ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Dự báo nồng độ bụi mịn $PM_{2.5}$ tại Hà Nội bằng mô hình học sâu và phương pháp xử lý dữ liệu thiếu dựa trên Masked Autoencoder

Sinh viên: Lê Minh Đức - 22028267
Hà Tiến Đông - 22028111
Nguyễn Tuấn Anh - 22028303
Bùi Đức Anh - 22028071

Môn học: Trí tuệ nhân tạo
Lớp học phần: INT3401E 2

Giảng viên hướng dẫn: PSG.TS Nguyễn Thị Nhật Thanh
ThS. Hoàng Gia Anh Đức

HÀ NỘI - 2025

Mục lục

Mục lục

Chương 1

Tóm tắt bài toán

1.1 Giới thiệu bài toán

Ô nhiễm không khí, đặc biệt là hạt bụi mịn $PM_{2.5}$, đang trở thành một vấn đề nghiêm trọng tại miền Bắc Việt Nam, nơi có mật độ dân cư cao và nhiều hoạt động công nghiệp. $PM_{2.5}$ (hạt bụi có đường kính nhỏ hơn 2.5 µm) có khả năng xâm nhập sâu vào phổi, gây ảnh hưởng xấu đến sức khỏe cộng đồng, bao gồm các bệnh về hô hấp, tim mạch và thậm chí là ung thư. Ngoài ra, $PM_{2.5}$ còn góp phần làm suy giảm chất lượng môi trường, ảnh hưởng đến hệ sinh thái và tầm nhìn. Mục tiêu của nghiên cứu này là xây dựng và đánh giá các mô hình học máy nhằm dự báo nồng độ $PM_{2.5}$ xung quanh khu vực Hà Nội, dựa trên dữ liệu từ các trạm quan trắc, dữ liệu khí tượng và dữ liệu địa hình. Việc dự báo chính xác chỉ số $PM_{2.5}$ là rất quan trọng để cảnh báo kịp thời cho người dân, hỗ trợ các cơ quan quản lý môi trường đưa ra các biện pháp ứng phó, và góp phần giảm thiểu nguy cơ sức khỏe do ô nhiễm không khí.

1.2 Tổng quan giải pháp

Để giải quyết bài toán, nhóm xác định những khó khăn chính của dữ liệu gồm có:

- Tính chất chuỗi thời gian và không gian phức tạp.
- Thiếu dữ liệu rải rác giữa các trạm, đặc biệt ở các đặc trưng viễn thám.

Để khắc phục những khó khăn do tính chất chuỗi thời gian – không gian phức tạp và việc thiếu hụt dữ liệu tại các trạm quan trắc, giải pháp của nhóm được triển khai qua ba giai đoạn chính.

Đầu tiên, trong bước tiền xử lý và phân chia dữ liệu, nhóm thực hiện khám phá trực quan mối quan hệ giữa $PM_{2.5}$ và các yếu tố khí tượng theo mùa, rồi chia tập train, validation và test theo thứ tự thời gian nhằm ngăn chặn rò rỉ thông tin từ tương lai.

Tiếp theo, bước thiết kế và lựa chọn đặc trưng tập trung vào việc tổng hợp 28 đặc trưng mới - từ chu kỳ trong tuần, vector gió, chỉ số nhiệt độ và điểm sương, đến đặc trưng địa hình và khoảng cách địa lý - và sử dụng kết hợp các phương pháp đánh giá như phân tích tương quan, F-ANOVA, Feature Importance trên mô hình Random Forest/XGBoost, Permutation Variable, giá trị SHAP và Mutual Information để chốt lại 15 đặc trưng đóng góp nhiều nhất.

Cuối cùng, ở giai đoạn xây dựng mô hình, nhóm chọn kiến trúc LSTM đa lớp làm mô hình dự báo, kết hợp với một Masked Autoencoder được thiết kế đặc biệt để tận dụng mối quan hệ không gian (giữa các trạm), thời gian (giữa các ngày) và cục bộ (giữa các đặc trưng trong cùng mẫu) nhằm điền dữ liệu khuyết. Việc kết hợp chặt chẽ hai thành phần này không chỉ đảm bảo không rò rỉ thông tin tương lai, mà còn nâng cao độ chính xác và độ tin cậy khi dự báo nồng độ $PM_{2.5}$.

Chương 2

Phát biểu bài toán

2.1 Dữ liệu

2.1.1 Dữ liệu đầu vào

Dữ liệu được sử dụng trong nghiên cứu bao gồm ba loại chính: dữ liệu trạm quan trắc, dữ liệu khí tượng, và dữ liệu địa hình. Dưới đây là mô tả chi tiết:

Dữ liệu trạm quan trắc

- **Nguồn gốc:** Dữ liệu được thu thập từ các trạm quan trắc không khí quanh khu vực Hà Nội (ví dụ: trạm Station 53), do cơ quan quản lý môi trường thành phố cung cấp.
- **Thông số:**
  - **ID trạm (ID):** Mã định danh duy nhất.
  - **Kinh độ (lon):** Tọa độ Đông - Tây.
  - **Vĩ độ (lat):** Tọa độ Bắc - Nam.
- **Thời gian thu thập:** Từ ngày 1/1/2020 đến 31/12/2021.

Dữ liệu khí tượng

- **Nguồn gốc:** Dữ liệu dự báo thời tiết từ hệ thống GFS (Global Forecast System) của NOAA (Cơ quan Quản lý Khí quyển và Đại dương Quốc gia Mỹ).
- **Thông số:**
  - **Nồng độ $PM_{2.5}$ (pm25):** Đo bằng $\mu g/m^3$, ghi nhận hàng ngày.
  - **Nhiệt độ (TMP, TX, TN):** Trung bình, tối đa và tối thiểu, đơn vị °C.

- **Độ ẩm tương đối (RH):** Tỷ lệ phần trăm (%).
- **Áp suất (PRES2M):** Áp suất tại độ cao 2m, đơn vị hPa.
- **Tổng lượng mưa (TP):** Đơn vị mm.
- **Tốc độ gió (WSPD):** Đơn vị m/s.
- **Hướng gió (WDIR):** Góc từ 0 - 360°.

Dữ liệu địa hình

- **Nguồn gốc:** Dữ liệu địa hình được tổng hợp từ ảnh vệ tinh và bản đồ số.
- **Thông số:**
  - **Độ cao (DEM):** So với mực nước biển, đơn vị mét.
  - **Khoảng cách tới biển:** Đơn vị km.
  - **Đặc trưng tổng hợp (SQRT_SEA_DEM_LAT):** Kết hợp độ cao, vĩ độ và khoảng cách tới biển.

$$\text{SQRT\_SEA\_DEM\_LAT} = \sqrt{\frac{distance\ to\ sea}{DEM}} \cdot Latitude$$

2.1.2 Dữ liệu đầu ra

Mục tiêu: Dự đoán nồng độ $PM_{2.5}$ ($\mu g/m^3$) của ngày tiếp theo (ngày t) dựa trên dữ liệu từ các ngày trước đó.

2.2 Phương pháp đánh giá

2.2.1 Chia dữ liệu huấn luyện/đánh giá/kiểm tra

Trong bài toán dự báo $PM_{2.5}$ - một bài toán chuỗi thời gian - việc chia dữ liệu đóng vai trò quan trọng trong việc đảm bảo tính thực tế và khách quan của mô hình. Chúng tôi đã thử nghiệm hai phương pháp chia dữ liệu như sau:

- **Chia ngẫu nhiên:** Các mẫu dữ liệu được chọn ngẫu nhiên để tạo thành các tập train, validation và test. Tuy nhiên phương pháp này có thể dẫn đến rò rỉ thông tin từ tương lai vào quá khứ, làm sai lệch đánh giá hiệu suất thực tế của mô hình.

- **Chia theo trình tự thời gian:** Dữ liệu được chia dựa trên thứ tự thời gian để mô phỏng quá trình dự báo thực tế:
  - **Train:** Từ 01/01/2020 đến 31/05/2021 (18 tháng, khoảng 77% dữ liệu).
  - **Validation:** Từ 01/06/2021 đến 31/07/2021 (2 tháng, khoảng 9% dữ liệu).
  - **Test:** Từ 01/08/2021 đến 31/12/2021 (4 tháng, khoảng 14% dữ liệu).

Tập dữ liệu	**Số ngày**	**Số mẫu**
Train	517	8844
Validation	92	1349
Test	122	1315

Bảng 2.1: Thống kê tập dữ liệu chia theo trình tự thời gian

Phương pháp chia theo trình tự thời gian cho hiệu suất thấp hơn so với chia ngẫu nhiên. Điều này phản ánh sự thay đổi về phân bố dữ liệu theo thời gian (ví dụ: xu hướng mùa vụ hoặc thay đổi khí hậu). Do đó, chúng tôi chọn phương pháp chia theo trình tự thời gian để đảm bảo đánh giá mô hình phản ánh đúng khả năng dự báo trong thực tế.

2.2.2 Các độ đo

MSE (Mean Squared Error): Đo lường sai số bình phương trung bình.

$$\text{MSE} = \frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2$$

$$\text{MSE} \in [0, +\infty), \text{lower is better.}$$

MAE (Mean Absolute Error): Đo lường sai số tuyệt đối trung bình.

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

$$\text{MAE} \in [0, +\infty), \text{lower is better.}$$

R^2 (Coefficient of determination): Tỷ lệ biến thiên trong biến phụ thuộc có thể dự đoán được từ các biến độc lập.

$$R^2 = 1 - \frac{S_{\text{residual}}}{S_{\text{total}}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

$$R^2 \in [-\infty, 1], \text{higher is better.}$$

2.2.3 Phương pháp đánh giá

Đánh giá trên tập Test

- Dữ liệu được tổ chức thành các mảng:
  - **X**: Kích thước [ws, `num_features`], chứa giá trị đặc trưng từ ngày $t - \text{ws}$ đến ngày $t - 1$.
  - **y**: Kích thước `[1, 1]`, chứa giá trị $PM_{2.5}$ của ngày t.
- Loại bỏ các mẫu có giá trị bị thiếu trong **X** hoặc **y**.

Đánh giá trên tập Train và Validation

- **Cách 1:** Sử dụng dữ liệu gốc, áp dụng quy trình như trên với ws cố định.
- **Cách 2:** Sử dụng dữ liệu sau khi Imputation, đánh giá trên toàn bộ dữ liệu với ws cố định.

Chương 3

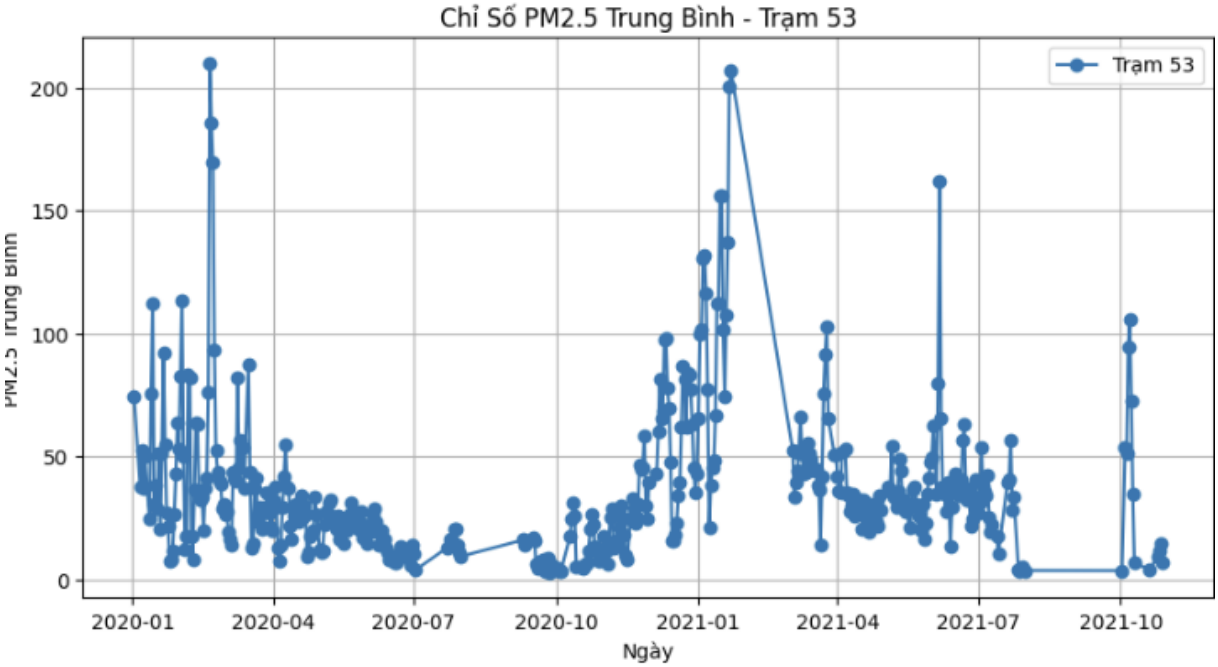
Phương pháp tiền xử lý dữ liệu

3.1 Phân tích dữ liệu trực quan

3.1.1 Các đặc trưng phụ thuộc vào thời gian

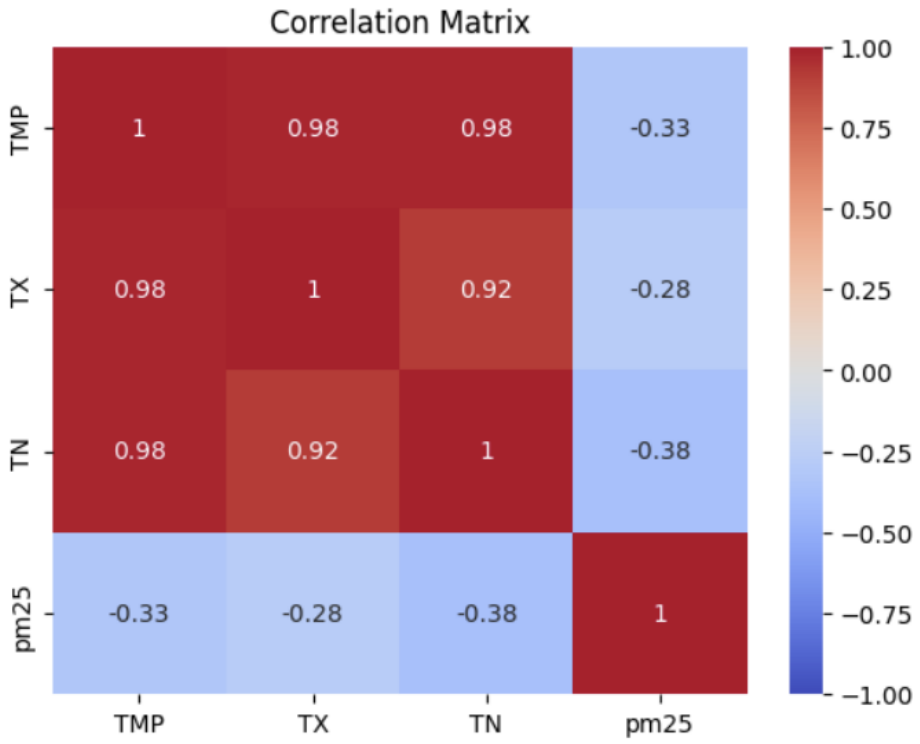
$PM_{2.5}$ (pm25): Nồng độ $PM_{2.5}$ xung quanh khu vực Hà Nội cho thấy sự biến động rõ rệt theo các mùa, đặc biệt tăng cao vào các tháng mùa đông (Hình 3.1):

