# Dự báo chất lượng không khí tại Việt Nam bằng phương pháp chuỗi thời gian

1st Huỳnh Lê Phong
*Khoa Hệ thống thông tin*
*Trường Đại học Công nghệ Thông tin*
21520086@gm.uit.edu.vn

2nd Nguyễn Quốc Trạng
*Khoa Hệ thống thông tin*
*Trường Đại học Công nghệ Thông tin*
21521556@gm.uit.edu.vn

3rd Nguyễn Triệu Vy
*Khoa Hệ thống thông tin*
*Trường Đại học Công nghệ Thông tin*
21522812@gm.uit.edu.vn

4th Vũ Thị Phương Linh
*Khoa Hệ thống thông tin*
*Trường Đại học Công nghệ Thông tin*
20521541@gm.uit.edu.vn

5th Đỗ Đình Đăng Khoa
*Khoa Hệ thống thông tin*
*Trường Đại học Công nghệ Thông tin*
21522218@gm.uit.edu.vn

***Tóm tắt nội dung*—Nghiên cứu này tập trung vào phân tích chất lượng không khí ở ba thành phố lớn tại Việt Nam: Hà Nội, Hạ Long và Việt Trì. Mục tiêu của nghiên cứu là sử dụng các kỹ thuật phân tích chuỗi thời gian tiên tiến để dự báo và hiểu chỉ số PM2.5. Phương pháp phân tích bao gồm cả các thuật toán truyền thống và tiên tiến như Hồi quy Tuyến tính, Mô hình ARIMA, Mạng Nơ-ron Hồi quy (RNN), Đơn vị Hồi quy Cổng (GRU), LSTM, VAR, Rừng Ngẫu nhiên, Mô hình Mã hóa Dày Chuỗi Thời gian (TiDE), Autoformer, Mạng Nơ-ron Đa Tầng (MLP). Nhóm chúng em đã sử dụng ba tỷ lệ khác nhau giữa các tập dữ liệu train:test, cụ thể là 7:3, 8:2 và 9:1, để đánh giá hiệu suất của các mô hình dưới các điều kiện khác nhau. Sau đó so sánh hiệu suất của các mô hình khác nhau dựa trên ba chỉ số: Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), và Mean Absolute Error (MAE). Cuối cùng, ba mô hình có hiệu suất tốt nhất được sử dụng để dự báo chất lượng không khí cho 90 ngày tiếp theo, thể hiện hiệu quả của chúng trong việc dự báo chất lượng không khí. Bằng cách sử dụng các tập dữ liệu thời gian thực, nghiên cứu này nhằm dự báo, khám phá các mẫu, xu hướng và mối quan hệ giữa các chỉ số chất lượng không khí. Ngoài ra, nghiên cứu cũng nhằm đóng góp vào việc hiểu rõ hơn về việc giải quyết các vấn đề ô nhiễm không khí, thúc đẩy môi trường đô thị khỏe mạnh và bền vững hơn.***

***Index Terms*—Chất lượng không khí, ô nhiễm không khí, phân tích chuỗi thời gian, dự báo, Hồi quy Tuyến tính, ARIMA, RNN, GRU, LSTM, VAR, Random Forest, TiDE, Autoformer, MLP, Hà Nội, Hạ Long, Việt Trì, Việt Nam.**

## I. Giới thiệu

Trong những thập kỷ gần đây, sự chú ý toàn cầu ngày càng tập trung vào suy thoái môi trường, đặc biệt là ô nhiễm không khí, do tác động sâu rộng của nó đối với sức khỏe con người, hệ sinh thái và sự ổn định của khí hậu. Hạt phấn mịn (PM2.5 và PM10) cùng với các khí như ozone (O3), dioxide nitơ (NO2), dioxide lưu huỳnh (SO2), và carbon monoxide (CO) đóng vai trò quan trọng trong sự suy giảm chất lượng không khí. Trong bối cảnh này, nghiên cứu của Nhóm nhằm khám phá động lực chất lượng không khí trong môi trường đô thị, đặc biệt là ở ba thành phố lớn của Việt Nam: Hà Nội, Hạ Long và Việt Trì. Mỗi thành phố đại diện cho một ngữ cảnh kinh tế-xã hội và địa lý khác biệt, mang lại cái nhìn sâu sắc về các thách thức và cơ hội của quản lý ô nhiễm không khí. Mục tiêu tổng thể là sử dụng các kỹ thuật phân tích chuỗi thời gian tiên tiến để dự báo và phân tích các chỉ số về chất lượng không khí. Nhóm đã sử dụng một loạt các thuật toán, từ các phương pháp truyền thống như Hồi quy Tuyến tính và Mô hình Trung bình Chuyển động Tích hợp Tự động (ARIMA) đến các kiến trúc học sâu tiên tiến như Mạng Nơ-ron Hồi quy (RNN), Đơn vị Hồi quy Cổng (GRU), Bộ nhớ Ngắn hạn - Dài hạn (LSTM), Hồi quy Vector (VAR), Rừng Ngẫu nhiên, Mô hình Mã hóa Dày Chuỗi Thời gian (TiDE), Autoformer, và Mạng Nơ-ron Đa Tầng (MLP). Bằng cách tận dụng những kỹ thuật này nhằm mục đích khám phá các mẫu, xu hướng và dự báo dữ liệu chất lượng không khí trong tương lai. Các tập dữ liệu sử dụng bao gồm các số liệu đo thời gian thực về các chỉ số chất lượng không khí chính, bao gồm PM2.5, PM10, O3, NO2, SO2 và CO, được thu thập trong một khoảng thời gian dài. Những tập dữ liệu này cho phép khám phá các biến thể thời gian, sự đa dạng không gian và tương tác phức tạp giữa các chất gây ô nhiễm. Các kết quả của nhóm đóng góp vào sự hiểu biết rộng lớn hơn về việc giải quyết các thách thức đa mặt do ô nhiễm không khí gây ra, thúc đẩy môi trường đô thị khỏe mạnh và bền vững hơn. Ngoài ra, điều quan trọng là nhấn mạnh về vấn đề ô nhiễm không khí cấp bách mà Việt Nam đang phải đối mặt, đặc biệt là ở các thành phố lớn như Hà Nội, nơi chất lượng không khí đã nằm trong số tồi tệ nhất ở Đông Nam Á. Các yếu tố như mức độ cao của PM2.5 và các chất gây ô nhiễm khác đã gây ra những lo ngại sức khỏe đáng kể và tác động tiêu cực đến GDP của đất nước.

