

# DỰ BÁO CHẤT LƯỢNG KHÔNG KHÍ DỰA VÀO CÁC MÔ HÌNH THỐNG KÊ, HỌC MÁY VÀ HỌC SÂU

ĐOÀN NGỌC TUẤN<sup>1</sup>, DOĀN CÔNG TRÍ<sup>2</sup>, TRẦN LÊ TÚ<sup>3</sup>, TRẦN QUỐC HƯNG<sup>4</sup>, NGUYỄN PHƯỚC HUY<sup>5</sup>

**Tóm tắt nội dung**—Trong bối cảnh biến đổi khí hậu ngày càng trở nên đáng lo ngại, việc dự đoán và đánh giá khí hậu trở thành một yếu tố quan trọng để giúp quản lý nguồn lực và phát triển bền vững. Đặc biệt, ở Việt Nam, một quốc gia nằm trong khu vực có đặc điểm khí hậu phức tạp, việc dự đoán khí hậu có thể đóng vai trò quan trọng trong việc xây dựng các chiến lược quản lý môi trường và phát triển kinh tế. Trong nghiên cứu này, chúng tôi tiếp cận vấn đề này bằng cách sử dụng dữ liệu chuỗi thời gian về khí hậu của ba tỉnh lớn tại Việt Nam: Hà Nội, Đà Nẵng và Việt Trì. Chúng tôi áp dụng một loạt các mô hình dự đoán như VAR (Vector Autoregression), Linear Regression áp dụng CalendarFourier và DeterministicProcess, SES (Simple Exponential Smoothing), DLinear (Dynamic Linear Model), và NBEATS (Neural Basis Expansion Analysis Time Series) để so sánh và đánh giá hiệu suất dự đoán của mỗi mô hình trên dữ liệu thực tế.

**Index Terms**—Keywords - Time series , statistical method, forecasting AQI index, Linear Regression, VAR, DLinear, SES, NBEATS, Linear regression CalendarFourier and DeterministicProcess, ARIMA, RNN, GRU, LSTM

## I. GIỚI THIỆU

Sự gia tăng của ô nhiễm không khí không chỉ ảnh hưởng đến sức khỏe con người mà còn gây ra nhiều vấn đề khác liên quan đến môi trường và kinh tế. Đặc biệt là ở Việt Nam, một quốc gia nằm trong khu vực có đặc điểm khí hậu phức tạp. Trong bối cảnh này, việc dự đoán và đánh giá chất lượng không khí trở thành một mối quan tâm hàng đầu. Và việc lựa chọn một mô hình dự đoán phù hợp và chính xác không phải là việc đơn giản. Mục tiêu của nghiên cứu này là áp dụng các phương pháp thống kê, mô hình học máy và học sâu để dự đoán chất lượng không khí. Chúng tôi đã lựa chọn và áp dụng một loạt các mô hình dự đoán như VAR (Vector Autoregression), Linear Regression áp dụng CalendarFourier và DeterministicProcess, SES (Simple Exponential Smoothing), DLinear (Dynamic Linear Model), và NBEATS (Neural Basis Expansion Analysis Time Series) để so sánh và đánh giá hiệu suất dự đoán của mỗi mô hình trên dữ liệu thực tế. Câu hỏi đặt ra cho nghiên cứu là “Các mô hình đã lựa chọn có đạt được hiệu suất dự đoán như mong đợi hay không?”. Nghiên cứu sẽ tập trung vào việc áp dụng các phương pháp dự đoán chất lượng không khí trên dữ liệu

thực tế của ba thành phố lớn ở Việt Nam: Hà Nội, Đà Nẵng và Việt Trì. Phạm vi của nghiên cứu sẽ giới hạn trong việc so sánh và đánh giá hiệu suất của các mô hình đã lựa chọn. Kết quả của nghiên cứu sẽ cung cấp thông tin quan trọng cho quyết định quản lý chất lượng không khí và phát triển bền vững tại các thành phố lớn. Ngoài ra, nó cũng có thể đề xuất các phương pháp mới và hiệu quả hơn trong việc dự đoán chất lượng không khí, mang lại lợi ích lớn cho việc quản lý môi trường và kinh tế.

## II. CÁC NGHIÊN CỨU LIÊN QUAN

Nur Izzah Jamil và đồng nghiệp (2019) đã tiến hành một phân tích toàn diện về chỉ số chất lượng không khí (Air Pollutant Index - API) tại Petaling Jaya, Malaysia[1]. Bằng cách sử dụng thuật toán Single Exponential Smoothing (SES), Double Exponential Smoothing (DES) và một số thuật toán khác để đánh giá và phân tích dữ liệu, họ đã đề xuất một phương pháp hiệu quả để đo lường và dự đoán mức độ ô nhiễm không khí trong khu vực. [1]

Jalin và đồng nghiệp (2023) đã đề xuất một phương pháp kết hợp xử lý ngoại lai (Outlier Correction), phân tích tín hiệu được cải tiến (Optimized Decomposition) và mô hình dự báo DLinear để dự báo tốc độ gió nhiều bước (multi-step-ahead wind speed forecast) tiên tiến, có độ chính xác cao. [2]

Taesung và đồng nghiệp đề xuất một phương pháp mới sử dụng mạng nơ-ron nhân tạo học sâu (deep neural networks) N-BEASTS[3] để xử lý các giá trị thiếu trong dữ liệu chất lượng không khí theo chuỗi thời gian. [3]

Linear Regression là một trong những phương pháp phổ biến và lâu đời được sử dụng cho bài toán dự đoán chất lượng không khí. A. Loganathan\*, P. Sumithra, V. Deneshkumar đã đề xuất phương pháp ước tính mô hình hồi quy tuyến tính bội dựa trên thông tin liên quan đến chỉ số chất lượng không khí được ghi lại tại một trạm quan trắc ở Chennai, Ấn Độ. [4]

Mô hình Vector AutoRegression (VAR) có thể được áp dụng để dự đoán chất lượng không khí (AQI) với một số lợi ích quan trọng. Khaerun Nisa SH, Irfan Irfani, Utriweni Mukhaiyar đã đưa ra nghiên cứu dự đoán mức độ ô nhiễm không khí ở Jakarta bằng phương pháp phân tích Vector Autoregression (VAR) trên dữ liệu chuỗi thời gian về mức độ ô nhiễm không khí (AQI) và nồng độ hạt vật chất (PM2.5) [5]

Linear Regression kết hợp với CalendarFourier và DeterministicProcess giúp cải thiện khả năng dự đoán chất lượng không khí (AQI) bằng cách xử lý thông tin về thời gian và tích hợp các yếu tố không ngẫu nhiên. Trong nghiên cứu này, các mô hình Hồi quy tuyến tính đa biến dự đoán Chỉ số chất lượng không khí ngắn hạn của Thị trấn Amravati nằm ở bang Maharashtra [6]

Bhalgat và cộng sự. (2019) đã trình bày một mô hình tích hợp để dự đoán mức độ ô nhiễm không khí bằng cách sử dụng mạng lưới thần kinh nhân tạo và kriging trong nghiên cứu của họ. Mô hình này sử dụng giao thức hồi quy tuyến tính và perceptron đa lớp (ANN) để dự đoán ngày hôm sau. Mô hình AR và ARIMA dự đoán thành công giá trị SO<sub>2</sub>, nhưng cần nhiều nghiên cứu hơn để dự đoán PM2.5 và tính toán AQI [7]

Ong và cộng sự. (2016) đã kết nối RNN để dự đoán PM2.5 bằng thông tin cảm biến tự nhiên, từ đó đưa ra kết quả chính xác. Nghiên cứu của Kurt và Oktay (2010) về việc dự kiến ô nhiễm không khí với hệ thống thần kinh cho thấy ưu thế và khả năng đạt được của chiến lược. Liang và cộng sự. (2015) đã xuất bản một tập dữ liệu chứa ước tính PM2.5 vừa được ước tính ở Bắc Kinh. Sau đó, Liang và cộng sự. (2016) đã phân phối một bộ dữ liệu lớn hơn để phân tích yếu tố độc hại ở 5 khu vực thành thị của Trung Quốc. Tất cả các tập dữ liệu được sử dụng ở trên đều là thông tin theo điểm, không cho phép chúng tôi trình bày theo cách thể hiện không gian [8]

