Sự cacbonyl hóa được coi là một biến đổi sau dịch mã không thể đảo ngược và được coi là một dấu ấn sinh học của stress oxy hóa. Nó đóng một vai trò quan trọng không chỉ trong việc điều phối các quá trình sinh học khác nhau mà còn liên quan đến một số bệnh như bệnh Alzheimer, tiểu đường và bệnh Parkinson. Tuy nhiên, vì các công nghệ thử nghiệm tốn kém và tốn thời gian để phát hiện các vị trí cacbonyl hóa trong protein, một phương pháp tính toán chính xác để dự đoán các vị trí cacbonyl hóa là một vấn đề cấp bách có thể hữu ích cho việc phát triển thuốc. Trong nghiên cứu này, một công cụ tính toán mới có tên là predCar-Site đã được phát triển để dự đoán các vị trí cacbonyl hóa protein bằng cách (1) kết hợp thông tin liên kết trình tự vào thành phần axit amin giả chung, (2) cân bằng ảnh hưởng của tập dữ liệu huấn luyện lệch theo Khác nhau Phương pháp Chi phí lỗi và (3) xây dựng một công cụ dự đoán bằng cách sử dụng máy vectơ hỗ trợ làm bộ phân loại. Công cụ dự đoán predCar-Site này đạt được điểm AUC (diện tích dưới đường cong) trung bình là 0,9959, 0,9999, 1 và 0,9997 khi dự đoán các vị trí cacbonyl hóa của K, P, R và T tương ứng. Tất cả các kết quả thử nghiệm cùng với AUC được tìm thấy từ mức trung bình của 5 lần chạy hoàn chỉnh xác thực chéo 5 lần và những kết quả đó cho thấy hiệu suất tốt hơn đáng kể so với các dự đoán hiện có. Một máy chủ web thân thiện với người dùng của predCar-Site có sẵn tại: https/…(ko còn tồn tại)

# Intro

Sự đa dạng về cấu trúc và chức năng của **protein** cũng như tính dẻo và tính động của tế bào sống **bị chi phối đáng kể bởi các biến đổi sau dịch mã (PTM)** [1]. Không chỉ vậy, PTM còn có nhiệm vụ mở rộng mã di truyền và điều hòa sinh lý tế bào. [2, 3]. Một loạt các **PTM như hydroxyl hóa, nitrat hóa, sulfhydryl hóa, cacbonyl hóa và glutathionyl hóa** đã được **tạo ra từ stress oxy hóa** [4] **là kết quả** trực tiếp của sự **mất cân bằng** trong sản xuất và phân hủy các loại phản ứng oxy (reactive oxygen species-**ROS**) và các loại phản ứng nitơ (reactive nitrogen species-**RNS**) [5]. **Stress oxy hóa** có thể xảy ra khi **sinh dư thừa các loại phản ứng oxy (ROS)** với **khả năng giải độc của tế bào** và làm suy yếu khả năng sửa chữa tổn thương [5, 6, 7, 8].

Trong số nhiều loại PTM do stress oxy hóa gây ra, **sự cacbonyl hóa protein được coi là dấu ấn sinh học cho stress oxy hóa** do một số đặc điểm riêng biệt của nó như **hình thành tương đối sớm, ổn định và không thể đảo ngược** [9, 10]. Tuy nhiên, **mật độ cacbonyl hóa** của protein **tăng** lên cùng với **sự tăng của stress oxy hóa bên ngoài**, lão hóa và béo phì là dấu hiệu của giai đoạn đầu của bệnh [11, 12]. Nhiều loại bệnh chính của con người bao gồm **bệnh Alzheimer, tiểu đường, bệnh Parkinson, suy thận mãn tính, bệnh phổi mãn tính, nhiễm trùng huyết** có liên quan đến quá trình carbolnyl hóa protein [9, 13].

Do đó, việc xác định các vị trí cacbonyl hóa trong protein đã trở thành một câu hỏi quan trọng trong sinh lý học và bệnh học tế bào, do đó, giúp cung cấp một số bằng chứng có giá trị cho cả nghiên cứu y sinh và phát triển thuốc [5, 6].

