

CAFA 6 Protein Function Prediction Report

Trung Hau Tran, Trung Hieu Pham, Nam Khanh Pham
VNU University of Engineering and Technology
Hanoi, Vietnam

Tóm tắt nội dung—Bài báo cáo này trình bày phương pháp của chúng tôi cho CAFA 6 Protein Function Prediction. Cuộc thi này tập trung vào việc xây dựng mô hình học máy có khả năng dự đoán được chức năng sinh học của protein chỉ dựa trên trình tự axit amin của các protein đó. Dữ liệu gồm các chuỗi protein và trình tự axit amin tương ứng, cùng với các nhãn chức năng sinh học được gán với Gene Ontology (GO).

Chúng tôi áp dụng phương pháp học máy và học sâu, kết hợp các kỹ thuật trích xuất đặc trưng, lựa chọn mô hình embedding phù hợp và kỹ thuật ensemble mô hình để xây dựng mô hình dự đoán chức năng protein.

Điểm cuối cùng đạt được là 0.358, top 56/460 đội (tính đến ngày 9/8/2025).

I. GIỚI THIỆU

Protein được coi là những cỗ máy phân tử đảm nhiệm hầu hết các chức năng thiết yếu của sự sống, từ xúc tác các phản ứng sinh hóa, vận chuyển vật chất, truyền tín hiệu đến việc tạo nên cấu trúc tế bào. Trong những thập kỷ gần đây, sự bùng nổ của công nghệ giải trình tự gen thế hệ mới (Next-Generation Sequencing - NGS) đã tạo ra một lượng dữ liệu khổng lồ về trình tự protein. Các cơ sở dữ liệu như UniProt hiện chứa hàng trăm triệu chuỗi protein từ đa dạng các loài sinh vật.

Tuy nhiên, tốc độ xác định chức năng của các protein này bằng thực nghiệm (như tinh thể học tia X hay cộng hưởng từ hạt nhân) lại chậm hơn rất nhiều do chi phí cao và quy trình phức tạp. Sự chênh lệch ngày càng lớn giữa lượng dữ liệu trình tự đã biết và lượng dữ liệu chức năng đã được xác thực tạo nên một thách thức lớn trong sinh học tính toán, được gọi là "Khoảng trống chú giải" (The Annotation Gap). Việc phát triển các thuật toán máy tính để dự đoán tự động chức năng protein (Automated Function Prediction - AFP) trở thành một nhu cầu cấp thiết để lấp đầy khoảng trống này.

Để thúc đẩy sự phát triển của các thuật toán AFP, cuộc thi CAFA (Critical Assessment of Functional Annotation) đã được tổ chức như một thử thách toàn cầu, nơi các nhà nghiên cứu tranh tài để dự đoán chức năng của một tập hợp các protein chưa được chú giải. CAFA 6 là phiên bản mới nhất của chuỗi thử thách này, đặt ra yêu cầu cao hơn về độ chính xác và khả năng tổng quát hóa của mô hình trên các loài sinh vật khác nhau.

Trong báo cáo này, chúng tôi trình bày các phương pháp đã áp dụng khi tham gia CAFA 6, bao gồm mô hình hóa bài toán, xây dựng khung quy trình tiền xử lý dữ liệu, tiếp cận bài toán theo ba hướng: Không học máy (non-ML), học máy (ML) và học sâu (DL).

II. BACKGROUND

A. CAFA 6 Protein Function Prediction

B. Tập dữ liệu

Dữ liệu sử dụng trong nghiên cứu này được cung cấp bởi CAFA 6 gồm các tệp tin chính đóng vai trò đầu vào và nhãn mục tiêu cho quá trình huấn luyện:

- **train_sequences.fasta** (Dữ liệu thô):
 - Trình tự axit amin (primary sequences) của khoảng 140.000 protein.
 - Định dạng: FASTA. Mỗi mục gồm mã định danh (EntryID) và chuỗi ký tự đại diện cho các axit amin. (Ví dụ: M, K, T, L, ...). Đây là dữ liệu đầu chính cho mô hình.
- **train_terms.tsv** (Dữ liệu nhãn):
 - Chứa thông tin gán nhãn chức năng cho các protein trong tập huấn luyện.
 - Cấu trúc: EntryID <-> GO Terms (Mã chức năng) <-> Aspect (Nhóm chức năng: BPO, MFO, CCO).
 - Đặc điểm: Một protein có thể tương ứng với nhiều nhãn GO khác nhau (Multi-label).
- **go-basic.obo** (Cấu trúc Ontology):
 - File định nghĩa cấu trúc đồ thị của Gene Ontology, mô tả mối quan hệ cha-con giữa các thuật ngữ chức năng.
- **train_taxonomy.tsv** (Thông tin loài):
 - Cung cấp mã định danh loài (Taxon ID) cho từng protein. Dữ liệu này giúp mô hình phân biệt đặc điểm sinh học giữa các loài khác nhau (ví dụ: vi khuẩn vs. động vật có vú).