- **Mùa đông (tháng 10 - tháng 1):** Nồng độ $PM_{2.5}$ thường vượt quá 150 µg/m³, đôi khi đạt mức trên 200 µg/m³. Các yếu tố chính bao gồm:
  - **Nghịch nhiệt:** Nhiệt độ thấp vào mùa đông làm tăng độ ổn định của không khí, tạo lớp nghịch nhiệt ngăn cản sự phân tán của $PM_{2.5}$, giữ các chất ô nhiễm gần mặt đất.
  - **Hoạt động đốt rơm rạ:** Sau vụ thu hoạch lúa, việc đốt rơm rạ thải ra lượng lớn $PM_{2.5}$.
  - **Gió mùa Đông Bắc:** Gió từ hướng Bắc mang theo ô nhiễm từ các khu công nghiệp ở miền Nam Trung Quốc và các nguồn địa phương (giao thông, công nghiệp), kết hợp với địa hình núi phía Bắc và Tây Bắc giữ lại chất ô nhiễm.
- **Mùa hè (tháng 4 - tháng 9):** Nồng độ $PM_{2.5}$ giảm đáng kể, dao động từ 20 - 50 µg/m³, do:
  - **Nhiệt độ cao:** Không khí đối lưu mạnh hơn, giúp phân tán $PM_{2.5}$.
  - **Gió mùa Đông Nam:** Gió từ biển Đông mang không khí sạch hơn, làm giảm ô nhiễm.
  - **Lượng mưa:** Mưa rửa trôi bụi mịn thông qua hiệu ứng lắng ướt.



Hình 3.1: Biểu đồ chỉ số $PM_{2.5}$ trung bình Trạm 53

Nhiệt độ (TMP): Nhiệt độ có mối quan hệ tỉ lệ nghịch với nồng độ $PM_{2.5}$ (Hình 3.2):



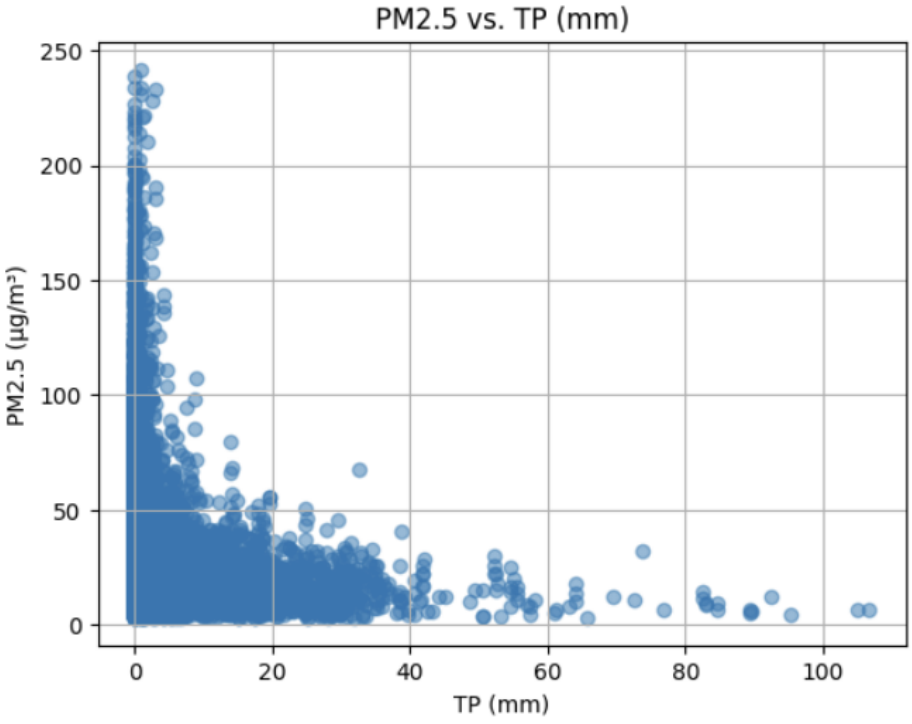
Hình 3.2: Biểu đồ tương quan giữa các yếu tố nhiệt và chỉ số $PM_{2.5}$

- **Mùa đông:** Nhiệt độ thấp (thường dưới 20°C) làm không khí ổn định, tương ứng với mức $PM_{2.5}$ cao do hiện tượng nghịch nhiệt.
- **Mùa hè:** Nhiệt độ cao (trên 30°C) thúc đẩy đối lưu, phân tán bụi mịn, làm giảm $PM_{2.5}$.

  Giải thích khoa học: Nhiệt độ thấp làm giảm độ cao của lớp ranh giới khí quyển (planetary, boundary layer), hạn chế sự khuếch tán dọc của $PM_{2.5}$, trong khi nhiệt độ cao tăng cường đối lưu, giúp phân tán chất ô nhiễm.

Lượng mưa (TP): Lượng mưa ảnh hưởng mạnh đến $PM_{2.5}$ thông qua hiệu ứng lắng ướt:

- **TP thấp (0 - 20 mm):** $PM_{2.5}$ dao động từ 50 - 250 µg/m³, với nhiều điểm tập trung ở mức 50 - 150 µg/m³, cho thấy ô nhiễm cao khi mưa ít hoặc không mưa.
- **TP trung bình (20 - 60 mm)**: $PM_{2.5}$ giảm dần, chủ yếu dưới 100 µg/m³, với cụm điểm quanh 0 - 50 µg/m³.
- **TP cao (60 - 100 mm):** $PM_{2.5}$ thường dưới 50 µg/m³, đôi khi gần 0 µg/m³.

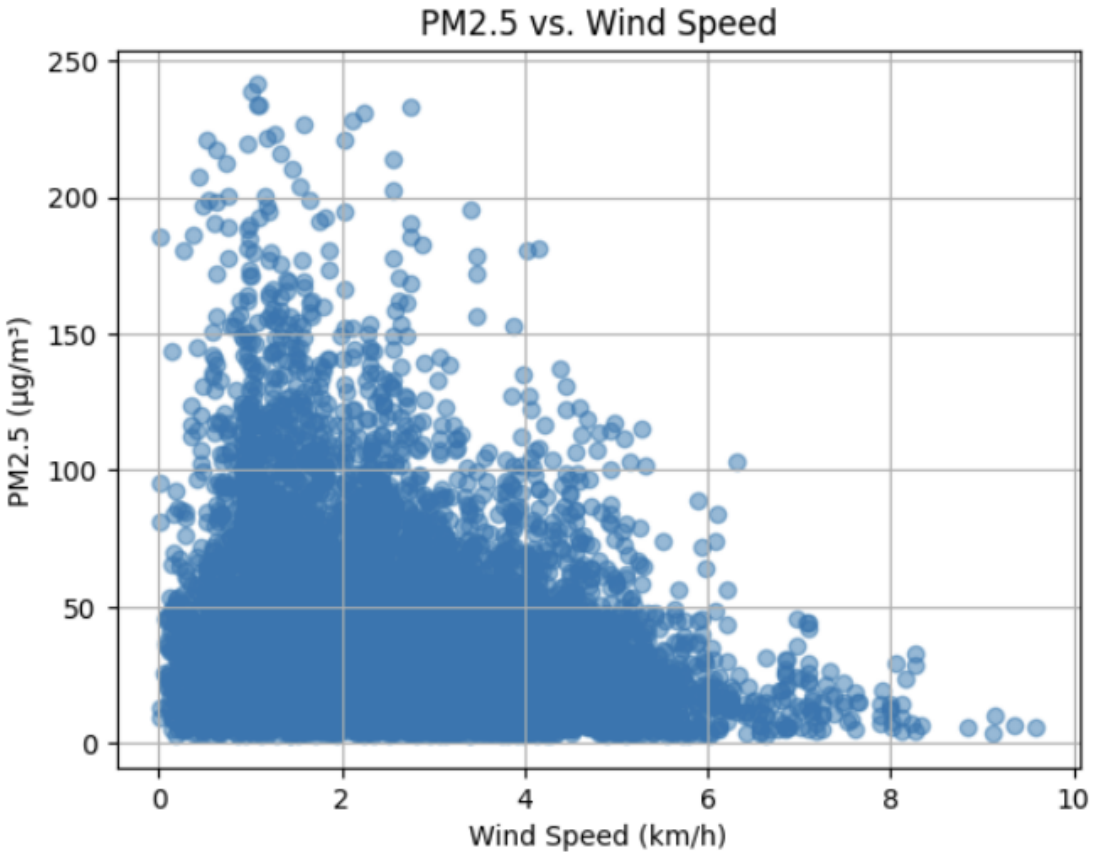


Hình 3.3: Biểu đồ tương quan nghịch giữa $PM_{2.5}$ và lượng mưa.

Xu hướng: Mối quan hệ tiêu cực giữa TP và $PM_{2.5}$. Một lượng mưa nhỏ (khoảng 20 mm) đã có thể giảm đáng kể $PM_{2.5}$, nhưng hiệu quả giảm dần khi lượng mưa tăng thêm do các hạt bụi mịn đã được rửa trôi gần hết (Hình 3.3).

Tốc độ gió (WSPD) và Hướng gió (WDIR):

- **Tốc độ gió (WSPD):**
  - **WSPD thấp (0 - 4 km/h):** $PM_{2.5}$ biến động mạnh (50 - 250 µg/m³), do các yếu tố khác như giao thông, khí thải công nghiệp, hoặc độ ẩm chiếm ưu thế.
  - **WSPD cao (6 - 10 km/h):** $PM_{2.5}$ ổn định ở mức thấp hơn, cho thấy gió là yếu tố chính giúp phân tán lượng bụi mịn tại các nguồn phát thải.

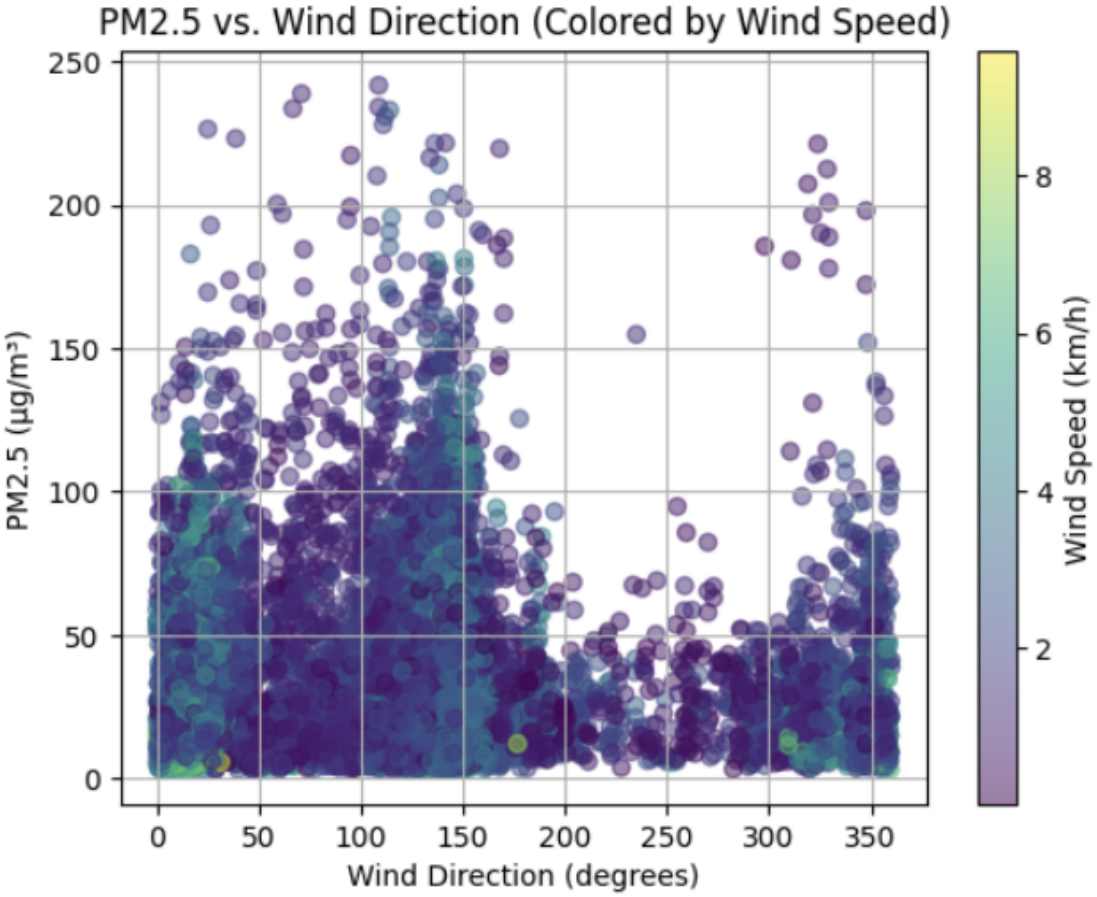


Hình 3.4: Biểu đồ tương quan nghịch giữa $PM_{2.5}$ và tốc độ gió

- **Hướng gió (WDIR):**

– **Gió mùa Đông Bắc (0° - 180°, tháng 10 - tháng 3):** Gió từ hướng Bắc và Đông Bắc mang theo ô nhiễm từ các khu công nghiệp ở Trung Quốc và nguồn địa phương (đồng bằng sông Hồng), kết hợp với tốc độ gió thấp và địa hình núi, làm $PM_{2.5}$ tích tụ (lên đến 250 µg/m³).

