cải thiện rõ rệt về khả năng phân biệt hai lớp trên dữ liệu *Arabidopsis thaliana* - dữ liệu hạn chế(Hình 4.7 (D)), thể hiện qua việc hình thành hai cụm dữ liệu tách biệt hơn so với trước khi áp dụng chắt lọc tri thức. Embedding động đóng vai trò nền tảng giúp mô hình biểu diễn giàu ngữ nghĩa hơn từ dữ liệu thô, trong khi quá trình chắt lọc tri thức giúp Học viên học được những đặc trưng tinh chỉnh, có giá trị phân biệt cao từ "mô hình Giáo viên".

Kết hợp embedding động và chắt lọc tri thức mở ra hướng tiếp cận hứa hẹn trong việc xây dựng các mô hình nhẹ hơn nhưng vẫn đảm bảo hiệu suất cao, đặc biệt với các bài toán phân loại PTM trên loài *Arabidopsis thaliana* hoặc các loài khác trong tương lai.

**So sánh hiệu suất dự đoán của mô hình học chắt lọc tri thức với "mô hình Giáo viên" và "mô hình Học viên" học không có hướng dẫn của giáo viên.**

Kết quả trong Bảng 4.7 và Bảng 4.8 thể hiện rõ hiệu quả của mô hình học chắt lọc tri thức (KD_ArapUbi) so với các mô hình đối chứng khác trong cả hai kịch bản đánh giá kiểm thử chéo và kiểm thử độc lập. Cụ thể, trong thử nghiệm kiểm thử chéo (Bảng 4.7), mô hình học chắt lọc tri thức đạt (ACC = 0.863, MCC = 0.729 và AUC = 0.932), cao hơn đáng kể so với "mô hình Giáo viên" (ACC = 0.844, MCC = 0.691, AUC = 0.917) và "mô hình Học viên" không có hướng dẫn (ACC = 0.835, MCC = 0.673, AUC = 0.917). Xu hướng tương tự được duy trì trong bộ kiểm tra độc lập (Bảng 4.8), mô hình học chắt lọc tri thức tiếp tục khẳng định về hiệu suất dự đoán vượt trội hơn với (ACC = 0.865, MCC = 0.734 và AUC = 0.927).

Những kết quả này cho thấy cơ chế chắt lọc tri thức thực sự mang lại hiệu quả, khi "mô hình Học viên" không chỉ tiếp thu được thông tin tổng quát từ dữ liệu gốc mà còn học được những tri thức tiềm ẩn, tinh chỉnh từ "mô hình Giáo viên", giúp cải thiện năng lực phân loại, đặc biệt trong việc giảm thiểu sai sót nhầm lẫn giữa các lớp (thể hiện qua MCC cao hơn).

**Bảng 4.7 Kết quả kiểm thử chéo của các mô hình**

| Mô hình | ACC | SEN | SPE | MCC | AUC |
|---|---|---|---|---|---|
| Mô hình Giáo viên | 0.844 | 0.812 | 0.882 | 0.691 | 0.917 |
| Mô hình Học viên | 0.835 | 0.805 | 0.872 | 0.673 | 0.917 |
| Mô hình học chắt lọc tri thức KD_ArapUbi | 0.863 | 0.835 | 0.897 | 0.729 | 0.932 |

**Bảng 4.8 Kết quả kiểm thử độc lập của các mô hình**

| Mô hình | ACC | SEN | SPE | MCC | AUC |
|---|---|---|---|---|---|
| Mô hình Giáo viên | 0.844 | 0.804 | 0.896 | 0.695 | 0.919 |
| Mô hình Học viên | 0.841 | 0.799 | 0.895 | 0.688 | 0.917 |
| Mô hình học chắt lọc tri thức KD_ArapUbi | 0.865 | 0.830 | 0.908 | 0.734 | 0.927 |

**Đánh giá khả năng nén mô hình và tiết kiệm tài nguyên của phương pháp chắt lọc tri thức**

Bên cạnh hiệu suất dự đoán, một yếu tố quan trọng khác của mô hình học chắt lọc tri thức là khả năng giảm số lượng tham số, từ đó giúp tiết kiệm bộ nhớ và tài nguyên tính toán. Trong mô hình này, cả Giáo viên và Học viên đều sử dụng kiến trúc Bi-LSTM, tuy nhiên số lượng nút của lớp Bi-LSTM trong "mô hình Giáo viên" gấp đôi so với Học viên. Cụ thể, "mô hình Giáo viên" có 346,186 tham số với dung lượng 1351.68 KB, trong khi "mô hình Học viên" chỉ có 174,540 tham số với dung lượng 681.80 KB, tức giảm gần 50% số lượng tham số.

**Bảng 4.9 So sánh mức sử dụng tài nguyên giữa mô hình gốc và KD_ArapUbi**

| Mô hình | Số lượng tham số | Bộ nhớ |
|---|---|---|
| Mô hình Giáo viên | 346,186 | 1351.68 KB |
| Mô hình Học viên | 174,540 | 681.80 KB |
| Mô hình KD_ArapUbi | 174,540 | 681.80 KB |

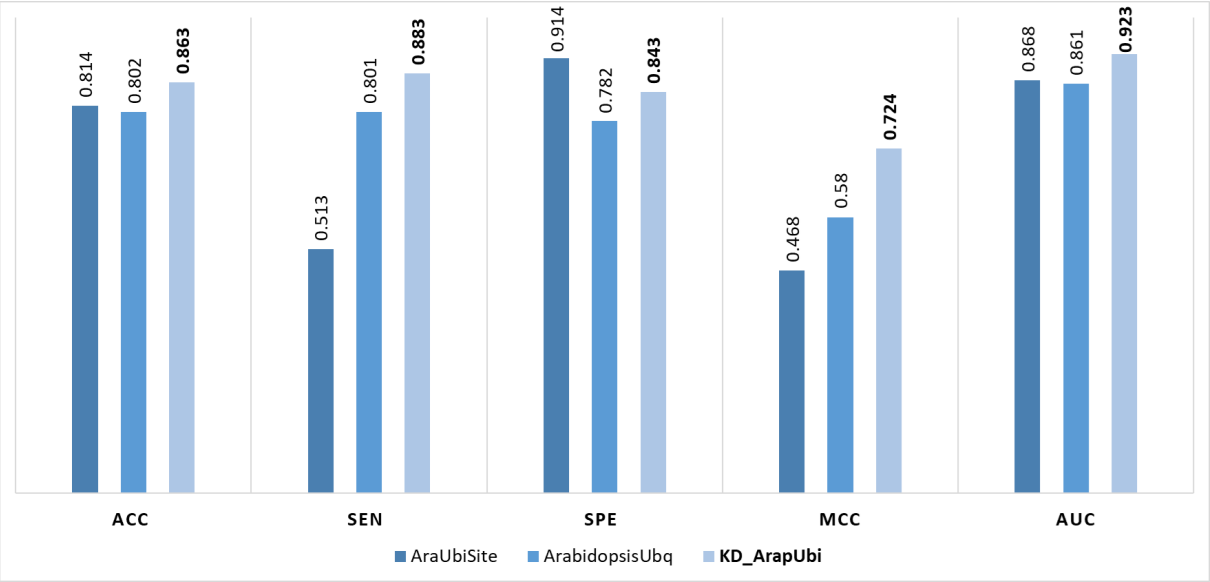
Mặc dù kích thước nhỏ hơn, "mô hình Học viên" khi được huấn luyện với chiến

lược chắt lọc tri thức vẫn đạt hiệu suất cao, tiệm cận "mô hình Giáo viên". Việc giảm số lượng tham số không chỉ giúp giảm chi phí lưu trữ mà còn làm tăng tốc độ suy luận, đặc biệt hữu ích khi triển khai trên các thiết bị có tài nguyên hạn chế hoặc khi xử lý các tập dữ liệu lớn. Điều này minh chứng cho tính hiệu quả và khả năng ứng dụng thực tế của phương pháp chắt lọc tri thức trong bài toán dự đoán vị trí ubiquitin hoá protein.

