BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC PHENIKAA

o0o

BÁO CÁO

HỌC PHẦN: Lập trình phân tích dữ liệu bằng python

TÌM HIỂU VỀ PHÂN TÍCH DỮ LIỆU CHUỖI THỜI GIAN

Họ tên	MSSV	Lớp
Nguyễn Dương Tuấn Nguyên	22010091	K16.KHMT-TN
Hồ Xuân Hùng	22010493	K16.KHMT-TN

GVHD: Ths. Nguyễn Ngọc Hùng

HÀ NỘI - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC PHENIKAA

o0o

BÁO CÁO

HỌC PHẦN: Lập trình phân tích dữ liệu bằng python

TÌM HIỂU VỀ PHÂN TÍCH DỮ LIỆU CHUỖI THỜI GIAN

GVHD: ThS. Nguyễn Ngọc Hùng

Họ tên	MSSV	Lớp
Nguyễn Dương Tuấn Nguyên	22010091	K16.KHMT-TN
Hồ Xuân Hùng	22010493	K16.KHMT-TN

HÀ NỘI - 2025

Mục lục

Danh sách hình ảnh

Chương 1

Giới thiệu

1.1 Nêu vấn đề

Trong những năm gần đây, chất lượng không khí đã trở thành một trong những mối quan tâm hàng đầu tại nhiều quốc gia trên thế giới, đặc biệt là tại các đô thị lớn. Một trong những chỉ số quan trọng để đánh giá mức độ ô nhiễm không khí là nồng độ bụi mịn PM2.5 – các hạt có đường kính nhỏ hơn hoặc bằng 2.5 micromet, có khả năng xâm nhập sâu vào hệ hô hấp và gây ảnh hưởng nghiêm trọng đến sức khỏe con người.

Việc theo dõi và dự báo xu hướng chỉ số PM2.5 theo thời gian là vô cùng cần thiết nhằm hỗ trợ các cơ quan chức năng đưa ra các cảnh báo sớm và chính sách kiểm soát ô nhiễm kịp thời. Để làm được điều này, cần áp dụng các phương pháp phân tích dữ liệu chuỗi thời gian (time-series analysis) nhằm mô hình hóa và dự đoán giá trị của PM2.5 trong tương lai.

1.2 Công cụ sử dụng

Đề tài sử dụng hai phương pháp chính trong phân tích chuỗi thời gian để dự đoán nồng độ PM2.5:

- **ARIMA (AutoRegressive Integrated Moving Average)**: là một trong những mô hình thống kê kinh điển và phổ biến nhất trong phân tích chuỗi thời gian. ARIMA phù hợp với dữ liệu có tính tuần hoàn, có thể biến đổi thành chuỗi dừng thông qua sai phân.
- **LSTM (Long Short-Term Memory)**: là một dạng mạng nơ-ron

hồi tiếp (Recurrent Neural Network - RNN), đặc biệt hiệu quả trong việc học và ghi nhớ thông tin chuỗi dài hạn. LSTM có khả năng xử lý tốt các chuỗi dữ liệu phức tạp và phi tuyến.

Trong quá trình thực nghiệm, các công cụ và thư viện chính của Python được sử dụng gồm:

- `pandas, numpy, matplotlib, seaborn`: xử lý dữ liệu và trực quan hóa.
- `statsmodels`: triển khai mô hình ARIMA.
- `scikit-learn`: chuẩn hóa dữ liệu và đánh giá mô hình.
- `TensorFlow/Keras`: xây dựng và huấn luyện mô hình LSTM.

Thông qua việc áp dụng hai hướng tiếp cận này, báo cáo hướng tới việc so sánh hiệu quả giữa mô hình truyền thống (ARIMA) và mô hình học sâu (LSTM) trong bài toán dự báo chỉ số ô nhiễm không khí PM2.5.

Chương 2

Cơ sở lý thuyết

2.1 Các khái niệm

Dữ liệu chuỗi thời gian: là tập hợp các quan sát được thu thập theo trình tự thời gian, thường tại các thời điểm cách đều nhau (ví dụ: ngày, tháng, năm).

Tính dừng (stationarity): Một chuỗi được gọi là dừng khi các đặc trưng thống kê như kỳ vọng, phương sai và hiệp phương sai không thay đổi theo thời gian. Tính dừng là điều kiện cần thiết để áp dụng nhiều mô hình thống kê trong phân tích chuỗi thời gian.

2.2 Các mô hình phổ biến

Các mô hình trong phân tích chuỗi thời gian thường được chia thành hai nhóm chính: **(1) Mô hình thống kê** và **(2) Mô hình học máy/học sâu**.