– **Gió mùa Đông Nam (180° - 360°, tháng 5 - tháng 9):** Gió từ biển Đông và khu vực nông thôn sạch hơn, với tốc độ cao hơn (4 - 8 km/h), giúp phân tán $PM_{2.5}$, giảm nồng độ xuống dưới 50µg/m³.



Hình 3.5: Biểu đồ $PM_{2.5}$, hướng gió và tốc độ gió

Giải thích khoa học: Gió mạnh tăng cường khuếch tán ngang, làm giảm nồng độ $PM_{2.5}$, trong khi gió yếu kết hợp với địa hình trũng (đồng bằng sông Hồng) làm chất ô nhiễm tích tụ. Hình 3.5 cho thấy $PM_{2.5}$ giảm rõ rệt khi gió thổi từ hướng 200° đến 300° (gió từ biển vào đất liền). Ngoài ra, ở Hình 3.4, có thể thấy ở mức gió > 6 km/h, hàm lượng bụi mịn tương đối thấp, nồng độ $PM_{2.5}$ luôn dưới 50µg/m³.

Độ ẩm tương đối (RH): Có mối tương quan thuận giữa RH và $PM_{2.5}$:

- RH thấp (30 - 50%): $PM_{2.5}$ thường dưới 50 µg/m³.
- RH cao (70 - 100%): $PM_{2.5}$ biến động lớn, có thể lên đến 250 µg/m³.

  Giải thích khoa học: Độ ẩm cao thúc đẩy sự hình thành các hạt aerosol thứ cấp (secondary aerosols) từ phản ứng hóa học giữa các tiền chất ô nhiễm trong không khí, làm tăng $PM_{2.5}$.

Dữ liệu viễn thám:
Ngoài các đặc trưng được đo tại các trạm, nhóm sử dụng dữ liệu viễn thám được thu thập từ vệ tinh Sentinel-5 Precursor do Cơ quan Vũ trụ Châu Âu phóng vào ngày 13 tháng 10 năm 2017 để theo dõi tình trạng ô nhiễm không khí, sử dụng cảm biến Tropomi. Các dữ liệu được sử dụng: CO, NO_2, O_3, AAI (chỉ số hấp thụ khí) và Cloud Coverage.

- CO: Cùng sinh ra từ các hoạt động đốt nhiên liệu → thường tăng/giảm đồng thời với $PM_{2.5}$, là chỉ thị cho các nguồn ô nhiễm giao thông hoặc công nghiệp
- NO_2: Là tiền chất phản ứng tạo ra các dạng hạt nitrate - thành phần chính trong $PM_{2.5}$ thứ cấp.
- O_3: Là một sản phẩm của phản ứng quang hóa giữa các hợp chất hữu cơ dễ bay hơi, thúc đẩy quá trình oxi hóa khí VOCs để tạo ra các chất hữu cơ thứ cấp (SOA), thành phần chính của $PM_{2.5}$.
- AAI: Chỉ số cao thường đi kèm với nồng độ $PM_{2.5}$ cao, nhất là khi có cháy rừng hoặc bụi mịn hấp thụ ánh sáng.
- Cloud Coverage: Ảnh hưởng đến ánh sáng và phản ứng quang hóa, khi mây thưa, ánh sáng mạnh có thể tăng $PM_{2.5}$ thứ cấp

3.1.2 Các đặc trưng không phụ thuộc vào thời gian

Các đặc trưng không phụ thuộc vào thời gian cung cấp thông tin về bối cảnh địa lý và xã hội của các trạm quan trắc tại khu vực Hà Nội, giúp mô hình phân biệt được sự khác biệt về mức độ ô nhiễm $PM_{2.5}$ giữa các vị trí mà không cần dựa vào

yếu tố thời gian. Bộ dữ liệu được thu thập từ các trạm quan trắc xung quanh khu vực Hà Nội, và các đặc trưng bao gồm:

- **Vị trí địa lý:** Tọa độ vĩ độ (latitude) và kinh độ (longitude) của mỗi trạm quan trắc. Hà Nội là một đô thị lớn với mật độ dân cư và hoạt động kinh tế tập trung, dẫn đến sự biến thiên rõ rệt về nồng độ $PM_{2.5}$ giữa các khu vực. Ví dụ: các trạm gần trung tâm thành phố, khu công nghiệp hoặc các tuyến đường lớn thường ghi nhận mức $PM_{2.5}$ cao hơn so với vùng ngoại ô do tác động từ giao thông và các nguồn phát thải.
- **Độ cao trạm:** Độ cao của trạm quan trắc so với mực nước biển (DEM - Digital Elevation Model). Mặc dù địa hình Hà Nội tương đối bằng phẳng, sự khác biệt nhỏ về độ cao vẫn có thể ảnh hưởng đến sự phân tán của $PM_{2.5}$. Các trạm ở vị trí thấp hơn có xu hướng tích tụ chất ô nhiễm nhiều hơn do không khí lạnh và bụi mịn lắng xuống khu vực trũng.

Các đặc trưng này giúp mô hình nhận diện sự khác biệt không gian trong nồng độ $PM_{2.5}$ mà không phụ thuộc vào các yếu tố thời gian như thời tiết hay chu kỳ ngày. Chẳng hạn, một trạm ở trung tâm Hà Nội thường có mức $PM_{2.5}$ cao hơn trạm ở ngoại ô, ngay cả khi điều kiện khí hậu giống nhau. Khi kết hợp với các đặc trưng phụ thuộc thời gian (như nhiệt độ, gió), chúng tạo ra một bức tranh toàn diện, giúp mô hình dự báo chính xác hơn.

3.2 Thiết kế đặc trưng

Nhóm đã thiết kế các đặc trưng mới từ dữ liệu ban đầu để cung cấp thông tin bổ sung và cải thiện khả năng dự báo $PM_{2.5}$. Các đặc trưng này được xây dựng dựa trên mối quan hệ vật lý giữa các yếu tố môi trường và $PM_{2.5}$, cũng như các mẫu chu kỳ trong dữ liệu chuỗi thời gian. Sau quá trình thiết kế, chúng tôi thu được tổng cộng 28 đặc trưng. Dưới đây là mô tả chi tiết về các đặc trưng mới thu được:

- **Chu kỳ ngày (sin_day, cos_day):** Mã hóa chu kỳ ngày trong tuần (thứ Hai đến Chủ Nhật) thành giá trị sin và cos. Đặc trưng này giúp mô hình nhận diện các mẫu lặp lại theo ngày, ví dụ như nồng độ $PM_{2.5}$ tăng cao vào các ngày làm việc do lưu lượng giao thông lớn hơn so với cuối tuần.

- **Vector gió (WDIR_x, WDIR_y, wind_u, wind_v):** Chuyển đổi hướng gió (WDIR) thành các thành phần vector theo trục x và y (hoặc u, v trong hệ tọa độ gió). Điều này giúp mô hình hiểu được tác động của hướng gió đến sự phân tán $PM_{2.5}$. Chẳng hạn, gió từ hướng Bắc có thể mang theo bụi mịn từ các khu công nghiệp phía Bắc Hà Nội, làm tăng nồng độ $PM_{2.5}$ tại các trạm phía Nam.

- **Chỉ số nhiệt (heat_index):** Kết hợp nhiệt độ (TMP) và độ ẩm tương đối (RH) để tính chỉ số nhiệt, phản ánh cảm giác nhiệt thực tế. Chỉ số này có thể liên quan đến sự tích tụ $PM_{2.5}$ trong điều kiện thời tiết nóng ẩm, thường gặp ở Hà Nội vào mùa hè.

- **Điểm sương (dew_point):** Được tính từ nhiệt độ (TMP) và độ ẩm tương đối (RH), điểm sương biểu thị mức độ bão hòa của không khí. Điểm sương cao thường gắn với sương mù, làm chậm quá trình phân tán $PM_{2.5}$, đặc biệt trong mùa đông ở Hà Nội.

- **day_of_week, month, season, day_of_year ,is_weekend:** Mã hóa ngày trong tuần, tháng, mùa, năm (1: xuân, 2: hạ, 3: thu, 4: đông), và biến nhị phân cuối tuần (0/1) để bắt các mẫu chu kỳ dài hạn và ngắn hạn.

- **distance_to_hanoi:** Khoảng cách từ trạm quan trắc đến trung tâm Hà Nội (tính bằng km), phản ánh mức độ đô thị hóa và ảnh hưởng của nguồn ô nhiễm trung tâm.

- **temp_wind:** Tích của nhiệt độ (TMP) và tốc độ gió (WSPD), biểu thị tác động kết hợp của hai yếu tố này đến sự phân tán $PM_{2.5}$.

- **rh_pressure:** Tích của độ ẩm (RH) và áp suất (PRES2M), liên quan đến sự hình thành aerosol thứ cấp.

- **wspd_squared:** Bình phương tốc độ gió (WSPD), giúp mô hình nhận diện hiệu ứng phi tuyến của gió mạnh trong việc giảm $PM_{2.5}$.

3.3 Lựa chọn đặc trưng

Để xác định tập hợp đặc trưng tối ưu từ 28 đặc trưng ban đầu, chúng tôi đã áp dụng nhiều phương pháp đánh giá tầm quan trọng của đặc trưng, bao gồm cả các

phương pháp tuyến tính và phi tuyến tính. Quy trình này đảm bảo rằng các đặc trưng được chọn có đóng góp thực sự vào hiệu suất dự báo.

Phương pháp sử dụng gồm có:

- **Correlation Analysis (R-squared-like):** Đo lượng tương quan tuyến tính giữa từng đặc trưng và $PM_{2.5}$.
- **F-ANOVA:** Đánh giá sự khác biệt thống kê giữa các nhóm giá trị của đặc trưng và $PM_{2.5}$
- **Model-Based Feature Importance (Random Forest, XGBoost):** Sử dụng tầm quan trọng của đặc trưng từ mô hình Random Forest và XGBoost.
- **Permutation Importance:** Đo lường mức giảm hiệu suất của mô hình khi hoán đổi ngẫu nhiên giá trị của một đặc trưng.
- **SHAP Values (Random Forest, XGBoost):** Phân tích đóng góp của từng đặc trưng vào dự đoán của mô hình bằng giá trị SHAP (SHapley Additive exPlanations).
- **Mutual Information (Information Gain for Regression):** Đo lường lượng thông tin chung giữa đặc trưng và $PM_{2.5}$, phù hợp với cả mối quan hệ phi tuyến.

Quy trình:

- Tính điểm tầm quan trọng của 28 đặc trưng thu được sau quá trình Feature Engineering bằng từng phương pháp trên.
- Chuẩn hóa điểm số từ các phương pháp và lấy trung bình để xếp hạng các đặc trưng.
- Thử nghiệm cùng một mô hình với số lượng đặc trưng giảm dần (từ 28 xuống 10, loại bỏ dần dần các đặc trưng có thứ hạng thấp) và đánh giá hiệu suất trên tập validation bằng các chỉ số MSE, MAE, và R^2 Score.

Kết quả:

Sau khi thử nghiệm, mô hình đạt hiệu suất tốt nhất khi sử dụng 15 đặc trưng được xếp hạng cao nhất. Danh sách các đặc trưng này bao gồm:

- **pm25**: Giá trị $PM_{2.5}$.
- **TN**: Nhiệt độ thấp nhất trong ngày.
- **dew_point**: Điểm sương.
- **heat_index**: Chỉ số nhiệt.
- **TMP**: Nhiệt độ trung bình.
- **sin_day**: Thành phần sin của ngày trong tuần.
- **PRES2M**: Áp suất khí quyển tại độ cao 2m.
- **distance_to_hanoi**: Khoảng cách đến trung tâm Hà Nội.
- **temp_wind**: Tích nhiệt độ và tốc độ gió.
- **cos_day**: Thành phần cos của ngày trong tuần.
- **TP**: Tổng lượng mưa.
- **TX**: Nhiệt độ cao nhất trong ngày.
- **wind_u**: Thành phần gió theo hướng u.
- **rh_pressure**: Tích độ ẩm và áp suất.
- **SQRT_SEA_DEM_LAT**: Đặc trưng tổng hợp từ độ cao, vĩ độ, và khoảng cách đến biển.