## 4.3 So sánh mô hình đề xuất với các công cụ hiện có về dự đoán *Arabidopsis thaliana*

Để đánh giá khả năng và tính thực tiễn của mô hình dự đoán đề xuất, việc so sánh hiệu suất của mô hình đề xuất với các công cụ dự đoán *Arabidopsis thaliana* hiện khác là cần thiết. AraUbiSite [13] và ArabidopsisUbq [72] là hai công cụ dự đoán *Arabidopsis thaliana* mới nhất. Do đó, NCS sử dụng hai công cụ này để so sánh với mô hình đề xuất trên cùng tập kiểm thử độc lập. Kết quả trong so sánh hiển thị trong Hình 4.8 cho thấy mô hình KD_ArapUbi đề xuất có hiệu suất tốt hơn, đạt độ chính xác 0.863, MCC 0.724 và AUC 0.923. Kết quả này khẳng định khả năng mạnh mẽ của mô hình trong dự đoán các vị trí Ubiquitination.

Hiệu suất vượt trội của mô hình KD_ArapUbi so với các công cụ dự đoán AraUbiSite, ArabidopsisUbq đến từ kiến trúc học chắt lọc tri thức kết hợp với embedding động. Cụ thể, KD_ArapUbi sử dụng một "mô hình Giáo viên" mạnh mẽ, đã được huấn luyện trên tập dữ liệu đa loài giàu tính khái quát, để truyền đạt tri thức hữu ích cho "mô hình Học viên" nhẹ hơn, tối ưu hóa riêng cho *Arabidopsis thaliana*. Thông qua quá trình chắt lọc tri thức, Học viên không chỉ học từ nhãn thật mà còn tiếp thu thêm “tri thức mềm” (soft labels) từ "mô hình Giáo viên", giúp tăng cường khả năng phân biệt các vị trí Ubiquitination. Kiến trúc Bi-LSTM của "mô hình Học viên" được lựa chọn nhờ khả năng học ngữ cảnh hai chiều trong chuỗi protein, đảm bảo hiệu quả cao khi kết hợp với cơ chế chắt lọc tri thức.




**Hình 4.8 So sánh mô hình đề xuất và các công cụ dự đoán *Arabidopsis thaliana***

Embedding động đóng vai trò bổ trợ quan trọng trong việc nâng cao hiệu quả mô hình. Khác với các phương pháp embedding tĩnh truyền thống, embedding động cho phép mô hình tự học biểu diễn tối ưu của các axit amin dựa trên ngữ cảnh và cấu trúc chuỗi protein, từ đó giảm thiểu mất mát thông tin và tăng tính linh hoạt. Đặc biệt, khi tích hợp vào mô hình học chắt lọc tri thức, embedding động giúp kết nối hiệu quả giữa tri thức tổng quát từ "mô hình Giáo viên" và biểu diễn đặc trưng tinh chỉnh phù hợp với dữ liệu *Arabidopsis thaliana*. Chính sự kết hợp giữa kiến trúc học sâu Bi-LSTM, embedding động và cơ chế chắt lọc tri thức đã tạo nên lợi thế vượt trội cho mô hình KD_ArapUbi, giúp nó đạt hiệu suất cao hơn so với các công cụ dự đoán hiện có.

## 4.4 Phân tích so sánh tổng thể bốn mô hình đề xuất trong luận án

Trong luận án, bốn mô hình dự đoán PTM (RSX_SUMO, CLW_SUMO, CBiL-Succsite và KD_ArapUbi) được đề xuất với kiến trúc và mục tiêu khác nhau, phản ánh sự đa dạng trong cách tiếp cận bài toán.

**Bảng 4.10 Đánh giá so sánh bốn mô hình được đề xuất trong luận án**

| Mô hình | RSX_SUMO | CLW_SUMO | CBiLSuccsite | KD_ArapUbi |
|---|---|---|---|---|
| Kỹ thuật xây dựng mô hình | Học máy tổ hợp (XGBoost, SVM, RF) | Học sâu lai (CNN–LSTM) | Học sâu lai (CNN–BiLSTM) | Học chắt lọc tri thức |
| Phương pháp mã hoá dữ liệu và trích chọn đặc trưng | Vector đặc trưng thủ công (AAIndex, CKSAAP, BLOSUM62, Word2Vec) | Embedding tĩnh (Word2Vec tiền huấn luyện) | Embedding động (lớp embedding được huấn luyện) | Embedding động (lớp embedding được huấn luyện) |
| Tổng số tham số | 4,387,000 ( 40.3MB) | 5,859,281 (22.35 MB) | 451,025 (1.72 MB) | 174,538 (681.79 KB) |
| Tham số huấn luyện được | 4,387,000 ( 40.3MB) | 302,081 (1.15 MB) | 451,025 (1.72 MB) | 174,538 (681.79 KB) |
| Thời gian huấn luyện | Cao | Cao | Trung bình | Thấp |
| Phù hợp với dữ liệu | Hạn chế | Vừa,lớn | Vừa | Hạn chế,vừa |

Dựa trên Bảng 4.10, có thể nhận thấy: Bốn mô hình được đề xuất trong luận án thể hiện sự đa dạng về kỹ thuật xây dựng mô hình dự đoán và phương pháp mã hoá đặc trưng, qua đó phản ánh các hướng tiếp cận khác nhau trong dự đoán vị trí biến đổi sau dịch mã.

Cụ thể, RSX_SUMO sử dụng tổ hợp các thuật toán học máy cổ điển (XGBoost, SVM, RF) cùng với tập đặc trưng lý hóa được trích chọn bằng phương pháp thủ công như AAIndex, CKSAAP và BLOSUM62, giúp mô hình duy trì được tính diễn giải cao và phù hợp với các tập dữ liệu có quy mô hạn chế. Trong khi đó, CLW_SUMO và CBiLSuccsite khai thác sức mạnh của mô hình học sâu lai, kết hợp giữa mạng CNN và LSTM/BiLSTM nhằm tự động trích xuất đặc trưng ngữ cảnh trong chuỗi amino acid. Đặc biệt, CLW_SUMO tận dụng embedding Word2Vec tiền huấn luyện, giúp tăng khả năng biểu diễn ngữ nghĩa nhưng đòi hỏi chi phí tính toán cao hơn. Đặc biệt, CBiLSuccsite được cải tiến bằng việc sử dụng embedding động được huấn luyện trực tiếp trong mô hình, qua đó nâng cao khả năng thích ứng với dữ liệu đặc thù và giảm độ phức tạp so với CLW_SUMO. Cuối cùng, KD_ArapUbi đại diện cho hướng tiếp cận hiện đại hơn bằng việc sử dụng kỹ thuật học chắt lọc tri thức và phương pháp mã hóa embedding động được huấn luyện trực tiếp trong mô hình. Mô hình này có số lượng tham số nhỏ gọn, tối ưu về hiệu quả tính toán và khả năng khái quát hóa, đặc biệt phù hợp với các bài toán có quy mô dữ liệu vừa và nhỏ.

Nhìn chung, theo trình tự phát triển, các mô hình trong luận án không chỉ mở rộng phạm vi ứng dụng sang nhiều loại PTM khác nhau mà còn thể hiện sự tiến bộ tuần tự về mặt kỹ thuật và hiệu quả từ việc cải thiện độ chính xác, tối ưu cách biểu diễn dữ liệu, đến tiết kiệm tài nguyên và rút ngắn thời gian huấn luyện. Điều này cho thấy một định hướng phát triển nhất quán, đồng thời khẳng định tiềm năng ứng dụng cao của các mô hình được đề xuất trong cả nghiên cứu lý thuyết lẫn thực tiễn sinh học tính toán.