2.2.1 Mô hình thống kê

Mô hình **ARIMA (Auto-Regressive Integrated Moving Average)** là một dạng hồi quy tuyến tính mở rộng, giả định rằng chuỗi dữ liệu đầu vào là chuỗi dừng.

Mô hình ARIMA gồm ba thành phần:

- **Auto Regression (AR):** mô hình hóa giá trị hiện tại dựa trên các

giá trị trong quá khứ. AR bậc p được biểu diễn như sau:

$$AR(p) = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t \tag{2.2.1}$$

với $w_t \sim WN(0, \sigma_w^2)$ và ϕ_i là các hệ số mô hình.

- **Moving Average (MA):** mô hình hóa sai số dự báo dựa trên các sai số trong quá khứ. MA bậc q được biểu diễn như sau:

$$MA(q) = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}?? \tag{2.2.2}$$

với $w_t \sim WN(0, \sigma_w^2)$ và θ_j là các hệ số mô hình.

- **Integrated (I):** là quá trình lấy sai phân để biến chuỗi không dừng thành chuỗi dừng. Ví dụ:

$$\text{Bậc 1: } I(1) = \Delta x_t = x_t - x_{t-1} \tag{2.2.3}$$

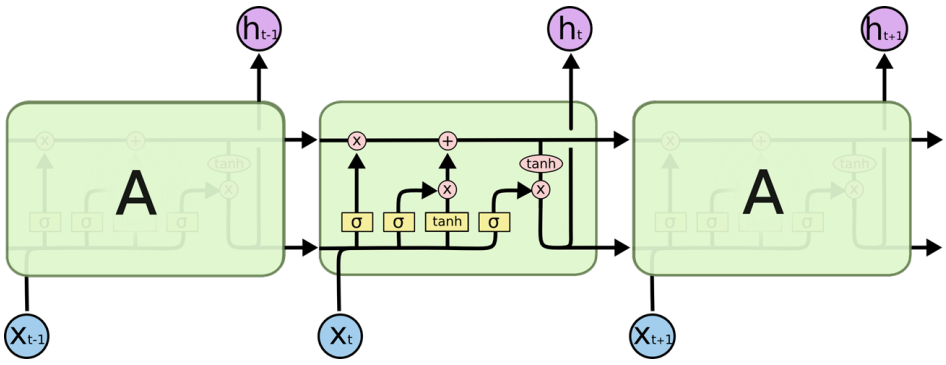
$$\text{Bậc d: } I(d) = \Delta^d x_t = \underbrace{\Delta(\Delta(\ldots\Delta(x_t)))}_{d \text{ lần}} \tag{2.2.4}$$

Phương trình tổng quát của mô hình ARIMA(p, d, q) là:

$$\Delta^d x_t = \phi_1 \Delta^d x_{t-1} + \cdots + \phi_p \Delta^d x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q} \tag{2.2.5}$$

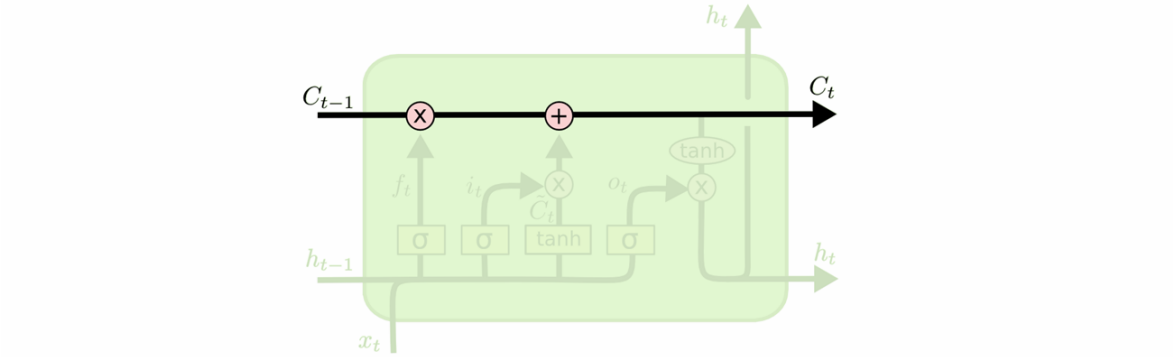
2.2.2 Mô hình Long Short Term Memory (LSTM)

Mạng nơ-ron **Long Short-Term Memory (LSTM)** là một biến thể của mạng nơ-ron hồi tiếp (RNN), được thiết kế để khắc phục hiện tượng mất dần gradient khi huấn luyện chuỗi dài. Cấu trúc của LSTM vẫn có dạng chuỗi, nhưng bên trong có thêm các cơ chế điều khiển thông tin.**??**



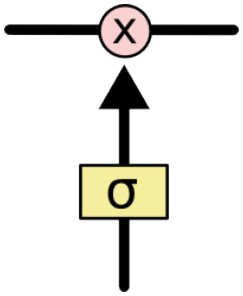
Hình 2.1: Cấu trúc chuỗi trong LSTM

Trong hình 2.1, các hình tròn màu hồng biểu diễn các phép toán (như cộng, nhân vô hướng), còn các khối màu vàng biểu thị hàm kích hoạt phi tuyến (sigmoid, tanh). Hai đường nhập thể hiện phép gộp kết quả; hai đường phân nhánh biểu thị việc sao chép thông tin truyền đến các phần khác trong mạng.