So sánh hiệu suất:

Kết quả ở bảng 3.1 cho thấy hiệu quả của hai phương pháp: Random Forest kết hợp Optuna và LSTM khi áp dụng trên ba tập đặc trưng khác nhau:

1. 10 đặc trưng gốc (Baseline)
2. 28 đặc trưng sau quá trình Feature Engineering
3. 15 đặc trưng sau Feature Selection

Đặc trưng	Phương pháp	Window size	MSE	MAE	R^2
10 đặc trưng gốc (Baseline)	Random Forest + Optuna	1	170.1297	7.4871	-0.0060
		2	148.9057	7.1343	0.1420
	LSTM	1	163.5203	8.0873	0.4801
		2	**127.2114**	7.1967	0.5806
		3	187.2501	**6.4893**	**0.6308**
28 đặc trưng sau Feature Engineering (Baseline)	Random Forest + Optuna	1	162.8759	7.3866	-0.0646
		2	150.9253	7.0764	0.0178
	LSTM	1	167.3468	7.9034	0.4679
		2	141.7286	7.4265	0.5328
		3	**109.3189**	**6.8146**	**0.5374**
15 đặc trưng sau feature selection	Random Forest + Optuna	1	166.1201	7.3956	-0.0546
		2	154.5476	7.0957	-0.0110
	LSTM	1	155.9376	7.7650	0.5042
		2	105.0816	6.5629	**0.6536**
		3	**85.8078**	**6.3807**	0.6369

Bảng 3.1: Kết quả trên tập validation của các mô hình hồi quy

Đáng chú ý, mô hình LSTM sử dụng 15 đặc trưng sau khi đã trải qua cả hai giai đoạn xử lý đặc trưng đạt hiệu suất tốt nhất với **MSE = 85.8078**, **MAE = 6.3807** và $R^2 = 0.6536$. So với mô hình trên tập 10 đặc trưng gốc (MSE = 127.2114, $R^2 = 0.6308$), hiệu suất đã được cải thiện rõ rệt. Điều này chứng minh rằng việc lựa chọn đặc trưng hợp lý không chỉ giúp giảm độ phức tạp của mô hình mà còn nâng cao khả năng tổng quát hoá, nhờ loại bỏ các đặc trưng dư thừa hoặc nhiễu.

Như vậy, tập đặc trưng gồm 15 thuộc tính là sự cân bằng tối ưu giữa việc giữ lại thông tin quan trọng và đơn giản hoá mô hình, góp phần cải thiện độ chính xác và độ tin cậy trong dự đoán.

Chương 4

Phương pháp xây dựng mô hình

4.1 Lựa chọn mô hình hồi quy

Để giải quyết bài toán, nhóm chủ yếu sử dụng các mô hình học sâu nhờ khả năng khai thác và học biểu diễn đặc trưng linh hoạt, đồng thời xây dựng các mô hình học máy dựa trên luật (Rule-based) làm Baseline. Trước khi cài đặt mô hình, nhóm khảo sát và đánh giá các kiến trúc, mô hình bao gồm: Long Short Term Memory (LSTM), Convolutional Neural Networks (CNN), Transformers và các mô hình học máy truyền thống.

4.1.1 Mô hình LSTM

- **Ưu điểm:**
  - Xử lý tốt thông tin cục bộ và phụ thuộc ngắn hạn, phù hợp với độ dài ngữ cảnh nhỏ (3–30 timestep).
  - Giảm thiểu hiện tượng tiêu biến/bùng nổ gradient so với RNN truyền thống bằng các cổng thông tin gồm cổng quên, cổng đầu vào, cổng đầu ra, giúp mô hình lựa chọn đặc trưng toàn cục hiệu quả.
  - Kích thước mô hình nhỏ gọn, tốc độ hội tụ nhanh.
- **Nhược điểm:**
  - Hạn chế trong việc nắm bắt các mối quan hệ dài hạn phức tạp. Tuy nhiên kết quả thực nghiệm cho thấy các mối quan hệ dài hạn gần như không cần thiết cho bài toán này.
- **Phù hợp với:** Dữ liệu có tính chất tuần hoàn ngắn và số lượng đặc trưng vừa phải.

4.1.2 Mô hình CNN

- **Ưu điểm:**
  - Khả năng trích xuất đặc trưng toàn cục thông qua các phép tích chập và pooling.
  - Không bị ảnh hưởng bởi hiện tượng tiêu biến hoặc bùng nổ gradient.
- **Nhược điểm:**
  - Mất mát thông tin cục bộ do phép tích chập làm mịn thông tin của từng timestep. Do vậy mô hình khó nắm bắt được các mối quan hệ phức tạp của những chuỗi dài mà chỉ nắm bắt được các đặc trưng toàn cục.
- **Phù hợp với:** Bài toán yêu cầu mô hình nhẹ và xử lý chuỗi có ngữ cảnh dài, cần nắm bắt đặc trưng toàn cục.

4.1.3 Mô hình Transformers

- **Ưu điểm:**
  - Nắm bắt đồng thời các mối quan hệ dài hạn và cục bộ nhờ cơ chế self-attention.
  - Xử lý linh hoạt dữ liệu khuyết thiếu thông qua cơ chế Masking.
  - Không gặp hiện tượng tiêu biến hoặc bùng nổ gradient.
  - Khả năng của mô hình scale (gần như) tuyến tính theo độ lớn dữ liệu và độ lớn mô hình.
- **Nhược điểm:**
  - Yêu cầu lượng dữ liệu huấn luyện lớn và mô hình lớn.
  - Độ phức tạp tính toán cao, dễ overfitting với dữ liệu nhỏ.
- **Phù hợp với:** Bài toán cần nắm bắt sự phụ thuộc phức tạp giữa các timestep, hoặc các bài toán có dữ liệu lớn.

Tiêu chí	LSTM	CNN	Transformers
Context Length	Ngắn (3 - 30)	Linh hoạt	Linh hoạt
Nắm bắt đặc trưng	Cục bộ	Toàn cục	Cục bộ và toàn cục
Các vấn đề gradient	Ít	Không	Không
Độ phức tạp	Thấp	Thấp	Cao
Dung lượng dữ liệu	Vừa phải	Vừa phải	Lớn

Bảng 4.1: Bảng so sánh các kiến trúc học sâu

4.1.4 Các mô hình học máy truyền thống

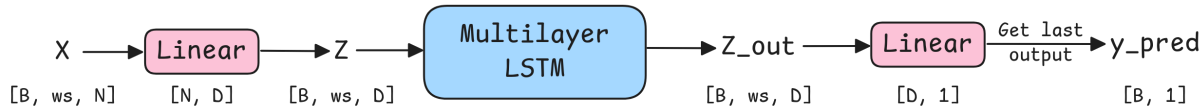
Nhóm cũng thử nghiệm với một số mô hình học máy làm Baseline như Random Forest và XGBoost. Các tham số của mô hình được tối ưu dựa trên hiệu suất trên tập validation sử dụng thư viện **Optuna**.

Kết luận: Bảng 4.1 so sánh các kiến trúc xử lý chuỗi phổ biến. Với dữ liệu đầu vào có context length ngắn (window_size ≤ 30), kích thước tập train hạn chế, và yêu cầu mô hình nhẹ, LSTM được lựa chọn làm kiến trúc chính. Transformers sẽ được ứng dụng riêng cho Impute dữ liệu nhờ khả năng xử lý missing data, sẽ được trình bày trong mục sau.

4.2 Cài đặt mô hình

4.2.1 Thống nhất ký hiệu

- B: Số lượng mẫu trong một batch (batch size).
- ws: Độ dài của cửa sổ thời gian (window size).
- N: Số đặc trưng (features) của mỗi ngày.
- M: Tổng số mẫu trong toàn bộ Dataset.
- D: Kích thước tiềm ẩn của mạng học sâu.



Hình 4.1: Kiến trúc mô hình học sâu.

4.2.2 Kế hoạch cài đặt

Dataset

- Dataset tiền xử lý và trả về một sample là một tensor $\mathbf{x} \in \mathbb{R}^{\text{ws} \times N}$ tương ứng với dữ liệu của ws ngày liên tiếp.
- Loại bỏ các sample có dữ liệu thiếu.
- Trả về các sample với các giá trị ws khác nhau như một cách augment dữ liệu.

Mô hình

Kiến trúc của mô hình được thiết kế đơn giản và trực quan (Hình 4.1).

- Kiến trúc chính: Mô hình LSTM nhiều lớp.
- Theo sau LSTM là một lớp Fully Connected để dự đoán giá trị đầu ra.
- Đầu vào: $\mathbf{X} \in \mathbb{R}^{B \times \text{ws} \times N}$
- Đầu ra: $\mathbf{y}_{\text{pred}} \in \mathbb{R}^{\text{B} \times 1}$

Thiết lập Tham số và Huấn luyện

Do điều kiện thời gian hạn chế nên nhóm không dành nhiều thời gian để tune tham số cho LSTM, mà sẽ dùng chung cho cả bài một cấu hình tham số cố định vừa đủ tốt.

- **Optimizer**: AdamW
  - Learning rate: $\text{lr} = 1 \times 10^{-3}$
  - Weight decay: 1×10^{-6}
- **Kiến trúc LSTM**:

  - Số chiều của lớp ẩn: 100
  - Số lớp LSTM: 2
- **Số epochs**: 20
- **Batch size**: 32

Do mục tiêu chính của báo cáo là so sánh các chiến lược tiền xử lý, thuật toán và phương thức huấn luyện, thay vì tối ưu hóa triệt để hiệu suất, chúng tôi không áp dụng các kỹ thuật như early stopping hay lr scheduler để đơn giản hóa quy trình huấn luyện và đảm bảo tính tái lập của các thực nghiệm.

Đánh giá

- Sử dụng chỉ số **MSE** để chọn mô hình tốt nhất trên tập validation.
- Kết quả cuối cùng của báo cáo là **MSE** trên tập test.

4.3 Kết quả đánh giá sơ bộ

Đặc trưng	Phương pháp	Window size	MSE	MAE	R^2
Đặc trưng gốc	Random Forest + Optuna	1	177.5926	8.9607	0.6902
		2	170.6864	**8.7215**	0.6708
	LSTM	1	198.9063	10.1925	0.7265
		2	**159.0900**	8.8113	**0.7835**
		3	174.8565	9.2526	0.7636
Sau feature selection	Random Forest + Optuna	1	168.9792	**8.7406**	0.6752
		2	169.9284	8.7503	0.6548
	LSTM	1	166.4096	9.3297	0.7712
		2	**153.1854**	8.7500	**0.7915**
		3	170.8373	9.0925	0.7690

Bảng 4.2: Kết quả trên tập test của các mô hình hồi quy.

Bảng 4.2 tổng hợp kết quả đánh giá hiệu suất dự báo của các mô hình hồi quy trên tập test, với hai tập đặc trưng: đặc trưng gốc và đặc trưng sau khi lựa chọn. Các mô hình được so sánh gồm LSTM và Random Forest kết hợp với tối ưu hóa siêu tham số bằng Optuna. Đồng thời, bảng 4.3 phân tích ảnh hưởng của độ dài cửa sổ thời gian (ws) tới hiệu suất của mô hình LSTM.

Đặc trưng	Phương pháp	Window size	MSE	MAE	R2
Sau Feature Selection	LSTM	1	166.4096	9.3297	0.7712
		2	**153.1854**	**8.7500**	**0.7915**
		3	170.8373	9.0925	0.7690
		4	174.3943	8.9587	0.7658
		5	166.9047	8.8865	0.7789
		7	162.8317	8.9201	0.7825
		14	175.8731	9.2094	0.7651
		28	185.1304	9.9640	0.7608

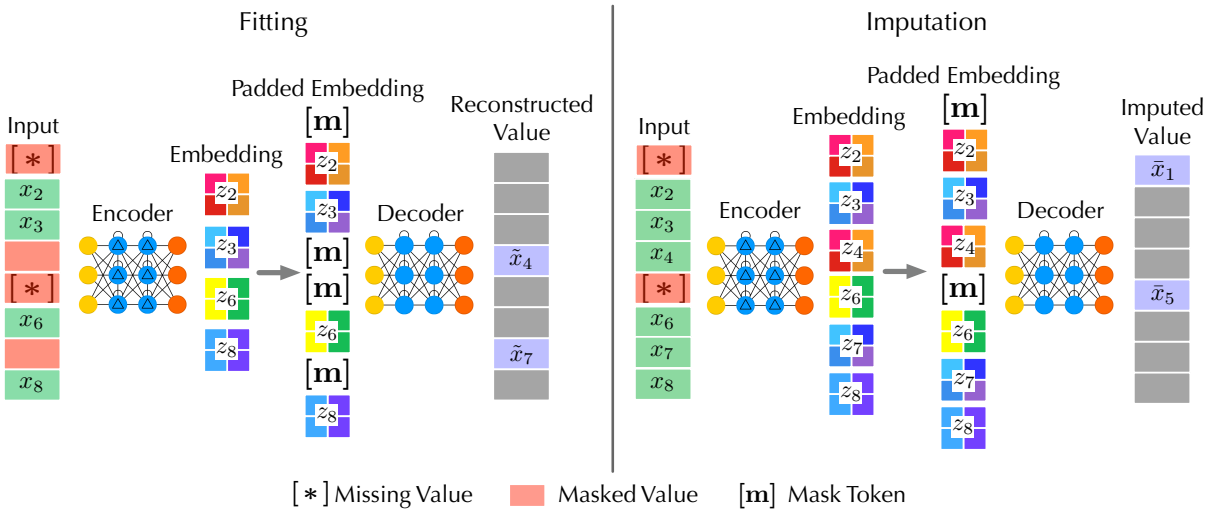
Bảng 4.3: Kết quả trên tập test với các giá trị ws khác nhau.