Bài toán dự đoán nồng độ chất ô nhiễm không khí là bài toán dự đoán chuỗi thời gian điển hình với nhiều biến đầu vào. Với mạng thần kinh tái phát GRU (Gated Recurrent Unit) có thể tìm hiểu thông tin phụ thuộc dài hạn và chúng ta có thể sử dụng nó trong lĩnh vực dự đoán chuỗi thời gian. Trong khi đó, nghiên cứu của Xinxing Zhou và các cộng sự đã xác nhận rằng GRU hoạt động tốt hơn trong việc hội tụ về thời gian CPU, cập nhật tham số và khai quật hóa so với LSTM (Mạng bộ nhớ ngắn dài). Họ đã sử dụng GRU để thiết lập mô hình dự đoán nồng độ trung bình mỗi giờ PM2.5 cho Bắc Kinh, bao gồm mô hình quanh năm và bốn mô hình tương ứng với bốn mùa xuân, hạ, thu và đông, để cải thiện độ chính xác của Dự đoán nồng độ PM2.5. [9]

RNN có xu hướng tạo ra độ dốc khi xử lý chuỗi thời gian dài nên độ chính xác của nó thường kém. Để giải quyết vấn đề này, LSTM lần đầu tiên được giới thiệu bởi Hochreiter, S. et al. và tái xuất hiện như một kiến trúc thành công. Mạng nơ-ron LSTM là một biến thể của cấu trúc RNN. Ý tưởng chính của nó là giới thiệu một cơ chế chọn cổng thích ứng. Nó xác định mức độ mà đơn vị LSTM vẫn ở trạng thái trước đó. Nó cũng có thể ghi nhớ các tính năng được trích xuất của dữ liệu đầu vào hiện tại. Mặc dù có nhiều biến thể của LSTM được đề xuất nhưng kiến trúc tiêu chuẩn của LSTM được áp dụng trong bài báo này để dự đoán chất lượng không khí. [10]

### III. TÀI NGUYÊN

#### A. BỘ DỮ LIỆU

Mục đích của nghiên cứu là dự báo chất lượng không khí tại 3 thành phố: Hà Nội, Việt Trì và Đà Nẵng. Vì vậy, một bộ dữ liệu đã được thu thập về các phép đo liên quan đến

chất lượng không khí ở ba thành phố này. Ba bộ dữ liệu đã được thu thập trên trang web AQICN, trong khoảng thời gian từ ngày 1 tháng 1 năm 2019 đến ngày 10 tháng 3 năm 2024 và bao gồm các thuộc tính được mô tả sau:

Bảng I: MÔ TẢ THUỘC TÍNH

Thuộc tính	Mô tả
Date	Ngày đo chất lượng không khí (dd/mm/yyyy)
PM2.5	Nồng độ bụi mịn PM2.5 (microgram/m <sup>3</sup> )
PM10	Nồng độ bụi mịn PM10 (microgram/m <sup>3</sup> )
O3	Nồng độ khí Ozon (microgram/m <sup>3</sup> )
NO2	Nồng độ khí Nitrogen dioxide (microgram/m <sup>3</sup> )
SO2	Nồng độ khí Sulfur dioxide (microgram/m <sup>3</sup> )
CO	Nồng độ khí Carbon monoxide (mg/m <sup>3</sup> )

#### B. THÔNG KÊ MÔ TẢ

Bảng II: THÔNG KÊ MÔ TẢ AQI HÀ NỘI, ĐÀ NẴNG, VIỆT TRÌ

	AQIHanoi	AQIDaNang	AQIVietTri
Count	1979	1979	1979
Mean	111.63	81.48	78.59
Std	49.83	43.83	42.55
Min	15	10	5
25%	70	50	46.5
50%	110	74	72
75%	152.85	101.67	106.67
Max	267	362	228

**Nhận xét:** AQI ở thành phố Hà Nội cao hơn hẳn so với Đà Nẵng và Việt Trì, cho thấy mức độ ô nhiễm không khí ở khu vực này rất cao. Bên cạnh đó, độ biến động AQI của Hà Nội cũng khá cao, không ổn định về chất lượng không khí. AQI ở Đà Nẵng ở mức trung bình, có những ngày AQI đạt ngưỡng 362, ảnh hưởng nghiêm trọng tới đời sống con người. Ngược lại thì Việt Trì có AQI ở mức tương đối thấp, cho thấy chất lượng không khí ở đây tốt hơn nhiều so với hai khu vực còn lại.

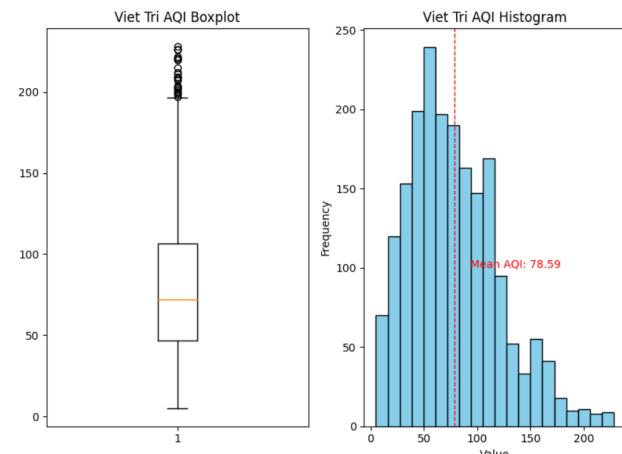
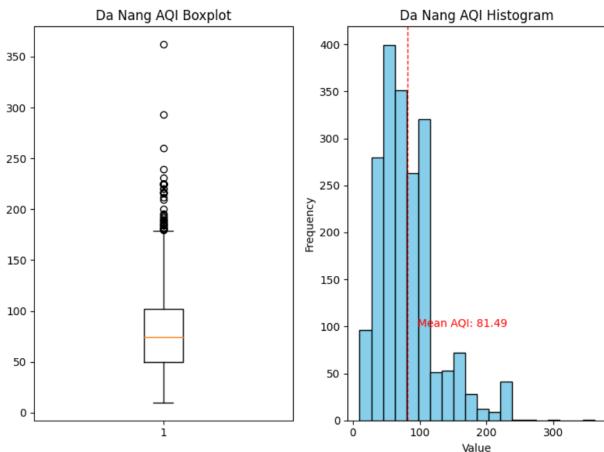
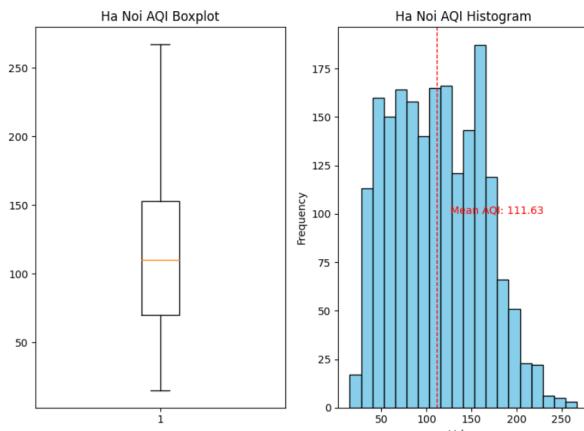


Fig1. Box Plot và Histogram Chart của AQI Việt Trì



**Fig2.** Box Plot và Histogram Chart của AQI Đà Nẵng



**Fig3.** Box Plot và Histogram Chart của AQI Hà Nội

### C. CÔNG CỤ

Trong nghiên cứu này, chúng tôi sử dụng ngôn ngữ lập trình Python cùng với các thư viện và công cụ phổ biến trong lĩnh vực thống kê, máy học và học sâu, với mục đích thực hiện phân tích thống kê và huấn luyện mô hình cũng như đánh giá hiệu suất của các mô hình dự đoán chất lượng không khí của chúng tôi một cách hiệu quả và linh hoạt.

### D. TỈ LỆ PHÂN CHIA DỮ LIỆU

Trong nghiên cứu này, các tập dữ liệu được chia thành hai tập train và test với các tỉ lệ 7:3; 8:2; 9:1 để sử dụng cho việc huấn luyện và kiểm tra các mô hình.

### E. CHỈ SỐ ĐÁNH GIÁ MÔ HÌNH

Trong nghiên cứu này, nhóm sử dụng các chỉ số: Root Mean Squared Error (**RMSE**), Mean absolute error (**MAE**), Mean absolute percentage error (**MAPE**) để đánh giá và so sánh các mô hình trên các tập dữ liệu.

