

# Phân tích và dự đoán đường chim bay

---

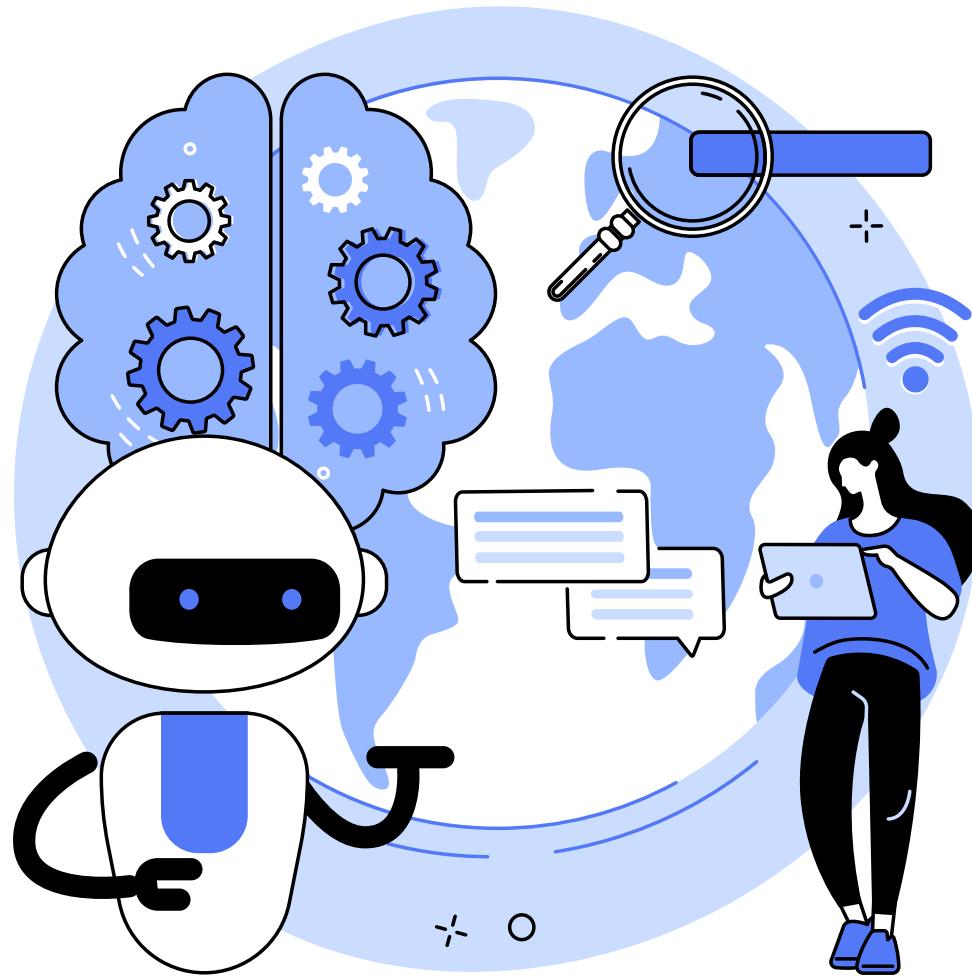
Khai thác dữ liệu và ứng dụng - CS313.Q12

GVHD: TS. Võ Nguyễn Lê Duy

# Thành viên nhóm 12

Họ và tên	MSSV	Công việc	Đóng góp
Trần Minh Vũ	23521819	Crawl data, EDA, Preprocessing, Slide	25%
Đinh Trần Duy Trưởng	23521688	Modeling, Testing, Demo, Slide	25%
Nguyễn Thanh Tuấn	23521724	Modeling, Testing, Slide	25%
Nguyễn Đăng Khoa	24520820	EDA, Preprocessing, Slide, Presentation	25%

# CONTENTS



- 1 INTRODUCTION
- 2 DATA & EDA
- 3 PREPROCESSING
- 4 MODELING
- 5 RESULT



1

# INTRODUCTION

# Introduction

## 1. Bối cảnh.

- Hành vi di cư của các loài chim là một trong những hiện tượng tự nhiên kỳ thú và phức tạp nhất. Việc hiểu và dự đoán được đường bay của chúng không chỉ mang lại giá trị khoa học mà còn có nhiều ứng dụng thực tiễn quan trọng.