III. PHƯƠNG PHÁP

A. Tiền xử lý dữ liệu

Dữ liệu thô ban đầu là các chuỗi trình tự protein (dạng văn bản) trong tệp FASTA. Để đưa vào các mô hình học máy, chúng tôi đã thử nghiệm và triển khai hai phương pháp tiếp cận tiền xử lý dữ liệu riêng biệt nhằm khai thác tối đa thông tin từ trình tự:

- **Phương pháp 1: Trích xuất đặc trưng thủ công (Manual Feature Engineering)** - Dựa trên kiến thức sinh học thống kê.
- **Phương pháp 2: Học biểu diễn ngữ nghĩa (Semantic Representation Learning)** - Dựa trên mô hình ngôn ngữ lớn (ESM-2).

1) Phương pháp 1: Trích xuất đặc trưng thủ công:

Ở phương pháp này, chúng tôi coi chuỗi protein là một tập hợp các thành phần hóa học và sử dụng công thức thống kê để biến đổi chuỗi thành vector số học. Phương pháp này tập trung vào đặc điểm thành phần (composition) thay vì thứ tự sắp xếp.

a. Xử lý dữ liệu thô

- Đọc dữ liệu: sử dụng thư viện Biopython để phân tích cú pháp tệp FASTA.
- Làm sạch ID: Tách phân tiêu đề (Header) của FASTA để lấy mã định danh chuẩn (EnrID).

b. Kỹ thuật tạo đặc trưng (Feature Engineering)

Chúng tôi xây dựng vector đặc trưng X_{manual} dựa trên các thuộc tính lý hóa của protein:

- Thành phần Axit Amin (Amino Acid Composition-AAC): tính toán tần suất xuất hiện của 20 loại axit amin chuẩn (A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V) trong mỗi chuỗi. Với công thức:

$$AAC_i = \frac{\text{Count}(AA_i)}{\text{Length}(\text{Sequence})} \quad (1)$$

- Độ dài chuỗi (Sequence Length): Thêm đặc trưng về tổng số lượng axit amin. Độ dài chuỗi thường tương quan với độ phức tạp của chức năng protein.

Kết quả là mỗi protein được biểu diễn bằng một vector chiều thấp (Low-dimensional vector) với kích thước $D \approx 21 - 25$.

2) Phương pháp 2: Học biểu diễn ngữ nghĩa (Semantic Representation Learning):

Ở phương pháp này, chúng tôi coi chuỗi protein là một "ngôn ngữ" và sử dụng mô hình học sâu đã được huấn luyện trước (Pre-trained Model) để trích xuất các đặc trưng ngữ nghĩa tiềm ẩn. Phương pháp này nắm bắt được ngữ cảnh, thứ tự và cấu trúc 3D của protein.

a. Mô hình nền tảng

Chúng tôi sử dụng mô hình ESM-2 (Evolutionary Scale Modeling) phiên bản 650 triệu tham số (esm2_t33_650M_UR50D). Đây là mô hình Transformer tiên tiến nhất hiện nay cho dữ liệu protein.

b. Quy trình embedding

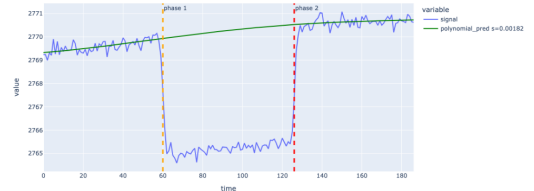
- **Bước 1: Mã hóa (tokenization)** - Mỗi axit amin được chuyển đổi thành token số nguyên. Các token đặc biệt <cls> và <eos> được thêm vào để đánh dấu bắt đầu và kết thúc chuỗi.
- **Bước 2: Suy luận (inference)** - Chuỗi token được đưa qua 33 lớp Transformer của ESM-2. Tại đây, mô hình sử dụng cơ chế Self-Attention để học mối quan hệ phức tạp giữa các axit amin, bất kể khoảng cách của chúng trong chuỗi.
- **Bước 3: Gộp (pooling)** - Chúng tôi lấy vector trạng thái ẩn (hidden states) tại lớp cuối cùng và

áp dụng kỹ thuật Mean Pooling (lấy trung bình) dọc theo chiều dài chuỗi để thu được một vector đại diện duy nhất cho toàn bộ protein.