## IV. PHƯƠNG PHÁP NGHIÊN CỨU

### A. VAR - VECTOR AUTOREGRESSION

Tự hồi quy vecto (VAR) là một mô hình thống kê được sử dụng trong kinh tế lượng, có thể được sử dụng khi hai hoặc

nhiều chuỗi thời gian ảnh hưởng lẫn nhau. Nghĩa là, mối quan hệ giữa chuỗi thời gian liên quan là hai chiều. Nó ước tính từng phương trình của từng biến chuỗi theo độ trễ của biến ( $p$ ) và tất cả các biến còn lại. n dự đoán tốt nhất biến y là hàm tuyến tính của các biến x.

Một phương trình mô hình AR( $p$ ) điển hình có dạng sau:

$$Y_t = \alpha_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \varepsilon_t$$

Trong đó:

- $\alpha_0$  là hệ số chặn
- $\beta_1, \dots, \beta_p$  là các hệ số hồi quy cho các biến độc lập.
- $\varepsilon$  là hệ số lỗi.

Đối với mô hình VAR với 2 chuỗi thời gian ( $Y_1, Y_2$ ), ta có hệ phương trình như sau:

$$Y_{1,t} = \alpha_0 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \cdots + \varepsilon_{1,t}$$

$$Y_{2,t} = \alpha_0 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \cdots + \varepsilon_{2,t}$$

Trong đó:

- $Y_{1,t}, Y_{2,t}$  lần lượt là độ trễ đầu tiên của chuỗi thời gian  $Y_1$  và  $Y_2$ .
- $\beta_{ij,k}$  Hệ số đo lường tác động của biến trễ  $Y_{i,t-k}$  on  $Y_{i,t}$
- $\varepsilon_{1,t}, \varepsilon_{2,t}$  là những hệ số lỗi.

### B. LINEAR REGRESSION APPLY CALENDARFOURIER AND DETERMINISTICPROCESS

#### 1) Linear Regression

Mô hình hồi quy tuyến tính là một phương pháp trong thống kê để dự đoán giá trị của biến phụ thuộc dựa trên 2 hoặc nhiều biến độc lập. Nó giả định mối quan hệ tuyến tính giữa các biến và sử dụng phương pháp hồi quy để ước lượng các hệ số.

#### 2) CalendarFourier

Lớp CalendarFourier trong statsmodels là một công cụ mạnh mẽ để mô hình các thành phần xác định dựa trên thời gian trong lịch. Đặc biệt, nó hữu ích khi làm việc với dữ liệu chuỗi thời gian có các mẫu theo mùa, được sử dụng để tạo ra các thành phần xác định dựa trên thời gian trong chuỗi thời gian bằng cách sử dụng chuỗi Fourier. Thường được sử dụng để mô hình hóa các yếu tố theo chu kỳ trong chuỗi thời gian, ví dụ như yếu tố hàng năm hoặc hàng tháng.

$$\text{Fourier terms} = [\sin(t), \cos(t), \sin(2t), \cos(2t), \dots, \sin(kt), \cos(kt)]$$

#### 3) DeterministicProcess

DeterministicProcess là một lớp chung có thể được sử dụng để chỉ định thành phần xác định trong mô hình chuỗi thời gian DeterministicProcess nhằm mục đích tạo ra các tính năng được sử dụng trong mô hình Hồi quy để xác định xu hướng và tính chu kỳ. Nó lấy DatetimeIndex và một vài tham số khác rồi trả về một DataFrame có đầy đủ các tính năng cho mô hình.

#### 4) Linear Regression áp dụng calendarFourier and DeterministicProcess

Xử lý các thành phần chu kỳ: CalendarFourier được sử dụng để mô hình các thành phần chu kỳ, như các mẫu mùa vụ, trong dữ liệu chuỗi thời gian bằng cách sử dụng chuỗi Fourier. Điều

này cho phép mô hình nắm bắt được các biến động mùa vụ trong dữ liệu, DeterministicProcess nhằm mục đích tạo ra các đặc trưng được sử dụng trong mô hình Hồi quy để xác định xu hướng và tính chu kỳ

Trong phân tích chuỗi thời gian và hồi quy tuyến tính, chúng ta sử dụng CalendarFourier và DeterministicProcess để mô hình hóa các yếu tố xác định dựa trên thời gian.

### C. ARIMA

ARIMA là viết tắt của "Autoregressive Integrated Moving Average", là một phương pháp dự báo thống kê được sử dụng rộng rãi trong phân tích chuỗi thời gian. Mô hình này tích hợp các thành phần Autoregressive (AR), Moving Average (MA), và Differencing (I) để bắt các mối quan hệ giữa giá trị hiện tại và quá khứ của một chuỗi thời gian.

#### Autoregressive (AR):

AR (Tự hồi quy) - là quá trình tìm mối quan hệ giữa dữ liệu hiện tại và p dữ liệu quá khứ trước đó (Gọi là lag):

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t$$

Trong đó:

- $y_t$  là giá trị hiện tại.
- $c$  là thuật ngữ hằng số.
- $p$  là số lượng đơn đặt hàng.
- $\phi$  là hệ số tự hồi quy.
- $\epsilon_t$  là thuật ngữ lỗi.

**Moving Average (MA):** MA (Trung bình di động) - là quá trình tìm mối quan hệ giữa dữ liệu hiện tại và q phần lõi quá khứ trước đó.

$$y_t = c + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} + \epsilon_t$$

**Differencing (I):** So sánh sự khác nhau giữa d quan sát (Hiệu giữa giá trị hiện tại và d giá trị trước đó):

Nếu  $d = 0$ :  $\Delta Y_t = Y_t$

Nếu  $d = 1$ :  $\Delta Y_t = Y_t - Y_{t-1}$

Nếu  $d = 2$ :  $\Delta Y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t + 2Y_{t-1} + Y_{t-2}$

Sau khi kết hợp chúng, chúng ta sẽ có ARIMA ( $p, d, q$ ) được biểu diễn như sau:

$$\begin{aligned} \Delta Y_t &= c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \\ &\quad \dots + \theta_q \epsilon_{t-q} + \epsilon_t \end{aligned}$$

Mô hình ARIMA nắm bắt mối quan hệ giữa giá trị hiện tại và quá khứ của một chuỗi thời gian, tích hợp các thành phần tự hồi quy, trung bình di động và đạo hàm. Chúng được sử dụng rộng rãi để dự báo giá trị tương lai của các chuỗi thời gian khác nhau, bao gồm dữ liệu kinh tế, giá cổ phiếu và mô hình thời tiết.

### D. LINEAR REGRESSION

Phân tích hồi quy là một công cụ để xây dựng các mô hình toán học và thống kê mô tả mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập hoặc biến giải thích, tất cả đều là số. Kỹ thuật thống kê này được sử dụng để tìm

phương trình dự đoán tốt nhất biến y dưới dạng hàm tuyến tính của biến x. Mô hình hồi quy tuyến tính bội có dạng:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

Trong đó:

- $Y$  là biến phụ thuộc (Biến mục tiêu)
- $X_1, X_2, \dots, X_k$  là các biến độc lập
- $\beta_0$  là hệ số chặn.
- $\beta_1, \dots, \beta_k$  là các hệ số hồi quy cho các biến độc lập.
- $\varepsilon$  là hệ số lỗi.

### E. SES - SIMPLE EXPONENTIAL SMOOTHING

Simple exponential smoothing (SES) là một phương pháp dự đoán dữ liệu theo thời gian. Nó sử dụng tổng có trọng lượng của dữ liệu trước trong dây dữ liệu theo thời gian để đoán các dữ liệu trong tương lai.

Simple exponential smoothing có công thức như sau:

$$F_t = F_{t-1} + a \cdot (A_{t-1} - F_{t-1})$$

Với  $a$  là smoothing factor, và  $0 \leq a \leq 1$ .  $F_t$  (giá trị dự đoán ở thời gian  $t$ ) là trung bình có trọng lượng của  $A_{t-1}$  (giá trị thực tế tại thời gian  $t-1$ ) và  $F_{t-1}$  (giá trị dự đoán tại thời gian  $t-1$ ).