Dựa trên bảng 4.2, có thể thấy rằng mô hình LSTM đạt hiệu suất tốt hơn so với Random Forest trên hầu hết các thiết lập, đặc biệt khi kết hợp với tập đặc trưng đã được lựa chọn. Mô hình LSTM với window size = 2 đạt kết quả tốt nhất với **MSE = 153.18**, **MAE = 8.75**, và R^2 **= 0.7915**.

Ngoài ra, kết quả từ bảng 4.3 cho thấy hiệu suất mô hình nhạy cảm với độ dài cửa sổ thời gian. Giá trị ws tối ưu nằm trong khoảng từ 2 đến 7. Khi ws vượt quá 14, độ chính xác có xu hướng giảm đáng kể, cho thấy rằng việc tăng quá nhiều độ dài chuỗi đầu vào có thể gây nhiễu và làm giảm khả năng tổng quát hóa của mô hình.

4.4 Phương pháp điền dữ liệu khuyết sử dụng học sâu

Nhóm mong muốn tận dụng tối đa dữ liệu viễn thám, tuy nhiên các đặc trưng này lại bị thiếu với tần suất cao và phân bố rải rác trong bộ dữ liệu. Để khắc phục vấn đề này, nhóm đề xuất một phương pháp điền dữ liệu khuyết hiệu quả dựa trên học sâu. Kế thừa ý tưởng Masked Autoencoder áp dụng cho kiến trúc Transformers đã được ứng dụng nhiều trong thị giác máy (Masked Image Modeling[1][2]) và xử lý ngôn ngữ tự nhiên (Masked Language Modeling[3]), nhóm định hướng ứng dụng kỹ thuật tương tự cho dữ liệu của bài toán này. Trên thực tế đã có nghiên cứu Remasker[4] (ICLR 2024) cài đặt ý tưởng này cho dữ liệu dạng bảng với các đặc trưng liên tục, tuy nhiên chỉ tận dụng mối quan hệ sample wise (mối quan hệ các features trong từng sample) mà không tận dụng được mối quan hệ về thời gian và không gian (vị trí các trạm). Nhóm quyết định mở rộng ý tưởng này để tận dụng được những mối quan hệ phức tạp của dữ liệu.

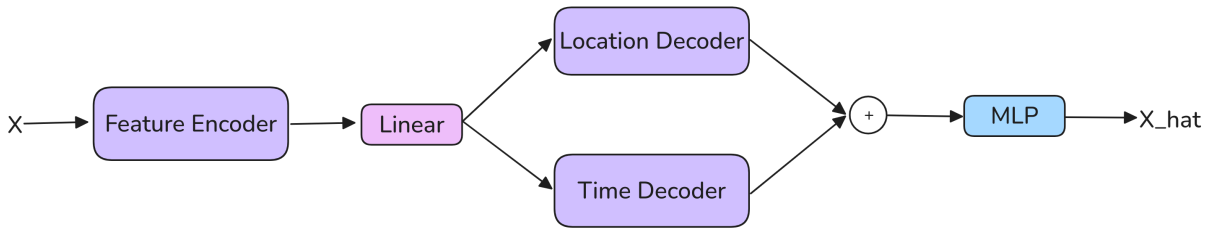


Hình 4.2: Framework tổng thể của giải pháp Remasker

4.4.1 Sơ lược về Remasker

Về ý tưởng chung, Remasker sử dụng kiến trúc Transformers để suy luận các mẫu có đặc trưng khuyết bằng cách mask các đặc trưng khuyết đi. Trong quá trình huấn luyện, bên cạnh những đặc trưng khuyết bị mask, Remasker sẽ mask ngẫu nhiên thêm một số các đặc trưng không bị khuyết, và huấn luyện mô hình khôi phục lại các đặc trưng bị mask đó. Quy trình tổng thể được thể hiện ở hình 4.2. Tóm lược thuật toán như sau:

- Định nghĩa mô hình: Enc, Dec là các mô hình Transformers nhiều lớp.
- Đầu vào:
  - X là batch đầu vào có chiều $[B, N]$, các đặc trưng thiếu nhận giá trị 0.
  - mask_org là mask có chiều $[B, N]$ biểu thị vị trí những đặc trưng bị thiếu.
- Suy luận khi huấn luyện:
  - Tạo m để mask ngẫu nhiên thêm một số đặc trưng.
  - $Z = \text{Enc}(\text{Embedding}(X), m)$ là đặc trưng được trích xuất bởi mô hình Encoder.
  - $\hat{X} = \text{Dec}(Z)$ là dự đoán suy luận bởi mô hình Decoder.
- Hàm mục tiêu: Là MSE của X với $\hat{X}$ tại những vị trí bị mask thêm bởi m.



Hình 4.3: Kiến trúc mô hình MAE đề xuất.

- Suy luận khi impute: $\hat{X} = \text{Dec}(\text{Enc}(\text{Embedding}(X), \text{mask\_org}))$.

Với hướng tiếp cận này mô hình sẽ tự học cách Impute dựa trên tất cả những ô dữ liệu khả dụng, trong khi bỏ qua những ô dữ liệu thiếu.

4.4.2 Ứng dụng Masked Autoencoder cho bài toán $PM_{2.5}$

Để thuận tiện cho mô hình, nhóm coi tất cả các đặc trưng đều khuyết theo loại missing at random (MAR). Nhóm mong muốn tận dụng triệt để 3 loại mối quan hệ để điền dữ liệu khuyết:

- Quan hệ không gian: Sử dụng dữ liệu trong ngày của trạm khác để fill trạm hiện tại.
- Quan hệ thời gian: Sử dụng dữ liệu quá khứ hoặc tương lai để fill một ngày bị thiếu.
- Quan hệ cục bộ: Đối với một ngày, tại một trạm, sử dụng dữ liệu các đặc trưng khác để fill một đặc trưng thiếu.

Nhóm đề xuất một kiến trúc MAE suy luận tận dụng các mối quan hệ này. Nhóm sẽ chỉ trình bày cách suy luận mô hình, về phần huấn luyện sẽ thực hiện quy trình giống như Remasker. Kiến trúc tổng quan được mô tả trong hình 4.3. Tóm lược quá trình suy luận như sau:

- Định nghĩa 3 khối Transformers sau:
  - $\text{Enc}_{\text{feature}}$ suy luận mối quan hệ cục bộ giữa các đặc trưng.
  - Dec_{loc} suy luận mối quan hệ không gian.
  - Dec_{time} suy luận mối quan hệ thời gian.

- Đầu vào:
  - X có chiều $[T, S, N]$, các giá trị thiếu được điền 0.
  - m là mặt nạ có chiều $[T, S, N]$ biểu thị vị trí những đặc trưng thiếu.
- Suy luận:
  - $Z_0 = \text{Embedding}(X)$ có chiều $[T, S, N, D_0]$, xếp lại thành $[T \cdot S, N, D_0]$.
  - $Z_1 = \text{Enc}_{\text{features}}(Z_0, m)$ có chiều $[T \cdot S, N, D_0]$, xếp lại thành $[T \cdot S, N \cdot D_0]$.
  - $Z_2 = \text{Linear}(Z_1)$ có chiều $[T \cdot S, D]$.
  - Xếp Z_2 thành $[S, T, D]$, suy luận $Z_3 = \text{Dec}_{\text{time}}(Z_2)$ có chiều $[S, T, D]$, xếp lại thành $[T, S, D]$.
  - Xếp Z_2 thành $[T, S, D]$, suy luận $Z_4 = \text{Dec}_{\text{loc}}(Z_2)$ có chiều $[T, S, D]$.
  - $\hat{X} = \text{Linear}(Z_3 + Z_4)$ có chiều $[T, S, N]$.

Dữ liệu huấn luyện sẽ là các mẫu X xếp chồng T ngày liên tục của S trạm được gom ngẫu nhiên. Mô hình MAE sẽ được huấn luyện và tinh chỉnh theo MSE trên tập validation để lấy mô hình cho kết quả tốt nhất. Sau đó MAE được đánh giá dựa trên hiệu suất trên test của mô hình LSTM được đề xuất ở mục 4.2, sử dụng dữ liệu được impute bởi MAE. Để tăng độ chính xác cho dữ liệu đầu ra trước khi đưa vào LSTM, các mẫu của tập train sẽ được impute sử dụng chủ yếu dữ liệu trong tương lai. Ngoài ra để tránh rò rỉ thông tin từ tương lai về quá khứ, mỗi mẫu của tập validation và tập test sẽ được impute chỉ sử dụng dữ liệu quá khứ trước khi đưa vào LSTM để đánh giá. Những mẫu có nhãn pm2.5 bị khuyết sẽ được loại bỏ khỏi đánh giá. Như vậy kể cả khi mô hình MAE sử dụng cả đặc trưng pm2.5, thông tin cũng không bị leak từ tương lai về quá khứ, đảm bảo khách quan cho đánh giá mô hình.

4.5 Kết quả đánh giá mô hình Impute

4.5.1 Xây dựng baseline sử dụng Iterative Impute

Để đánh giá mức độ hiệu quả của mô hình MAE, nhóm thực nghiệm so sánh với một Baseline sử dụng Iterative Impute kết hợp các mô hình học máy và tiến hành thử nghiệm nhằm lựa chọn mô hình phù hợp cho phương pháp này. Hiệu quả được

đánh giá dựa trên các chỉ số MSE, MAE và hệ số R^2 của mô hình Random Forest trên tập validation sử dụng dữ liệu đã được impute.

Bên cạnh đó nhóm cũng sử dụng chỉ số đo độ tương đồng phân bố như sau:

$$\text{mean\_log\_pvalue} = \frac{1}{N}\sum_{i=1}^{N}\log(p_i),$$

trong đó p_i là giá trị p thu được từ kiểm định Kolmogorov-Smirnov cho đặc trưng thứ i so sánh phân bố dữ liệu gốc và phân bố sau khi impute, chỉ số này càng cao càng cho thấy phân bố sau impute càng giống phân bố gốc. Kết quả thực nghiệm như bảng 4.4.

Mô hình	**MSE**	**MAE**	**R^2**	**Mean Log p-value**
XGBoost	161.5642	7.3385	-0.0899	-6.5944
RandomForest	163.7412	7.3633	-0.0597	-9.7698
LassoCV	164.0943	7.3534	-0.0598	-9.5506

Bảng 4.4: Bảng so sánh hiệu năng của các mô hình hồi quy trên validation cho phương pháp Iterative Impute.

4.5.2 Kết quả thực nghiệm của MAE

Kết quả thực nghiệm của các phương pháp impute như bảng 4.5. Kết quả cho thấy mô hình MAE cho dự đoán lỗi thấp hơn so với phương pháp Iterative Impute.

Phương pháp	**Mô hình**	**MSE**	**MAE**	**R^2**
Baseline không impute	LSTM ws=1	166.4096	9.3297	0.7712
	LSTM ws=2	153.1854	8.7500	**0.7915**
	LSTM ws=3	170.8373	9.0925	0.7690
IterativeImpute + XGBoost	LSTM ws=1	168.7103	8.9783	0.7680
	LSTM ws=2	179.9618	9.4129	0.7551
	LSTM ws=3	199.3296	9.5842	0.7305
IterativeImpute + LassoCV	LSTM ws=1	173.1613	9.4941	0.7619
	LSTM ws=2	154.8397	8.7341	0.7893
	LSTM ws=3	162.3418	8.6808	0.7805
MAE	LSTM ws=1	152.0666	8.7963	0.7368
	LSTM ws=2	**133.5258**	**8.3105**	0.7700
	LSTM ws=3	144.6031	8.5308	0.7493

Bảng 4.5: Kết quả trên tập test của các phương pháp điền dữ liệu khuyết.

4.6 Các phương pháp tối ưu MAE

Dựa trên các kết quả từ mục trước, chúng tôi nhận thấy hiệu suất tổng thể của mô hình LSTM phụ thuộc chủ yếu vào chất lượng dữ liệu sau khi được impute. Do đó, chúng tôi tập trung vào việc cải thiện độ chính xác của MAE. Hai hướng tiếp cận chính được đề xuất như sau:

- **Tinh chỉnh siêu tham số MAE bằng Optuna:** Sử dụng Optuna để tự động tìm kiếm tổ hợp các siêu tham số tối ưu cho MAE, từ tỉ lệ masking, mức dropout, đến số lớp, số chiều trong kiến trúc Encoder–Decoder. Việc này giúp MAE học được biểu diễn dữ liệu tiềm ẩn tốt hơn trước khi truyền sang LSTM.
- **Ứng dụng Bilevel Optimization cho bài toán Impute:** Huấn luyện đồng thời một mô hình MAE để impute và một mô hình LSTM để dự đoán. Mục

tiêu là huấn luyện MAE để, sau khi impute, LSTM đạt kết quả dự đoán tốt nhất trên tập validation.

Trong các mục sau, nhóm sẽ trình bày chi tiết quá trình tinh chỉnh siêu tham số MAE bằng Optuna, cũng như cơ chế Bilevel optimization cho bài toán Impute.