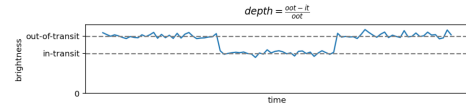
Kết quả là mỗi protein được biểu diễn bằng một vector với kích thước $D = 1280$ (High-dimensional vector). Vector này chứa đựng thông tin phong phú về cấu trúc và chức năng của protein.

B. Tiếp cận theo hướng không dùng học máy

Trong bối cảnh bài toán này, có một phương pháp non-ML mang lại hiệu quả đáng kinh ngạc so với các phương pháp sử dụng thuật toán Machine Learning, Deep Learning. Chúng tôi tạm gọi nó là heuristic method. Có một số lý do khiến model này hiệu quả. Thứ nhất, chúng tôi nhận thấy khi lấy trung bình tín hiệu trên từng bước thời gian sau khi xử lý tín hiệu, ta được Hình 1 và Hình 2.



Hình 1. Trung bình tín hiệu trên từng bước thời gian



Hình 2. Insight về hiện tượng transit

Có thể quan sát thấy rằng tín hiệu thu được thường có dạng lõm ở phần giữa, phản ánh hiện tượng hành tinh đi qua phía trước ngôi sao chủ (transit), khiến cường độ ánh sáng giảm xuống do bị che khuất. Độ suy giảm này đạt cực đại khi hành tinh nằm tại vị trí trung tâm của ngôi sao trên đường đi của nó.

Dựa trên hiện tượng vật lý này, thay vì huấn luyện một mô hình học máy, chúng tôi đề xuất một phương pháp tiếp cận thực nghiệm, trong đó tìm kiếm trực tiếp một hệ số khuếch đại tối ưu cho từng phổ tín hiệu.

Cụ thể, với mỗi phổ, chúng tôi thực hiện phát hiện khoảng thời gian xảy ra transit, bằng cách xác định hai điểm mốc là **phase1** và **phase2**—tương ứng với thời điểm bắt đầu và kết thúc của quá trình transit. Quá trình này được thực hiện thông qua hàm **phase_detector**, dựa trên đạo hàm bậc nhất của tín hiệu. Đạo hàm được sử dụng bởi vì ta thấy phase1 là nơi tín hiệu giảm nhanh nhất, phase2 là điểm tín hiệu tăng nhất. Do đó, ta dùng đạo hàm bậc nhất để tìm điểm có đạo hàm cực tiểu và cực đại trong khoảng có tiềm năng.

Sau khi xác định được đoạn tín hiệu chứa hiện tượng transit, chúng tôi tìm một hệ số khuếch đại s sao cho khi nhân đoạn tín hiệu này với $(1 + s)$ thì toàn bộ tín hiệu trở nên “mượt” nhất có thể khi được xấp xỉ bằng một đa thức. Độ “mượt” này được định lượng bằng các hàm sai số dùng trong hồi quy giữa tín hiệu đã điều chỉnh và đường cong hồi quy. Cụ thể, trước tiên ta sẽ dùng hồi quy đa thức để fit phần out-of-transit với một đường đa thức nào đó. Sau đó, dùng phương pháp tối ưu hóa Nelder-Mead để tìm s sao cho hàm lỗi là nhỏ nhất khi nhân phần in-transit với $(1+s)$. Sau nhiều thử nghiệm, chúng tôi nhận thấy rằng sử dụng đa thức bậc ba với hồi quy (degree = 3) sẽ cho kết quả ổn định và chính xác nhất.