Hình 2.2: Đường đi của ô trạng thái trong LSTM

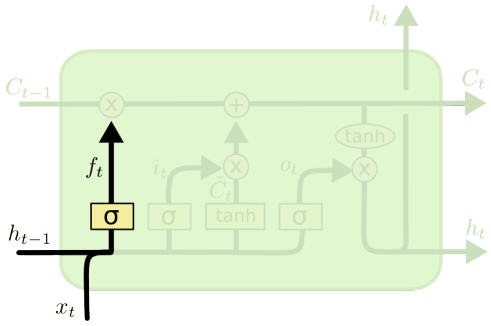
Ô trạng thái (cell state) là đường truyền nằm ngang xuyên suốt chuỗi thời gian. Nó hoạt động như một "băng chuyền", giúp thông tin được truyền ổn định. LSTM điều chỉnh dòng thông tin bằng các **cổng (gates)** – bao gồm hàm sigmoid và phép nhân.



Hình 2.3: Một cổng điều khiển thông tin

Hàm sigmoid đưa ra giá trị từ 0 đến 1, biểu thị mức độ thông tin được phép đi qua.

Bước 1 – Tầng quên (Forget Gate): quyết định giữ lại bao nhiêu thông tin từ trạng thái cũ C_{t-1}.



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

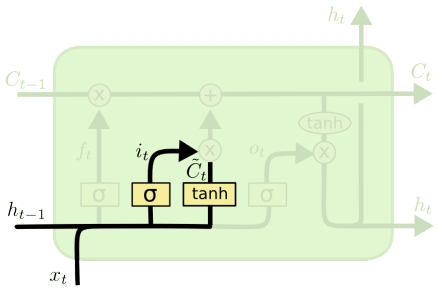
Hình 2.4: Tầng quên – kiểm soát thông tin bị loại bỏ

Bước 2 – Tầng vào (Input Gate): xác định thông tin mới nào được thêm vào trạng thái. Bao gồm:

- Tầng sigmoid sinh ra i_t
- Tầng tanh sinh ra $\overline{C}_t$

Cập nhật trạng thái:

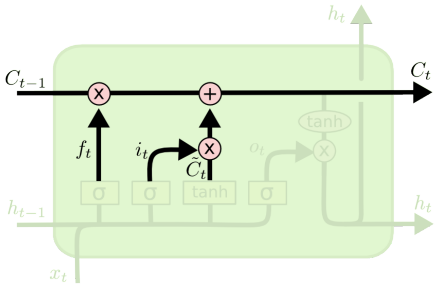
$$C_t = f_t * C_{t-1} + i_t * \overline{C}_t$$



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 2.5: Cập nhật trạng thái mới

Bước 3 – Tính toán ô trạng thái mới:

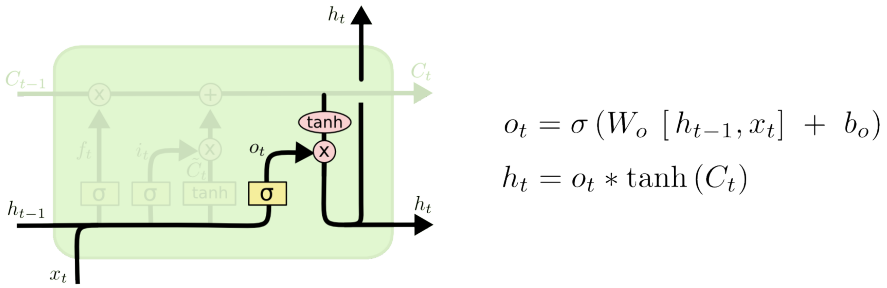


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Hình 2.6: Tính toán ô trạng thái mới

Bước 4 – Tính toán đầu ra: Đầu ra h_t được tính từ trạng thái mới thông qua một tầng sigmoid và hàm tanh:

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



Hình 2.7: Tầng đầu ra của LSTM

Chương 3

Thực nghiệm và kết quả

3.1 Thực nghiệm

3.1.1 Chuẩn bị dữ liệu

Dữ liệu được sử dụng trong bài toán được lấy từ trang `Kaggle`, cụ thể là bộ dữ liệu có tên **Air Quality Data in India** truy cập tại địa chỉ: https://www.kaggle.com/datasets/fedesoriano/air-quality-data-in-india. Bộ dữ liệu chứa thông tin về thời gian và chỉ số PM2.5 đánh giá mức độ bụi mịn trong không khí tại Ấn Độ.