## 2. Mục tiêu.

- Dự đoán quỹ đạo di cư của chim trong tương lai gần

## 3. Ý nghĩa bài toán.

- Bảo tồn sinh học
- Hỗ trợ chống săn bắt trái phép
- An toàn hàng không

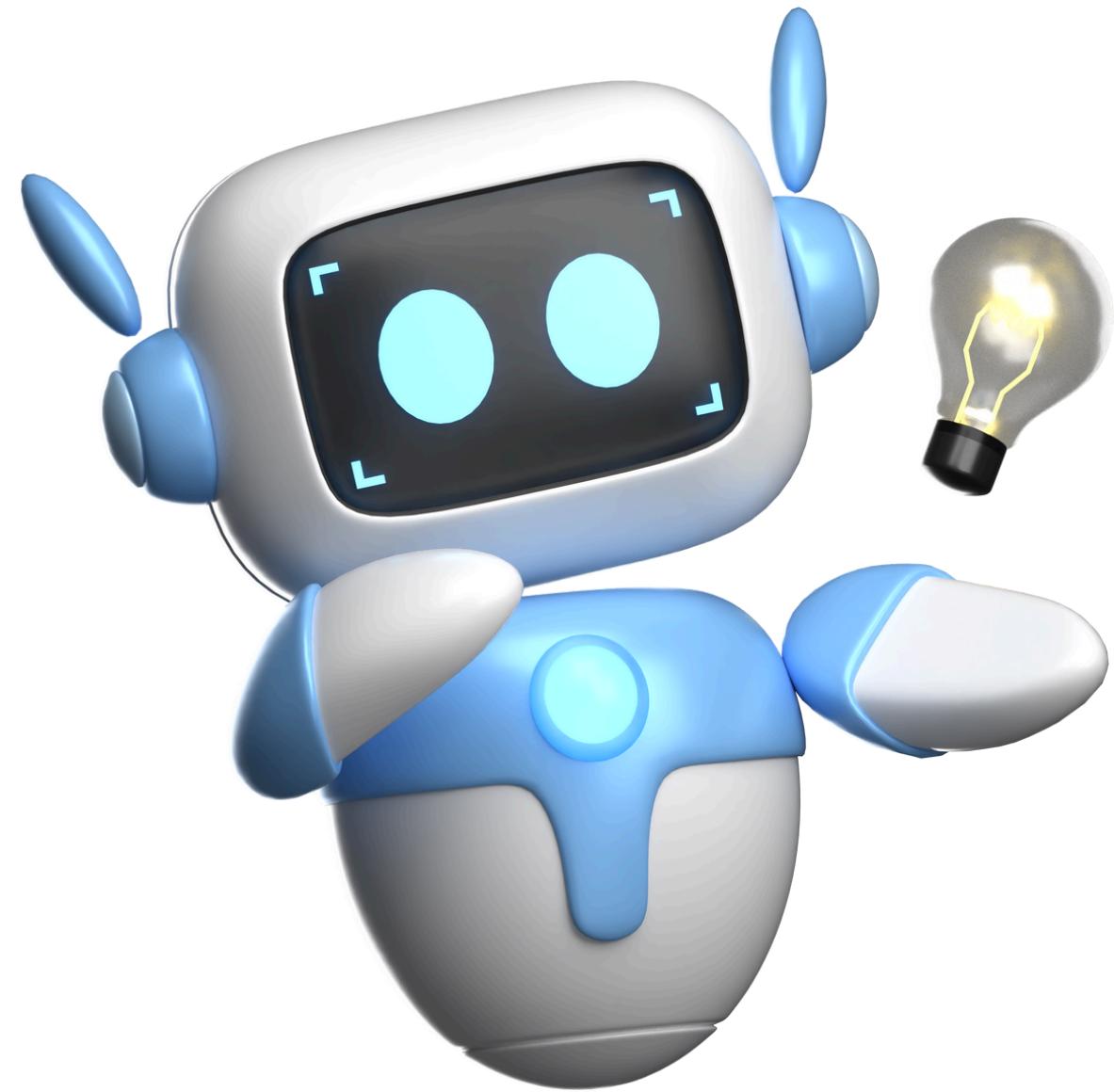
# Introduction

## 1. Input

- Dữ liệu lịch sử về vị trí (kinh độ, vĩ độ) của đối tượng và các yếu tố ảnh hưởng khác như nhiệt độ, tốc độ di chuyển, hướng di chuyển trong quá khứ (48h trước).

## 2. Output.

- Dự đoán vị trí của đối tượng trong tương lai (24h tới).





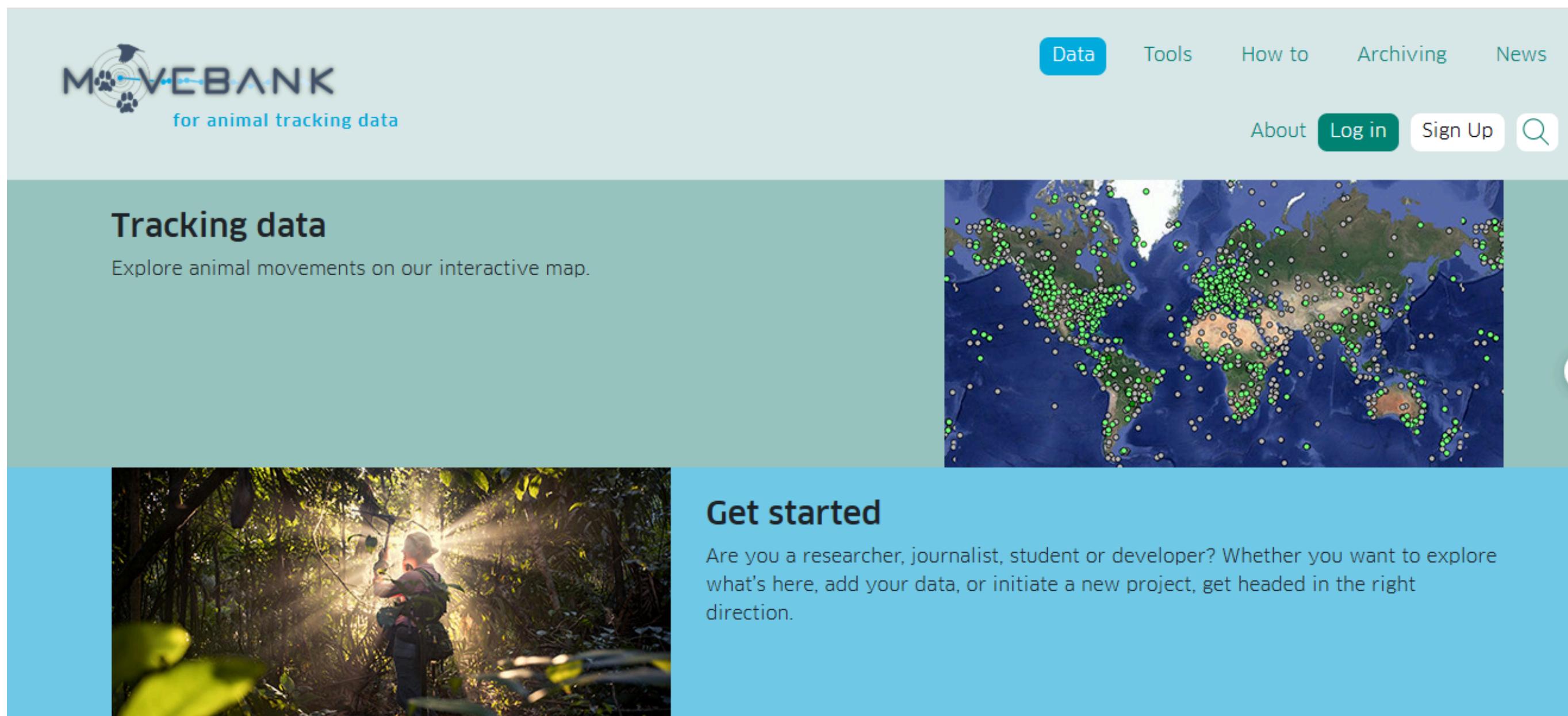
2

# DATA & EDA

# DATA

## 1. Nguồn dữ liệu

- Dữ liệu được crawl bằng API từ trang web [movebank.org](https://movebank.org)
- Đây là dữ liệu của loài chim Falco naumanni sống ở Senegal



# DATA

## 2. Thông tin chung

- Dữ liệu bao gồm 30904 điểm dữ liệu và 16 đặc trưng.
- Không có feature nào NULL.
- Các đặc trưng quan trọng: timestamp, location-long, location-lat, temperature, speed, heading, height,...