## II. Nghiên cứu liên quan

Trong những năm gần đây, đã có một sự gia tăng đáng kể trong nghiên cứu nhằm dự báo chất lượng không khí bằng cách sử dụng một loạt các kỹ thuật máy học, học sâu và thống kê.

Evgeniy Marinov, Dessislava Petrova-Antonova và Simeon Malinov, trong một nghiên cứu [1], tập trung vào việc cải thiện

độ chính xác của việc dự báo chất lượng không khí thông qua việc áp dụng phương pháp ARIMA. Nghiên cứu của họ, dựa trên dữ liệu từ các trạm giám sát chất lượng không khí tại Thành phố Sofia, Bulgaria, từ ngày 1 tháng 1 năm 2015 đến ngày 31 tháng 12 năm 2019, đã cho thấy hiệu quả của các mô hình ARIMA trong việc dự báo nồng độ ô nhiễm như CO, NO2, O3 và PM2.5.

S. H. Khaerun Nisa, Irfan Irfani và Utriweni Mukhaiyar, trong nghiên cứu của họ [2], đã khám phá việc dự báo mức độ ô nhiễm không khí tại Jakarta bằng phương pháp phân tích Vector Autoregressive (VAR). Bằng cách phân tích dữ liệu chuỗi thời gian về chỉ số chất lượng không khí (AQI) và nồng độ PM2.5 thu thập từ ngày 16 tháng 8 đến ngày 25 tháng 9 năm 2023, để huấn luyện, và từ ngày 25 tháng 9 đến ngày 1 tháng 10 năm 2023, để kiểm tra, họ đã xác định mô hình VAR(2) tối ưu cho việc dự báo ô nhiễm không khí chính xác.

Khawaja Hassan Waseem và cộng sự đã khám phá tác động của các yếu tố khí hậu đối với nồng độ PM2.5 và phát triển mô hình dự báo trong một nghiên cứu gần đây [3]. Nghiên cứu của họ, kéo dài trong 30 ngày và 72 giờ cho các dự đoán hàng ngày và hàng giờ tương ứng, đã kết hợp dữ liệu chất lượng không khí, chất gây ô nhiễm và điều kiện khí hậu từ nhiều thành phố ở Pakistan. Sử dụng các mô hình máy học và học sâu bao gồm FbProphet và LSTM, họ đã phát hiện mô hình mã hóa-giải mã LSTM vượt trội so với các mô hình khác, đạt được cải thiện đáng kể về độ chính xác dự báo.

Hai tác giả, Marwa Winis Misbah Esager và Kamil Demirberk Ünlü [4], đã áp dụng mô hình LSTM (Long Short-Term Memory) để dự báo nồng độ hàng giờ của hạt phấn mịn PM2.5 tại Tripoli, Libya. Họ đã sử dụng 100 epochs để huấn luyện mô hình, và kết quả tốt nhất đã được đạt được khi số nút được đặt là 20. Kết quả RMSE trên tập kiểm tra cho thấy mức độ lỗi thấp, khoảng 0.0146, chứng tỏ mô hình dự báo có độ chính xác khá cao.

Ngoài ra, các nghiên cứu gần đây cũng đã khám phá việc sử dụng các kỹ thuật mô hình hóa tiên tiến để cải thiện dự báo chất lượng không khí. Ví dụ, Abhimanyu Das và các đồng nghiệp đã giới thiệu TiDE (Time-series Dense Encoder) trong công việc của họ [5]. TiDE, một mô hình mới được tùy chỉnh cho dự báo chuỗi thời gian dài hạn, cho thấy khả năng hứa hẹn trong việc xử lý các biến đổi phi tuyến tính trong dữ liệu chuỗi thời gian. Các đánh giá thực nghiệm của TiDE đã cho thấy sự vượt trội hoặc tương đương với các phương pháp hiện tại trên các chỉ số dự báo dài hạn phổ biến trong thế giới thực, đồng thời tự hào về tốc độ suy luận và huấn luyện nhanh hơn đáng kể.

Ngược lại, nghiên cứu trước đó của Ong và đồng nghiệp [6] đã khám phá việc sử dụng các phương pháp dựa trên RNN để dự báo mức độ chất lượng không khí, trình bày một kỹ thuật động để tiền huấn luyện mô hình tập trung vào dự báo chuỗi thời gian nhiều bước trước. Tương tự, Lim và đồng nghiệp [7] đã sử dụng RNN để dự báo các chất gây ô nhiễm không khí khác nhau nhưng không tìm thấy sự khác biệt hoặc cải thiện đáng kể so với các mô hình truyền thống.

Bốn tác giả, Haixu Wu, Jiehui Xu, Jianmin Wang và Mingsheng Long, đã phát triển một mô hình gọi là Autoformer [8]. Mô hình này được thiết kế để giải quyết vấn đề dự báo chuỗi thời gian dài hạn, một thách thức đáng kể trong các ứng dụng thực tế như dự báo thời tiết cực đoan và lập kế hoạch tiêu thụ năng lượng dài hạn. Autoformer vượt trội so với các mô hình dựa trên Transformer trước đó bằng cách tích hợp các khối Phân rã và Mơ hình Tương quan Tự động dựa trên tính chu kỳ của chuỗi thời gian, cho phép mô hình khám phá và tổng hợp thông tin ở mức con-chuỗi. Kết quả thực nghiệm trên sáu thử nghiệm khác nhau, bao gồm năm ứng dụng thực tế từ năng lượng đến dịch bệnh, cho thấy rằng Autoformer đạt được độ chính xác hàng đầu với cải thiện tương đối 38% so với các phương pháp hiện tại.

Những phương pháp và kết quả đa dạng này nhấn mạnh những nỗ lực tiếp tục để nâng cao các kỹ thuật dự báo chất lượng không khí, tận dụng cả các phương pháp thống kê truyền thống và các mô hình máy học tiên tiến.