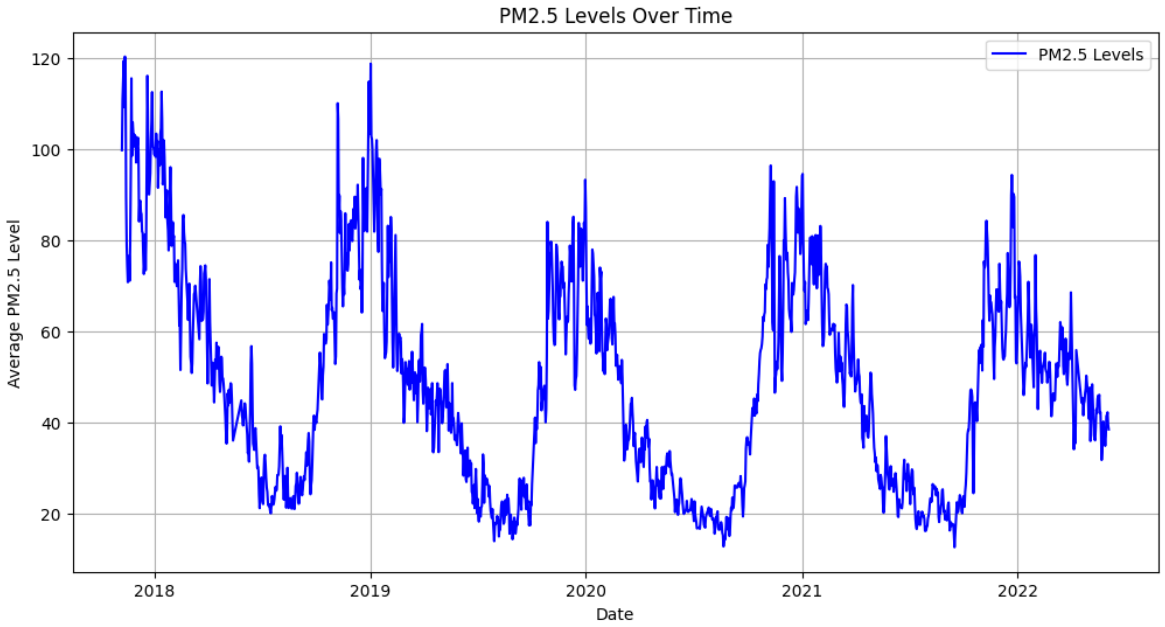
Trước khi có thể sử dụng thì cần phải thực hiện các bước tiền xử lý nhằm giúp đảm bảo chất lượng dữ liệu đầu vào cho các mô hình dự báo và giảm thiểu sai số do dữ liệu nhiễu hoặc thiếu với quy trình như sau:

- **Lọc dữ liệu:** Chỉ giữ lại các cột liên quan đến ngày tháng và chỉ số PM2.5 đồng thời gộp dữ liệu theo ngày(dữ liệu gốc có cả thời gian từng ngày).
- **Chuyển đổi định dạng thời gian:** Cột thời gian được chuyển đổi về kiểu `datetime` để thuận tiện cho việc xử lý chuỗi thời gian.
- **Kiểm tra dữ liệu thiếu:** Dữ liệu được kiểm tra xem có giá trị bị thiếu (`NaN`, `null`) trong cột PM2.5 hay không. Kết quả kiểm tra cho thấy không tồn tại giá trị bị thiếu trong dữ liệu được sử dụng cho thực nghiệm.
- **Sắp xếp dữ liệu theo thời gian:** Đảm bảo thứ tự thời gian từ cũ đến mới để mô hình hóa chuỗi thời gian chính xác.