## 4.5 Kết luận chương 4

Trong chương này, NCS đã đề xuất và phát triển mô hình KD_ArapUbi, một kiến trúc học chắt lọc tri thức ứng dụng cho bài toán dự đoán vị trí ubiquitination trên loài *Arabidopsis thaliana*. Bằng cách kết hợp giữa "mô hình Giáo viên" Bi-LSTM mạnh mẽ, embedding động và "mô hình Học viên" gọn nhẹ hơn, KD_ArapUbi đã đạt được hiệu suất vượt trội so với nhiều phương pháp truyền thống và công cụ dự đoán hiện có, đồng thời giảm đáng kể số lượng tham số cần huấn luyện.

Bên cạnh đó, việc so sánh và đánh giá bốn mô hình khác nhau (RSX_SUMO, CLW_SUMO, CBiLSuccsite và KD_ArapUbi) cho thấy sự đa dạng trong cách tiếp cận bài toán. Các mô hình học máy truyền thống đem lại độ ổn định và tốc độ xử lý nhanh, thích hợp cho dữ liệu nhỏ. Các mô hình học sâu lai khai thác tốt cả đặc trưng cục bộ và toàn cục, song đòi hỏi tài nguyên tính toán lớn hơn. Trong khi đó, hướng tiếp cận hiện đại với học chắt lọc tri thức đã chứng minh khả năng tối ưu hoá, vừa giảm chi phí tính toán, vừa duy trì hiệu suất cao, phù hợp cho các ứng dụng thực tiễn.

Từ đó, luận án khẳng định giá trị khoa học không chỉ ở việc đề xuất một mô hình mới có hiệu năng vượt trội, mà còn ở việc cung cấp góc nhìn toàn diện về ưu, nhược điểm và bối cảnh sử dụng của các mô hình dự đoán PTM. Đây là cơ sở quan trọng để định hướng phát triển các công cụ tính toán hiệu quả, phục vụ cho nghiên cứu và ứng dụng trong sinh học tính toán.

Một phần nội dung trong chương này đã được NCS công bố trên tạp chí dưới đây:

[CT8] Nguyen V. N., Tran T. X,. Nguyen T. T, N.Q.K. Le. (2024), Enhancing Ara- bidopsis thaliana ubiquitination site prediction through knowledge distillation and natural language processing. Methods. 232: p. 65-71. DOI: https://doi.org/10.1016/j.ymeth.2024.10.006. (SCIE Q1 IF: 4.2).

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong luận án này, NCS đã tập trung nghiên cứu và phát triển các mô hình cải tiến nhằm nâng cao hiệu suất dự đoán các vị trí sửa đổi sau dịch mã (PTM) trên protein. Cụ thể với việc đề xuất kiến trúc Học máy tổ hợp với đặc trưng lai ghép, một số kiến trúc Mô hình học sâu lai, học chắt lọc tri thức kết hợp kỹ thuật NLP mới giúp cải thiện hiệu suất của 3 PTM (SUMOylation, Succinylation và Ubiquitination).

**A. Các kết quả đạt được của luận án**

**Luận án có ba đóng góp chính sau:**

**(1) Cơ sở lý luận và tổng quan hệ thống:** Luận án đã hệ thống hóa, phân tích và so sánh các phương pháp từ truyền thống, học máy tổ hợp, học sâu lai cho đến kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) trong bài toán dự đoán PTM, qua đó xây dựng nền tảng khoa học vững chắc cho các nghiên cứu tiếp theo.

**(2) Khai thác NLP cho dữ liệu protein:** Luận án đã chứng minh khả năng ứng dụng và hiệu quả của các kỹ thuật NLP trong việc biểu diễn ngữ cảnh của chuỗi protein, giúp vượt qua hạn chế của đặc trưng thủ công và nâng cao độ chính xác trong dự đoán.

**(3) Đề xuất và phát triển mô hình mới:** Luận án đã đề xuất bốn mô hình PTM với hiệu suất cao, trong đó có các mô hình lai kết hợp học sâu với NLP và đặc biệt là mô hình áp dụng học chắt lọc tri thức cho Ubiquitination, phù hợp với bối cảnh dữ liệu hạn chế và môi trường tính toán hạn chế. Cụ thể, bốn đề xuất chính gồm:

- Đề xuất mô hình dự đoán vị trí PTM (SUMOylation) dựa trên học máy tổ hợp và các đặc trưng lai ghép.

- Đề xuất hai mô hình dự đón vị trí PTM (SUMOylation và Succinylation) dựa trên kỹ thuật học sâu lai ghép và kỹ thuật xử lý ngôn ngữ tự nhiên.

- Đề xuất mô hình dự đoán PTM (Ubiquitination) dựa trên học chắt lọc tri thức và kỹ thuật xử lý ngôn ngữ tự nhiên.

**B. Những điểm mới và ý nghĩa của các kết quả nghiên cứu**

Nghiên cứu đã đề xuất phương pháp mã hóa dữ liệu sử dụng kỹ thuật NLP, phù hợp với dữ liệu protein cấu trúc bậc 1, đồng thời tận dụng được sức mạnh tự động học đặc trưng của mô hình học sâu. Điều này giúp cải thiện đáng kể khả năng biểu diễn dữ liệu, tối ưu hóa quá trình huấn luyện mô hình và nâng cao hiệu suất dự đoán.

Bên cạnh đó, luận án đã đề xuất một số mô hình tiên tiến để dự đoán vị trí PTM với hiệu suất cao, trong đó có RSX_SUMO, CLW_SUMO, CBiLSuccSite và KD_ArapUbi.

Các mô hình này đều ứng dụng kỹ thuật mã hóa chuỗi protein bằng NLP, giúp khai thác tốt đặc trưng trình tự protein. Hơn nữa, kiến trúc của các mô hình đề xuất đã phát huy sức mạnh của nhiều mô hình học máy và học sâu, từ đó cải thiện hiệu suất tổng thể.

Một số mô hình học sâu lai đặc biệt mô hình học chắt lọc tri thức không chỉ giúp nâng cao độ chính xác mà còn có khả năng tự động học đặc trưng từ dữ liệu thô và thực hiện quá trình học end-to-end. Điều này giúp giảm thiểu sự phụ thuộc vào các phương pháp trích xuất đặc trưng thủ công, đồng thời đảm bảo tính tổng quát và hiệu quả của mô hình trong bài toán dự đoán vị trí PTM.Các mô hình đề xuất không chỉ mang tính học thuật mà còn có ý nghĩa thực tiễn, hỗ trợ các nhà nghiên cứu về sinh học phân tử, dược sĩ, bác sĩ rút ngắn thời gian trong việc phát hiện, phân tích các vị trí sửa đổi trên protein. Bên cạnh việc công bố kết quả nghiên cứu, NCS cũng chia sẻ dữ liệu và toàn bộ codes chương trình thực nghiệm lên nền tảng Github để đóng góp và hỗ trợ tích cực cho các nhà khoa học trong quá trình nghiên cứu, thực nghiệm có liên quan của họ.

Những kết quả nghiên cứu của luận án không chỉ đóng góp vào lĩnh vực dự đoán vị trí PTM mà còn khẳng định tính khả thi và hiệu quả của việc ứng dụng các mô hình Học máy tổ hợp, Mô hình học sâu lai, Học chắt lọc tri thức và kỹ thuật NLP với dữ liệu protein cấu trúc bậc 1 trong dự đoán vị trí PTM.