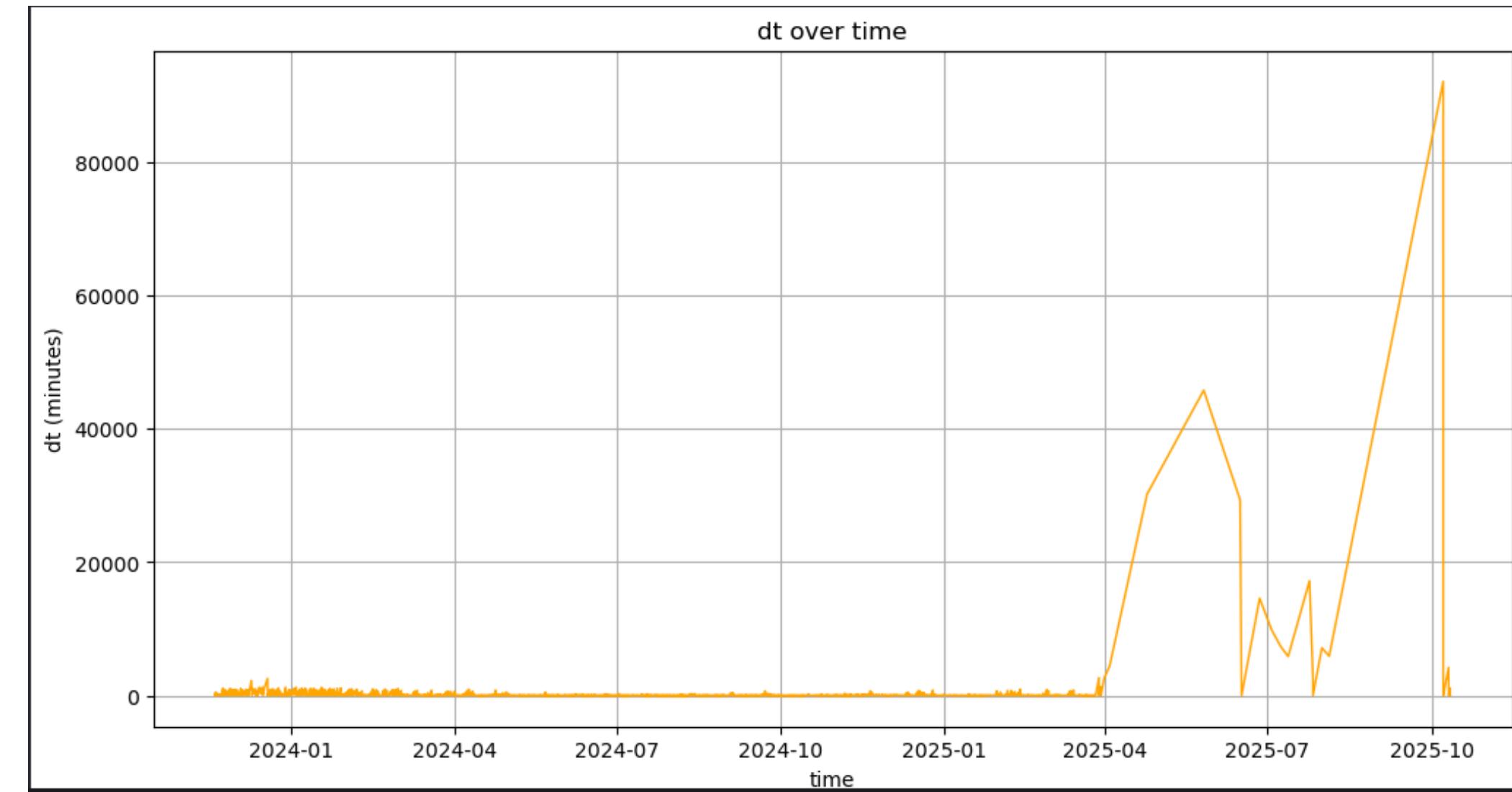
```
RangeIndex: 30904 entries, 0 to 30903
```

```
Data columns (total 16 columns):
```

#	Column	Non-Null Count	Dtype
0	event-id	30904	non-null
1	visible	30904	non-null
2	timestamp	30904	non-null
3	location-long	30904	non-null
4	location-lat	30904	non-null
5	external-temperature	30904	non-null
6	ground-speed	30904	non-null
7	heading	30904	non-null
8	height-above-msl	30904	non-null
9	import-marked-outlier	30904	non-null
10	gls:light-level	30904	non-null
11	sensor-type	30904	non-null
12	individual-taxon-canonical-name	30904	non-null
13	tag-local-identifier	30904	non-null
14	individual-local-identifier	30904	non-null
15	study-name	30904	non-null