$a$  (smoothing factor) là một tham số quan trọng trong SES, nó kiểm tra độ quan trọng của dữ liệu mới và dữ liệu cũ.  $a$  càng gần đến 1 thì độ quan trọng của dữ liệu mới càng cao và ngược lại  $a$  càng gần về 0 thì độ quan trọng của dữ liệu mới càng thấp.

Không có một cách cụ thể để tính toán  $a$ . Nhưng cũng có nhiều cách để tối ưu hóa độ chính xác của  $a$ . Ví dụ, có thể sử dụng phương pháp bình phương tối thiểu để tính giá trị của  $a$ , cụ thể là tìm  $a$  để tổng các  $(F_t - A_{t+1})^2$  là thấp nhất.

### F. DECOMPOSED LINEAR (D-LINEAR)

D-Linear sử dụng một phương pháp phân rã để chia dữ liệu thành thành phần xu hướng và mùa vụ. Sau đó, hai mạng tuyến tính một tầng được áp dụng cho mỗi thành phần và các đầu ra được cộng lại để có được dự đoán cuối cùng.

#### 1) Cấu trúc của DLinear

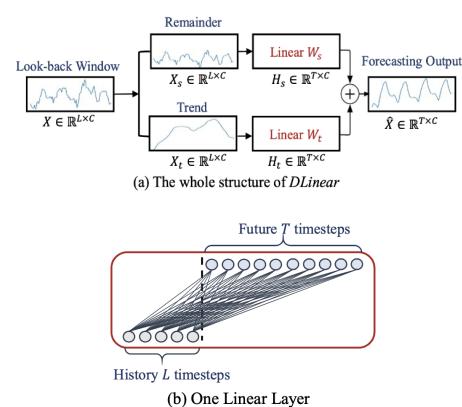


Fig4. Hình minh họa của mô hình Decomposition Linear

Cấu trúc tổng quan được thể hiện ở hình (a). Toàn bộ quá trình:  $\hat{H} = H_s + H_t$ , trong đó  $H_s = W_s X_s \in \mathbb{R}^{T \times C}$  và  $H_t = W_t X_t \in \mathbb{R}^{T \times C}$  là các thành phần xu hướng và phần dư đã được phân rã.  $W_s \in \mathbb{R}^{T \times C}$  và  $W_t \in \mathbb{R}^{T \times C}$  là hai lớp tuyến tính, như được minh họa trong Hình (b).

## 2) Kiến trúc Phân rã

Với độ dài- $L$  của chuỗi đầu vào  $X \in \mathbb{R}^{T \times C}$ :

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

trong đó  $X_s, X_t \in \mathbb{R}^{T \times C}$  đại diện cho phần mùa và phần xu hướng chu kỳ được trích xuất tương ứng.

## G. NEURAL BASIS EXPANSION ANALYSIS

### 1) Dữ liệu đầu vào

Xét bài toán dự báo đơn chuỗi thời gian:

Từ chuỗi thời gian có độ dài  $t$  là

$x = [y_{T-t+1}, y_{T-t+2}, \dots, y_T] \in \mathbb{R}^T$  ta cần dự báo giá trị của  $H$  bước tiếp theo là  $x = [y_{T+1}, y_{T+2}, \dots, y_{T+H}] \in \mathbb{R}^H$ . Ta kí hiệu  $\hat{y}_i$  là dự đoán của mô hình cho vectơ  $y$ . Kích thước của dữ liệu đầu vào là  $t = nH$  ( $n$  thường nằm trong khoảng từ [2, 7]) được gọi là khoảng thời gian xem lại (lookback period). Mô hình N-BEATS sẽ học và tìm hiểu chuỗi thời gian từ khoảng thời gian xem lại để dự đoán giá trị của  $H$  điểm tiếp theo.

### 2) Kiến trúc tổng quát của mô hình

Kiến trúc tổng quát của mô hình N-BEATS được mô tả như trong hình bên dưới. Mô hình bao gồm các block được xếp chồng lên nhau. Đầu vào của block đầu tiên là đầu vào tổng thể của mô hình hay còn gọi là khoảng thời gian xem lại  $x$ . Mỗi block có hai đầu ra. Một đầu ra (backcast) sẽ được làm đầu vào cho block tiếp theo. Đầu ra còn lại (forecast) sẽ được tổng hợp để đưa ra kết quả cuối cùng.

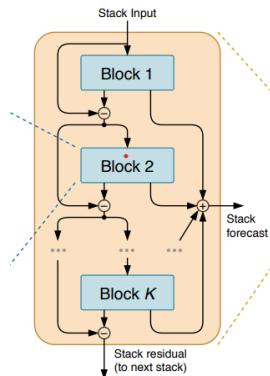


Fig5. Kiến trúc tổng quát của mô hình N-BEATS

Chi tiết hơn, với block thứ  $i$  trong mô hình:

- Nếu  $i = 1 = 1$ , tức là block đầu tiên, đầu vào chính là cửa sổ dữ liệu  $x$

- Nếu  $i > 1$ , đầu vào của block thứ  $i$  là:  $x^i = x^{i-1} - \hat{x}^{i-1}$

Block thứ  $i$  gồm hai đầu ra:

- Đầu ra backcast  $\hat{x}^i$  được làm đầu vào cho block tiếp theo.

- Đầu ra focecast  $\hat{y}^i$  dùng để tổng hợp cho kết quả dự đoán cuối cùng:

$$\hat{y} = \sum_{i=1}^R \hat{y}^i$$

Từ đây, ta thấy được đầu vào của block sẽ không chứa mà block trước đó đã dự đoán được. Điều này giúp các block sau tập trung vào những phần thông tin chưa được dự đoán.

### 3) Kiến trúc của block

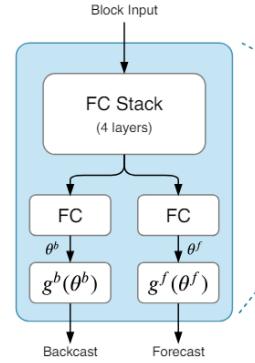


Fig6. Kiến trúc của block

Hình trên mô tả kiến trúc trong block thứ  $i$  của mô hình. Kiến trúc của block đơn giản chỉ bao gồm các lớp kết nối đầy đủ FC (fully connected). Các lớp kết nối đầy đủ đơn giản là một lớp ánh xạ tuyến tính với hàm kích hoạt RELU. Ví dụ:

$$FC(x^i) = \text{RELU}(Wx^i + b)$$

trong đó  $W$  và  $b$  lần lượt là các ma trận sẽ được học trong quá trình huấn luyện. Hàm RELU là hàm  $f(x) = \max(0, x)$ . Block bao gồm  $M$  lớp FC chung ( $M$  thường là 4). Sau đó, sẽ có hai lớp FC là  $M+1$  và  $M+2$  để chia đầu ra thành 2 nhánh. Đầu ra của hai lớp trên lần lượt là hai vectơ hệ số  $\theta^f$  và  $\theta^b$ . Cuối cùng, ta sẽ ánh xạ hai vectơ trên thành các đầu ra backcast và forecast thông qua các hàm  $g^b$  và  $g^f$ . Tùy vào dạng hàm  $g$  khác nhau mà chúng ta sẽ có các loại block với vai trò khác nhau. Cụ thể, mô hình N-BEATS giới thiệu ba loại block: trend block, seasonality block, generic block.

Generic block là kiến trúc không mang tính diền giải được trong chuỗi thời gian. Generic block đơn giản đặt các ánh xạ  $g^b$  và  $g^f$  là các ánh xạ tuyến tính đối với đầu ra của lớp trước.