Khối phổ- Mass spectrometry và sắc ký lỏng- liquid chromatography là những kỹ thuật phổ biến nhất để phân tích tính nhạy cảm với protein của PTMs oxy hóa và xác định vị trí cacbonyl hóa chính xác của nó gần đây. Cần lưu ý rằng trong số tất cả các phần còn lại của các phân tử protein, **bốn loại dư lượng** axit amin, cụ thể là lysine (K), proline (P), arginine (R) và threonine (T), đã được tìm thấy **dễ bị cacbonyl hóa** [16 –18]. Tuy nhiên, kỹ thuật thực nghiệm thuần túy để xác định vị trí chính xác của chất nền cacbonyl hóa rất tốn kém cũng như tốn kém thời gian, đặc biệt là đối với các bộ dữ liệu quy mô lớn [8, 14].

Trong bối cảnh này, nhu cầu cao là sử dụng các phương pháp tính toán để xác định các vị trí cacbonyl hóa một cách hiệu quả và chính xác [5, 6]. Gần đây, nhiều loại bộ phân loại tính toán khác nhau đã được phát triển để xác định các vị trí cacbonyl hóa thông qua các loại thuật toán học máy khác nhau [5, 6, 19, 20]. Tuy nhiên, để đáp ứng nhu cầu hiện tại nhằm sản xuất các công cụ thông lượng cao hiệu quả, cần có thêm nỗ lực để cải thiện hơn nữa chất lượng dự đoán [5, 6].

Trong sự phát triển của bộ phân loại tính toán, một trong những thách thức lớn là xử lý vấn đề **tập dữ liệu mất cân bằng** [6, 21], vì nó được tìm thấy trong hầu hết các tập dữ liệu cho loại dự đoán này, tập con **số âm lớn hơn nhiều so với số dương**. Như bức tranh thực tế là các vị trí không cacbonyl hóa luôn chiếm đa số so với các vị trí cacbonyl hóa, do đó, dự đoán tự nhiên sẽ thiên về các vị trí không phải cacbonyl hóa. Vấn đề ở đây là, đối với loại dự báo này có thể giải thích nhiều vị trí cacbonyl hóa là các vị trí không cacbonyl hóa [22, 23, 24]. Tuy nhiên**, thông tin về các vị trí cacbonyl hóa chủ yếu được mong muốn hơn là các vị trí không được cacbonyl hóa.** Do đó, điều quan trọng là phải **tìm ra một giải pháp hiệu quả để cân bằng loại hệ quả thiên vị này.**

Nghiên cứu hiện tại đã bắt đầu với nỗ lực giải quyết các vấn đề nêu trên và sau đó cố gắng phát triển một công cụ dự báo mạnh mẽ hơn bằng cách sử dụng **support vector machine.** Trong dự đoán của chúng tôi**, phương pháp Different Error Costs (DEC) đã được sử dụng để giải quyết vấn đề mất cân bằng dữ liệu.** Ở đây cần lưu ý rằng **trích chọn đặc trưng** được trích xuất bằng cách sử dụng mô hình ghép nối trình tự được đánh giá cao - **vctorized sequence-coupling model.** Trong các nghiên cứu gần đây, hiệu suất của PTMPred [19], CarSpred [5], và iCar-PseCp [6] trên một tập hợp lớn các protein đã được nghiên cứu trong [6]. Do đó, để so sánh hiệu suất của predCar-Site với các hệ thống đó (PTMPred [19], CarSpred [5] và iCar-PseCp [6]), chúng tôi sử dụng cùng một tập dữ liệu và sử dụng 10-fold cross-validation. Vì thông tin về **10-fold-crossvalidation trong các nghiên cứu trước đây [6] không có sẵn**, vì vậy chúng tôi đã **thực hiện 5 complete runs của 10 fold-crossvalidation**, trong đó **mỗi complete runs uses a different 10-way splits.** Việc sử dụng nhiều lần chạy với các mức phân chia khác nhau giúp xác nhận tính ổn định và ý nghĩa thống kê của kết quả. Cuối cùng, kết quả trung bình của tất cả các số liệu được tìm thấy từ nghiên cứu này đã được báo cáo. Kết quả thử nghiệm của chúng tôi chỉ ra rằng predCar-Site đạt được kết quả tốt hơn đáng kể so với kết quả được tìm thấy từ các hệ thống hàng đầu khác (PTMPred [19], CarSpred [5] và iCar-PseCp [6]).