## III. Tài nguyên

### A. Bộ dữ liệu

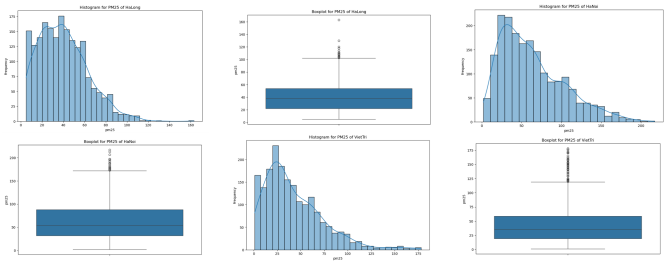
Bài viết sử dụng 3 bộ dữ liệu lấy từ dữ liệu chất lượng không khí trên trang web aqicn.org từ 01/03/2019 - 01/06/2024 ở 3 thành phố Hà Nội, Hạ Long và Việt Trì. Bộ dữ liệu bao gồm các thuộc tính cụ thể sau:

Bảng I
MÔ TẢ THUỘC TÍNH

| Thuộc tính | Mô tả |
|---|---|
| Ngày | Thời gian (YYYY-MM-DD) |
| Pm25 | Bụi mịn có đường kính từ 2.5 đến 10 micron |
| Pm10 | Bụi mịn có đường kính nhỏ hơn 2.5 micron |
| O3 | Ozon |
| NO2 | Đioxit nitơ |
| SO2 | Đioxit lưu huỳnh |
| CO | Carbon monoxit |

Bảng II
THỐNG KÊ MÔ TẢ CHO HẠ LONG, HÀ NỘI VÀ VIỆT TRÌ

| Thống kê | Hạ Long | Hà Nội | Việt Trì |
|---|---|---|---|
| Count | 1920 | 1920 | 1920 |
| Mean | 40.08 | 63.09 | 42.39 |
| Std Dev | 22.95 | 40.26 | 31.66 |
| Min | 5 | 2 | 1 |
| Q1 | 22 | 31 | 19 |
| Q2 (Median) | 38 | 54.5 | 35 |
| Q3 | 54 | 88 | 59 |
| Max | 163 | 217 | 178 |
| Mode | 5 | 24 | 1 |
| Variance | 527.01 | 1620.88 | 1002.69 |
| Kurtosis | 0.41 | 0.334 | 1.92 |
| Skewness | 0.685 | 0.902 | 1.24 |
| CV | 0.572 | 0.638 | 0.74 |



Hình 1. Biểu đồ histogram và boxplot

### B. Công cụ

Trong quá trình nghiên cứu và phân tích dữ liệu, Nhóm đã tận dụng một loạt các công cụ phân tích thống kê trong Python để khám phá sâu hơn các mẫu dữ liệu và rút ra những kết luận có ý nghĩa. Các công cụ chính bao gồm: Darts, numpy, pandas, sklearn, matplotlib.pyplot,... Sử dụng các công cụ phân tích thống kê này đã giúp Nhóm em hiểu sâu hơn về dữ liệu và đưa ra dự báo. Chi tiết các kết quả có thể được tìm thấy trong bảng mô tả và biểu đồ đi kèm.

### C. Tỷ lệ phân chia dữ liệu

Trong quá trình phân tích dữ liệu chuỗi thời gian, Nhóm em đã chia tập dữ liệu thành hai phần: tập huấn luyện và tập kiểm tra, với các tỷ lệ khác nhau như 70% cho huấn luyện và 30% cho kiểm tra, 80% cho huấn luyện và 20% cho kiểm tra, và 90% cho huấn luyện và 10% cho kiểm tra. Những tỷ lệ này giúp Nhóm đánh giá cách mà chúng ảnh hưởng đến hiệu suất của mô hình bằng cách xem xét sự phân phối của dữ liệu trong mỗi tập. Tỷ lệ phổ biến nhất là 7:3, chia 70% dữ liệu cho huấn luyện và 30% cho kiểm tra, tạo ra sự cân bằng giữa việc cung cấp đủ dữ liệu huấn luyện và đảm bảo các tập riêng biệt để điều chỉnh và đánh giá. Một lựa chọn khác là tỷ lệ 8:2, ưa chuộng việc sử dụng 80% dữ liệu cho huấn luyện, có ích cho các mô hình phức tạp yêu cầu tập dữ liệu huấn luyện lớn hơn. Trong một số trường hợp cụ thể, một cách tiếp cận thận trọng như tỷ lệ 9:1 có thể được ưa chuộng, đặc biệt khi xử lý tập dữ liệu lớn và một mô hình đơn giản. Tỷ lệ này đảm bảo có đủ dữ liệu huấn luyện trong khi vẫn giữ được một tập kiểm tra đáng kể để đánh giá hiệu suất.

### D. Đánh giá mô hình

Trong việc đánh giá hiệu suất của các mô hình, Nhóm em sử dụng ba chỉ số là Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), và Root Mean Squared Error (RMSE). Thuật toán có giá trị thấp nhất trong ba chỉ số này sẽ cho thấy mức độ chính xác tốt nhất. Dưới đây là các công thức để tính MAE, MAPE và RMSE.

*Với các tham số:*

- $n$: Số lượng điểm dữ liệu.
- $y_i$: Giá trị thực tế.
- $\hat{y}_i$: Giá trị dự đoán.

*MAE (Lỗi tuyệt đối trung bình):*

MAE là trung bình của các giá trị tuyệt đối của sai số giữa giá trị dự đoán và giá trị thực tế. Nó đo lường độ lớn của sai số trung bình và không quan tâm đến hướng của sai số. MAE càng nhỏ, mô hình càng chính xác.

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

*RMSE (Căn bậc hai của trung bình bình phương sai số):*

RMSE là căn bậc hai của trung bình của các bình phương của sai số giữa giá trị dự đoán và giá trị thực tế. Nó đo lường độ lớn của sai số trung bình và cung cấp một con số tương đối về mức độ sai lệch giữa dự đoán và giá trị thực tế. RMSE càng nhỏ, mô hình càng chính xác.

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

*MAPE (Lỗi phần trăm tuyệt đối trung bình):*

MAPE là trung bình của tỉ lệ phần trăm của sai số tuyệt đối so với giá trị thực tế. Nó thường được sử dụng để đo lường tỷ lệ phần trăm trung bình của sai số so với giá trị thực tế, cung cấp cái nhìn về mức độ chính xác của mô hình dự đoán. MAPE càng nhỏ, mô hình càng chính xác.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{3}$$

## IV. Phương pháp

### A. LINEAR REGRESSION

Hồi quy tuyến tính là một kỹ thuật quan trọng trong thống kê, cho phép chúng ta khám phá và mô hình hóa mối quan hệ giữa các biến. Giúp chúng ta hiểu rõ hơn về cách các yếu tố độc lập ảnh hưởng đến một biến phụ thuộc. Trong hồi quy tuyến tính đa biến, chúng ta có thể xem xét tác động của nhiều biến độc lập đến một biến phụ thuộc, giúp tạo ra các dự đoán hoặc giải thích phức tạp hơn về thực tế. Một mô hình hồi quy tuyến tính đa biến có dạng:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \epsilon$$

Trong đó:

- $Y$: Biến phụ thuộc.;
- $X_1, X_2, \ldots, X_k$: Các biến độc lập.;
- $\beta_0$: Hệ số chặn;
- $\beta_1, \beta_2, \ldots, \beta_k$: Các hệ số hồi quy.;
- $\epsilon$: Thành phần sai số.