Trend block là loại block để nhận biết và phân tích được tính xu hướng của dữ liệu. Một đặc điểm của xu hướng thường phần lớn là các hàm đơn điệu hoặc thay đổi chậm. Để tìm hiểu được tính xu hướng của dữ liệu, các hàm  $g^b$  và  $g^f$  sẽ là các hàm đa thức với bậc  $p$  nhỏ theo thời gian. Khi đó, các hệ số  $\theta^f$  và  $\theta^b$  đóng vai trò là các hệ số của đa thức. Ví dụ đầu ra forecast của trend block thứ  $i$  là:

$$\hat{y}^i = \sum_{j=1}^R \theta_j^f t^j$$

Trong đó,  $\theta_j^f$  là phần tử thứ  $j$  trong vectơ hệ số  $\theta^f$  và  $t$  là phần tử trong vectơ lướt thời gian:

$$t = \frac{[0, 1, 2, \dots, H-2, H-1]^T}{H}$$

Cuối cùng, loại seasonality block dùng để nhận biết tính mùa của dữ liệu. Đặc điểm chính của tính mùa là dữ liệu thường

có tính chu kỳ. Do đó, để mô phỏng tính mùa, các hàm  $g^b$  và  $g^f$  sẽ là các loại hàm tuần hoàn, ví dụ  $y_t = y_{t+\delta}$  trong đó  $\delta$  là chu kỳ mùa. Sử dụng chuỗi Fourier để xây dựng các hàm tuần hoàn trên, ta có đầu ra forecast của seasonality block thứ  $i$  là:

$$y^i = \sum_{j=0}^{\lfloor \frac{H}{2} - 1 \rfloor} \theta_j^i \cos(2\pi i t) + \theta_{j+\lfloor \frac{H}{2} \rfloor}^i \sin(2\pi i t)$$

Sau khi giới thiệu ba loại block trên, mô hình N-BEATS được xây dựng dựa trên số lượng và thứ tự của mỗi loại block.

#### H. Gated Recurrent Units (GRU)

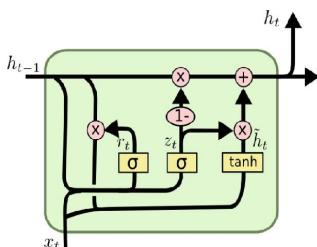
Gated Recurrent Units được giới thiệu bởi Kyunghyun Cho và các đồng nghiệp vào năm 2014 như một giải pháp cho vấn đề vanishing gradient. GRU sử dụng cơ chế cổng để kiểm soát luồng thông tin. Những cổng này xác định thông tin nào nên được chuyển đến đầu ra và thông tin nào nên tiếp tục được giữ lại trong trạng thái nội bộ của mạng, cho phép mô hình nắm bắt tốt hơn các phụ thuộc cho các chuỗi có độ dài đa dạng.

Kiến trúc GRU:

Cổng Cập Nhật: Cổng cập nhật giúp mô hình xác định bao nhiêu thông tin từ quá khứ (từ các bước thời gian trước) cần được truyền đến tương lai. Điều này rất quan trọng để mô hình có thể nắm bắt các phụ thuộc dài hạn và quyết định cái gì cần giữ trong bộ nhớ.

Cổng Đặt Lại: Cổng đặt lại quyết định bao nhiêu thông tin từ quá khứ cần được quên đi. Nó cho phép mô hình quyết định mức độ quan trọng của mỗi đầu vào đối với trạng thái hiện tại và hữu ích cho việc đưa ra dự đoán.

Các cổng này là các vector chứa giá trị từ 0 đến 1. Các giá trị này được tính bằng cách sử dụng hàm kích hoạt sigmoid. Giá trị gần với 0 có nghĩa là cổng đóng và không có thông tin nào được truyền qua, trong khi giá trị gần với 1 có nghĩa là cổng mở và tất cả thông tin được truyền qua.



**Fig7.** Sơ đồ luồng kiến trúc của GRU.

Các phương trình được sử dụng để tính toán cổng đặt lại, cổng cập nhật và trạng thái ẩn của một GRU như sau:

$$\text{Reset gate: } r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

$$\text{Update gate: } z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

$$\text{Candidate hidden state: } \hat{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h)$$

$$\text{Hidden state: } h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t$$

Trong :

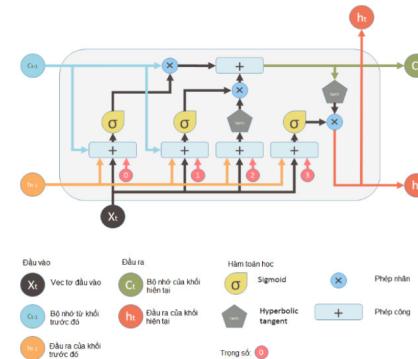
- $h_t$  là vector lớp ẩn.

- $x_t$  là vector đầu vào.

- $b_z, b_r, b_h$  là những bias vector.
- $W_z, W_r, W_h$  là các ma trận tham số.
- $\sigma, \tanh$  là những hàm kích hoạt.

#### I. LONG SHORT-TERM MEMORY

Mô hình đề xuất dựa trên mô hình mạng thần kinh sâu LSTM, đây là một dạng đặc biệt của RNN (Recurrent Neural Network - Mạng thần kinh hồi quy). LSTM được giới thiệu bởi Hochreiter và Schmidhuber (1997) nhằm giải quyết các bài toán về phụ thuộc xa (long-term dependency)



**Fig8.** Cấu trúc của mô hình LSTM.

Theo Olah (2015), mọi mạng hồi quy đều có dạng là một chuỗi các mô đun lặp đi lặp lại của một mạng thần kinh, mỗi mô đun này thường có cấu trúc đơn giản được gọi là một tầng “tanh”. LSTM cũng có kiến trúc dạng chuỗi như vậy và thay vì chỉ có 1 tầng mạng thần kinh như RNN chuẩn thì chúng có tới 4 tầng và tương tác với nhau một cách đặc biệt. Cấu trúc của mô hình mạng thần kinh LSTM được thể hiện ở Hình 1. Cốt lõi của LSTM bao gồm trạng thái tế bào (cell state) và cổng (gate). Trạng thái tế bào giống như băng chuyền, chạy xuyên suốt qua tất cả các nút mạng giúp thông tin được truyền đạt dễ dàng, còn cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid. Một LSTM gồm có 3 cổng để duy trì hoạt động trạng thái của tế bào.

Bước đầu tiên của mô hình LSTM được gọi là tầng cổng quên (forget gate layer). Bước này sẽ quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Đầu vào cho bước này là  $h_{t-1}$  (giá trị đầu ra tại thời điểm  $t-1$ ) và  $x_t$  (dữ liệu đầu vào); đầu ra  $f_t$  là một số trong khoảng từ 0 đến 1 cho mỗi số trong trạng thái tế bào  $C_{t-1}$ .

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

Trong đó:  $\sigma$  là hàm sigmoid,  $W_f$  và  $b_f$  lần lượt là trọng số và tham số của tầng cổng quên.

Các bước tiếp theo sẽ quyết định thông tin lưu vào trạng thái tế bào và cập nhật giá trị cho trạng thái. Bao gồm một tầng sigmoid hay còn được gọi là cổng vào (input gate layer,  $i_t$ ) và một véc tơ giá trị được tạo từ tầng tanh.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \hat{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * \hat{C}_t \end{aligned}$$

Trong đó:  $C_{t-1}$  và  $C_t$  là trạng thái tế bào lần lượt ở thời điểm  $t - 1$  and  $t$ ;  $W_C$  và  $b_C$  lần lượt là trọng số và tham số của trạng thái tế bào.

Ở bước cuối cùng, giá trị đầu ra ( $h_t$ ) sẽ được quyết định bởi trạng thái của tế bào muốn xuất ra (output gate,  $o_t$ ).

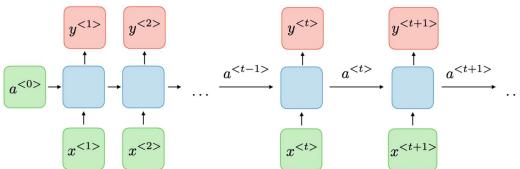
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

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

### J. RNN - Recurrent Neural Network

RNN là một loại kiến trúc mạng nơ-ron trong lĩnh vực trí tuệ nhân tạo AI và học máy. RNN được thiết kế để xử lý dữ liệu chuỗi, nơi thông tin từ các bước thời gian trước đó được giữ lại để ảnh hưởng đến các bước thời gian sau.

Gọi  $t$  là thời điểm xét,  $t - 1$  là thời điểm quá khứ sau một đơn vị so với  $t$ :

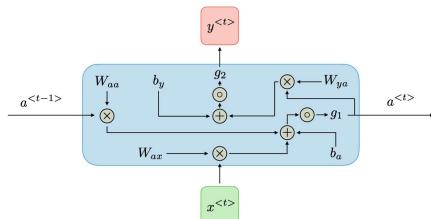


**Fig9.** Mô hình RNN đơn giản

$a^{<t>}$  là trạng thái thời điểm  $t$ .