Để rõ ràng hơn, ta giả sử đoạn tín hiệu trong khoảng [phase1:phase2] tương ứng với pha transit có giá trị trung bình là it (in-transit), và phần còn lại có giá trị trung bình là oot (out-of-transit). Khi áp dụng hệ số khuếch đại s vào vùng transit, ta có:

$$\text{Adjusted in-transit signal} = it \cdot (1 + s) \quad (2)$$

Giả định rằng việc điều chỉnh này làm cho mức tín hiệu trong vùng transit tiệm cận với mức tín hiệu ngoài transit, tức là:

$$it \cdot (1 + s) = oot \quad (3)$$

Từ đó, ta suy ra:

$$s = \frac{oout - it}{it} \quad (4)$$

Mặt khác, transit depth (mức tiêu cần dự đoán) được định nghĩa là:

$$\text{transit depth} = \frac{oout - it}{oot} \quad (5)$$

Ta thay s vào biểu thức trên:

$$\text{transit depth} = \frac{s \cdot it}{oot} = \frac{s \cdot it}{it(1 + s)} = \frac{s}{1 + s} = 1 - \frac{1}{1 + s} \quad (6)$$

Vậy:

$$\boxed{\text{transit depth} = 1 - \frac{1}{1 + s}} \quad (7)$$

Công thức này cho thấy mối quan hệ chặt chẽ giữa hệ số khuếch đại s và transit depth (mức tiêu cần dự đoán), giúp diễn giải rõ ràng giá trị đầu ra của mô hình non-ML theo cách có ý nghĩa vật lý.

C. Tiếp cận theo hướng sử dụng học máy

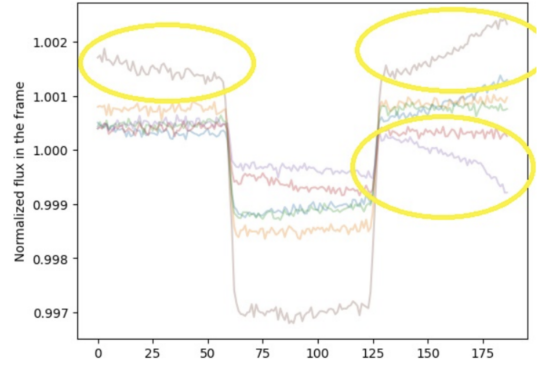
Chúng tôi đã thử một số phương pháp học máy khác nhau, gồm: K-Nearest Neighbors (KNN), Gaussian Process Regressor, phương pháp denoise bằng Auto Encoder. Tuy nhiên, kết quả thu được không khả quan, do dữ liệu có quá nhiều nhiễu và không đủ thông tin để mô hình hóa chính xác.

IV. THỰC NGHIỆM

A. Tối ưu các hệ số mô hình non-ML

Ý tưởng của phương pháp non-ML đơn giản nhưng để thực nghiệm hiệu quả thì cần phải chọn các tham số phù hợp dựa vào các quan sát trên dữ liệu.

Như đã đề cập ở trên, chúng tôi nhân đoạn tín hiệu transit với $1 + s$ và kỳ vọng đoạn tín hiệu thu được sẽ mượt như khi không có hành tinh nào đi qua sao chủ. Tín hiệu này sẽ được xấp xỉ bằng một đa thức. Việc chọn bậc của đa thức này ảnh hưởng lớn đến độ chính xác của mô hình. Nhìn vào dữ liệu như hình 3, ta thấy rằng sau khi

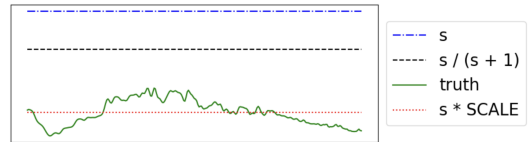


Hình 3. Hình dạng phổ biến của flux theo thời gian trong tập dữ liệu

nhân phần lõm ở giữa với $1 + s$ thì tín hiệu không phải lúc nào cũng là 1 đường tuyến tính. Hình dạng phổ biến là bậc 1, 2, 3. Chúng tôi thực nghiệm với các giá trị bậc đa thức khác nhau cho kết quả như bảng A..

Bậc đa thức	1	2	3	4	5	10
Điểm	0.314	0.315	0.322	0.310	0.304	0.115

Sau khi fit tín hiệu với một đa thức bậc 3, ta tìm s tối ưu (hiệu trị tuyệt đối nhỏ nhất), chúng tôi dự đoán transit depth dựa vào s .