4.6.1 Tinh chỉnh siêu tham số MAE bằng Optuna

Nhóm đánh giá chất lượng của MAE dựa trên khả năng impute, thông qua một tập validation cố định. Tỉ lệ mask được giữ nguyên giữa các trial để đảm bảo tính nhất quán. Mỗi trial được huấn luyện trong 5000 bước lặp do giới hạn tài nguyên. Cách thức tìm kiếm tham số như sau:

- **Không gian tìm kiếm:**
  - `ft_embed_dim`: từ 4 đến 32 (bước 4)
  - `ft_enc_nhead`: {1, 2, 4}
  - `ft_enc_num_layers`: từ 1 đến 6
  - `mae_hidden_dim`: từ 32 đến 256 (bước 32)
  - `mlp_ratio`: từ 1.0 đến 16.0
  - `dim_feedforward`: tính bằng `mae_hidden_dim` × `mlp_ratio`
  - `mae_nhead`: {1, 2, 4, 8}
  - `mae_num_layers`: từ 1 đến 6
  - `mae_dropout`: từ 0.0 đến 0.5
  - `mask_ratio`: từ 0.1 đến 0.5 (tỷ lệ mask khi huấn luyện)
  - `lr`: log-scale từ 10^{-5} đến 10^{-3}
  - `weight_decay`: log-scale từ 10^{-8} đến 10^{-3}
- **Sampler và Pruner:**
  - `TPESampler`: tìm kiếm dựa trên phân phối Bayes
  - `MedianPruner`: cắt trial kém trước khi hoàn thành

4.6.2 Ứng dụng Bilevel Optimization cho bài toán điền dữ liệu khuyết

Dựa trên ý tưởng Bilevel Optimization ứng dụng cho các bài toán về dữ liệu [5][6], chúng tôi thiết lập một cơ chế tối ưu mới để huấn luyện MAE. Gọi mô hình MAE là teacher T với tham số θ_T và mô hình LSTM là student S với tham số θ_S. Giả sử cặp dữ liệu $(x_{\text{train}}, y_{\text{train}})$ và $(x_{\text{val}}, y_{\text{val}})$ lần lượt là một batch trong tập train và validation. Định nghĩa $T_x(x;\theta_T)$ là các đặc trưng được điền bởi teacher; $T_y(x;\theta_T)$ là nhãn (ở đây là $\text{PM}_{2.5}$ ngày tiếp theo) được điền bởi teacher; $S(x;\theta_S)$ là dự đoán của student trên dữ liệu x. Chúng tôi quy ước rằng nếu một đặc trưng x_j không bị thiếu thì $\big[T_x(x;\theta_T)\big]_j = x_j$; hoặc nếu nhãn y không bị thiếu thì $T_y(x;\theta_T) = y$.

Cách hình thành bài toán tối ưu

Khi có được dữ liệu được điền $T_x(x_{\text{train}};\theta_T)$ bởi teacher, dữ liệu này sẽ được dùng để huấn luyện student như sau:

$$\theta_S^* = \arg\min_{\theta_S} \Big[\underbrace{\text{MSE}\big(S\big(T_x(x_{\text{train}};\theta_T);\theta_S\big),\, T_y(x_{\text{train}};\theta_T)\big)}_{:=\mathcal{L}_{\text{train}}(\theta_T,\theta_S)} + \lambda\|\theta_S\|^2 \Big].$$

Với một mô hình teacher đủ tốt, có khả năng tạo ra dữ liệu impute chất lượng cao, ta kỳ vọng rằng các tham số tối ưu θ_S^* của student sẽ có khả năng tổng quát hóa tốt, thể hiện qua việc đạt được loss thấp trên tập validation, cụ thể là $\text{MSE}\big(S\big(T_x(x_{\text{val}};\theta_T);\theta_S^*\big),\, y_{\text{val}}\big) := \mathcal{L}_{\text{val}}(\theta_T, \theta_S^*)$.

Trong bài toán này, để ý rằng tham số tối ưu của student θ_S^* luôn phụ thuộc vào tham số của teacher θ_T thông qua dữ liệu được impute $T_x(x_{\text{train}};\theta_T)$. Do đó, có thể biểu diễn sự phụ thuộc này một cách tường minh dưới dạng hàm $\theta_S^*(\theta_T)$. Như vậy hàm loss trên validation $\mathcal{L}_{\text{val}}(\theta_T, \theta_S^*)$ cũng là một hàm của θ_T. Bài toán tối ưu tổng thể có thể được phát biểu dưới dạng bài toán hai tầng như sau:

$$\min_{\theta_T} \mathcal{L}_{\text{val}}\big(\theta_T, \theta_S^*(\theta_T)\big) \tag{4.1}$$

$$\text{where } \theta_S^*(\theta_T) = \arg\min_{\theta_S} \Big[\mathcal{L}_{\text{train}}(\theta_T, \theta_S) + \lambda\|\theta_S\|^2\Big] \tag{4.2}$$

Về mặt trực quan, teacher sẽ được điều chỉnh để tạo dữ liệu đầu ra giúp cải thiện hiệu suất của student. Tuy nhiên bài toán tối ưu này tiềm ẩn một vấn đề lớn: teacher có thể tạo artifact - đưa ra những dữ liệu đầu ra không thực lợi dụng tín hiệu từ student.

Phân tích vấn đề artifact

Để hiểu nguyên nhân xuất hiện của artifact, chúng ta phân tích hàm loss (4.1):

$$\mathcal{L}_{\text{val}}(\theta_T, \theta_S^*(\theta_T)) = \text{MSE}\big(S\big(T_x(x_{\text{val}}; \theta_T); \theta_S^*(\theta_T)\big),\, y_{\text{val}}\big),$$

Khi khai triển gradient của hàm loss theo tham số của teacher, ta thu được:

$$\frac{d\mathcal{L}_{\text{val}}}{d\theta_T} = \underbrace{\frac{\partial \mathcal{L}_{\text{val}}}{\partial \theta_T}\bigg|_{\theta_S=\theta_S^*(\theta_T)}}_{(A)} + \underbrace{\frac{\partial \mathcal{L}_{\text{val}}}{\partial \theta_S}\bigg|_{\theta_S=\theta_S^*(\theta_T)} \frac{\partial\, \theta_S^*(\theta_T)}{\partial \theta_T}}_{(B)}.$$

Trong đó, hạng tử (A) thể hiện gradient trực tiếp của teacher thông qua dữ liệu đầu ra được điền bởi teacher, cho phép huấn luyện teacher nhằm minimize hàm loss trên tập validation. Mã giả thực hiện cập nhật tương ứng với hạng tử (A) được mô tả dưới đây:

```
# 1. Impute the validation data with teacher and predict with
↪  student.
imputed_x_val = teacher(x_val)
with torch.no_grad():
  for p in student.parameters():
    p.requires_grad = False  # Freeze student's parameters

y_pred   = student(imputed_x_val)
loss_val = criterion(y_pred, y_val) # MSE

# 2. Backprop only the teacher parameters
teacher_optimizer.zero_grad()
loss_val.backward()
teacher_optimizer.step()
```

Có thể thấy, hạng tử (A) cho phép teacher khai thác trực tiếp các tín hiệu adversarial từ hàm loss của student trên validation. Điều này sẽ khiến teacher tạo ra các artifact làm lộ thông tin của label vào các đặc trưng được điền, làm $\mathcal{L}_{\text{val}}$ sụt giảm về 0. Nói cách khác, khi teacher biết được nhãn y_{val}, teacher sẽ làm lộ nhãn đó qua $T_x(x_{\text{val}})$, thông qua hạng tử (A). Hệ quả là teacher sẽ đưa ra dữ liệu không thực, lệch lạc nhằm mục đích giảm $\mathcal{L}_{\text{val}}$.

Giải quyết vấn đề artifact

Chúng tôi đề xuất không sử dụng dữ liệu impute của teacher trong hàm loss trên validation của student (hàm loss 4.1), tức là sẽ không impute x_{val} nữa. Hàm loss mới sẽ trở thành:

$$\mathcal{L}_{\text{outer}}\big(\theta_T, \theta_S^*(\theta_T)\big) = \text{MSE}\big(S(x_{\text{val}}; \theta_S^*(\theta_T)),\, y_{\text{val}}\big)$$

Như vậy bài toán tối ưu mới yêu cầu sử dụng student có khả năng suy luận đặc trưng thiếu thay vì phụ thuộc hoàn toàn vào các đặc trưng được điền của teacher. Để mô phỏng được phân bố dữ liệu khuyết trong dữ liệu, chúng tôi đề xuất sử dụng dropout đặc trưng cho dữ liệu đầu ra của teacher ở mục tiêu tối ưu (4.2). Như vậy mục tiêu tối ưu mới trở thành:

$$\theta_S^*(\theta_T) = \arg\min_{\theta_S}\Big[\mathcal{L}_{\text{inner}}(\theta_T, \theta_S) + \lambda\|\theta_S\|^2\Big]$$
$$\text{with } \mathcal{L}_{\text{inner}}(\theta_T, \theta_S) = \text{MSE}\big(S\big(D_p(T_x(x_{\text{train}}; \theta_T)); \theta_S\big),\, T_y(x_{\text{train}}; \theta_T)\big)$$

trong đó D_p là phép dropout áp dụng lên các đặc trưng với tỉ lệ drop là p. Khai triển gradient của hàm loss mới:

$$\frac{d\mathcal{L}_{\text{outer}}}{d\theta_T} = \underbrace{\frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_T}\bigg|_{\theta_S=\theta_S^*(\theta_T)}}_{=0} + \frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_S}\bigg|_{\theta_S=\theta_S^*(\theta_T)} \frac{\partial\, \theta_S^*(\theta_T)}{\partial \theta_T}\,.$$

Có thể thấy rằng thành phần tạo tín hiệu adversarial (thành phần gradient trực tiếp lên θ_T) đã bị triệt tiêu, teacher không có cách làm lộ trực tiếp phân bố nhãn qua các đặc trưng được điền, và buộc phải học cách tạo dữ liệu có tính tổng quát hơn. Hàm loss cuối cùng của teacher gồm hàm loss outer và hàm loss Masked Autoencoder:

$$\mathcal{L}_{\text{teach}}(\theta_T) = \mathcal{L}_{\text{outer}}(\theta_T, \theta_S^*(\theta_T)) + \mathcal{L}_{\text{mae}}(\theta_T)$$

Về mặt trực quan, dữ liệu được điền khuyết bởi teacher trong phương pháp mới có thể được xem như một dạng dữ liệu tăng cường, trong đó dropout đặc trưng được áp dụng để tạo ra các biến thể của dữ liệu khuyết, thay vì tái tạo đầy đủ toàn bộ đặc trưng ban đầu. Chi tiết cách tính gradient và thuật toán được mô tả ở phần phụ lục A, B.

4.6.3 Kết quả thực nghiệm

Phương pháp	**Mô hình**	**MSE**	**MAE**	**R^2**
Baseline không impute	LSTM ws=1	166.4096	9.3297	0.7712
	LSTM ws=2	153.1854	8.7500	**0.7915**
	LSTM ws=3	170.8373	9.0925	0.7690
IterativeImpute + XGBoost	LSTM ws=1	168.7103	8.9783	0.7680
	LSTM ws=2	179.9618	9.4129	0.7551
	LSTM ws=3	199.3296	9.5842	0.7305
IterativeImpute + LassoCV	LSTM ws=1	173.1613	9.4941	0.7619
	LSTM ws=2	154.8397	8.7341	0.7893
	LSTM ws=3	162.3418	8.6808	0.7805
MAE	LSTM ws=1	152.0666	8.7963	0.7368
	LSTM ws=2	133.5258	8.3105	0.7700
	LSTM ws=3	144.6031	8.5308	0.7493
MAE (Tuned)	LSTM ws=1	172.5271	9.4448	0.6990
	LSTM ws=2	132.9732	8.2810	0.7717
	LSTM ws=3	**128.5674**	8.0774	0.7772
MAE (Tuned) + Bilevel	LSTM ws=1	147.3512	8.6277	0.7429
	LSTM ws=2	129.2690	**8.0320**	0.7781
	LSTM ws=3	129.4254	8.0931	0.7757

Bảng 4.6: Kết quả trên tập test của các phương pháp điền dữ liệu khuyết.

Kết quả thực nghiệm của các phương pháp như bảng 4.6. Có thể rút ra một số phân tích như sau:

- **Baseline không impute vẫn đạt hiệu quả cạnh tranh:** Dù không sử dụng bất kỳ kỹ thuật điền dữ liệu nào, mô hình baseline đạt kết quả khá tốt. Đặc biệt với window size = 2, LSTM đạt $R^2 = 0.7915$, cao nhất toàn bảng. Điều này cho thấy dữ liệu gốc đã có cấu trúc đủ mạnh để hỗ trợ dự đoán khi thiếu hụt không quá nghiêm trọng.
- **Các phương pháp impute truyền thống không mang lại nhiều cải thiện:** Cả hai phương pháp IterativeImpute + XGBoost và IterativeImpute

+ LassoCV đều cho kết quả kém ổn định. Nhiều trường hợp MSE và MAE cao hơn baseline, cho thấy chúng không tận dụng tốt đặc trưng thời gian–không gian của dữ liệu môi trường.

- **MAE giúp cải thiện hiệu suất rõ rệt:** Việc áp dụng MAE cho kết quả tốt hơn rõ rệt so với các phương pháp truyền thống. Mô hình LSTM kết hợp MAE (chưa tinh chỉnh) ở window size = 2 đạt MSE = 133.53 và MAE = 8.31, cho thấy MAE có khả năng điền dữ liệu chất lượng giúp mô hình dự đoán.

- **Tinh chỉnh siêu tham số với Optuna mang lại cải thiện đáng kể:** Khi MAE được tinh chỉnh bằng Optuna, hiệu suất tiếp tục cải thiện. MSE giảm còn 128.57 với window size = 3, đây là kết quả tốt nhất về MSE toàn bảng.

- **Bilevel Optimization giúp mô hình ổn định và tổng quát tốt hơn:** Khi áp dụng bilevel optimization, kết quả tiếp tục được nâng cao. MAE giảm xuống chỉ còn 8.03 tại window size = 2 – thấp nhất trong tất cả các thiết lập. Ngoài ra, R^2 duy trì ở mức cao cho mọi window size, cho thấy khả năng tổng quát tốt hơn của dữ liệu được điền khuyết.

- **Tổng kết:** Phương pháp **MAE (Tuned) + Bilevel** đạt kết quả toàn diện nhất xét trên cả ba tiêu chí MSE, MAE và R^2. Điều này cho thấy MAE phù hợp cho bài toán điền dữ liệu khuyết trong dữ liệu môi trường, và Bilevel Optimization là công cụ hữu hiệu để nâng cao khả năng tổng quát hóa.