$y^{<t>}$  là giá trị output thời điểm  $t$ .

$x^{<t>}$  là giá trị input thời điểm  $t$ .



**Fig10.** Mô hình RNN đơn giản

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

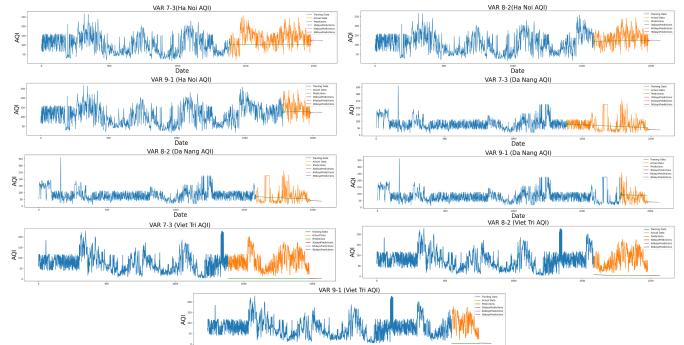
$$\tilde{y}_t = y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Với  $W_{ax}$ ,  $W_{aa}$ ,  $W_{ya}$  là các trọng số, thường là random,  $g_1$ ,  $g_2$  là các hàm kích hoạt,  $b_a$ ,  $b_y$  là các bias.