Hình 4. Mối quan hệ giữa s , mức tiêu cần dự đoán

Nhận thấy rằng, nếu lấy giá trị dự đoán theo $\frac{s}{1+s}$ cho ra kết quả thấp hơn so với lấy $s * SCALE$. Hơn nữa, khi ta tính giá trị $SCALE = \text{truth.mean}/s$ với mỗi điểm dữ liệu, thì giá trị này sẽ gần bằng nhau cho tất cả các hành tinh. Chạy thực nghiệm cho thấy, sử dụng $SCALE = 0.9396$ cho điểm cao nhất.

Dựa vào công thức tính điểm, để tối ưu điểm không chỉ cần dự đoán chính xác mà còn cần dự đoán độ không chắc

chấn phù hợp. Giả định rằng, giá trị dự đoán của mô hình khớp với trung bình của quang phổ cần dự đoán, như vậy

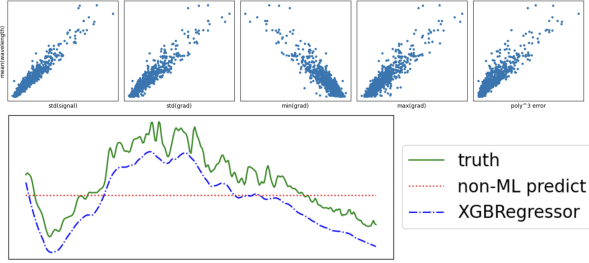
$$\sigma^* = \underset{\sigma}{\operatorname{argmax}} \sum_{i=1}^{283} \left(\log(\sigma^2) + \frac{(\text{truth}[i] - \mu_{\text{truth}})^2}{\sigma^2} \right) \quad (8)$$

Dễ thấy, hàm mục tiêu là tổng 2 hàm lồi,

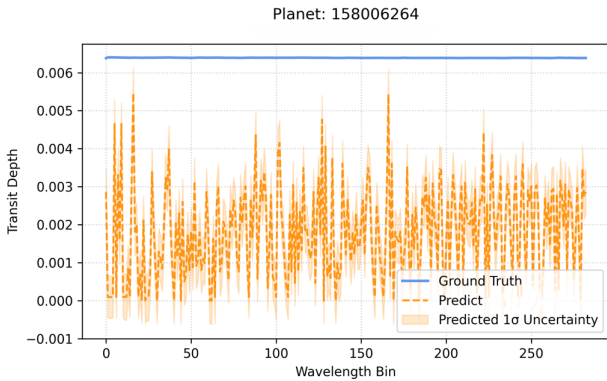
$$(\sigma^*)^2 = \sum_{i=1}^{283} (\text{truth}[i] - \mu_{\text{truth}})^2 = (\sigma_{\text{truth}})^2 \quad (9)$$

Nhận thấy rằng, giá trị σ tối ưu cho từng hành tinh là khác nhau và có liên quan đến tín hiệu theo trục thời gian. Chúng tôi sử dụng mô hình phụ LinearRegression với đầu vào là tín hiệu, đầu ra là σ_{truth} , và thực hiện huấn luyện mô hình này trên tập dữ liệu huấn luyện. Kết quả tăng 0.004 so với việc sử dụng giá trị σ cố định cho tất cả các hành tinh.