# Exploratory Data Analysis

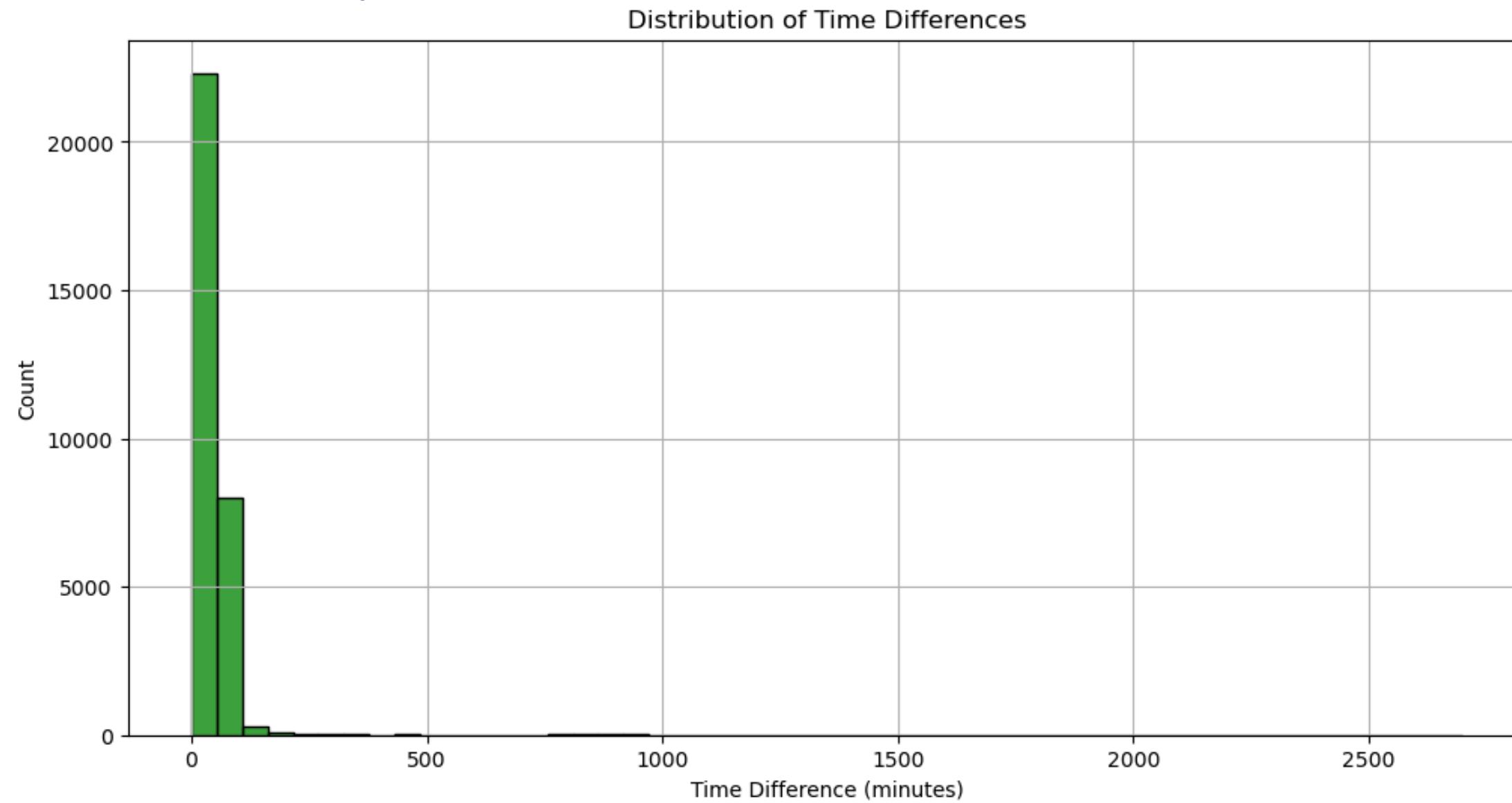
## 1. Xử lý timestamp



- Từ tháng 4/2025 khoảng cách thời gian những các bản ghi trở nên rất lớn.  
Chứng tỏ có sự ngừng hoặc gián đoạn trong quá trình thu thập dữ liệu.  
→ Do đó chúng em chỉ lấy dữ liệu đến 31/3/2025.