**C. Hướng phát triển của luận án**

**Thứ nhất: Nâng cao độ chính xác của mô hình**

Mặc dù các mô hình trong luận án đã đạt được kết quả đáng khích lệ trong việc dự đoán các vị trí sửa đổi sau dịch mã (PTM), vẫn còn những tiềm năng để cải thiện độ chính xác. Trong các nghiên cứu tiếp theo, cần xem xét kết hợp thêm các kỹ thuật học sâu tiên tiến hơn, tối ưu hóa kiến trúc mô hình hoặc kết hợp thêm thông tin đặc trưng sinh học để cải thiện chất lượng dự đoán.

**Thứ hai: Xử lý vấn đề dữ liệu mất cân bằng**

Dữ liệu PTM, đặc biệt khi mở rộng sang các loại PTM khác hoặc các loài khác, thường gặp phải tình trạng mất cân bằng nghiêm trọng giữa số lượng mẫu dương tính và âm tính. Trong nghiên cứu này, NCS sử dụng các bộ dữ liệu đã được cân bằng theo các nghiên cứu trước. Tuy nhiên, các hướng nghiên cứu tiếp theo cần tập trung khai thác và so sánh hiệu quả của các phương pháp xử lý dữ liệu mất cân bằng như oversampling, undersampling, áp dụng trọng số cho hàm mất mát, hoặc sử dụng các kỹ thuật như focal loss. Điều này không chỉ giúp cải thiện hiệu quả dự đoán mà còn tăng tính ứng dụng khi triển khai trên các bộ dữ liệu thực tế.

**Thứ ba: Mở rộng mô hình cho dự đoán các PTM khác**

Luận án đã tập trung vào một số PTM tiêu biểu như SUMOylation, Succinylation

và Ubiquitination. Các nghiên cứu tiếp theo có thể mở rộng phạm vi nghiên cứu sang các loại PTM khác như Methylation, Acetylation, Phosphorylation,... để xây dựng một hệ thống dự đoán PTM toàn diện hơn.

**Thứ tư: Phát triển phần mềm và công cụ hỗ trợ nghiên cứu**

Việc triển khai các mô hình dự đoán vị trí PTM dưới dạng phần mềm hoặc công cụ dễ sử dụng cho các nhà sinh học và nghiên cứu viên sẽ giúp ứng dụng rộng rãi các phương pháp trong thực tiễn, góp phần hỗ trợ các nghiên cứu trong lĩnh vực sinh học phân tử và phát triển dược phẩm.

Trong quá trình thực hiện luận án, với hiểu biết còn hạn chế, NCS rất mong nhận được góp ý của các thầy cô để luận án hoàn thiện tốt nhất. NCS xin chân thành cảm ơn!

Thái Nguyên, ngày .... tháng .... năm 2025
Nghiên cứu sinh

**Trần Thị Xuân**

# DANH MỤC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ

[CT1] Le N.Q.K, Tran T.X, Nguyen P.A. Nguyen V.N, et al. (2023), Recent progress in machine learning approaches for predicting carcinogenicity in drug development. Expert Opinion on Drug Metabolism & Toxicology. p 621-628, DOI: https://doi.org/10.1080/17425255.2024.2356162.**(SCIE Q1, IF: 3.9)- Bổ trợ- Chương 1**

[CT2] Le N.Q.K, Nguyen V.N, Nguyen T.T, Tran T.X, at al. (2024), Enhancing Protein Sequence Classification with a Fuzzy Neural Network: A Study in Anticancer Peptide Identification, International Conference on Fuzzy Theory and Its Applications (iFUZZY), Kagawa, Japan. pp. 1-6, DOI: https://doi.org/10.1109/iFUZZY63051.2024.10662887. **- Bổ trợ - Chương 1**

[CT3] Tran T.X, Nguyen V.N, and Le N.Q.K. (2023) Incorporating Natural Language-Based and Sequence-Based Features to Predict Protein SUMOylation Sites. Conference on Information Technology and its Applications. DOI: https://doi.org/10.1007/978-3-031-36886-8_7. **(Indexed: Scopus Q4)- Liên quan trực tiếp - Chương 2**

[CT4] Tran T.X., Le N.Q.K., and Nguyen V.N. (2024), CLW-SUMO: A hybrid deep learning model for predicting protein SUMOylation sites. Journal of Computer Science and Cybernetics. DOI: https://doi.org/10.15625/1813-9663/19626. **(Tạp chí Tin học điều khiển 1.25đ) - Liên quan trực tiếp - Chương 3**

[CT5] Tran T.X, Le N.Q.K, and Nguyen V.N. (2025), Integrating CNN and Bi-LSTM for protein succinylation sites prediction based on Natural Language Processing technique. Computers in Biology and Medicine. 186: p. 109664. DOI: https://doi.org/10.1016/j.compbiomed.2025.109664. **(SCIE Q1, IF: 7.0)- Liên quan trực tiếp - Chương 3**

[CT6] Tran T.X., Nguyen T.T., Le N.Q.K., et al. (2024). A novel deep learning approach for the prediction of Arabidopsis thaliana ubiquitination sites. Proceedings of the 13th International Conference on Information Technology and Its Applications (CITA 2024), pp. 48–57. DOI: https://elib.vku.udn.vn/handle/123456789/4010. **(Scopus Q4) – Liên quan trực tiếp – Chương 3**

[CT7] Tran T.X, Nguyen T.T, Le N.Q.K, and Nguyen V.N. (2025), A hybrid deep learning and Natural Language Processing Model for Plant Ubiquitination Site Prediction, The 3rd International Conference on Advances in Information and

Communication Technology. ICTA 2024. DOI: https://doi.org/10.1007/978-3-031-80943-9_49. **(Indexed: Scopus Q4)- Liên quan trực tiếp - Chương 3**

[CT8] Nguyen V. N., Tran T. X,. Nguyen T. T, N.Q.K. Le. (2024), Enhancing Arabidopsis thaliana ubiquitination site prediction through knowledge distillation and natural language processing. Methods. 232: p. 65-71. DOI: https://doi.org/10.1016/j.ymeth.2024.10.006. **(SCIE Q1 IF: 4.2)- Liên quan trực tiếp - Chương 4**

# TÀI LIỆU THAM KHẢO


[1] Mathworks - matlab and simulink, 2025. Truy cập ngày 10/6/2025.

[2] Nghiên cứu, kết hợp mô hình học máy ứng dụng phân tích dự đoán protein sửa đổi sau dịch mã. Báo cáo tổng kết Đề tài Khoa học và Công nghệ cấp Đại học Thái Nguyên, 2025. Mã số: ĐH2023-TN08-05.

[3] H. Abou-Abbass, H. Abou-El-Hassan, H. Bahmad, K. Zibara, A. Zebian, R. Youssef, J. Ismail, R. Zhu, S. Zhou, X. Dong, et al. Glycosylation and other ptms alterations in neurodegenerative diseases: Current status and future role in neurotrauma. *Electrophoresis*, 37(11):1549–1561, 2016.

[4] F. Ali, J. S. Dar, A. R. Magray, et al. Posttranslational modifications of proteins and their role in biological processes and associated diseases. pages 1–35. Elsevier, 2019. DOI:https://doi.org/10.1016/B978-0-12-811913-6.00001-1.

[5] A. Alkhulaifi, F. Alsahli, and I. Ahmad. Knowledge distillation in deep learning and its applications. *PeerJ Computer Science*, 7:e474, 2021. DOI:`https://doi.org/10.7717/peerj-cs.474`.

[6] M. Alleyn, M. Breitzig, R. Lockey, et al. The dawn of succinylation: a post-translational modification. *American Journal of Physiology-Cell Physiology*, 314(2):C228–C232, 2018. DOI:`https://doi.org/10.1152/ajpcell.00148.2017`.

[7] R. Aslebagh, K. L. Wormwood, D. Channaveerappa, A. G. N. Wetie, A. G. Woods, and C. C. Darie. Identification of posttranslational modifications (ptms) of proteins by mass spectrometry. *Advancements of mass spectrometry in biomedical research*, pages 199–224, 2019.