**Để đưa ra một công cụ dự đoán** thống kê dựa trên trình tự hữu ích cho một hệ thống sinh học như đã được chứng minh trong một loạt các ấn phẩm gần đây [6, 21, 30-37], các **quy tắc năm bước của Chou [29] cần được tuân thủ**:

* (i) cấu trúc hoặc chọn một **tập dữ liệu chuẩn hợp lệ để train và test** công cụ dự đoán
* (ii) **xây dựng các mẫu trình tự** sinh học **với một biểu thức toán học hiệu quả** có thể phản ánh thực sự mối tương quan nội tại của chúng với mục tiêu được dự đoán
* (iii) giới thiệu hoặc phát triển một thuật toán mạnh mẽ để vận hành dự đoán
* (iv) thực hiện đúng các thử nghiệm xác thực chéo để đánh giá khách quan độ chính xác dự đoán của nó
* (v) thiết lập một máy chủ web thân thiện với người dùng hoặc gói phần mềm có thể truy cập được cho công chúng

# Phương pháp cụ thể

## Benchmark Dataset

Bộ dữ liệu điểm chuẩn của iCar-PseCp [6] đã được sử dụng trong nghiên cứu này. Bộ dữ liệu của iCar-PseCp được lấy từ 230 chuỗi protein cacbonyl hóa từ người [15, 38-41] và 20 chuỗi protein cacbonyl hóa từ Photobacterium và Escherichia coli [17, 39, 42, 43].

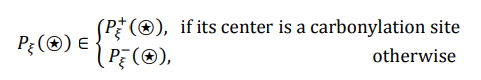
Trong iCar-PseCp [6], theo sơ đồ của Chou, một mẫu peptit thường được biểu thị:



Trong đó ký hiệu ⍟ đại diện cho mã axit amin sinlge K, P, R hoặc T; ξ là một số nguyên,

* đại diện cho up-stream residues axit amin thứ từ trung tâm;
* đại diện cho down-stream residues axit amin

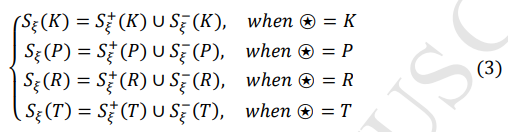
Mỗi mẫu (2 ξ +1)-tuple peptide chia thành 2 loại:



: true carbonylation với K, P, R, or T ở vị trí trung tâm

: false carbonylation với K, P, R, or T ở vị trí trung tâm

Cuối cùng dataset tại iCar-PseCp sẽ có dạng:

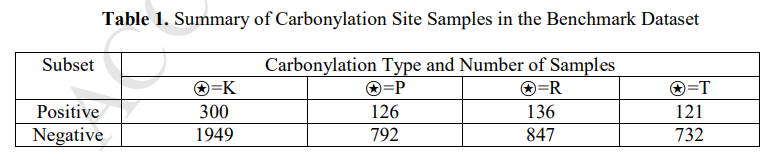


Trong đó chỉ gồm các mẫu true carbonylation , và chỉ gồm các mẫu false carbonylation , và ∪ là biểu thức hợp.