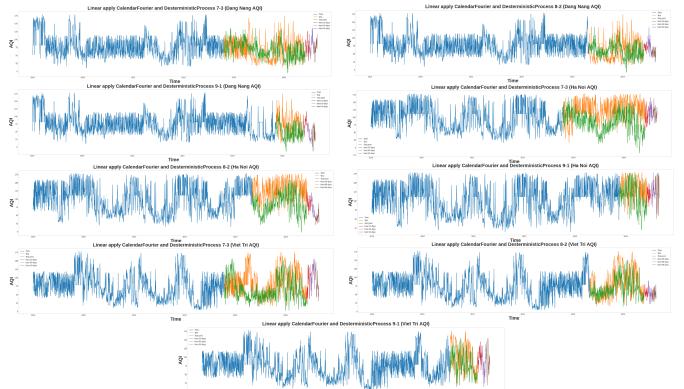
## V. KẾT QUẢ

### A. THIẾT LẬP MÔ HÌNH

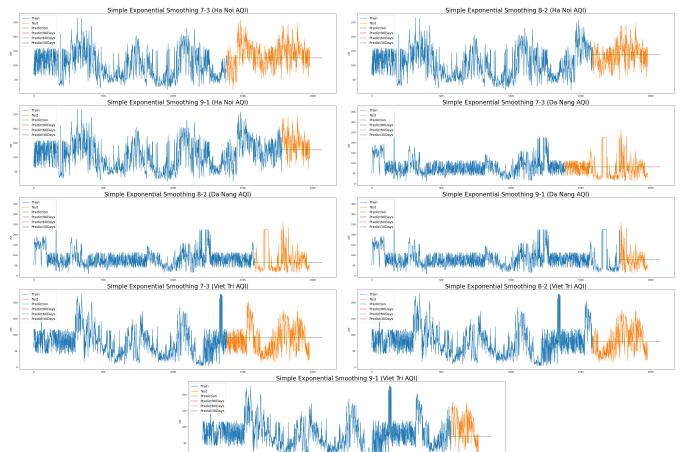
#### 1) VAR



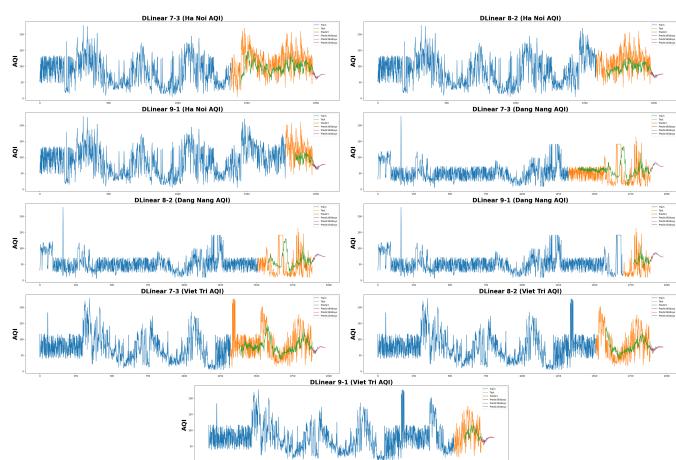
#### 2) LINEAR REGRESSION APPLY CALENDARFOURIER. DETERMINISTICPROCESS



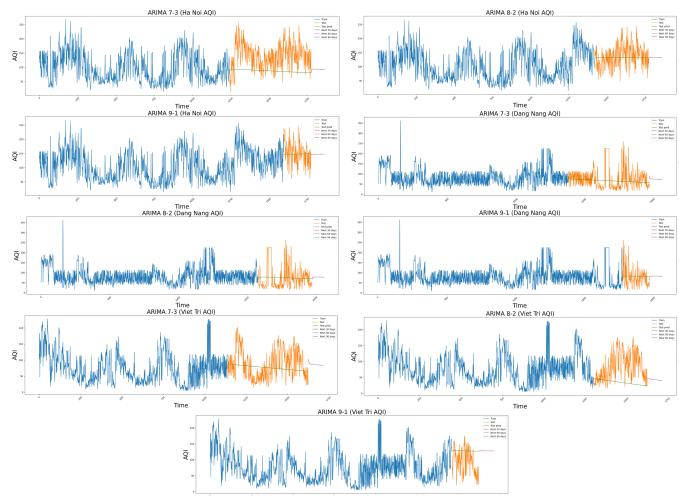
#### 3) SES



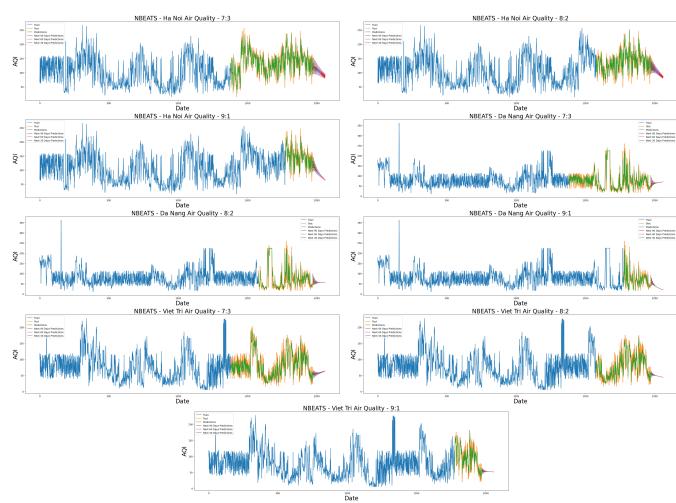
#### 4) DLINEAR



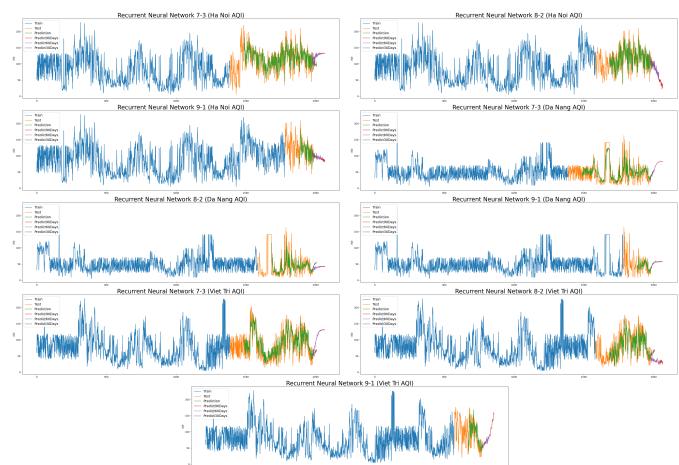
#### 7) ARIMA



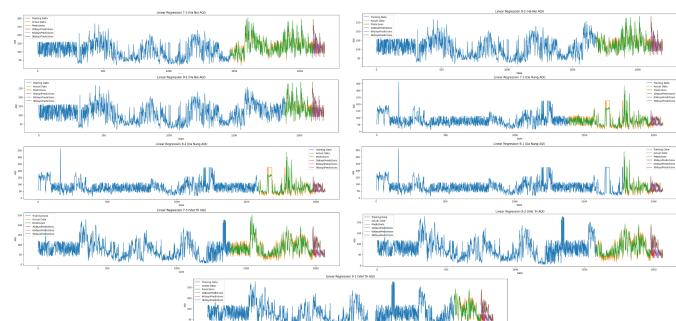
#### 5) NBEATS



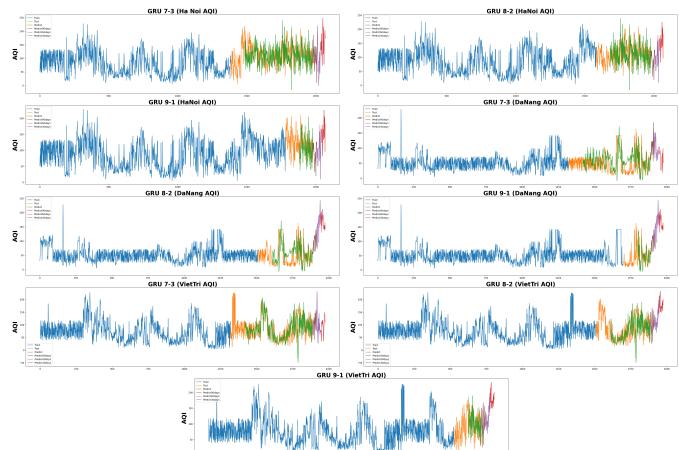
#### 8) RNN



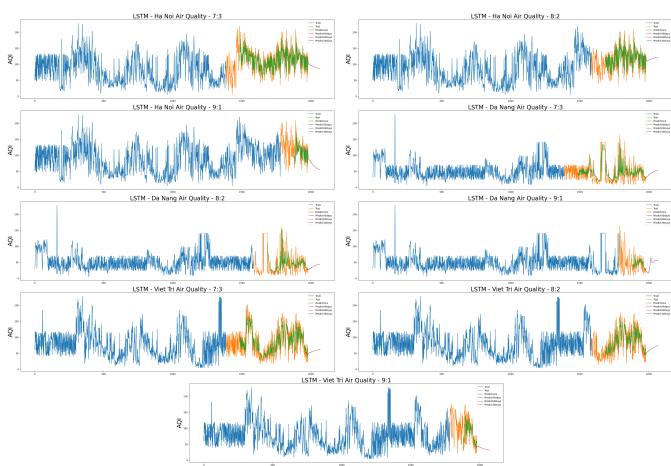
#### 6) LINEAR REGRESSION



#### 9) GRU



## 10) LSTM



## B. ĐÁNH GIÁ MÔ HÌNH

Bảng IV: ĐÁNH GIÁ CÁC MÔ HÌNH TRÊN DATASET AQI ĐÀ NẴNG

Model	Ratio	RMSE	MAE	MAPE (%)
LinearRegression	7:3	17.85	11.54	18.43
	8:2	21.68	14.06	23.77
	<b>9:1</b>	<b>20.33</b>	<b>10.25</b>	<b>14.64</b>
ARIMA	<b>7:3</b>	<b>52.04</b>	<b>36.12</b>	<b>57.96</b>
	8:2	59.32	45.17	88.03
	<b>9:1</b>	44.37	34.99	74.22
RNN	7:3	23.04	16.57	49.52
	8:2	24.90	17.86	61.15
	<b>9:1</b>	<b>16.50</b>	<b>12.84</b>	<b>44.35</b>
LSTM	7:3	23.57	17.09	55.26
	8:2	26.83	19.18	67.74
	<b>9:1</b>	<b>16.62</b>	<b>12.99</b>	<b>46.38</b>
GRU	7:3	37.09	29.20	99.73
	8:2	41.10	36.48	123.78
	<b>9:1</b>	<b>32.08</b>	<b>24.31</b>	<b>77.66</b>
VAR	7:3	<b>52.28</b>	<b>36.08</b>	<b>57.09</b>
	8:2	61.21	42.81	69.05
	<b>9:1</b>	45.88	37.19	83.90
DLINEAR	7:3	<b>41.17</b>	<b>30.42</b>	<b>107.03</b>
	8:2	49.78	40.45	152.76
	<b>9:1</b>	39.03	31.56	123.40
LN APPLY CF DP	7:3	39.8	32.84	66.39
	8:2	41.56	35.04	78.52
	<b>9:1</b>	<b>41.56</b>	<b>39.03</b>	<b>65.54</b>
NBEAST	7:3	35.60	24.07	42.58
	8:2	39.24	25.36	48.45
	<b>9:1</b>	<b>40.08</b>	<b>28.11</b>	<b>45.34</b>
SES	7:3	51.27	39.14	77.31
	8:2	60.40	42.36	72.00
	<b>9:1</b>	<b>44.00</b>	<b>34.06</b>	<b>70.00</b>

Bảng III: ĐÁNH GIÁ CÁC MÔ HÌNH TRÊN DATASET AQI HÀ NỘI

Model	Ratio	RMSE	MAE	MAPE (%)
LinearRegression	7:3	13.74	10.45	7.65
	<b>8:2</b>	<b>12.63</b>	<b>10.57</b>	<b>7.46</b>
	<b>9:1</b>	14.53	11.92	7.89
ARIMA	7:3	69.16	58.91	39.62
	8:2	38.67	31.65	25.40
	<b>9:1</b>	<b>38.24</b>	<b>30.16</b>	<b>22.64</b>
RNN	7:3	27.00	20.97	21.29
	8:2	30.63	24.30	23.79
	<b>9:1</b>	<b>26.09</b>	<b>20.66</b>	<b>19.84</b>
LSTM	7:3	26.61	20.60	20.20
	8:2	29.34	22.87	23.07
	<b>9:1</b>	<b>26.70</b>	<b>21.03</b>	<b>19.44</b>
GRU	<b>7:3</b>	<b>42.62</b>	<b>33.77</b>	<b>31.32</b>
	8:2	45.49	36.48	33.85
	<b>9:1</b>	42.79	35.92	32.67
VAR	<b>7:3</b>	<b>135.21</b>	<b>128.06</b>	<b>93.93</b>
	8:2	136.55	131.46	94.37
	<b>9:1</b>	146.71	141.73	95.21
DLINEAR	7:3	42.81	34.24	29.02
	8:2	38.75	31.65	27.34
	<b>9:1</b>	<b>38.12</b>	<b>30.36</b>	<b>24.91</b>
LN APPLY CF DP	7:3	62.9	52.58	44.94
	8:2	61.15	51.45	39.23
	<b>9:1</b>	<b>51.85</b>	<b>41.42</b>	<b>31.19</b>
NBEAST	7:3	31.26	24.38	20.77
	8:2	32.28	24.87	20.48
	<b>9:1</b>	<b>34.81</b>	<b>26.20</b>	<b>19.59</b>
SES	7:3	44.81	36.61	32.00
	8:2	37.94	31.23	26.31
	<b>9:1</b>	<b>45.43</b>	<b>37.41</b>	<b>24.85</b>

Bảng V: ĐÁNH GIÁ CÁC MÔ HÌNH TRÊN DATASET AQI VIỆT TRÌ

Model	Ratio	RMSE	MAE	MAPE (%)
LinearRegression	7:3	8.73	6.86	10.75
	8:2	8.02	7.04	13.06
	<b>9:1</b>	<b>7.81</b>	<b>6.90</b>	<b>8.39</b>
ARIMA	<b>7:3</b>	<b>45.42</b>	<b>37.29</b>	<b>55.33</b>
	8:2	65.86	52.51	55.20
	<b>9:1</b>	50.67	41.02	76.39
RNN	7:3	29.36	23.50	38.35
	8:2	30.16	24.28	38.36
	<b>9:1</b>	<b>33.21</b>	<b>26.50</b>	<b>39.00</b>
LSTM	7:3	27.97	22.33	34.86
	8:2	29.02	23.62	36.91
	<b>9:1</b>	<b>28.97</b>	<b>22.76</b>	<b>32.70</b>
GRU	<b>7:3</b>	<b>34.55</b>	<b>22.26</b>	<b>34.98</b>
	8:2	42.91	29.13	47.93
	<b>9:1</b>	40.72	31.61	42.71
VAR	7:3	92.34	82.76	96.16
	<b>8:2</b>	<b>85.50</b>	<b>75.23</b>	<b>91.29</b>
	<b>9:1</b>	103.90	96.03	96.49
DLINEAR	7:3	38.33	30.09	45.11
	8:2	36.26	29.23	52.82
	<b>9:1</b>	<b>40.65</b>	<b>33.19</b>	<b>40.16</b>
LN APPLY CF DP	7:3	51.75	40.23	49.5
	8:2	51.75	32.933	47.07
	<b>9:1</b>	<b>50.05</b>	<b>40.53</b>	<b>45.99</b>
NBEAST	7:3	28.95	22.73	33.43
	8:2	28.03	21.81	35.09
	<b>9:1</b>	<b>30.23</b>	<b>23.33</b>	<b>31.20</b>
SES	7:3	41.34	34.80	60.82
	8:2	40.12	33.96	59.50
	<b>9:1</b>	<b>48.50</b>	<b>40.11</b>	<b>42.59</b>