# Exploratory Data Analysis

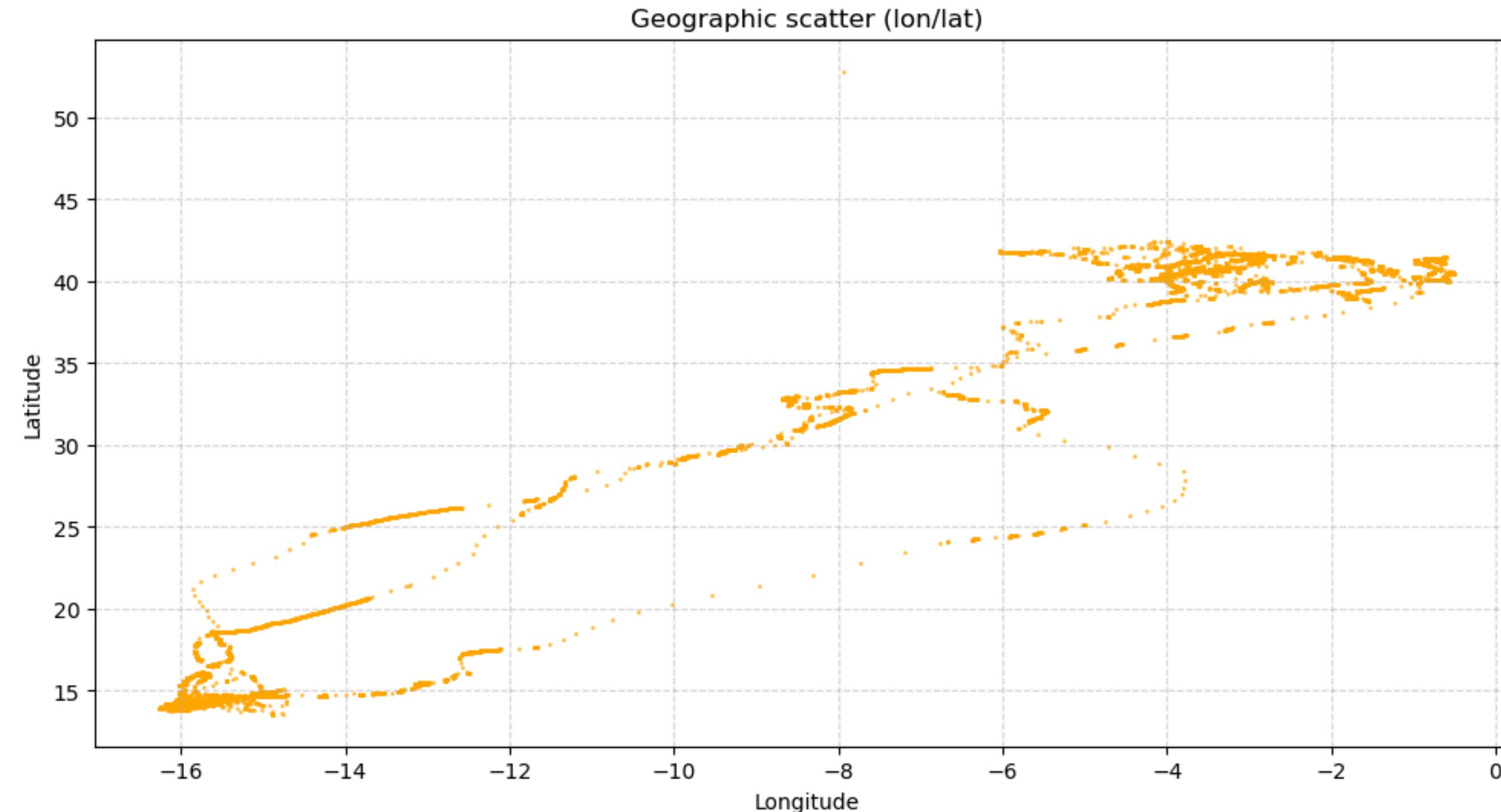
## 1. Xử lý timestamp



- Số data points sau khi xóa: 30875
- Sau khi xóa các bản ghi có sự gián đoạn lớn, dữ liệu gần như được thu thập liên tục qua các ngày.