Trong công việc của iCar-PseCp, cửa sổ peptit (2 ξ +1) -tuple được sử dụng để thu thập đoạn peptit có K, P, R hoặc T ở trung tâm. Ở đây cần lưu ý rằng nếu phần upstream và downstream trong trình tự protein nhỏ hơn ξ hoặc lớn hơn L-ξ (L là độ dài của trình tự protein liên quan) thì axit amin thiếu đã được lấp đầy bằng một lượng residues X (với các giá trị đặc tính hóa lý được gán = 0)

Sau khi áp dụng một số quy trình sàng lọc dựa trên một số ràng buộc đối với các mẫu peptit đã thu thập đó, ví dụ, xem xét kích thước cửa sổ, <= 30% nhận dạng trình tự theo cặp đối với bất kỳ peptit nào khác, iCar-PseCp cuối cùng đã xây dựng một tập dữ liệu chuẩn [6]. Quy trình chi tiết về việc xây dựng tập dữ liệu điểm chuẩn của iCar-PseCp được giải thích trong [6].

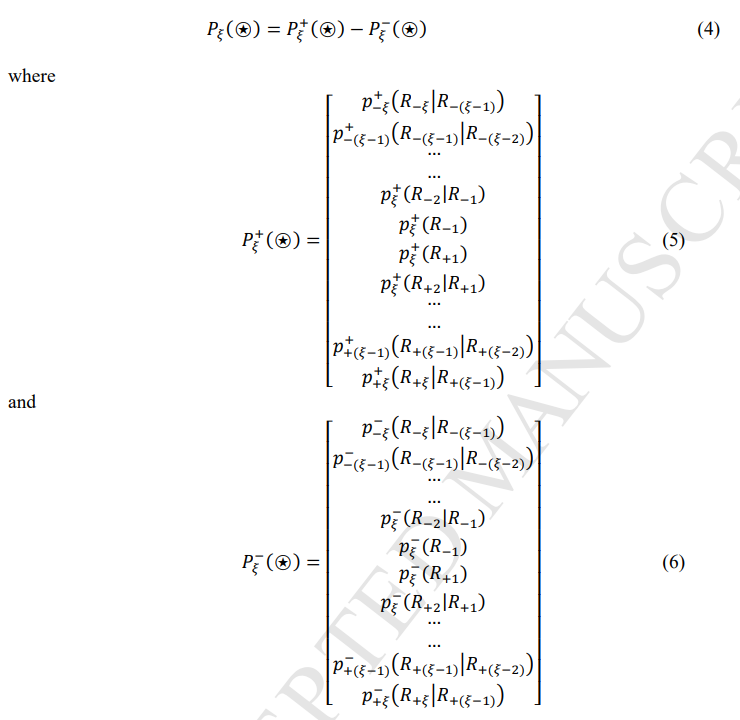
Lưu ý rằng, tùy thuộc vào một số thử nghiệm sơ bộ, kích thước cửa sổ được chọn là 15 (2 ξ +1) trong nghiên cứu của iCar-PseCp, trong đó ξ = 7. Do đó, tập dữ liệu điểm chuẩn do iCar-PseCp thu được cho), ), ), and ) có sẵn tại các tài liệu bổ sung trực tuyến (http://research.ru.ac.bd/predCar-Site/) dưới dạng Thông tin hỗ trợ S1, S2, S3 và S4, tương ứng. Cần lưu ý rằng các tài liệu bổ sung trực tuyến đã xuất bản của chúng tôi được lấy từ công việc của iCar-PseCp [6]. Tóm tắt về tập dữ liệu điểm chuẩn này được đưa ra trong Bảng 1.



## Trích các đặc trưng

Các đặc trưng thích hợp của trình tự hoặc mẫu protein đóng vai trò rất quan trọng đối với việc dự đoán vị trí cacbonyl hóa, do đó, nó thu hút nhiều sự chú ý của các nhà khoa học rằng làm thế nào để chọn ra các đặc điểm cốt lõi và thiết yếu của các mẫu protein. Vì hầu hết các thuật toán học máy hiện có chỉ có thể xử lý vector mà không phải mẫu tuần tự, một trong những vấn đề quan trọng trong tin sinh học là làm thế nào để tách vector từ chuỗi sinh học mà vẫn giữ được các đặc điểm trình tự đáng kể [37]. Trong bài báo này, để tránh làm mất hoàn toàn thông tin về mẫu trình tự cho protein, **general PseAAC của Chou** đã được áp dụng để chiết xuất tính năng từ đoạn peptit bằng cách sử dụng mô hình nối trình tự [28, 44] được mô tả ngắn gọn bên dưới.