**Nhận xét:** Dựa vào các chỉ số đánh giá độ chính xác trên từng mô hình và trên cả 3 bộ dữ liệu AQI Hà Nội, Đà Nẵng, Việt Trì, ta thấy những mô hình có độ chính xác cao nhất là LinearRegression, LSTM và NBEATS. Các mô hình này hoạt động tốt trên bộ dữ liệu AQI Hà Nội, thích hợp để dự đoán AQI của các tỉnh thành phố trong tương lai.

## VI. KẾT LUẬN

### A. KẾT LUẬN TỔNG QUAN

Chất lượng không khí là một yếu tố quan trọng ảnh hưởng đến sức khỏe con người và môi trường, đòi hỏi các phương pháp dự báo chính xác để giúp các cơ quan quản lý và cộng đồng có thể đưa ra các biện pháp ứng phó kịp thời. Nghiên cứu của chúng tôi đã chứng minh rằng việc sử dụng các mô hình thống kê, học máy và học sâu có thể cải thiện đáng kể độ chính xác của dự báo chất lượng không khí.

Trong quá trình nghiên cứu, nhóm đã triển khai và so sánh nhiều mô hình khác nhau, bao gồm các phương pháp thống kê truyền thống, các thuật toán học máy như Linear Regression, cùng với các mạng nơ-ron sâu như LSTM và GRU,... Kết quả cho thấy các mô hình học sâu thể hiện độ chính xác vượt trội, đặc biệt việc xử lý các dữ liệu thời gian, dự báo các biến động phức tạp của chất lượng không khí.

Mặc dù các mô hình học sâu đã chứng minh được tiềm năng đáng kể trong việc dự báo chất lượng không khí, vẫn còn nhiều yếu tố bên ngoài như thay đổi thời tiết, phát thải từ các hoạt động công nghiệp, và các biến động trong lối sống con người có thể ảnh hưởng đến kết quả dự báo. Vì vậy, cần có thêm các nghiên cứu để tích hợp các yếu tố này vào mô hình dự báo nhằm cải thiện độ chính xác và độ tin cậy của dự báo.

### B. NHỮNG KHÓ KHĂN GẶP PHẢI

Trong quá trình thực hiện dự án, nhóm đã đối mặt với một số khó khăn:

**Bộ dữ liệu:** Nguồn thu thập dữ liệu về chất lượng không khí còn hạn chế dẫn đến chất lượng dữ liệu không được như mong muốn, còn nhiều thiếu sót, vì thế cần phải áp dụng nhiều phương pháp tiền xử lý dữ liệu dẫn đến bộ dữ liệu không còn độ chính xác cao.

**Tiền xử lý và xây dựng mô hình:** xây dựng mô hình về dự đoán chất lượng không khí đòi hỏi nhiều kiến thức chuyên môn về lĩnh vực này để có thể mang lại kết quả tốt nhất cho mô hình. Vì chưa có nhiều kinh nghiệm trong lĩnh vực này nên nhóm còn gặp đôi chút khó khăn trong việc lựa chọn các phương pháp tiền xử lý cũng như các tham số hợp lý để huấn luyện mô hình.

**Dánh giá hiệu suất mô hình:** nhóm đã áp dụng nhiều chỉ số cũng như các thuật toán để đánh giá mô hình nhưng kết quả cho thấy vẫn có nhiều mô hình không có độ chính xác cao, chưa đạt yêu cầu.

### C. DỰ ĐỊNH TƯƠNG LAI

Kế hoạch cho tương lai sẽ tập trung vào việc thăm dò và nghiên cứu các thuật toán mới khác ngoài các thuật toán hiện tại để áp dụng vào dự đoán chất lượng không khí trong dữ liệu time series. Mục tiêu là mở rộng kiến thức và hiểu biết về các phương pháp khác nhau, từ đó tăng cường khả năng áp dụng hiệu quả trong lĩnh vực này.

## LỜI CẢM ƠN

Nhóm chúng em xin gửi lời cảm ơn sâu sắc đến Phó Giáo sư Tiến sĩ Nguyễn Đình Thuân và Trợ giảng Nguyễn Minh Nhựt vì kiến thức chuyên môn vô giá, sự hướng dẫn tận tâm trong suốt thời gian thực hiện đồ án này. Đồ án nà không chỉ cung cấp cho nhóm chúng em cơ hội nâng cao kỹ năng làm việc nhóm, kỹ năng hợp tác, học hỏi lẫn nhau mà còn cho phép chúng em áp dụng kiến thức của mình vào các lĩnh vực thực tế.

Trong suốt quá trình thực hiện đồ án, nhóm chúng em đã tận dụng kiến thức đã có và nghiên cứu các khái niệm mới để mang lại kết quả tốt nhất. Mặc dù đã cố gắng hết sức, chúng em thừa nhận rằng không thể tránh khỏi những thiếu sót. Do đó, chúng em rất mong nhận được phản hồi, nhận xét của các thầy, điều này sẽ giúp nhóm chúng em hoàn thiện kiến thức và nâng cao các nỗ lực trong tương lai.

Ngoài ra, chúng em muôn bày tỏ lòng biết ơn đến các thành viên trong nhóm và bạn bè đã hỗ trợ, giúp đồ án được thực hiện thành công. Lời cảm ơn chân thành của chúng em xin gửi đến tất cả những người đã đồng hành cùng chúng em trong suốt thời gian hoàn thành đồ án này.

## Tài liệu

- [1] Jamil, Nur & Amit, Norani & Yusof, Noreha. (2020). Model Evaluation on Air Pollutant Index (API) in Petaling Jaya, Malaysia. 29. 1959-1966
- [2] Liu, Jialin. "Multi-Step-Ahead Wind Speed Forecast Method Based on Outlier Correction, Optimized Decomposition, and DLinear Model." MDPI, 17 6 2023, .
- [3] Kim, Taesung. "Missing Value Imputation of Time-Series Air-Quality Data via Deep Neural Networks." MDPI,
- [4] A. Loganathan, P. Sumithra, and V. Deneshkumar, "Estimation of Air Quality Index Using Multiple Linear Regression," Applied Ecology and Environmental Sciences, vol. 10, no. 12 (2022): 717-723. doi: 10.12691/aees-10-12-3.
- [5] Khaerun Nisa SH, Irfan Irfani, Utriweni 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)
- [6] Aarthi, P. Gayathri, N. R. Gomathi, et.al., (2020), "Air Quality Prediction Through Regression Model", International Journal of Scientific & Technology Research Vol 9, pp 923-928.
- [7] H. N. Mahendra , S. Mallikarjunaswamy, D. Mahesh Kumar, Shilpi Kumari, Shubhali Kashyap, Sapna Fulwani and Aishee Chatterjee\* (2022) Assessment and Prediction of Air Quality Level Using ARIMA Model: A Case Study of Surat City, Gujarat State, India
- [8] Athira Va , Geetha Pb , Vinayakumar Rab, Soman K P (2018, DeepAir-Net: Applying Recurrent Networks for Air Quality Prediction
- [9] Xinxing Zhou et al 2019 J. Phys, Air Pollutant Concentration Prediction Based on GRU Method, <https://iopscience.iop.org/article/10.1088/1742-6596/1168/3/032058>
- [10] Zhehua Du and Xin Lin 2020, Air Quality Prediction Based on Neural Network Model of Long Short-term Memory