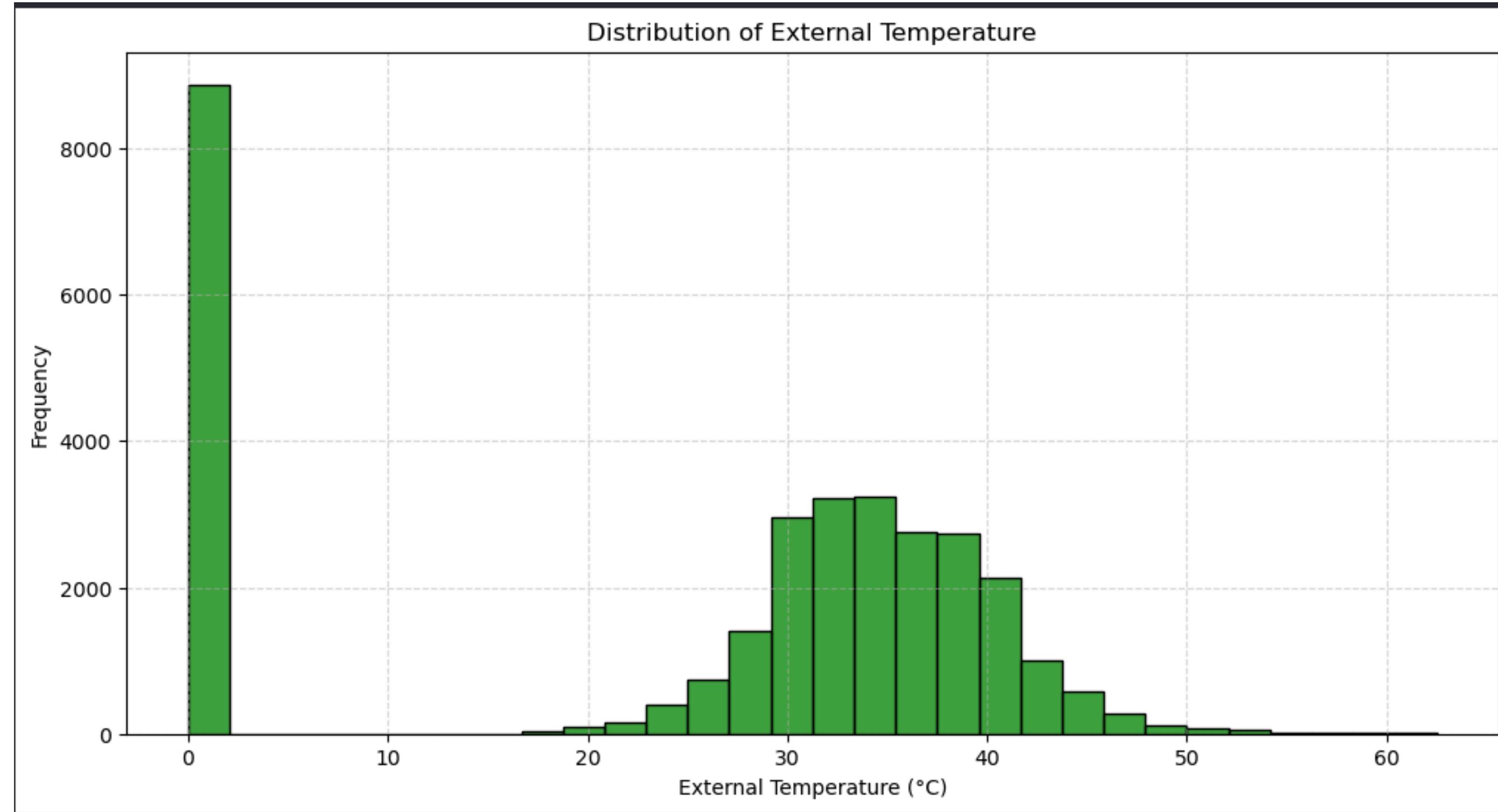
# Exploratory Data Analysis

## 2. EDA đa biến



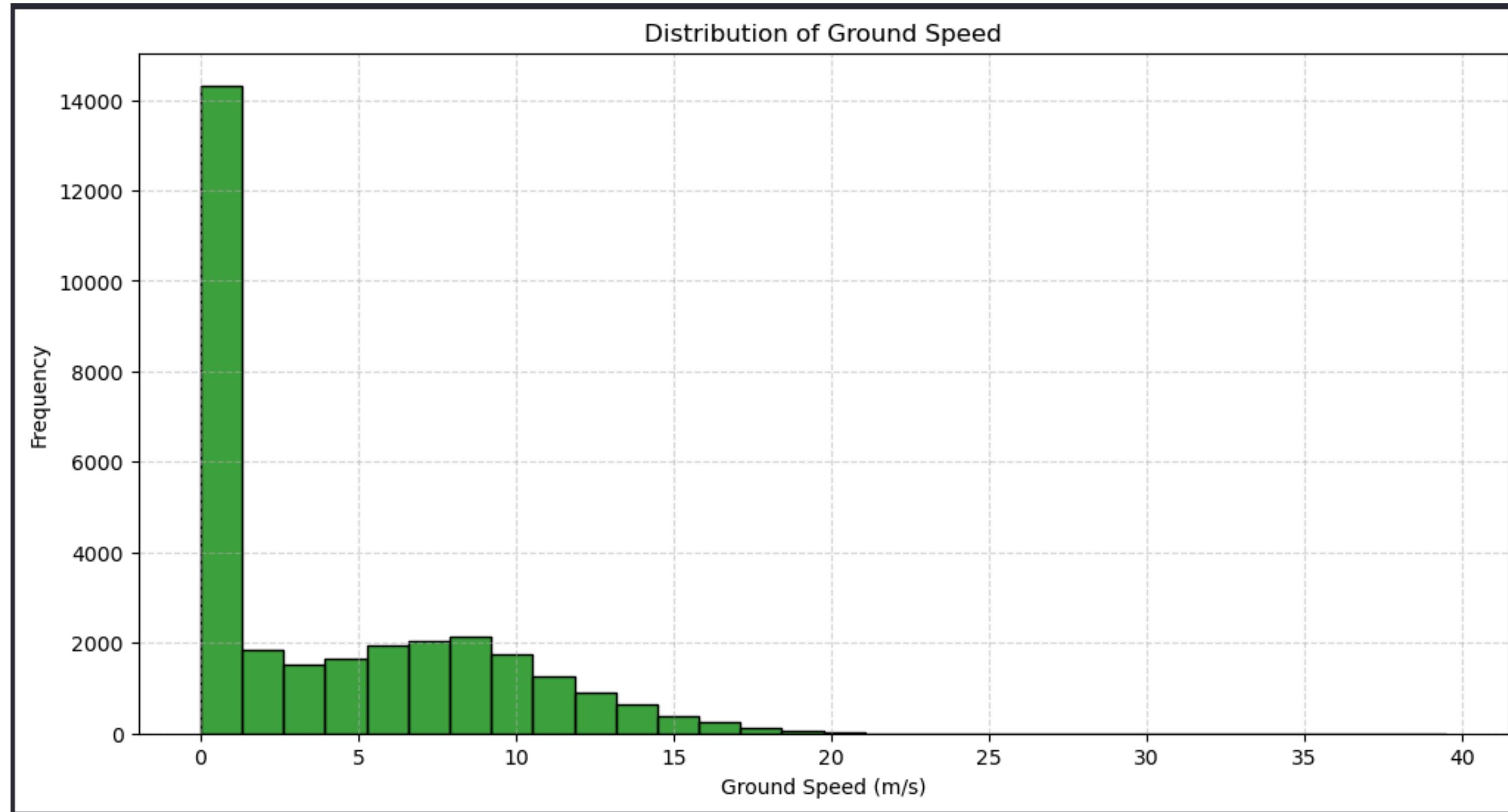
# Exploratory Data Analysis

## 2. EDA đơn biến



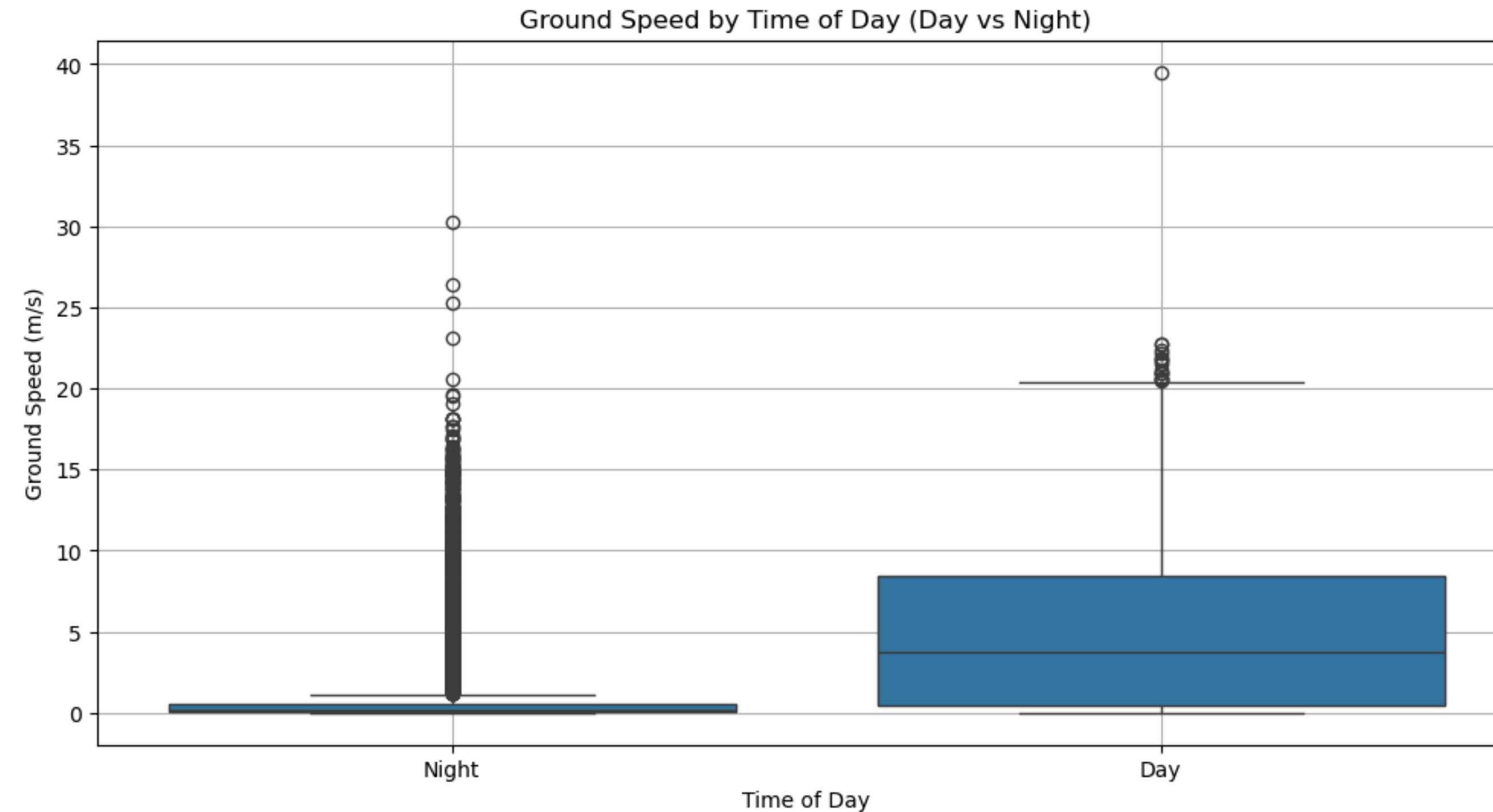
# Exploratory Data Analysis

## 2. EDA đơn biến



# Exploratory Data Analysis

## 2. EDA đa biến

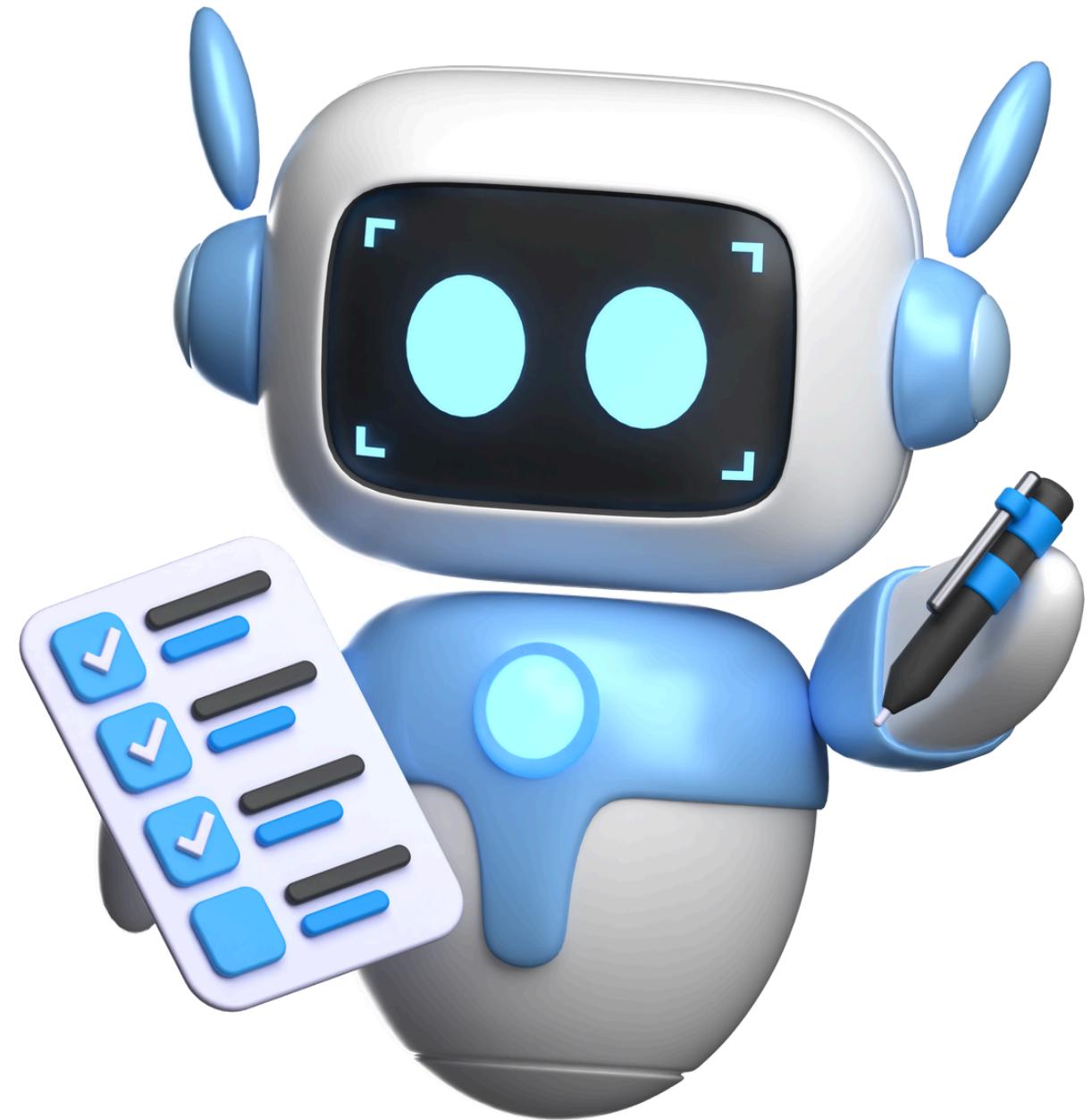




3

# PREPROCESSING

# PREPROCESSING



1

Resampling

2

Features selection

3

Features extraction

4

Features standard

# PREPROCESSING

## 1. Resampling

- Resampling là một bước quan trọng trong phân tích dữ liệu thời gian giúp điều chỉnh tần suất dữ liệu cho phù hợp với yêu cầu phân tích.
- Ý nghĩa việc resampling:
  - Tăng tính đồng đều của dữ liệu: Nếu khoảng thời gian thu thập dữ liệu không đều nhau hoặc không liên tục, resampling sẽ giúp dữ liệu trở nên đều đặn hơn, giúp mô hình học máy dễ dàng xử lý và phân tích.