### B. ARIMA

Phương pháp ARIMA (Autoregressive Integrated Moving Average) sử dụng một mô hình AR (Autoregressive) kết hợp với một mô hình MA (Moving Average) để thực hiện dự báo chuỗi thời gian. Các tham số chính cần xem xét bao gồm:

- Số lượng quan sát trước đó (p);

- Độ chênh lệch (d);
- Kích thước của trung bình chuyển động (q).

Mô hình AR hiển thị sự phụ thuộc của một quan sát vào một giai đoạn thời gian trước đó. Mô hình AR thu được p quan sát trước đó như sau:

$$y_t = \alpha + \sum_{i=1}^{p} \phi_i y_{t-i} + e_t \quad (5)$$

trong đó $y_t$ là biến dự đoán cho thời điểm t từ phân phối chuẩn và $y_{t-i}$ xác định p quan sát trước của cùng một chuỗi thời gian. $\phi_i$ biểu thị các hệ số hồi quy, $\alpha$ là một hằng số và $e_t$ là thuật ngữ lỗi ngẫu nhiên. Thứ tự p cho mô hình AR(p) được lựa chọn dựa trên các đỉnh quan trọng của PACF (Partial Autocorrelation Function). Một chỉ báo bổ sung là sự giảm chậm chạp của ACF (Autocorrelation Function).

MA thực hiện dự báo dựa trên các trung bình chuyển động của các thuật ngữ lỗi ngẫu nhiên trước đó như sau:

$$y_t = \mu + \sum_{i=1}^{q} \theta_i e_{t-i}$$

trong đó $\theta_t$ đại diện cho các hệ số hồi quy, q là thứ tự của trung bình chuyển động, và $\mu$ là một hằng số. Thứ tự q cho mô hình MA(q) được lấy từ ACF, nếu nó có một đoạn cắt sắc sau lags q. PACF giảm chậm trong trường hợp này.

Mô hình ARIMA có thể được xác định như sau:

$$\phi_p(B)(1-B)^d y_t = \theta_q(B) e_t \quad (7)$$

trong đó B là toán tử backshift, p là thứ tự tự hồi quy, d là thứ tự chênh lệch và q là thứ tự của trung bình chuyển động.

### C. VAR

Mô hình Vector Autoregression (VAR) là một công cụ mạnh mẽ trong phân tích dữ liệu thời gian đa biến. Thay vì chỉ tập trung vào một biến duy nhất như các mô hình hồi quy tuyến tính thông thường, VAR cho phép chúng ta đánh giá các tương tác phức tạp giữa nhiều biến số cùng một lúc. Điều này giúp chúng ta hiểu rõ hơn về cách các biến ảnh hưởng lẫn nhau qua thời gian, và tạo ra dự báo linh hoạt dựa trên các kịch bản khác nhau của tương lai. VAR đã trở thành một công cụ không thể thiếu trong lĩnh vực kinh tế lượng, tài chính và các lĩnh vực khác đòi hỏi sự hiểu biết sâu sắc về mối quan hệ giữa các biến số thời gian. Công thức chung của mô hình VAR với $p$ lags (trễ) được biểu diễn như sau:

$$Y_t = c + A_1 Y_{t-1} + A_2 Y_{t-2} + \ldots + A_p Y_{t-p} + \epsilon_t$$

Trong đó:

- $Y_t$: là một vector chứa các biến phụ thuộc tại thời điểm $t$;
- $c$: là một vector hằng số;
- $A_1, A_2, \ldots, A_p$: là các ma trận hệ số tương ứng với các lags;
- $\epsilon_t$: là vector của thành phần sai số tại thời điểm $t$.

### D. Random Forest

**Định nghĩa**

Random Forest là phương pháp xây dựng một tập hợp (ensemble) cây quyết định (decision tree) cho các bài toán phân lớp (classification) và hồi quy (regression). Cây quyết định thường có độ lệch thấp và phương sai cao, và Random Forest khai thác điều này bằng cách lấy trung bình các cây để cải thiện hiệu suất.

**Thách thức**

Thách thức chính của Random Forest là tạo ra tính ngẫu nhiên để giảm mối tương quan giữa các cây ($\rho(x)$) và duy trì độ lệch thấp.

**Thuật toán**

Thuật toán "Bagging" (Bootstrap Aggregating) là cơ chế chính của Random Forest. Mỗi cây quyết định được xây dựng từ một tập dữ liệu huấn luyện khác nhau và không bị cắt tỉa, tăng tính đa dạng và giảm lỗi. *Dự đoán hồi quy:* Lấy giá trị trung bình của dự đoán từ tất cả các cây để giảm phương sai và cải thiện hiệu suất. *Dự đoán phân lớp:* Lấy đa số phiếu bầu cho nhãn lớp từ tất cả các cây, giúp khắc phục sai sót của từng cây và tăng độ chính xác.

### E. Time-series Dense Encoder

**TiDE** (Time-series Dense Encoder) là một mô hình dự báo chuỗi thời gian. Mô hình hoạt động mã hóa chuỗi thời gian quá khứ cùng với hiệp phương sai bằng cách sử dụng "dense MLP" (Multi-Layer Perceptron). Sau đó, mô hình giải mã (decode) chuỗi thời gian được mã hóa (encode) cùng với các hiệp phương sai trong tương lai.

Kiến trúc tổng quan được trình bày ở Hình 1. Đầu vào (input) của mô hình là dữ liệu quá khứ và phương sai của một chuỗi thời gian tại một thời điểm $(y^{(i)}_{1:L}, x^{(i)}_{1:L}, a^{(i)})$ và ánh xạ tới dự đoán của chuỗi thời gian $\hat{y}^{(i)}_{L+1:L+H}$. Thành phần chính của mô hình là residual block MLP.

**Residual Block:** Là một thành phần quan trọng của kiến trúc TiDE vì nó cho phép mô hình nắm bắt các tính chất phi tuyến tính vốn có trong dữ liệu chuỗi thời gian, đồng thời duy trì các mối quan hệ tuyến tính nhằm giúp cải thiện hiệu suất dự báo dài hạn.