Chương 5

Trực quan hóa nồng độ $PM_{2.5}$

Chương này sẽ nói về cách sử dụng kết hợp MAE và LSTM để tận dụng triệt để mối quan hệ về không gian, thời gian để điền dữ liệu khuyết. Dữ liệu trực quan hóa là bộ test với 153 ngày từ 2021-08-01 đến 2021-12-31 cho 26 trạm.

5.1 Cơ chế điền dữ liệu của MAE

Trong quá trình cài đặt, nhóm đã lược bỏ hết positional encoding cho các token, mà thay vào đó đưa vào mô hình 4 đặc trưng thay thế là lat, lon, sin_day, cos_day. Khi không có positional encoding, MAE (cấu thành bởi các khối Transformers và các lớp trích xuất đặc trưng ở mức token-wise) sẽ đối xử các token như nhau, đặc trưng được trích xuất sẽ không phụ thuộc vào vị trí đặt token trong mẫu đầu vào, mà chỉ phụ thuộc vào các đặc trưng thay thế cho positional encoding như đề cập ở trên. Với cơ chế này, MAE có thể điền dữ liệu cho mọi tổ hợp không gian và thời gian của các mẫu. Như vậy bài toán trở thành tìm cách xếp dữ liệu khuyết của S trạm trong khoảng thời gian T ngày liên tục sao cho hiệu quả điền dữ liệu là tốt nhất.

5.2 Điền dữ liệu nồng độ $PM_{2.5}$ tại những nơi có trạm

Trước khi tạo được heatmap thì nhóm nhắm đến điền dữ liệu $PM_{2.5}$ cho tất cả các trạm trước. Nhóm kết hợp sử dụng các mô hình MAE và mô hình LSTM để điền dữ liệu $PM_{2.5}$ vào những ngày thiếu cho tất cả các trạm. Để điền dữ liệu cho ngày $t+1$ của trạm s, quy trình sẽ như sau:

- Tìm $S-1$ trạm có vị trí địa lý gần nhất với trạm s. Nếu $S-1$ trạm đó đều không có dữ liệu, tìm đến các trạm khác xa hơn.
- Sử dụng MAE để điền dữ liệu cho ngày t tại trạm s dùng dữ liệu quá khứ của S trạm.

- Sử dụng LSTM để dự đoán $PM_{2.5}$ của ngày $t+1$. Sau đó lại sử dụng MAE để điền các đặc trưng còn lại cho ngày $t+1$.

Với quy trình này, nhóm có được một bộ dữ liệu test đầy đủ cho tất cả các ngày của các trạm.

5.3 Điền dữ liệu nồng độ $PM_{2.5}$ tại những nơi không có trạm

Với cơ chế điền dữ liệu của MAE như trên thì chỉ cần cung cấp đầy đủ 4 đặc trưng lat, lon, sin_day, cos_day cho một điểm bất kỳ, sẽ có thể điền dữ liệu cho điểm đó, kể cả khi điểm đó không hề xuất hiện trong bộ dữ liệu huấn luyện. Nhóm đề xuất rằng để lỗi điền dữ liệu không quá lớn, nhóm chỉ sử dụng MAE để điền dữ liệu cho những điểm có khoảng cách đến trạm gần nhất bé hơn một threshold $\mathrm{D}_{\mathrm{impute}}$. Những điểm còn lại sẽ được nội suy bằng phương pháp đánh trọng số theo nghịch đảo khoảng cách (Inverse Distance Weighting). Với một điểm $\mathrm{P} = (x, y)$, giá trị $PM_{2.5}$ của điểm đó sẽ được tính như sau:

$$\mathrm{PM}_{2.5}(\mathrm{P}) = \frac{\sum_{s=1}^{S} \left(w_s(\mathrm{P}) \cdot \mathrm{PM}_{2.5}(\mathrm{P}_{\mathrm{station}_s}) \right)}{\sum_{s=1}^{S} w_s(\mathrm{P})},$$

Trong đó

$$w_s(\mathrm{P}) = \frac{1}{\mathrm{D}(\mathrm{P}, \mathrm{P}_{\mathrm{station}_s})^2},$$

Với $\mathrm{D}(\cdot, \cdot)$ là khoảng cách Euclidean giữa 2 tọa độ, $\mathrm{P}_{\mathrm{station}_s}$ là tọa độ của trạm s, và $\mathrm{PM}_{2.5}(\mathrm{P})$ là giá trị $PM_{2.5}$ tại điểm P.

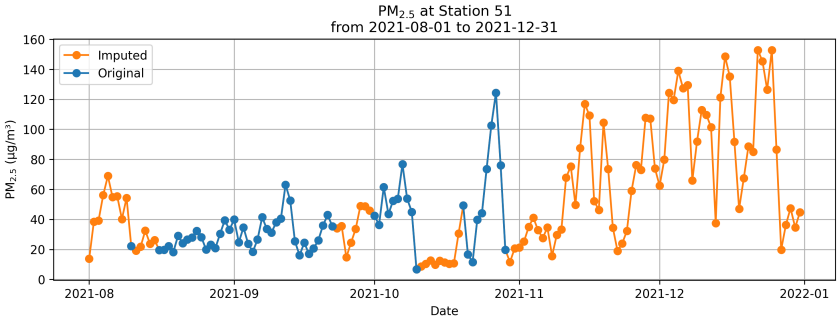
5.4 Kết quả điền dữ liệu của một số trạm

Sau khi áp dụng quy trình điền dữ liệu nêu trên, nhóm đã thu được kết quả đầy đủ cho từng ngày tại 26 trạm quan trắc từ 2021-08-01 đến 2021-12-31. Hình 5.1 minh họa diễn biến nồng độ $PM_{2.5}$ tại một số trạm tiêu biểu, trong đó có cả các trạm từng bị khuyết dữ liệu hoàn toàn hoặc phần lớn thời gian.

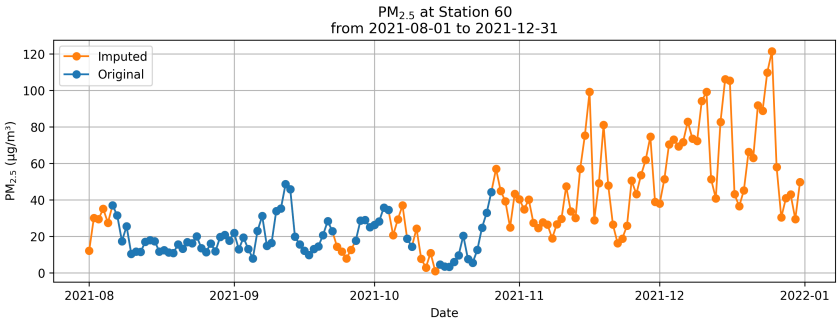
Ví dụ, trạm 156 là một trong những trạm có tỷ lệ thiếu dữ liệu rất cao, nhưng nhờ mối tương quan mạnh mẽ giữa trạm này và các trạm lân cận, mô hình MAE vẫn có thể tái hiện lại diễn biến tương đối hợp lý của $PM_{2.5}$ trong suốt giai đoạn. Có thể thấy các đỉnh và đáy trong chuỗi thời gian của trạm này được khớp khá tốt

với các trạm gần kề. Điều này cho thấy tính hiệu quả của phương pháp điền dựa trên không gian và thời gian.

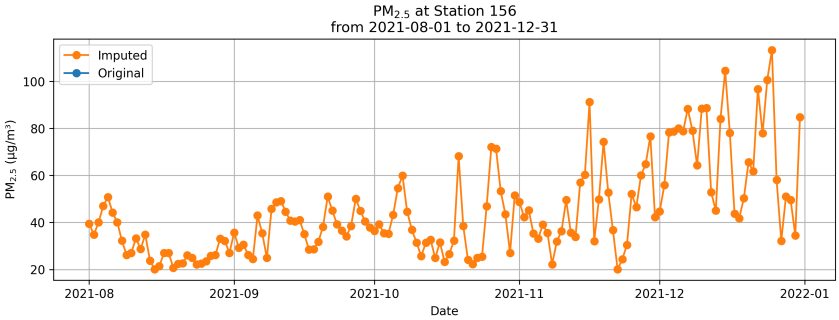
Trong khi đó, trạm 51 và trạm 60 có lượng dữ liệu quan trắc tương đối đầy đủ hơn, vì vậy các đoạn được điền đóng vai trò "nối mạch" vào các phần bị thiếu. Điều đáng chú ý là độ trơn mượt của chuỗi thời gian sau khi điền không gây ra hiện tượng đột biến bất thường, cho thấy mô hình đã học được phân bố và tính thời vụ của dữ liệu gốc.



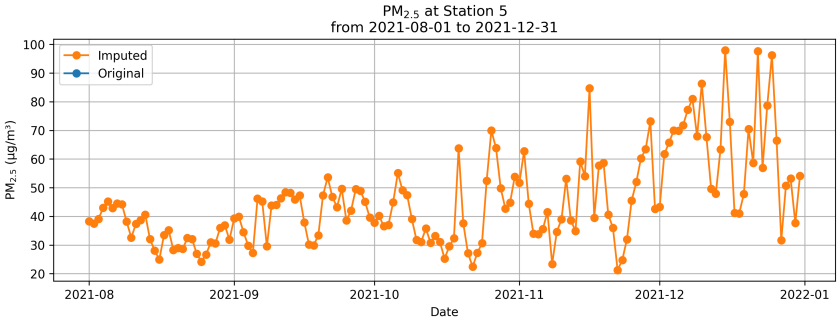
(a) Trạm 51



(b) Trạm 60



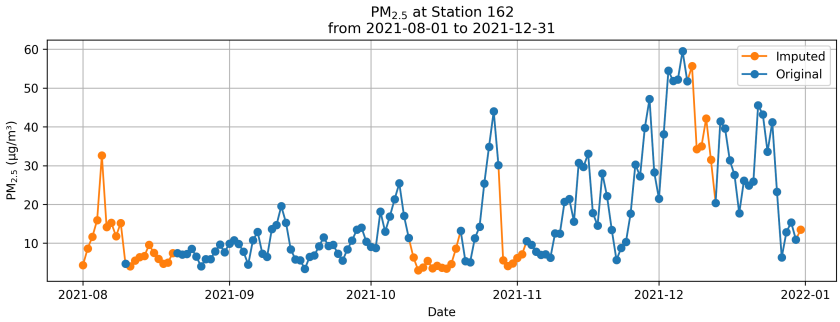
(c) Trạm 156



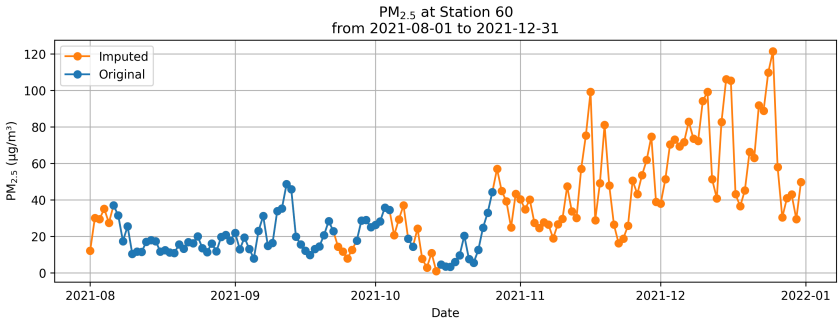
(d) Trạm 5

Hình 5.1: $PM_{2.5}$ tại các trạm: Trạm 51, Trạm 60, Trạm 156 và Trạm 5

So sánh kết quả điền dữ liệu của các trạm trong cùng một khu vực địa lý, có thể thấy mô hình cho ra kết quả tốt, không có sự đột biến về giá trị hoặc xu hướng tăng, giảm nồng độ $PM_{2.5}$ giữa các trạm có khoảng cách gần. Hình 5.2 trạm 162 và 163 ở Bắc Ninh với khoảng cách 7,7km.



(a) Trạm 162



(b) Trạm 163

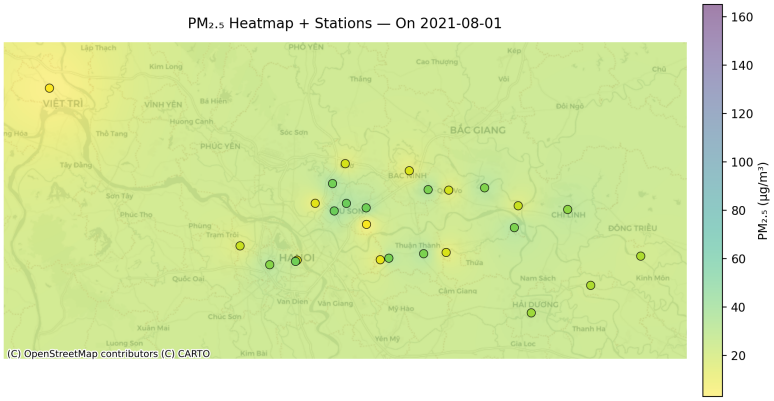
Hình 5.2: $PM_{2.5}$ tại các trạm: Trạm 162 và Trạm 163

Việc áp dụng kết hợp LSTM và MAE cũng thể hiện vai trò rõ rệt trong những ngày mà toàn bộ các trạm đều thiếu dữ liệu (ví dụ, những ngày bị mất điện, trạm tạm dừng hoạt động). Trong trường hợp này, LSTM đóng vai trò "dẫn dắt" xu hướng từ dữ liệu đã có, còn MAE đảm nhận phần điền chi tiết dựa trên cấu trúc không gian.

Tổng thể, kết quả này giúp hoàn thiện tập dữ liệu đầu vào cho bước trực quan hóa tiếp theo bằng heatmap, đồng thời tăng độ tin cậy của phân tích chất lượng không khí liên vùng.