3.1.2 Trực quan hóa dữ liệu

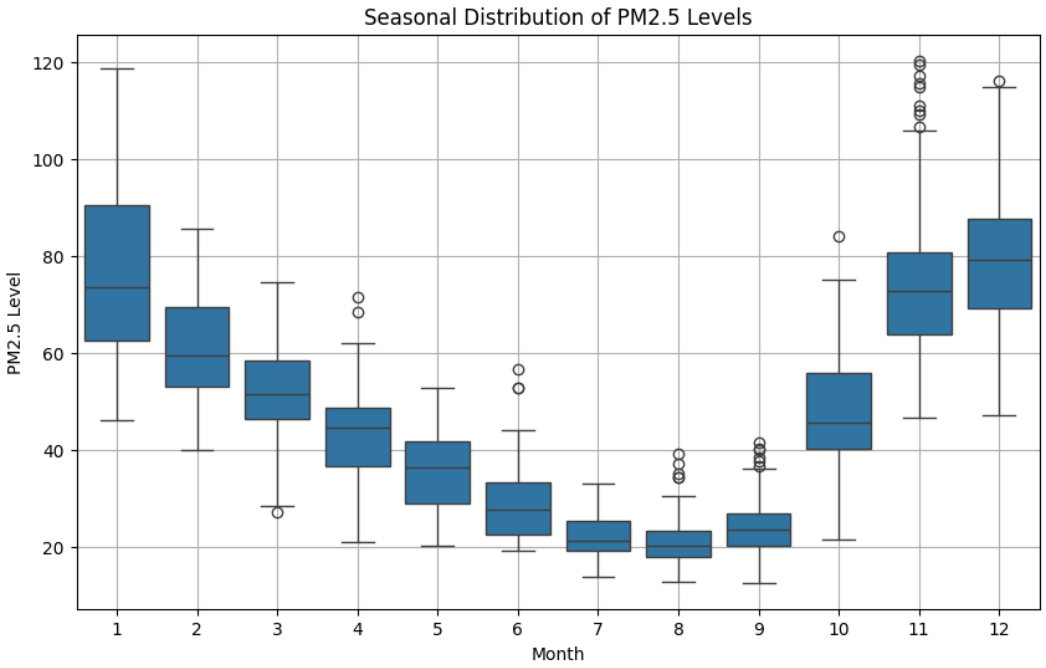
Trong phần này, bằng cách sử dụng các kĩ thuật trực quan hóa dữ liệu, ta có thể rút ra những tri thức cần thiết của dữ liệu. Qua đó, có thể đưa ra một số nhận xét trước khi đưa vào mô hình.

Tổng quan dữ liệu cho thấy một xu hướng mùa vụ qua các năm cùng với sự nhiễu trong dũ liệu, giả thuyết đặt ra là dữ liệu là chuỗi không dừng(ta sẽ kiểm chứng điều này sau).

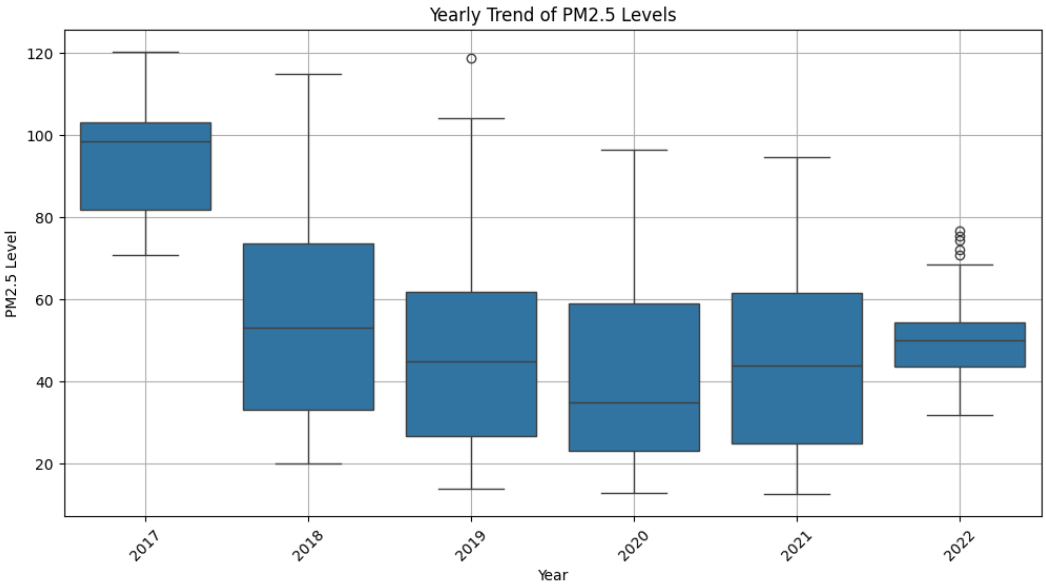


Hình 3.1: Tổng quan dữ liệu

Ta xem xét kĩ hơn tính chất theo chu kỳ của dữ liệu qua biểu đồ hộp(box plot) theo tháng và theo năm, nhận thấy có sự suy giảm về chỉ số PM2.5 theo tháng(thấp nhất vào các tháng hè như 7,8 và 9) và suy giảm theo các năm từ 2018 trở về sau.