Mô hình được chia thành hai phần: mã hóa (encoding) và giải mã (decoding):

**Mã hóa (Encoding):** Nhiệm vụ của bước mã hóa là ánh xạ dữ liệu quá khứ và phương sai của chuỗi thời gian thành một biểu diễn dày đặc (dense). Quá trình mã hóa có hai bước chính

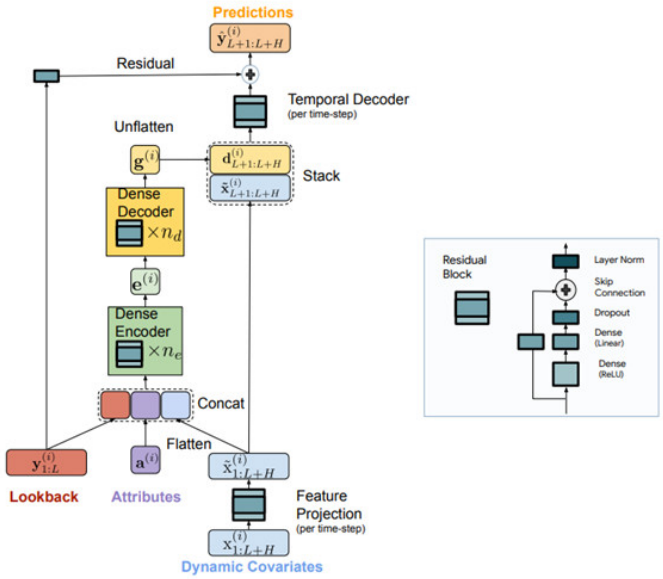
Feature Projection: Sử dụng residual block để ánh xạ $x^{(i)}_t$ tại mỗi time-step, hoạt động này được mô tả như sau

$$\tilde{x}^{(i)}_t = \text{ResidualBlock}(x^{(i)}_t) \quad (4)$$

**Giải mã (Decoding):** Việc giải mã trong mô hình ánh xạ các biểu diễn ẩn được mã hóa thành các dự đoán trong tương lai của chuỗi thời gian. Nó cũng bao gồm hai hoạt động, bộ giải mã dày đặc (dense decoder) và bộ giải mã tạm thời (temporal decoder).

### F. RNN

Mạng nơ-ron hồi tiếp (RNN) là một loại mạng có khả năng xử lý dữ liệu tuần tự. Điểm đặc biệt của RNN là các nơ-ron trong mạng có thể kết nối với nhau theo chu kỳ. Nhờ vậy, RNN




Hình 2. Tổng quan về kiến trúc TiDE

có thể học được mối quan hệ giữa các giá trị trong một chuỗi thời gian.

Công thức biểu diễn của RNN như sau:

$$h_t = f(x_t, h_{t-1}) \tag{5}$$

Trong đó:

- $h_t$: trạng thái của RNN tại thời điểm $t$
- $x_t$: đầu vào của RNN tại thời điểm $t$
- $h_{t-1}$: trạng thái của RNN tại thời điểm $t-1$
- $f$: hàm kích hoạt của RNN

## G. GRU

Trong lĩnh vực học máy và đặc biệt là xử lý ngôn ngữ tự nhiên (NLP), mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNNs) đóng vai trò quan trọng. Tuy nhiên, RNN truyền thống gặp phải vấn đề về vanishing gradient, làm giảm hiệu quả học của mạng khi xử lý các chuỗi dữ liệu dài. Để giải quyết vấn đề này, các kiến trúc RNN cải tiến như Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU) đã được đề xuất. Bài viết này tập trung vào Gated Recurrent Unit (GRU), một biến thể đơn giản hơn của LSTM nhưng vẫn rất hiệu quả trong việc duy trì thông tin dài hạn.

Điểm nổi bật của GRU là sự kết hợp giữa các cơ chế cập nhật và quên trong một đơn vị duy nhất, giúp giảm thiểu số lượng tham số cần thiết và tăng tốc độ tính toán mà vẫn duy trì hiệu quả cao.

Một GRU bao gồm hai cổng chính: cổng cập nhật (update gate) và cổng xoá (reset gate). Cả hai cổng này cùng hoạt động để điều chỉnh thông tin nào cần được cập nhật và thông tin nào cần được quên.

Cho $x_t$ là đầu vào tại thời điểm $t$, $h_t$ là trạng thái ẩn tại thời điểm $t$, và $h_{t-1}$ là trạng thái ẩn từ bước trước đó, các công thức của GRU được định nghĩa như sau:

### 1) Cổng Cập Nhật

Cổng cập nhật $z_t$ quyết định phần nào của trạng thái ẩn trước đó cần được giữ lại và phần nào cần được cập nhật:

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

### 2) Cổng Xoá

Cổng xoá $r_t$ quyết định phần nào của trạng thái ẩn trước đó cần được quên:

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

### 3) Trạng Thái Ẩn Mới

Trạng thái ẩn mới $\tilde{h}_t$ được tính toán bằng cách sử dụng cổng xoá để điều chỉnh thông tin từ trạng thái ẩn trước đó:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}))$$

### 4) Cập Nhật Trạng Thái Ẩn Cuối Cùng

Trạng thái ẩn cuối cùng $h_t$ tại thời điểm $t$ được tính bằng cách kết hợp trạng thái ẩn trước đó và trạng thái ẩn mới theo trọng số của cổng cập nhật:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

### 5) Diễn Giải Các Cổng

- **Cổng cập nhật** $z_t$: Quyết định tỉ lệ mà trạng thái ẩn trước đó được giữ lại so với trạng thái ẩn mới.
- **Cổng xoá** $r_t$: Điều chỉnh thông tin nào của trạng thái ẩn trước đó được sử dụng để tính toán trạng thái ẩn mới.

## H. LSTM

Long Short-Term Memory (LSTM) là một loại mạng nơ-ron tái hiện (RNN) được thiết kế để xử lý và dự báo chuỗi thời gian, khắc phục các vấn đề gradient biến mất và gradient bùng nổ trong các RNN truyền thống. LSTM bao gồm các đơn vị nhớ (memory cell) có khả năng lưu trữ thông tin trong một khoảng thời gian dài.

### 1) Cấu Trúc LSTM

Mỗi đơn vị LSTM bao gồm ba cổng chính: cổng quên (forget gate), cổng đầu vào (input gate), và cổng đầu ra (output gate). Các cổng này điều khiển dòng thông tin qua đơn vị nhớ.