Bây giờ, dựa trên khái niệm về thông tin được ghép nối với trình tự [28, 44] vào general PseAAC, trình tự peptit của Eq. (1) có thể được xây dựng dưới dạng:

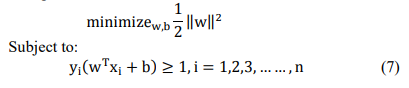


* là xác suất có điều kiện của axit amin xảy ra ở vị trí thứ nhất bên trái cho rằng láng giềng bên phải gần nhất của nó là
* Chỉ có , có xác suất không có điều kiện vì hàng xóm bên phải của và bên trái luôn là ⍟ (K, P, R, or T)

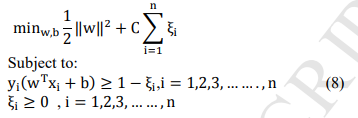
Tất cả các giá trị xác suất này có thể dễ dàng thu được từ tập dữ liệu chuẩn dương được đưa ra trong Supporting Information S1, S2, S3 và S4, tương ứng trong [44] ( ) ) . Tương tự với tập âm.

## SVM Classification

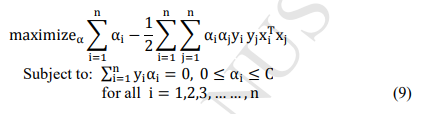
Xét bài toán tách tập các training vectors vào hai lớp riêng biệt , ,… trong đó và là nhãn lớp tương ứng 1<=i<=n. Nhiệm vụ chính của bài toán này là tìm một bộ phân loại có hàm quyết định f (x, θ) sao cho y = f (x, θ), trong đó y là nhãn lớp của x và θ là vectơ chưa biết tham số của chức năng quyết định. Máy vector hỗ trợ là một máy phân loại nổi tiếng và nó đã được ứng dụng rộng rãi trong nhiều bài toán phân loại. Về mặt hình học, thuật toán mô hình hóa SVM tìm thấy một siêu phẳng tối ưu với biên lớn nhất để tách hai lớp [45], điều này yêu cầu giải quyết vấn đề ràng buộc sau:



Để cho phép lỗi, vấn đề tối ưu hóa bây giờ trở thành:



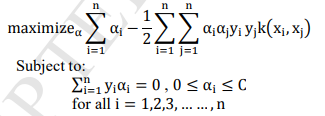
Sử dụng phương pháp nhân Lagrange, chúng ta có thể thu được công thức kép được biểu diễn dưới dạng các biến [45, 46]:



Cuối cùng, hàm phân biệt tuyến tính có dạng sau :



Trong nhiều ứng dụng, bộ phân loại phi tuyến tính cung cấp độ chính xác tốt hơn. Trong SVM, cách đơn giản để tạo bộ phân loại phi tuyến tính ra khỏi bộ phân loại tuyến tính là ánh xạ dữ liệu của chúng ta từ không gian đầu vào X sang không gian đặc trưng F bằng cách sử dụng hàm phi tuyến tính ∅: X → F. Trong không gian F, tối ưu hóa có dạng sau bằng cách sử dụng hàm nhân [45, 46, 47]:



Bây giờ, về mặt hàm nhân, hàm phân biệt có dạng sau:



Ở đây cần lưu ý rằng một hàm nhân và tham số của nó phải được chọn để xây dựng bộ phân loại SVM [45, 46, 47]. Trong công việc này, radial basis function kernel đã được sử dụng để xây dựng bộ phân loại SVM được định nghĩa dưới đây:



### Quản lý vấn đề tập dữ liệu mất cân bằng