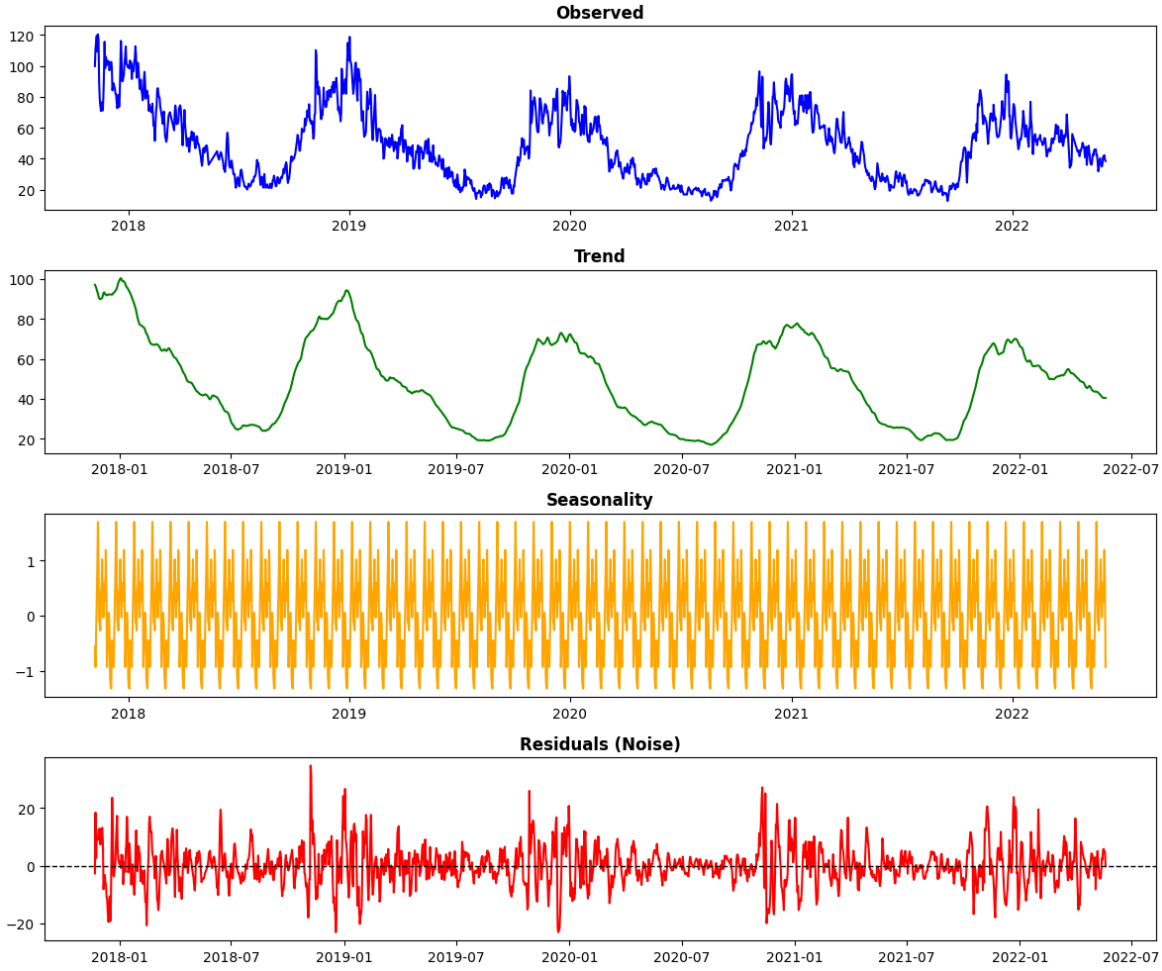


Hình 3.2: Biểu đồ hộp thể hiện tính mùa vụ theo tháng của dữ liệu



Hình 3.3: Biểu đồ hộp thể hiện tính mùa vụ theo năm của dữ liệu

Ta tiếp tục trực quan các thành phần của dữ liệu chuỗi thời gian như xu hướng(trend), tính mùa vụ(seasonality) hoặc tính chu kỳ(cyclic) và nhiễu(noise). Việc phân tách dữ liệu chuỗi thời gian(decomposition) giúp ta có thể hiểu rõ dữ liệu hơn phục vụ cho việc phân tích.