B. Kết quả



Hình 5. Kết quả của mô hình Regressor so với non ML

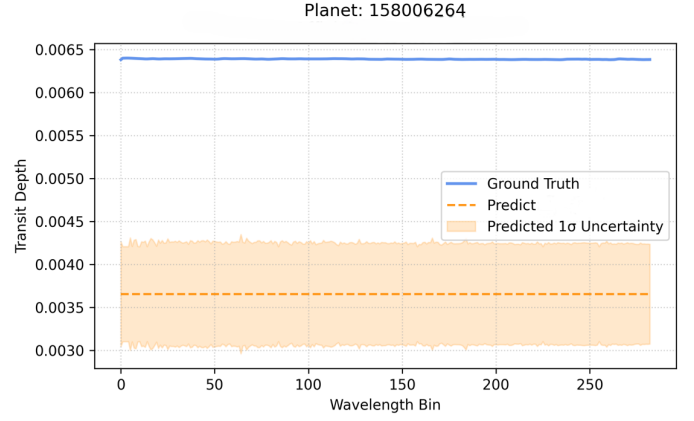


Hình 6. Các bước sóng chưa qua làm mịn

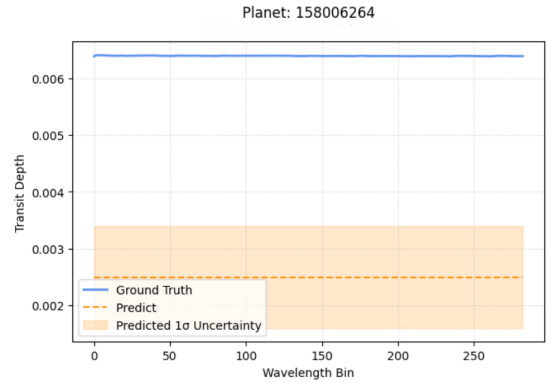
Phương pháp non-ML dự đoán quang phổ và LinearRegression dự đoán độ không chắc chắn cho điểm trên tập valid là 0.75, điểm trên tập test là 0.326. Đạt hạng 56/460 đội (tính đến ngày 9/8/2025).

Với các phương pháp ML cho dự đoán quang phổ,

Chúng tôi thử với mô hình Regressor, lấy thuộc tính là giá trị flux và một số thuộc tính của gradient của flux có liên quan mạnh đến độ sâu transit như hình 5



Hình 7. Các bước sóng đã qua làm mịn



Hình 8. Các bước sóng dự đoán của mô hình non-ML

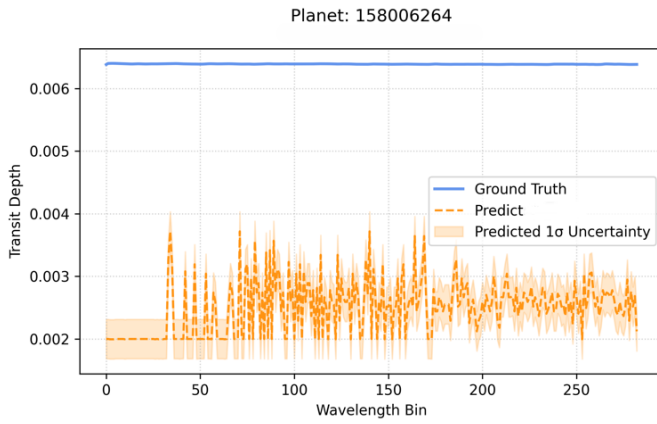
với điểm trên tập valid là 0.734, điểm trên tập test là 0.289, nguyên nhân có thể là do sự khác biệt trong phân phối dữ liệu giữa các tập và chọn thuộc tính chưa phù hợp.

Sự khác biệt về phân phối dữ liệu càng thể hiện rõ hơn khi chúng tôi thử với KNN, với khoảng cách trung bình các điểm dữ liệu trong tập train (với $k = 2$) là 50. Chúng tôi kết hợp KNN và phương pháp non-ML, với điểm dữ liệu trong tập test có khoảng cách với các điểm trong tập train dưới 40, chúng tôi nội suy từ tập train, những điểm lớn hơn thì chúng tôi sử dụng phương pháp non-ML. Kết quả test là 0.322, có hơn phương pháp non-ML nhưng độ chênh lệch quá nhỏ.

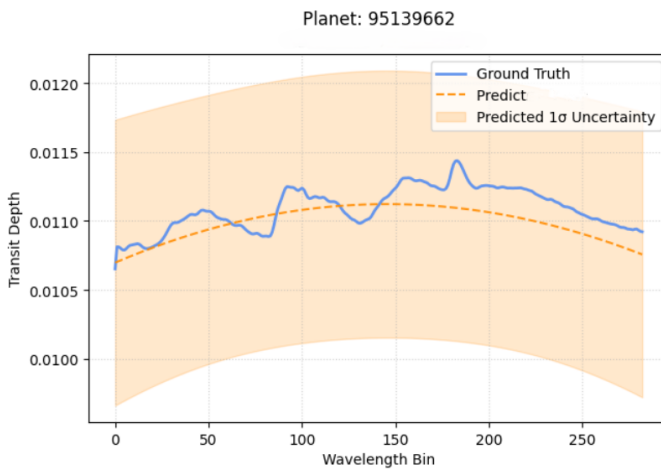
Chúng tôi cũng thử làm mịn kết quả dự đoán của 283 bước sóng bằng Gaussian Process (GP) và AutoEncoder (AE).