#### a) Cổng Quên (Forget Gate)

Cổng quên xác định lượng thông tin từ trạng thái trước đó cần được giữ lại. Công thức tính toán như sau:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

trong đó, $f_t$ là giá trị của cổng quên tại thời điểm $t$, $W_f$ là trọng số của cổng quên, $h_{t-1}$ là đầu ra từ bước thời gian trước, $x_t$ là đầu vào tại thời điểm $t$, và $b_f$ là bias của cổng quên.

*b) Cổng Đầu Vào (Input Gate)*

Cổng đầu vào xác định lượng thông tin mới được lưu trữ trong trạng thái nhớ. Công thức tính toán như sau:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

trong đó, $i_t$ là giá trị của cổng đầu vào, $W_i$ là trọng số của cổng đầu vào, $\tilde{C}_t$ là giá trị thông tin mới, $W_C$ là trọng số của thông tin mới, và $b_i$, $b_C$ lần lượt là các bias của cổng đầu vào và thông tin mới.

*c) Cập Nhật Trạng Thái Nhớ*

Trạng thái nhớ được cập nhật bằng cách kết hợp trạng thái cũ và thông tin mới:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

trong đó, $C_t$ là trạng thái nhớ tại thời điểm $t$ và $C_{t-1}$ là trạng thái nhớ từ bước thời gian trước.

*d) Cổng Đầu Ra (Output Gate)*

Cổng đầu ra xác định đầu ra của đơn vị LSTM. Công thức tính toán như sau:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

trong đó, $o_t$ là giá trị của cổng đầu ra, $W_o$ là trọng số của cổng đầu ra, $b_o$ là bias của cổng đầu ra, và $h_t$ là đầu ra của đơn vị LSTM tại thời điểm $t$.

*2) Quá Trình Hoạt Động của LSTM*

Quá trình hoạt động của LSTM bao gồm ba bước chính:

1) **Tính toán cổng quên:** Xác định lượng thông tin từ trạng thái nhớ trước đó cần được giữ lại.
2) **Tính toán cổng đầu vào và cập nhật trạng thái nhớ:** Xác định lượng thông tin mới được thêm vào và cập nhật trạng thái nhớ.
3) **Tính toán cổng đầu ra:** Xác định đầu ra của đơn vị LSTM dựa trên trạng thái nhớ cập nhật.

## *I. Multi-Layer Perceptron*

Multi-Layer Perceptron (MLP) là một loại mạng nơ-ron nhân tạo, là một phần của học sâu (deep learning). MLP bao gồm ít nhất ba lớp: một lớp đầu vào, một hoặc nhiều lớp ẩn, và một lớp đầu ra. Mỗi lớp chứa nhiều nơ-ron (neurons), và mỗi nơ-ron trong một lớp kết nối với tất cả các nơ-ron trong lớp kế tiếp.

MLP có thể được sử dụng cho nhiều nhiệm vụ khác nhau như phân loại, hồi quy và dự báo chuỗi thời gian. Mỗi nơ-ron trong MLP sử dụng một hàm kích hoạt phi tuyến tính để tạo ra đầu ra của nó, giúp mạng có khả năng học các mối quan hệ phi tuyến giữa đầu vào và đầu ra.

*1) Mô tả thuật toán*

Mạng Perceptron Đa Lớp là một loại mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) bao gồm nhiều lớp perceptron được sắp xếp theo cấu trúc phân tầng. Cấu trúc của MLP thường bao gồm ba loại lớp chính:

- **Lớp đầu vào (Input Layer)**: Đây là lớp nhận các giá trị đầu vào và truyền chúng vào các lớp tiếp theo. Số lượng nơ-ron trong lớp này tương ứng với số lượng tính năng của dữ liệu đầu vào.
- **Lớp ẩn (Hidden Layer)**: MLP có thể có một hoặc nhiều lớp ẩn. Các lớp này chịu trách nhiệm học các đặc trưng phức tạp của dữ liệu. Mỗi nơ-ron trong lớp ẩn nhận đầu vào từ tất cả các nơ-ron của lớp trước đó và áp dụng một hàm kích hoạt (activation function) để xác định giá trị đầu ra.
- **Lớp đầu ra (Output Layer)**: Lớp này cung cấp kết quả cuối cùng của mạng. Số lượng nơ-ron trong lớp đầu ra phụ thuộc vào số lượng lớp mục tiêu (target classes) trong bài toán phân loại hoặc số lượng biến mục tiêu trong bài toán hồi quy.

Công thức cơ bản cho một nơ-ron trong lớp ẩn hoặc lớp đầu ra là:

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right)$$

Trong đó:

- $y$ là đầu ra của nơ-ron.
- $f$ là hàm kích hoạt (chẳng hạn như hàm sigmoid, tanh, hoặc ReLU).
- $x_i$ là các đầu vào.
- $w_i$ là các trọng số liên kết với các đầu vào.
- $b$ là hệ số điều chỉnh (bias).

*2) Thuật Toán Gradient Descent*

Gradient descent là một phương pháp tối ưu hóa phổ biến được sử dụng để tìm cực tiểu của hàm mất mát (loss function). Trong ngữ cảnh của MLP, hàm mất mát đo lường sự khác biệt giữa dự đoán của mạng và giá trị thực tế. Công thức cập nhật trọng số trong gradient descent như sau:

$$\theta_i := \theta_i - \eta \frac{\partial J(\theta)}{\partial \theta_i}$$

trong đó, $\theta_i$ là trọng số cần cập nhật, $\eta$ là tốc độ học (learning rate), và $J(\theta)$ là hàm mất mát.

*3) Thuật toán Backpropagation*

Backpropagation là một phương pháp hiệu quả để tính gradient của hàm mất mát đối với các trọng số của MLP. Quá trình backpropagation bao gồm hai bước chính: lan truyền tiến (forward propagation) và lan truyền ngược (backward propagation).

Trong bước lan truyền tiến, chúng ta tính toán đầu ra của mạng cho một đầu vào cụ thể. Trong bước lan truyền ngược, chúng ta tính toán gradient của hàm mất mát đối với từng trọng số bằng cách áp dụng quy tắc dây chuyền (chain rule).

*4) Feedforward*

Feedforward là quá trình tính toán đầu ra của mạng từ đầu vào bằng cách đi qua các lớp nơ-ron. Quá trình này bao gồm việc tính toán đầu ra của mỗi nơ-ron trong lớp ẩn và lớp đầu ra.