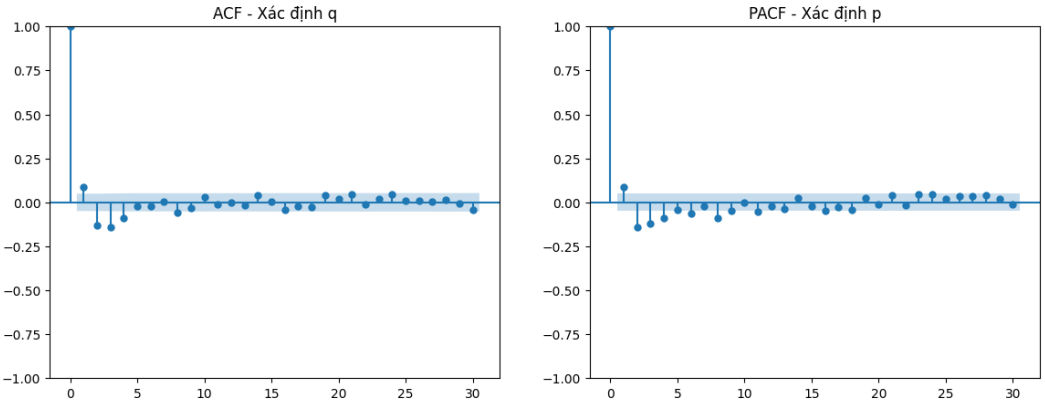


Hình 3.4: Phân rã các thành phần của dữ liệu(decomposition)

Nhận thấy một cách trực quan dữ liệu có tính chất chu kỳ(cyclic) và mùa vụ(seasonality) cao, các nhiễu(noise) là các whitenoise theo phân phối nào đó.

3.1.3 Xây dựng mô hình

Đối với mô hình ARIMA ta cần phải xem xét hệ số tự tương quan(ACF) và hệ số tự tương quan từng phần(PACF) để lựa chọn bậc của p và q đồng thời phải kiểm tra chuỗi dừng hay không trước và sau khi lấy sai phân để lựa chọn bậc của d. Ta xác định p và q lần lượt thông qua biểu đồ của ACF và PACF. Để xác định hệ số d ta kiểm tra tính dừng của chuỗi ban đầu và kiểm tra tính dừng của chuỗi sau khi lấy sai phần bậc d(thông thường sẽ lấy là 1) thông qua kiểm tra **ADF**.



Hình 3.5: Biểu đồ của ACF và PACF

```
ADF Statistic: -8.6604
p-value: 0.0000
Critical Values:
   1%: -3.4345
   5%: -2.8634
   10%: -2.5677
Chuỗi có tính dừng.
```

Hình 3.6: Kiểm tra tính dừng sau khi lấy sai phân bậc 1

Qua biểu đồ và kiểm tra ADF ta xác định được $p = 7, \quad d = 1, \quad q = 4$. Do đó mô hình **ARIMA**$(7, 1, 4)$ Đối với mô hình LSTM, nhóm lựa sử dụng Tensorflow thiết lập như sau:

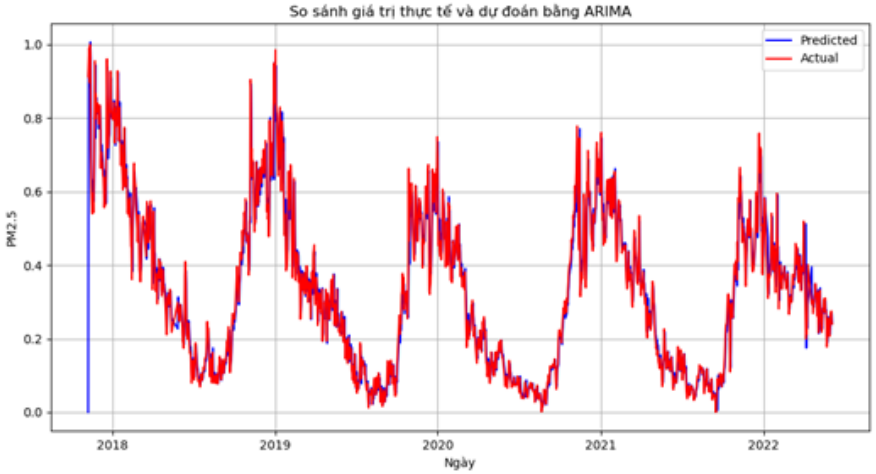
```
model = Sequential([
    LSTM(units=50, return_sequences=True, input_shape=(time_steps, 1)),
    Dropout(0.2),
    LSTM(units=50, return_sequences=False),
    Dropout(0.2),
    Dense(units=25),
    Dense(units=1)
])
```