[8] K. Bayoudh. A survey of multimodal hybrid deep learning for computer vision: Architectures, applications, trends, and challenges. *Information Fusion*, 105:102217, 2024. DOI:`https://doi.org/10.1016/j.inffus.2023.102217`.

[9] G. Beauclair, A. Bridier-Nahmias, J. Zagury, and et al. Jassa: a comprehensive tool for prediction of sumoylation sites and sims. *Bioinformatics*, 31(21):3483–3491, 2015. DOI:`https://doi.org/10.1093/bioinformatics/btv403`.

[10] G. Beauclair, A. Bridier-Nahmias, J.-F. Zagury, et al. Jassa: a comprehensive tool for prediction of sumoylation sites and sims. *Bioinformatics*, 31(21):3483–3491, 2015. DOI:https://doi.org/10.1093/bioinformatics/btv403.

[11] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

[12] C.-C. Chang, C.-H. Tung, C.-W. Chen, et al. Sumogo: Prediction of sumoylation sites on lysines by motif screening models and the effects of various post-translational modifications. *Scientific Reports*, 8(1):15512, 2018. DOI:https://doi.org/10.1038/s41598-018-33951-5.

[13] J. Chen, J. Zhao, S. Yang, et al. Prediction of protein ubiquitination sites in arabidopsis thaliana. *Current Bioinformatics*, 14(7):614–620, 2019. DOI:https://doi.org/10.2174/1574893614666190311141647.

[14] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. DOI:https://doi.org/10.1145/2939672.2939785.

[15] Y. Chen, Y. Gong, G. Ying, and C. Zhang. Sumohydro: a novel method for the prediction of sumoylation sites based on hydrophobic properties. *PLoS ONE*, 7(6):e39195, 2012. DOI:https://doi.org/10.1371/journal.pone.0039195.

[16] Y.-Z. Chen, Z. Chen, Y.-A. Gong, et al. Sumohydro: a novel method for the prediction of sumoylation sites based on hydrophobic properties. *PLoS ONE*, 7(6):e39195, 2012. DOI:https://doi.org/10.1371/journal.pone.0039195.

[17] Z. Chen, N. He, Y. Huang, et al. Integration of a deep learning classifier with a random forest approach for predicting malonylation sites. *Genomics, Proteomics & Bioinformatics*, 16(6):451–459, 2018. DOI:https://doi.org/10.1016/j.gpb.2018.08.004.

[18] Z. Chen, P. Zhao, F. Li, et al. ifeature: a python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, 34(14):2499–2502, 2018. DOI:https://doi.org/10.1093/bioinformatics/bty140.

[19] Z. Chen, P. Zhao, F. Li, et al. ilearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of dna, rna and protein sequence data. *Briefings in Bioinformatics*, 21(3):1047–1057, 2020. DOI:https://doi.org/10.1093/bib/bbz041.

[20] J. Cho and B. Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4802, 2019. DOI:10.48550/arXiv.1910.01348.

[21] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. DOI:https://doi.org/10.1007/BF00994018.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. DOI:https://aclanthology.org/N19-1423/.

[23] T. G. Dietterich et al. *Ensemble learning*, volume 2. 2002.

[24] S. Doll and A. L. Burlingame. Mass spectrometry-based detection and assignment of protein posttranslational modifications. *ACS chemical biology*, 10(1):63–71, 2015.

[25] S. Doll and A. L. Burlingame. Mass spectrometry-based detection and assignment of protein posttranslational modifications. *ACS Chemical Biology*, 10(1):63–71, 2015. DOI:10.1021/cb500904b.

[26] J. Donahue, L. A. Hendricks, S. Guadarrama, et al. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. DOI: 10.1109/CVPR.2015.7298878.

[27] L. Dou, X. Li, L. Zhang, et al. iglu_adaboost: identification of lysine glutarylation using the adaboost classifier. *Journal of Proteome Research*, 20(1):191–201, 2020. DOI:https://doi.org/10.1021/acs.jproteome.0c00314.

[28] A. Emamjomeh, B. Goliaei, J. Zahiri, et al. Predicting protein–protein interactions between human and hepatitis c virus via an ensemble learning method. *Molecular Biosystems*, 10(12):3147–3154, 2014. DOI:https://doi.org/10.1039/C4MB00410H.

[29] F. Esmaili, M. Pourmirzaei, S. Ramazi, S. Shojaeilangari, and E. Yavari. A review of machine learning and algorithmic methods for protein phosphorylation site prediction, 2023. DOI:https://doi.org/10.1016/j.gpb.2023.03.007.

[30] H. Fu, Y. Yang, X. Wang, et al. Deepubi: a deep learning framework for prediction of ubiquitination sites in proteins. *BMC Bioinformatics*, 20(1):1–10, 2019. DOI:https://doi.org/10.1186/s12859-019-2677-9.

[31] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR, 2018. DOI:80:1607-1616.

[32] I. Gandhi and M. Pandey. Hybrid ensemble of classifiers using voting. In *International Conference on Green Computing and Internet of Things (ICGCIoT)*. IEEE, 2015. DOI: 10.1109/ICGCIoT.2015.7380496.

[33] R. Geiss-Friedlander and F. Melchior. Concepts in sumoylation: a decade on. *Nature Reviews Molecular Cell Biology*, 8(12):947–956, 2007. DOI:https://doi.org/10.1038/nrm2293.

[34] G. Goldstein, M. Scheid, U. Hammerling, et al. Isolation of a polypeptide that has lymphocyte-differentiating properties and is probably represented universally in living cells. *Proceedings of the National Academy of Sciences*, 72(1):11–15, 1975. DOI:https://doi.org/10.1073/pnas.72.1.11.

[35] J. Gou, B. Yu, S. Maybank, et al. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. DOI:https://doi.org/10.1007/s11263-021-01453-z.

[36] Y. Gou, D. Liu, M. Chen, Y. Wei, X. Huang, C. Han, Z. Feng, C. Zhang, T. Lu, D. Peng, et al. Gps-sumo 2.0: an updated online service for the prediction of sumoylation sites and sumo-interacting motifs. *Nucleic Acids Research*, 52(W1):W238–W247, 2024.

[37] Z.-J. Han, Y.-H. Feng, B.-H. Gu, et al. The post-translational modification, sumoylation, and cancer. *International Journal of Oncology*, 52(4):1081–1094, 2018. DOI:https://doi.org/10.3892/ijo.2018.4280.

[38] M. Hasan, M. Khatun, M. Mollah, et al. A systematic identification of species-specific protein succinylation sites using joint element features information. *International Journal of Nanomedicine*, pages 6303–6315, 2017. DOI:https://doi.org/10.2147/IJN.S140875.

[39] M. Hasan and H. Kurata. Gpsuc: Global prediction of generic and species-specific succinylation sites by aggregating multiple sequence features. *PLoS ONE*, 13(10):e0200283, 2018. DOI:https://doi.org/10.1371/journal.pone.0200283.

[40] R. Hay. Sumo: a history of modification. *Molecular Cell*, 18(1):1–12, 2005. DOI: 10.1016/j.molcel.2005.03.012.

[41] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. DOI:https://doi.org/10.48550/arXiv.1503.02531.

[42] T. K. Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 278–282. IEEE, 1995. DOI:10.1109/ICDAR.1995.598994.

[43] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. DOI:https://doi.org/10.1007/978-3-642-24797-2_4.

[44] G. Huang, Q. Shen, G. Zhang, et al. Lstmcnnsucc: a bidirectional lstm and cnn-based deep learning method for predicting lysine succinylation sites. *BioMed Research International*, 2021, 2021. DOI:https://doi.org/10.1155/2021/9923112.