Như trong hình 6, các bước sóng chưa làm mịn có nhiều nhiễu và không đồng nhất. Để cải thiện, chúng tôi đã áp dụng Gaussian Process (GP) và AutoEncoder (AE) để làm mịn, khử nhiễu các bước sóng này. Kết quả là các bước sóng trở nên đồng nhất hơn, như trong hình 7.

Đồng thời, GP cũng sinh ra các ước lượng không chắc chắn cho các bước sóng, cho phép chúng tôi đánh giá độ tin cậy σ của các dự đoán. So với bước sóng được dự đoán bằng phương pháp non-ML, như trong hình 8, các

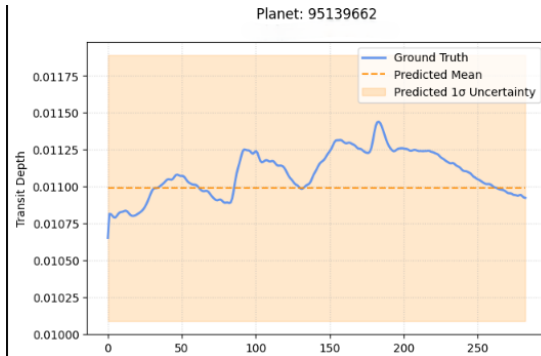


Hình 9. Các bước sóng đã qua làm mượt bằng PCA



Hình 10. Dự đoán của mô hình ML bám sát được hình dạng của ground truth

bước sóng này đã sát với ground truth hơn, 0.00356 so với 0.0024.



Hình 11. Dự đoán của mô hình non-ML không bám sát được hình dạng của ground truth

Một ý tưởng khác để làm mịn bước sóng là sử dụng PCA tìm ra những component chính của các bước sóng này. Kết quả làm mượt bằng PCA cho thấy những tiềm

năng của phương pháp này trong việc cải thiện độ chính xác của dự đoán. Bằng việc sử dụng 5 component chính, chúng tôi đã đạt được những cải thiện đáng kể trong việc giảm thiểu nhiễu và tăng cường độ chính xác cho các bước sóng dự đoán.

Dù việc sử dụng PCA đã mang lại những cải thiện đáng kể trên valid, nhưng vẫn còn nhiều thách thức trong việc xử lý dữ liệu nhiễu và không đồng nhất. PCA với 5 component đem lại kết quả thấp trên test set so với valid set có thể do các tập trong test set phân tán hơn, dẫn đến việc mô hình không thể tổng quát tốt hơn cho các bước sóng này.

So với sử dụng mean của toàn bộ quang phổ như non-ML, việc dự đoán từng bước sóng cải thiện việc bắt được cấu trúc của các bước sóng quang phổ phức tạp, như hình 10. So với việc dự đoán từng bước sóng, phương pháp non-ML chỉ cho ra được một giá trị trung bình cho toàn bộ quang phổ, như hình 11. Tuy nhiên, việc dự đoán từng bước sóng cũng làm tăng độ nhiễu và độ không đồng nhất của các bước sóng này, dẫn đến việc khó khăn trong việc tổng quát khiến những phương pháp chúng tôi thử nghiệm không đạt được kết quả tốt hơn so với non-ML.

V. KẾT LUẬN

Trong bài báo này, chúng tôi đã trình bày phương pháp non-ML để dự đoán quang phổ và linear regression để dự đoán độ không chắc chắn. Phương pháp này sử dụng các kỹ thuật xử lý tín hiệu truyền thống và cho kết quả tốt hơn các phương pháp ML (KNN, GP, AE) chúng tôi đã thử.

Tuy vậy, các phương pháp học máy hiện đang dùng cho kết quả tốt hơn trên tập valid. Do một vài nguyên nhân khiến điểm test chưa tốt. Chúng tôi sẽ tiếp tục nghiên cứu để cải thiện hơn nữa các phương pháp này trong tương lai.

PHỤ LỤC

A. Tỷ lệ đóng góp

Thành viên	Phan Bá Thọ	Nguyễn Quốc Huy	Trần Tuấn Anh
Tỷ lệ đóng góp	34%	33%	33%

Bảng I

TỈ LỆ ĐÓNG GÓP CỦA CÁC THÀNH VIÊN TRONG NHÓM