5.5 Kết quả vẽ heatmap dự báo

Sau khi thu được tập dữ liệu đầy đủ đã qua điền khuyết, nhóm tiến hành trực quan hóa nồng độ $PM_{2.5}$ bằng heatmap để kiểm tra tính hợp lý của phân bố không gian theo thời gian. Hình 5.3 minh họa kết quả dự báo cho ngày: ngày 2021-08-01.



Hình 5.3: Bản đồ $PM_{2.5}$ tại khu vực miền Bắc ngày 2021-08-01.

Hình 5.3 cho thấy sự chuyển tiếp hợp lý giữa các vùng nằm giữa các trạm có nồng độ $PM_{2.5}$ cao và các trạm có nồng độ $PM_{2.5}$ thấp. Các vùng có nồng độ $PM_{2.5}$ cao là nơi có nguồn phát thải ô nhiễm cục bộ, có diện tích nhỏ.

Kết luận

Trong quá trình nghiên cứu, nhóm đã tiến hành phân tích dữ liệu kỹ lưỡng, khám phá mối quan hệ giữa nồng độ $PM_{2.5}$ và các yếu tố khí tượng, địa hình, cũng như các yếu tố xã hội. Việc thiết kế và lựa chọn đặc trưng đóng vai trò quan trọng, dẫn đến việc xác định được các đặc trưng tối ưu giúp cải thiện hiệu suất của mô hình dự báo.

Nhóm đã thử nghiệm và so sánh nhiều kiến trúc mô hình học sâu, bao gồm LSTM, CNN và Transformers, cùng với các mô hình học máy truyền thống như Random Forest và XGBoost làm Baseline. Kết quả cho thấy mô hình LSTM, đặc biệt khi được huấn luyện trên các đặc trưng đã qua quá trình lựa chọn, đạt được hiệu suất dự báo tốt. So với mô hình chỉ sử dụng 10 đặc trưng gốc, sự cải thiện này chứng minh tầm quan trọng của việc tiền xử lý và lựa chọn đặc trưng.

Một đóng góp đáng chú ý khác là việc áp dụng và mở rộng kiến trúc Masked Autoencoder (MAE) dựa trên ý tưởng từ Remasker để xử lý vấn đề dữ liệu khuyết thiếu - một thách thức phổ biến trong các bài toán môi trường. Nhóm đã thiết kế một phiên bản MAE có khả năng khai thác đồng thời mối quan hệ không gian, thời gian và cục bộ giữa các đặc trưng và trạm đo, qua đó nâng cao hiệu quả tái tạo dữ liệu. Đặc biệt, quá trình huấn luyện MAE được tối ưu hóa bằng kỹ thuật Bilevel Optimization, giúp điều chỉnh quá trình mask và học biểu diễn một cách linh hoạt và thích ứng với mục tiêu điền khuyết. Kết quả thực nghiệm cho thấy MAE vượt trội hơn so với phương pháp Baseline như Iterative Impute, cả về độ chính xác điền dữ liệu và tác động tích cực đến các mô hình dự báo $PM_{2.5}$ sau đó.

Cuối cùng, nhóm đã tiến hành trực quan hóa nồng độ $PM_{2.5}$ trên khu vực miền Bắc bằng heatmap, dựa trên dữ liệu đã được điền khuyết bằng MAE và dự báo bằng LSTM. Các heatmap theo thời gian cho thấy sự phân bố hợp lý của nồng độ $PM_{2.5}$ và sự tương quan với các yếu tố khí tượng như tốc độ gió.

Các kết quả thực nghiệm đã chứng minh tính hiệu quả của các phương pháp được đề xuất, đặc biệt là vai trò của việc lựa chọn đặc trưng và khả năng của mô hình MAE trong việc xử lý dữ liệu khuyết thiếu, mở ra hướng tiếp cận tiềm năng cho việc dự báo và quản lý chất lượng không khí tại khu vực miền Bắc Việt Nam.

Tài liệu tham khảo

[1] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[2] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale, 2022.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[4] Tianyu Du, Luca Melis, and Ting Wang. Remasker: Imputing tabular data with masked autoencoding. In *The Twelfth International Conference on Learning Representations*, 2024.

[5] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11557–11568, June 2021.

[6] Marzi Heidari and Yuhong Guo. Bi-level optimization for semi-supervised learning with pseudo-labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16):17168–17176, Apr. 2025.

[7] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients, 2019.

[8] J. L. Nazareth. Conjugate gradient method. *WIREs Computational Statistics*, 1(3):348–353, 2009.

[9] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. 1994.

Phụ lục

A Khai triển gradient cập nhật tham số của Teacher

Ở mục này, chúng tôi trình bày chi tiết khai triển gradient hàm loss outer để cập nhật tham số của mô hình Teacher, như đã đề cập trong mục 4.6.2.

Nhắc lại bài toán tối ưu hai tầng:

$$\begin{aligned}
\mathcal{L}_{\text{outer}}(\theta_T, \theta_S^*) &= \text{MSE}\left(S(x_{\text{val}}; \theta_S^*(\theta_T)), y_{\text{val}}\right), \\
\theta_S^*(\theta_T) &= \arg\min_{\theta_S} \left[\mathcal{L}_{\text{inner}}(\theta_T, \theta_S) + \frac{\lambda}{2}\|\theta_S\|^2\right], \\
\mathcal{L}_{\text{inner}}(\theta_T, \theta_S) &= \text{MSE}\left(S(D_p(T_x(x_{\text{train}}; \theta_T)); \theta_S),\ T_y(x_{\text{train}}; \theta_T)\right).
\end{aligned}$$

Khai triển gradient của $\mathcal{L}_{\text{outer}}$:

$$\begin{aligned}
\frac{d\mathcal{L}_{\text{outer}}}{d\theta_T} &= \underbrace{\frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_T}}_{=0} + \frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_S} \cdot \frac{\partial \theta_S^*}{\partial \theta_T} \\
&= \frac{\partial \mathcal{L}_{\text{outer}}}{\partial \theta_S} \cdot \frac{\partial \theta_S^*}{\partial \theta_T}. \qquad (1)
\end{aligned}$$

Hạng tử đầu tiên có thể được tính bằng các thư viện autodiff. Chúng ta tập trung vào hạng tử thứ hai. Do $\theta_S^* = \arg\min_{\theta_S}\left[\mathcal{L}_{inner}(\theta_T, \theta_S) + \frac{\lambda}{2}\|\theta_S\|^2\right]$, nên theo điều kiện điểm cực tiểu, gradient tại $\theta_S = \theta_S^*$ bằng không, ta có:

$$\begin{aligned}
& \frac{\partial}{\partial \theta_S}\left[\mathcal{L}_{\text{inner}}(\theta_S) + \frac{\lambda}{2}\|\theta_S\|^2\right]\Bigg|_{\theta_S = \theta_S^*} = 0 \\
\Rightarrow\quad & \frac{\partial}{\partial \theta_S}\mathcal{L}_{\text{inner}}(\theta_S^*) + \lambda\theta_S^* = 0 \\
\Rightarrow\quad & \theta_S^* = -\frac{1}{\lambda}\frac{\partial}{\partial \theta_S}\mathcal{L}_{\text{inner}}(\theta_S^*),
\end{aligned}$$

Đạo hàm theo θ_T, ta có:

$$\begin{aligned}
&\frac{d\theta_S^*}{d\theta_T} = -\frac{1}{\lambda}\left(\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial \theta_T \partial \theta_S} + \frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial \theta_S^2} \cdot \frac{d\theta_S^*}{d\theta_T}\right) \\
\Rightarrow \quad &\left(I + \frac{1}{\lambda}\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial \theta_S^2}\right) \cdot \frac{d\theta_S^*}{d\theta_T} = -\frac{1}{\lambda}\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial \theta_T \partial \theta_S} \\
\Rightarrow \quad &\frac{d\theta_S^*}{d\theta_T} = -\left(\frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial \theta_S^2} + \lambda I\right)^{-1} \cdot \frac{\partial^2 \mathcal{L}_{\text{inner}}}{\partial \theta_T \partial \theta_S} \\
\Rightarrow \quad &\frac{d\theta_S^*}{d\theta_T} = -\left(\nabla^2_{\theta_S} \mathcal{L}_{\text{inner}} + \lambda I\right)^{-1} \cdot \left(\nabla_{\theta_T} \nabla_{\theta_S} \mathcal{L}_{\text{inner}}\right)^T . \qquad (2)
\end{aligned}$$

Thay (2) vào (1) để suy ra công thức cập nhật cuối cùng:

$$\frac{d\mathcal{L}_{\text{outer}}}{d\theta_T} = -\left(\nabla_{\theta_S} \mathcal{L}_{\text{outer}}\right)^T \cdot \left(\nabla^2_{\theta_S} \mathcal{L}_{\text{inner}} + \lambda I\right)^{-1} \cdot \left(\nabla_{\theta_T} \nabla_{\theta_S} \mathcal{L}_{\text{inner}}\right)^T .$$

Công thức cuối cùng cho thấy rằng để cập nhật teacher, ta cần tính một hạng tử nghịch đảo Hessian, vốn rất tốn kém trong thực nghiệm. Phương pháp xấp xỉ sẽ được thảo luận kỹ hơn ở phần tiếp theo.

B Chi tiết thuật toán và mã giả huấn luyện Bilevel

Trong phần này chúng tôi trình bày chi tiết về quy trình huấn luyện Bilevel áp dụng cho MAE. Thuật toán được xây dựng dựa trên phương pháp implicit MAML[7]. Trước tiên, MAE sẽ được tiền huấn luyện bằng $\mathcal{L}_{mae}$ trên bộ dữ liệu training. Để tham số của student thỏa mãn điều kiện điểm cực tiểu

$$\frac{\partial}{\partial \theta_S}\left[\mathcal{L}_{\text{inner}}(\theta_S) + \frac{\lambda}{2}\|\theta_S\|^2\right]\Bigg|_{\theta_S = \theta_S^*} = 0,$$

chúng tôi cũng tiến hành pretrain student với $\mathcal{L}_{\text{inner}}$ trên dữ liệu đầu ra của teacher một số bước cho trước. Tại mỗi bước huấn luyện bilevel, tham số tối ưu θ_S^* được ước lượng bằng cập nhật gradient 1 bước $\theta_S^* \approx \theta_S^{(t+1)} = \theta_S^{(t)} - \eta_S \cdot \nabla_{\theta_S} \mathcal{L}_{\text{inner}}$. Việc tính hạng tử nghịch đảo Hessian $\left(\nabla^2_{\theta_S} \mathcal{L}_{\text{inner}} + \lambda I\right)^{-1}$ là tốn kém và không ổn định về mặt tính toán, vì thế hạng tử này sẽ được xấp xỉ bằng phương pháp Conjugate Gradient[8][9]. Mã giả huấn luyện Bilevel được mô tả ở thuật toán 1.

Algorithm 1 Phương pháp huấn luyện Bilevel cho mô hình impute

Đầu vào: Tập training với giá trị thiếu $(x_{\text{train}}, m_{\text{train}})$, tập validation với giá trị thiếu $(x_{\text{val}}, m_{\text{val}})$, tỷ lệ mask của MAE p, hệ số regularization λ, learning rate η_S, η_T, số vòng lặp CG K

Đầu ra : $\theta_S^{(N)}$ ▷ *Tham số teacher cho suy luận imputation*

Khởi tạo $\theta_T^{(0)}$ và $\theta_S^{(0)}$

for $t = 0$ **to** $N - 1$ **do**

Điền giá trị thiếu cho tập train

$$\tilde{x}_{\text{train}} = T_x(x_{\text{train}}, m_{\text{train}}; \theta_T^{(t)})$$
$$\tilde{y}_{\text{train}} = T_y(x_{\text{train}}, m_{\text{train}}; \theta_T^{(t)})$$

Cập nhật student trên dữ liệu đã impute

$$\theta_S^{(t+1)} = \theta_S^{(t)} - \eta_S \cdot \nabla_{\theta_S} \text{MSE}(S(\tilde{x}_{\text{train}}; \theta_S^{(t)}),\ \tilde{y}_{\text{train}})$$

Tính các hàm loss trung gian

$$\mathcal{L}_{\text{inner}} = \mathcal{L}_{\text{train}} = \text{MSE}(S(\tilde{x}_{\text{train}}; \theta_S^{(t+1)}),\ \tilde{y}_{\text{train}})$$
$$\mathcal{L}_{\text{outer}} = \mathcal{L}_{\text{val}} = \text{MSE}(S(x_{\text{val}}; \theta_S^{(t+1)}),\ y_{\text{val}})$$

Tính toán các gradient trung gian

$$v = \nabla_{\theta_S} \mathcal{L}_{\text{outer}};\ H = \nabla^2_{\theta_S} \mathcal{L}_{\text{inner}}$$

Sử dụng thuật toán CG với K bước lặp để xấp xỉ

$$u_{\text{IG}} \approx (H + \lambda I)^{-1} v$$

Tính toán gradient bilevel của teacher

$$h_{\text{impl}}^{(t)} = -\nabla_{\theta_T} \left[(\nabla_{\theta_S} \mathcal{L}_{\text{inner}})^\top u_{\text{IG}} \right]$$

Tính toán gradient MAE trực tiếp

$$g_{\text{mae}}^{(t)} = \nabla_{\theta_T} \mathcal{L}_{\text{MAE}}$$

Cập nhật teacher

$$\theta_T^{(t+1)} = \theta_T^{(t)} - \eta_T \cdot (h_{\text{impl}}^{(t)} + g_{\text{mae}}^{(t)})$$

end

return $\theta_T^{(N)}$ ▷ *Trả về mô hình teacher cho suy luận imputation*