[45] G. Huang, Q. Shen, G. Zhang, et al. Lstmcnnsucc: A bidirectional lstm and cnn-based deep learning method for predicting lysine succinylation sites. *BioMed Research International*, 2021(1):9923112, 2021. DOI:https://doi.org/10.1155/2021/9923112.

[46] Y. Huang, B. Niu, Y. Gao, et al. Cd-hit suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5):680–682, 2010. DOI:https://doi.org/10.1093/bioinformatics/btq003.

[47] M. Jan, A. Awan, M. Khalid, et al. Ensemble approach for developing a smart heart disease prediction system using classification algorithms. *Research Reports in Clinical Cardiology*, pages 33–45, 2018. DOI:https://doi.org/10.2147/RRCC.S172035.

[48] J. Jia, G. Wu, M. Li, et al. psuc-edbam: Predicting lysine succinylation sites in proteins based on ensemble dense blocks and an attention module. *BMC Bioinformatics*, 23(1):450, 2022. DOI:https://doi.org/10.1186/s12859-022-05001-5.

[49] J. Jia, L. Zhang, Z. Liu, et al. psumo-cd: predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general pseaac. *Bioinformatics*, 32(20):3133–3141, 2016. DOI:https://doi.org/10.1093/bioinformatics/btw387.

[50] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019. DOI:https://doi.org/10.48550/arXiv.1909.10351.

[51] V. Jothi Prakash and N. Karthikeyan. Enhanced evolutionary feature selection and ensemble method for cardiovascular disease prediction. *Interdisciplinary Sciences: Computational Life Sciences*, 13(3):389–412, 2021. DOI:https://doi.org/10.1007/s12539-021-00430-x.

[52] S. Kawashima and M. Kanehisa. Aaindex: amino acid index database. *Nucleic Acids Research*, 28(1):374, 2000. DOI: https://doi.org/10.1093/nar/28.1.374.

[53] S. Khan, S. AlQahtani, S. Noor, et al. Pssm-sumo: deep learning based intelligent model for prediction of sumoylation sites using discriminative features. *BMC Bioinformatics*, 25(1):284, 2024. DOI:https://doi.org/10.1186/s12859-024-05917-0.

[54] J. W. Lee. Proteins, n.d. Truy cập ngày 5/6/2025.

[55] K. Lee, N. Pham, H. Min, et al. Dogpred: A novel deep learning framework for accurate identification of human o-linked threonine glycosylation sites. *Journal of Molecular Biology*, page 168977, 2025. DOI:https://doi.org/10.1016/j.jmb.2025.168977.

[56] M. Li, L. Fan, A. Cummings, et al. Hybrid quantum classical machine learning with knowledge distillation. In *ICC 2024–IEEE International Conference on Communications*. IEEE, 2024. DOI: 10.1109/ICC51166.2024.10622755.

[57] S. Li, M. Lin, Y. Wang, et al. Distilling a powerful student model via online knowledge distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 10.1109/TNNLS.2022.3152732.

[58] C.-M. Liu, V.-D. Ta, N. Le, et al. Deep neural network framework based on word embedding for protein glutarylation sites prediction. *Life*, 12(8):1213, 2022. DOI:https://doi.org/10.3390/life12081213.

[59] X. Liu, X. Wang, and S. Matwin. Improving the interpretability of deep neural networks with knowledge distillation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 905–912. IEEE, 2018. DOI: 10.1109/ICDMW.2018.00132.

[60] X. Liu, L.-L. Xu, Y.-P. Lu, et al. Deep_ksuccsite: A novel deep learning method for the identification of lysine succinylation sites. *Frontiers in Genetics*, 13:1007618, 2022. DOI:https://doi.org/10.3389/fgene.2022.1007618.

[61] Y. Liu, S. Jin, L. Song, et al. Prediction of protein ubiquitination sites via multi-view features based on extreme gradient boosting classifier. *Journal of Molecular Graphics and Modelling*, 107:107962, 2021. DOI:https://doi.org/10.1016/j.jmgm.2021.107962.

[62] Y. Liu, Z. Yu, C. Chen, et al. Prediction of protein crotonylation sites through lightgbm classifier based on smote and elastic net. *Analytical Biochemistry*, 609:113903, 2020. DOI:https://doi.org/10.1016/j.ab.2020.113903Getrightsandcontent.

[63] Z. Liu, L. Zhu, X.-P. Zhang, et al. Hybrid deep learning for plant leaves classification. In *Intelligent Computing Theories and Methodologies: 11th International Conference, ICIC 2015, Fuzhou, China, August 20-23, 2015, Proceedings, Part II*, volume 9226 of *Lecture Notes in Computer Science*, pages 345–354. Springer, 2015. DOI:https://doi.org/10.1007/978-3-319-22186-1_11.

[64] Y. Lopez, A. Dehzangi, H. Reddy, et al. C-isumo: a sumoylation site predictor that incorporates intrinsic characteristics of amino acid sequences. *Computational Biology and Chemistry*, 87:107235, 2020. DOI:https://doi.org/10.1016/j.compbiolchem.2020.1072.

[65] C. Lu, K. Huang, M. Su, and et al. Dbptm 3.0: an informative resource for investigating substrate site specificity and functional association of protein post-translational modifications. *Nucleic Acids Research*, 41(Database issue):D295–D305, 2013. DOI:https://doi.org/10.1093/nar/gks1229.

[66] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[67] M. Mann and O. N. Jensen. Proteomic analysis of post-translational modifications. *Nature Biotechnology*, 21(3):255–261, 2003. DOI:https://doi.org/10.1038/nbt0303-255.

[68] H. Marmor-Kollet, N. Kedersha, N. Knafo, N. Rivkin, Y. Danino, T. Moens, T. Olender, D. Sheban, and N. Cohen. Spatiotemporal proteomic analysis of stress granule disassembly using apex reveals regulation by sumoylation and links to als pathogenesis. *Molecular Cell*, 80:15, 2020. DOI: 10.1016/j.molcel.2020.10.032.

[69] L. Meng, W.-S. Chan, L. Huang, et al. Mini-review: recent advances in post-translational modification site prediction based on deep learning. *Computational and Structural Biotechnology Journal*, 20:3522–3532, 2022. DOI:`https://doi.org/10.1016/j.csbj.2022.06.045`.

[70] I. Mienye and Y. Sun. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10:99129–99149, 2022. DOI: 10.1109/ACCESS.2022.3207287.

[71] S. Mirzadeh, M. Farajtabar, A. Li, et al. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020. DOI:10.1609/aaai.v34i04.5963.

[72] M. Mosharaf, M. Hassan, F. Ahmed, et al. Computational prediction of protein ubiquitination sites mapping on arabidopsis thaliana. *Computational Biology and Chemistry*, 85:107238, 2020. DOI:`https://doi.org/10.1016/j.compbiolchem.2020.107238`.

[73] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. DOI:`https://doi.org/10.1080/09332480.2014.914768`.

[74] S. Müller, C. Hoege, G. Pyrowolakis, and S. Jentsch. Sumo, ubiquitin's mysterious cousin. *Nature Reviews Molecular Cell Biology*, 2(3):202–210, 2001. DOI:`https://doi.org/10.1038/35056591`.

[75] H. Nakashima, K. Nishikawa, and T. Ooi. The folding type of a protein is relevant to the amino acid composition. *The Journal of Biochemistry*, 99(1):153–162, 1986. DOI: `https://doi.org/10.1109/ACCESS.2021.3127881`.

[76] W. Ning, H. Xu, P. Jiang, et al. Hybridsucc: a hybrid-learning architecture for general and species-specific succinylation site prediction. *Genomics, Proteomics and Bioinformatics*, 18(2):194–207, 2020. DOI:`https://doi.org/10.1016/j.gpb.2019.11.010`.