Hình 3.7: Thiết lập mô hình LSTM

3.2 Kết quả thảo luận

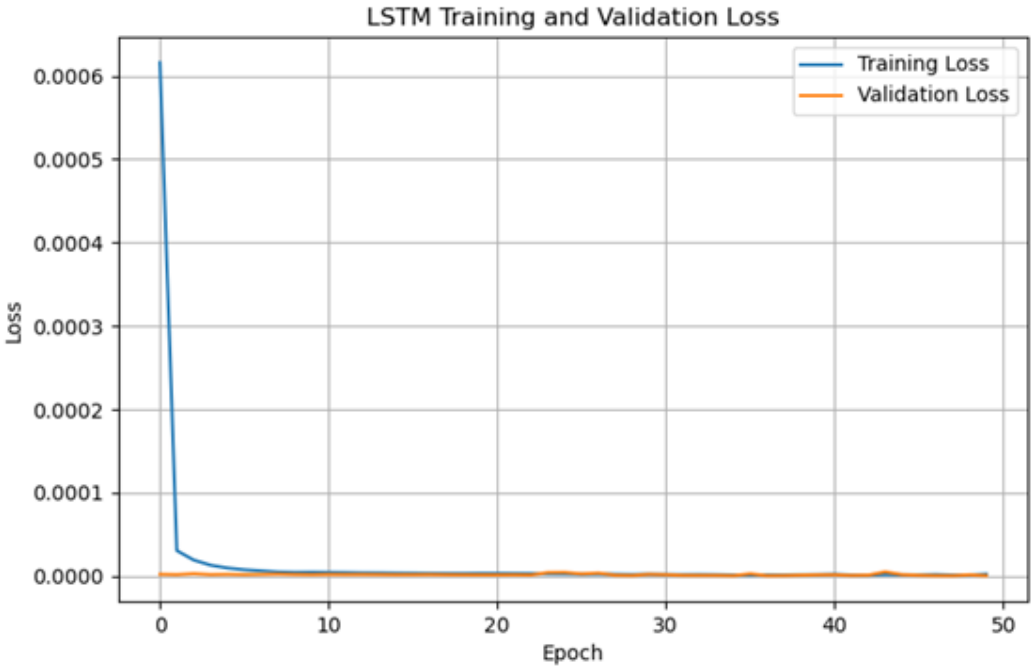
Kết quả sau khi áp dụng mô hình ARIMA cho thấy mô hình dự đoán quá khớp với dữ liệu, điều này chưa hẳn là tín hiệu tốt khi mô hình có thể

không đưa ra dự đoán ổn định với dữ liệu có tính biến động cao hơn.

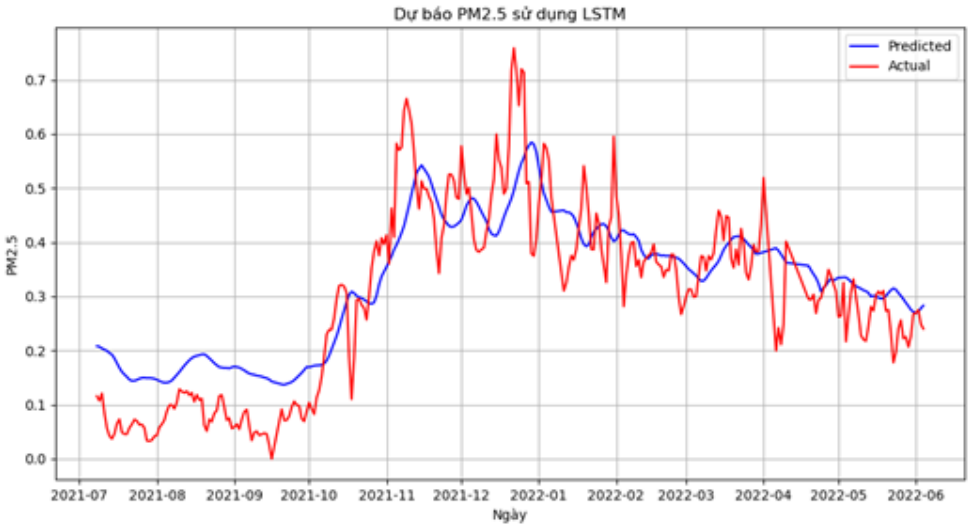


Hình 3.8: Dự đoán của mô hình ARIMA với dữ liệu

Xem xét kết quả của quá trình huấn luyện và dự đoán của mô hình LSTM cho thấy mô hình hoạt động tương đối ổn định khi đã có thể dự đoán tương đối xu hướng của dữ liệu và những sự dao động của dữ liệu.



Hình 3.9: Biểu đồ hàm mất mát của quá trình huấn luyện mô hình LSTM



Hình 3.10: Dự đoán của mô hình LSTM

Tài liệu tham khảo

[1] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Springer, New York, 4th edition, 2017.

[2] Phạm Đình Khánh. Lý thuyết về mạng lstm, April 2019. Accessed: 2025-07-09.