Giả sử $a^{(l)}$ là đầu ra của lớp $l$, quá trình feedforward được mô tả như sau:

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)}$$

$$a^{(l+1)} = \sigma(z^{(l+1)})$$

trong đó, $W^{(l)}$ và $b^{(l)}$ lần lượt là trọng số và bias của lớp $l$, $z^{(l+1)}$ là tổng trọng số, và $\sigma$ là hàm kích hoạt.

### J. Autoformer

Autoformer là một mô hình được thiết kế để dự báo chuỗi thời gian dài hạn, bao gồm ba thành phần chính: Khối phân rã chuỗi (Series Decomposition Block), Cơ chế tự tương quan (Auto-Correlation Mechanism), và Bộ mã hóa/giải mã (Encoder/Decoder).

#### 1) Khối Phân Rã Chuỗi

Khối phân rã chuỗi được sử dụng để tách chuỗi thời gian thành các thành phần xu hướng và mùa vụ. Phương pháp này sử dụng trung bình động để làm mượt các dao động định kỳ và làm nổi bật xu hướng dài hạn. Công thức cơ bản như sau:

$$X_t = \text{AvgPool}(\text{Padding}(X))$$

$$X_s = X - X_t$$

trong đó, $X_s$ và $X_t$ lần lượt là các phần mùa vụ và xu hướng của chuỗi.

#### 2) Cơ Chế Tự Tương Quan

Cơ chế tự tương quan được thiết kế để phát hiện các phụ thuộc dựa trên chu kỳ và tập hợp các chuỗi con tương tự từ các chu kỳ ngầm. Cơ chế này thay thế cho self-attention trong Transformer và có độ phức tạp $O(L \log L)$, trong đó $L$ là chiều dài chuỗi.

##### a) Công Thức Tự Tương Quan

Cơ chế tự tương quan bao gồm hai bước chính: tính toán tự tương quan và tổng hợp các giá trị tương quan.

$$\text{ACF}(X, \tau) = \frac{1}{L} \sum_{t=1}^{L-\tau} X_t \cdot X_{t+\tau}$$

trong đó, $\text{ACF}(X, \tau)$ là hàm tự tương quan, $X_t$ là giá trị tại thời điểm $t$, và $\tau$ là độ trễ.

Sau khi tính toán hàm tự tương quan, các giá trị được tổng hợp lại để tìm các chuỗi con tương tự, dựa trên công thức:

$$Y_{t+\tau} = \sum_{k=1}^{K} \alpha_k X_{t+\tau_k}$$

trong đó, $\alpha_k$ là trọng số tự tương quan, và $K$ là số lượng chuỗi con được chọn.

#### 3) Bộ Mã Hóa và Bộ Giải Mã

Bộ mã hóa và bộ giải mã của Autoformer bao gồm các khối phân rã chuỗi và cơ chế tự tương quan. Bộ mã hóa xử lý dữ liệu đầu vào từ chuỗi thời gian quá khứ, trong khi bộ giải mã tinh chỉnh các thành phần mùa vụ và xu hướng để đưa ra dự báo.

##### a) Bộ mã hóa (Encoder)

Bộ mã hóa bao gồm các khối phân rã và cơ chế tự tương quan để xử lý và trích xuất thông tin từ chuỗi đầu vào.

$$H^{(l)} = \text{Decompose}(X^{(l)})$$

$$Z^{(l)} = \text{AutoCorrelation}(H^{(l)})$$

trong đó, $H^{(l)}$ là đầu ra của lớp phân rã thứ $l$, và $Z^{(l)}$ là đầu ra của lớp tự tương quan thứ $l$.

##### b) Bộ giải mã (Decoder)

Bộ giải mã nhận các thành phần xu hướng và mùa vụ từ bộ mã hóa và tiếp tục tinh chỉnh, sử dụng các khối phân rã và cơ chế tự tương quan để dự báo chuỗi thời gian trong tương lai.

$$\hat{Y}^{(l)} = \text{Decompose}(Z^{(l)})$$

$$\hat{X}^{(l)} = \text{AutoCorrelation}(\hat{Y}^{(l)})$$

#### 4) Quá Trình Hoạt Động

Autoformer hoạt động theo quy trình sau:

1) **Đầu vào và Phân Rã Chuỗi:** Chuỗi thời gian quá khứ $X_{en}$ được đưa vào bộ mã hóa. Bộ mã hóa phân rã chuỗi này thành các thành phần xu hướng và mùa vụ.
2) **Bộ mã hóa (Encoder):** Áp dụng các khối phân rã và cơ chế tự tương quan để xử lý và trích xuất các thông tin cần thiết từ chuỗi đầu vào.
3) **Bộ giải mã (Decoder):** Nhận các thành phần xu hướng và mùa vụ từ bộ mã hóa và tiếp tục tinh chỉnh. Sử dụng các khối phân rã và cơ chế tự tương quan để dự báo chuỗi thời gian trong tương lai.

## V. Kết quả thí nghiệm

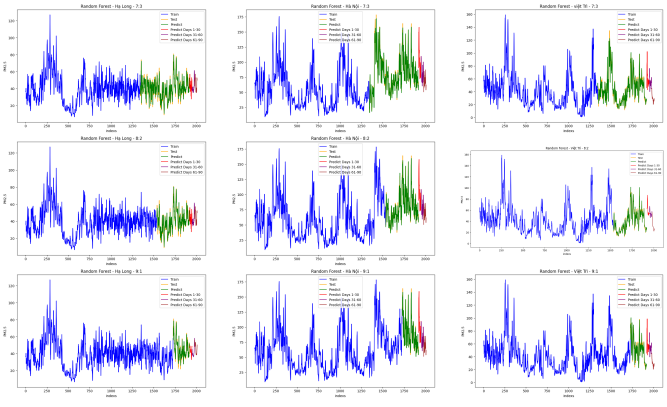
### A. Cài đặt mô hình

1. Linear Regression

2. ARIMA

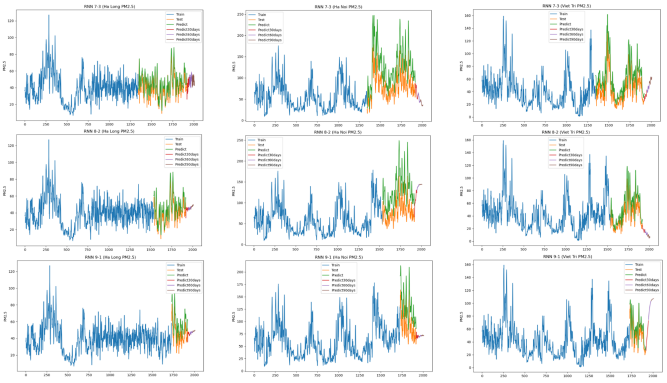
3. VAR

4. Random Forest



Hình 3. Kết quả dự báo của mô hình Random Forest

5. RNN



Hình 4. Kết quả dự báo của mô hình RNN

6. LSTM

7. GRU

8. LSTM

9. TiDE

10. Autoformer

*B. Đánh giá*

*C. Dự đoán PM.25 cho 90 ngày tiếp theo*

## VI. Kết luận

Thông qua các cuộc thử nghiệm ở cả 10 mô hình...