[77] D. Ofer, N. Brandes, and M. Linial. The language of proteins: Nlp, machine learning & protein sequences. *Computational and Structural Biotechnology Journal*, 19:1750–1758, 2021. DOI:`https://doi.org/10.1016/j.csbj.2021.03.022`.

[78] C.-Y. Ou, H. Pi, and C.-T. Chien. Control of protein degradation by e3 ubiquitin ligases in drosophila eye development. *Trends in Genetics*, 19(7):382–389, 2003. DOI:DOI:10.1016/S0168-9525(03)00146-XExternalLink.

[79] S. Pakhrin, N. Chauhan, S. Khan, et al. Prediction of human o-linked glycosylation sites using stacked generalization and embeddings from pre-trained protein language model. *Bioinformatics*, 40(11):btae643, 2024. DOI:https://doi.org/10.1093/bioinformatics/btae643.

[80] W. Park, D. Kim, Y. Lu, and M. Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976, 2019. DOI:10.1109/CVPR.2019.00409.

[81] N. Pham, Y. Zhang, R. Rakkiyappan, et al. Hotgpred: Enhancing human o-linked threonine glycosylation prediction using integrated pretrained protein language model-based features and multi-stage feature selection approach. *Computers in Biology and Medicine*, 179:108859, 2024. DOI:https://doi.org/10.1016/j.compbiomed.2024.10885.

[82] S. Pokharel, P. Pratyush, M. Heinzinger, et al. Improving protein succinylation sites prediction using embeddings from protein language model. *Scientific Reports*, 12(1):16933, 2022. DOI:https://doi.org/10.1038/s41598-022-21366-2.

[83] S. Pokharel, P. Pratyush, M. Heinzinger, R. Newman, and D. KC. Lmsuccsite: Improving protein succinylation sites prediction using embeddings from protein language model. 2022. DOI:https://doi.org/10.21203/rs.3.rs-1953874/v1.

[84] S. Pokharel, E. Sidorov, and D. Carageat. Nlp-based encoding techniques. *Machine Learning In Bioinformatics Of Protein Sequences: Algorithms, Databases And Resources For Modern Protein Bioinformatics*, page 81, 2022. DOI:https://doi.org/10.1142/9789811258589_0004.

[85] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020. DOI:https://doi.org/10.48550/arXiv.2010.16061.

[86] P. Pratyush, S. Pokharel, H. D. Ismail, S. Bahmani, and D. B. Kc. Lmptmsite: a platform for ptm site prediction in proteins leveraging transformer-based protein language models. In *Prediction of Protein Secondary Structure*, pages 261–297. Springer, 2024. DOI:https://doi.org/10.1007/978-1-0716-4196-5_16.

[87] A. Princz and S. N. Sumoylation in neurodegenerative diseases. *Gerontology*, 66:8, 2020. DOI:https://doi.org/10.1159/000502142.

[88] Quizlet. Gospodarka azotowa – biochemia diagram, 2025. Truy cập ngày 5/6/2025.

[89] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[90] S. Ramazi and J. Zahiri. Posttranslational modifications in proteins: resources, tools and prediction methods. *Database (Oxford)*, 2021. DOI:https://doi.org/10.1093/database/baab012.

[91] M. Ramesh, P. Gopinath, and T. Govindaraju. Role of post-translational modifications in alzheimer's disease. *ChemBioChem*, 21(8):1052–1079, 2020.

[92] J. Ren, X. Gao, C. Jin, and et al. Systematic study of protein sumoylation: Development of a site-specific predictor of sumosp 2.0. *Proteomics*, 9(12):3409–3412, 2009. DOI:https://doi.org/10.1002/pmic.200800646.

[93] A. Rojarath, W. Songpan, and C. Pong-inwong. Improved ensemble learning for classification techniques based on majority voting. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2016. DOI: 10.1109/ICSESS.2016.7883026.

[94] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. DOI:https://doi.org/10.48550/arXiv.1412.6550.

[95] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. DOI:https://doi.org/10.48550/arXiv.1910.01108.

[96] J. Seeler, O. B., K. Nacerddine, and A. Dejean. Sumo, the three rs and cancer. *Current Topics in Microbiology and Immunology*, 313:22, 2007. DOI:https://doi.org/10.1007/978-3-540-34594-7_4.

[97] A. Sharma, A. Lysenko, Y. López, et al. Hsesumo: Sumoylation site prediction using half-sphere exposures of amino acids residues. *BMC Genomics*, 19(9):1–7, 2019. DOI:https://doi.org/10.1186/s12864-018-5206-8.

[98] H.-B. Shen and K.-C. Chou. Pseaac: a flexible web server for generating various kinds of protein pseudo amino acid composition. *Analytical Biochemistry*, 373(2):386–388, 2008. DOI:https://doi.org/10.1016/j.ab.2007.10.01.

[99] P. Shrestha, J. Kandel, H. Tayara, et al. Post-translational modification prediction via prompt-based fine-tuning of a gpt-2 model. *Nature Communications*, 15(1):6699, 2024. DOI:https://doi.org/10.1038/s41467-024-51071-9.

[100] A. M. Silva, R. Vitorino, M. R. M. Domingues, et al. Post-translational modifications and mass spectrometry detection. *Free Radical Biology and Medicine*, 65:925–941, 2013. DOI:https://doi.org/10.1016/j.freeradbiomed.2013.08.1.

[101] N. Soylu and E. Sefer. Deepptm: protein post-translational modification prediction from protein sequences by combining deep protein language model with vision transformers. *Current Bioinformatics*, 19(9):810–824, 2024.

[102] K. Swatek and D. Komander. Ubiquitin modifications. *Cell Research*, 26(4):399–422, 2016. DOI:https://doi.org/10.1038/cr.2016.39.

[103] H. Tang, Q. Tang, Q. Zhang, et al. O-glythr: prediction of human o-linked threonine glycosites using multi-feature fusion. *International Journal of Biological Macromolecules*, 242:124761, 2023. DOI:https://doi.org/10.1016/j.ijbiomac.2023.124761.

[104] S. Teng, H. Luo, and L. Wang. Predicting protein sumoylation sites from sequence features. *Amino Acids*, 43:447–455, 2012. DOI:https://doi.org/10.1007/s00726-011-1100-2.

[105] Y. Teng, Y. Chen, X. Tang, et al. Pad2: A potential target for tumor therapy. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1878(5):188931, 2023. DOI:https://doi.org/10.1016/j.bbcan.2023.188931.

[106] N. Thapa, M. Chaudhari, S. McManus, et al. Deepsuccinylsite: a deep learning based approach for protein succinylation site prediction. *BMC Bioinformatics*, 21:1–10, 2020. DOI:https://doi.org/10.1186/s12859-020-3342-z.

[107] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996. DOI:https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[108] UniProt Consortium. P63165 - sumo1_human - small ubiquitin-related modifier 1 - homo sapiens (human), 2024. Truy cập ngày 7/6/2025.

[109] D. Velusamy and K. Ramasamy. Ensemble of heterogeneous classifiers for diagnosis and prediction of coronary artery disease with reduced feature subset. *Computer Methods and Programs in Biomedicine*, 198:105770, 2021. DOI:https://doi.org/10.1016/j.cmpb.2020.105770.

[110] VietJack. Cấu trúc của protein, 2025. Truy cập ngày 5/6/2025.

[111] C. Wang, X. Tan, D. Tang, et al. Gps-uber: a hybrid-learning framework for prediction of general and e3-specific lysine ubiquitination sites. *Briefings in Bioinformatics*, 23(2):bbab574, 2022. DOI:https://doi.org/10.1093/bib/bbab574.

[112] D. Wang, D. Liu, J. Yuchi, F. He, Y. Jiang, S. Cai, J. Li, and D. Xu. Musitedeep: a deep-learning based webserver for protein post-translational modification site prediction and visualization. *Nucleic Acids Research*, 48(W1):W140–W146, 2020. DOI:https://doi.org/10.1093/nar/gkaa275.