## Tài liệu


[1] E. Marinov, D. Petrova-Antonova, and S. Malinov, "Time Series Forecasting of Air Quality: A Case Study of Sofia City," *Atmosphere*, vol. 13, p. 788, 2022. [Online]. Available:https://www.mdpi.com/2073-4433/13/5/788

[2] K. N. Sh, I. Irfani, and U. Mukhaiyar, "Predicting Air Pollution Levels in Jakarta Using Vector Autoregressive Analysis," *Proceedings of the 5th International Conference on Statistics, Mathematics, Teaching, and Research 2023 (ICSMTR 2023)*, pp. 14-22, Atlantis Press, 2023. [Online]. Available:https://doi.org/10.2991/978-94-6463-332-0_3

[3] K. H. Waseem, H. Mushtaq, F. Abid, A. M. Abu-Mahfouz, A. Shaikh, M. Turan, and J. Rasheed, "Forecasting of Air Quality Using an Optimized Recurrent Neural Network," *Processes*, vol. 10, no. 10, p. 2117, 2022. [Online]. Available: https://doi.org/10.3390/pr1010211

[4] Citation: Esager, M.W.M.; Ünlü, K.D. "Forecasting Air Quality in Tripoli: An Evaluation of Deep Learning Models for Hourly PM2.5 Surface Mass Concentrations. *Atmosphere* 2023, 14, 478. [Online]. Available: https://doi.org/10.3390/atmos14030478

[5] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, "Long-term Forecasting with TiDE: Time-series Dense Encoder," *arXiv:2304.08424 [stat.ML]*, April 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2304.08424

[6] B. T. Ong, K. Sugiura, and K. Zettsu, "Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5," *Neural Comput Appl*, vol. 27, no. 6, pp. 1553–1566, 2016. [Online]. Available: https://doi.org/10.1007/s00521-015-1955-3


Bảng III
SỐ LIỆU HIỆU SUẤT TRÊN BỘ DỮ LIỆU PM2.5 Ở HÀ NỘI

| Model | Ratio | RMSE | MAE | MAPE (%) |
|---|---|---|---|---|
| Linear Regression | 7-3<br>8-2<br>9-1 | | | |
| ARIMA | 7-3<br>8-2<br>9-1 | | | |
| VAR | 7-3<br>8-2<br>9-1 | | | |
| Random Forest | 7-3<br>8-2<br>9-1 | 8.52<br>8.23<br>9.35 | 6.57<br>6.39<br>7.51 | 8.25<br>7.91<br>8.61 |
| RNN | 7-3<br>8-2<br>9-1 | 49.36<br>52.56<br>42.88 | 44.18<br>48.51<br>39.47 | 55.23<br>62.20<br>45.89 |
| MLP | 7-3<br>8-2<br>9-1 | | | |
| GRU | 7-3<br>8-2<br>9-1 | | | |
| LSTM | 7-3<br>8-2<br>9-1 | | | |
| TiDE | 7-3<br>8-2<br>9-1 | | | |
| Autoformer | 7-3<br>8-2<br>9-1 | | | |

Bảng IV
SỐ LIỆU HIỆU SUẤT TRÊN BỘ DỮ LIỆU PM2.5 Ở HẠ LONG

| Model | Ratio | RMSE | MAE | MAPE (%) |
|---|---|---|---|---|
| Linear Regression | 7-3<br>8-2<br>9-1 | | | |
| ARIMA | 7-3<br>8-2<br>9-1 | | | |
| VAR | 7-3<br>8-2<br>9-1 | | | |
| Random Forest | 7-3<br>8-2<br>9-1 | 5.29<br>4.67<br>4.71 | 4.20<br>3.70<br>3.74 | 11.60<br>10.88<br>8.61 |
| RNN | 7-3<br>8-2<br>9-1 | 8.76<br>9.21<br>13.58 | 6.94<br>7.38<br>12.08 | 19.54<br>20.93<br>28.23 |
| MLP | 7-3<br>8-2<br>9-1 | | | |
| GRU | 7-3<br>8-2<br>9-1 | | | |
| LSTM | 7-3<br>8-2<br>9-1 | | | |
| TiDE | 7-3<br>8-2<br>9-1 | | | |
| Autoformer | 7-3<br>8-2<br>9-1 | | | |

Bảng V
SỐ LIỆU HIỆU SUẤT TRÊN BỘ DỮ LIỆU PM2.5 Ở VIỆT TRÌ

| Model | Ratio | RMSE | MAE | MAPE (%) |
|---|---|---|---|---|
| Linear Regression | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |
| ARIMA | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |
| VAR | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |
| Random Forest | 7-3 | 6.31 | 4.58 | 10.88 |
| | 8-2 | 5.06 | 3.80 | 9.66 |
| | 9-1 | 5.90 | 4.64 | 8.39 |
| RNN | 7-3 | 19.97 | 16.70 | 41.66 |
| | 8-2 | 15.93 | 12.87 | 34.26 |
| | 9-1 | 14.03 | 11.57 | 25.25 |
| MLP | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |
| GRU | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |
| LSTM | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |
| TiDE | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |
| Autoformer | 7-3 | | | |
| | 8-2 | | | |
| | 9-1 | | | |


[7] Y. B. Lim, I. Aliyu, and C. G. Lim, "Air pollution matter prediction using recurrent neural networks with sequential data," In: *Proceedings of the 2019 3rd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence. ISMSI 2019*, pp. 40–44, Association for Computing Machinery, New York, NY, USA, 2019. [Online]. Available: https://doi.org/10.1145/3325773.3325788

[8] H. Wu, J. Xu, J. Wang, & M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," 2021.