[113] H. Wang, H. Li, W. Gao, et al. Prub-el: A hybrid framework based on deep learning for identifying ubiquitination sites in arabidopsis thaliana using ensemble learning strategy. *Analytical Biochemistry*, page 114935, 2022. DOI:https://doi.org/10.1016/j.ab.2022.114935.

[114] H. Wang, Z. Wang, Z. Li, and T.-Y. Lee. Incorporating deep learning with word embedding to identify plant ubiquitylation sites. *Frontiers in Cell and Developmental Biology*, 8:572195, 2020. DOI:https://doi.org/10.3389/fcell.2020.572195.

[115] H. Wang, H. Zhao, Z. Yan, et al. Mdcan-lys: A model for predicting succinylation sites based on multilane dense convolutional attention network. *Biomolecules*, 11(6):872, 2021. DOI:https://doi.org/10.3390/biom11060872.

[116] L. Wang and K.-J. Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):3048–3068, 2021. DOI:10.1109/TPAMI.2021.3055564.

[117] X. Wang, R. Yan, Y.-Z. Chen, et al. Computational identification of ubiquitination sites in arabidopsis thaliana using convolutional neural networks. *Plant Molecular Biology*, 105:601–610, 2021. DOI:https://doi.org/10.1007/s11103-020-01112-w.

[118] X. Wang, R. Yan, and Y. Wang. Computational identification of human ubiquitination sites using convolutional and recurrent neural networks. *Molecular Omics*, 17(6):948–955, 2021. DOI:DOIhttps://doi.org/10.1039/D0MO00183J.

[119] B. Wen, W. Zeng, Y. Liao, et al. Deep learning in proteomics. *Proteomics*, 20:21–22, 2020. DOI:https://doi.org/10.1002/pmic.201900335.

[120] D. Whitford. *Proteins: structure and function*. John Wiley & Sons, 2013. DOI:10.1007/978-1-4613-1787-6.

[121] K. Wilkinson. Protein ubiquitination: a regulatory post-translational modification. *Anti-Cancer Drug Design*, 2(2):211–229, 1987.

[122] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 461–470, 2015. DOI:https://doi.org/10.1145/2733373.28062.

[123] L. Yang, S. Miao, J. Zhang, et al. The growing landscape of succinylation links metabolism and heart disease. *Epigenomics*, 13(04):319–333, 2021. DOI:https://doi.org/10.2217/epi-2020-0273.

[124] Y. Yang, H. Wang, J. Ding, et al. iacet-sumo: identification of lysine acetylation and sumoylation sites in proteins by multi-class transformation methods. *Computers in Biology and Medicine*, 100:144–151, 2018. DOI:https://doi.org/10.1016/j.compbiomed.2018.07.006.

[125] D. Zhang and S. Wang. A protein succinylation sites prediction method based on the hybrid architecture of lstm network and cnn. *Journal of Bioinformatics and Computational Biology*, 20(02):2250003, 2022. DOI:https://doi.org/10.1142/S0219720022500032.

[126] L. Zhang, J. Song, A. Gao, et al. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019. DOI:10.1109/ICCV.2019.00381.

[127] Q. Zhao, Y. Xie, Y. Zheng, and et al. Gps-sumo: a tool for the prediction of sumoylation sites and sumo-interaction motifs. *Nucleic Acids Research*, 42(Web Server issue):W325–W330, 2014. DOI:https://doi.org/10.1093/nar/gku383.

[128] F.-F. Zhou, Y. Xue, G.-L. Chen, and X. Yao. Gps: a novel group-based phosphorylation predicting and scoring method. *Biochemical and Biophysical Research Communications*, 325(4):1443–1448, 2004. DOI:https://doi.org/10.1016/j.bbrc.2004.11.001.

[129] Y. Zhu, Y. Liu, Y. Chen, et al. Ressumo: A deep learning architecture based on residual structure for prediction of lysine sumoylation sites. *Cells*, 11(17):2646, 2022. DOI:https://doi.org/10.3390/cells11172646.

# PHỤ LỤC

**Algorithm 5.7** Bagging

**Đầu vào:** Tập dữ liệu huấn luyện $D$, tập dữ liệu kiểm thử $X_{\text{test}}$

**Đầu ra:** Dự đoán nhãn cho $X_{\text{test}}$ bằng mô hình tổ hợp $E$

1: Khởi tạo tập mô hình tổ hợp $E = \{E^{(1)}, E^{(2)}, \ldots, E^{(L)}\}$
2: Khởi tạo tập các bộ phân lớp đơn $C = \{C^{(1)}, C^{(2)}, \ldots, C^{(L)}\}$
3: Xác định số mẫu $n = |D|$
4: **for** $i = 1$ to $L$ **do**
5: Sinh mẫu bootstrap $S^{(i)}$ từ $D$ (lấy mẫu có hoàn lại)
6: Huấn luyện bộ phân lớp $C^{(i)}$ trên $S^{(i)}$, thu được mô hình $E^{(i)}$
7: **end for**
8: **for** $i = 1$ to $L$ **do**
9: $R^{(i)} = E^{(i)}(X_{\text{test}})$ ▷ Dự đoán nhãn trên tập kiểm thử
10: **end for**
11: **Result** = hàm tổ hợp $\{R^{(1)}, R^{(2)}, \ldots, R^{(L)}\}$ (đa số biểu quyết hoặc trung bình) =0

**Algorithm 5.8** Boosting

**Đầu vào:** Tập dữ liệu ban đầu $D$, tập dữ liệu kiểm thử $X_{\text{test}}$

**Đầu ra:** Nhãn dự đoán cho $X_{\text{test}}$ bằng mô hình tổ hợp $E$

Khởi tạo tập mô hình tổ hợp $E = \{E^{(1)}, E^{(2)}, \ldots, E^{(L)}\}$
Khởi tạo tập các mô hình cơ sở $C = \{C^{(1)}, C^{(2)}, \ldots, C^{(L)}\}$
Xác định số mẫu $n = |D|$
$S^{(1)}$ là một tập con ngẫu nhiên từ $D$
**for** $i = 1$ to $L$ **do**
    **if** $i > 1$ **then**
        $S^{(i)}$ là tập các mẫu bị phân loại sai bởi mô hình $E^{(i-1)}$ trên $S^{(i-1)}$
    **end if**
    Huấn luyện mô hình $C^{(i)}$ trên $S^{(i)}$ để tạo $E^{(i)}$
**end for**
**for** $i = 1$ to $L$ **do**
    $R^{(i)} = E^{(i)}(X_{\text{test}})$
**end for**
**Result =** tổ hợp $\{R^{(1)}, \ldots, R^{(L)}\}$ (đa số biểu quyết hoặc trọng số)

**Algorithm 5.9** Stacking

**Đầu vào:** Tập dữ liệu ban đầu $D$, tập dữ liệu kiểm thử $X_{\text{test}}$

**Đầu ra:** Nhãn dự đoán cho $X_{\text{test}}$ bằng mô hình tổ hợp $M$

Khởi tạo tập mô hình tổ hợp $E = \{E^{(1)}, E^{(2)}, \ldots, E^{(L)}\}$
Khởi tạo tập các mô hình cơ sở $C = \{C^{(1)}, C^{(2)}, \ldots, C^{(L)}\}$
Xác định mô hình phân loại meta $K$
**for** $i = 1$ to $L$ **do**
    Huấn luyện mô hình $E^{(i)}$ trên tập dữ liệu $D$
**end for**
$M = \{E^{(1)}, E^{(2)}, \ldots, E^{(L)}\} \cup \{K\}$ ▷ kết hợp mô hình meta với các mô hình con
**Result =** Dự đoán nhãn của $X_{\text{test}}$ bằng mô hình tổ hợp $M$