2023-11-25 10:01:51  
2023-11-25 15:02:01  
2023-11-25 16:02:26  
2023-11-25 18:01:05

Resampling  
→

2023-11-25 10:00:00  
2023-11-25 11:00:00  
2023-11-25 12:00:00  
2023-11-25 13:00:00  
2023-11-25 14:00:00  
2023-11-25 15:00:00  
2023-11-25 16:00:00  
2023-11-25 17:00:00  
2023-11-25 18:00:00

# PREPROCESSING

## 1. Resampling

- Resampling là một bước quan trọng trong phân tích dữ liệu thời gian giúp điều chỉnh tần suất dữ liệu cho phù hợp với yêu cầu phân tích.
- Ý nghĩa việc resampling:
  - Giảm nhiễu: Resample giúp làm mịn dữ liệu và giảm ảnh hưởng của các biến động nhỏ không quan trọng như sự thay đổi vị trí nhỏ giữa các bản ghi liên tiếp nhau.



# PREPROCESSING

## 1. Resampling

Quy trình resampling:

- Cắt phiên (Segmentation):
  - Phân chia dữ liệu thành các phiên dựa trên ngưỡng 12h. Khi khoảng cách thời gian giữa các bản ghi > 12 giờ, sẽ bắt đầu một phiên mới.
- Resampling trong từng phiên theo tần suất là 1h. Dùng Interpolate để nội suy tuyến tính cho các điểm dữ liệu mới sau khi resample.

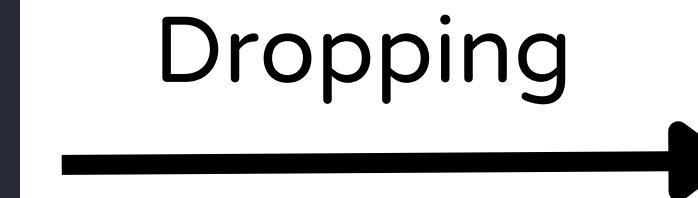
Sau khi resampling, dữ liệu chỉ còn 10597 data points

# PREPROCESSING

## 2. Features selection

- Xóa những features không quan trọng như event-id, visible, ...

```
0  event-id  
1  visible  
2  timestamp  
3  location-long  
4  location-lat  
5  external-temperature  
6  ground-speed  
7  heading  
8  height-above-msl  
9  import-marked-outlier  
10 gls:light-level  
11 sensor-type  
12 individual-taxon-canonical-name  
13 tag-local-identifier  
14 individual-local-identifier  
15 study-name
```



```
0  timestamp  
1  location-long  
2  location-lat  
3  external-temperature  
4  ground-speed  
5  heading  
6  height-above-msl  
7  gls:light-level
```

# PREPROCESSING

## 3. Features extraction

- Chuyển từ tọa độ địa lý (kinh độ, vĩ độ) sang UTM:
  - Tọa độ địa lý không thể đo khoảng cách chính xác khoảng cách trên trái đất vì trái đất hình cầu.
  - Chuyển sang hệ UTM cho phép tính toán khoảng cách chính xác bằng đại số tuyến tính.

```
wgs84 = pyproj.CRS("EPSG:4326") # Hệ tọa độ WGS84 (Lat/Lon)
utm = pyproj.CRS("EPSG:32628") # Hệ tọa độ UTM cho khu vực Senegal (EPSG:32628)
transformer = pyproj.Transformer.from_crs(wgs84, utm, always_xy=True)
# Chuyển đổi lat/lon thành x_m, y_m
data[['x_m', 'y_m']] = data.apply(
    lambda row: pd.Series(transformer.transform(row['location-lat'], row['location-long'])),
    axis=1)
```

# PREPROCESSING

## 3. Features extraction

- Tạo đặc trưng sin/cos cho các đặc trưng có tính chất tuần hoàn như thời gian và hướng di chuyển giúp mô hình nhận biết rằng đầu và cuối chu kỳ là gần nhau thay vì là hai giá trị xa nhau.

```
# Trích xuất giờ trong ngày, ngày trong tháng và tháng trong năm
hour_of_day = data['timestamp'].dt.hour
day_of_month = data['timestamp'].dt.day
month_of_year = data['timestamp'].dt.month

# Lấy số ngày trong tháng để tính toán tuần hoàn cho ngày
days_in_month = data['timestamp'].dt.days_in_month

# Tạo đặc trưng sin và cos cho giờ trong ngày
data['sin_hour'] = np.sin(2 * np.pi * hour_of_day / 24)
data['cos_hour'] = np.cos(2 * np.pi * hour_of_day / 24)

# Tạo các đặc trưng sin/cos cho ngày trong tháng
data['sin_day'] = np.sin(2 * np.pi * day_of_month / days_in_month)
data['cos_day'] = np.cos(2 * np.pi * day_of_month / days_in_month)

# Tạo các đặc trưng sin/cos cho tháng trong năm
data['sin_month'] = np.sin(2 * np.pi * month_of_year / 12)
data['cos_month'] = np.cos(2 * np.pi * month_of_year / 12)
```

```
# Áp dụng sin/cos transformation cho heading (hướng di chuyển)
data['sin_heading'] = np.sin(2 * np.pi * data['heading'] / 360)
data['cos_heading'] = np.cos(2 * np.pi * data['heading'] / 360)
```

# PREPROCESSING

## 3. Features extraction

- Tạo các đặc trưng mới:
  - Khoảng cách giữa hai vị trí liên tiếp → distance.
  - Thời điểm trong ngày (sáng, chiều, tối) → time\_of\_day.
  - Mùa trong năm (xuân, hạ, thu, đông) → season.

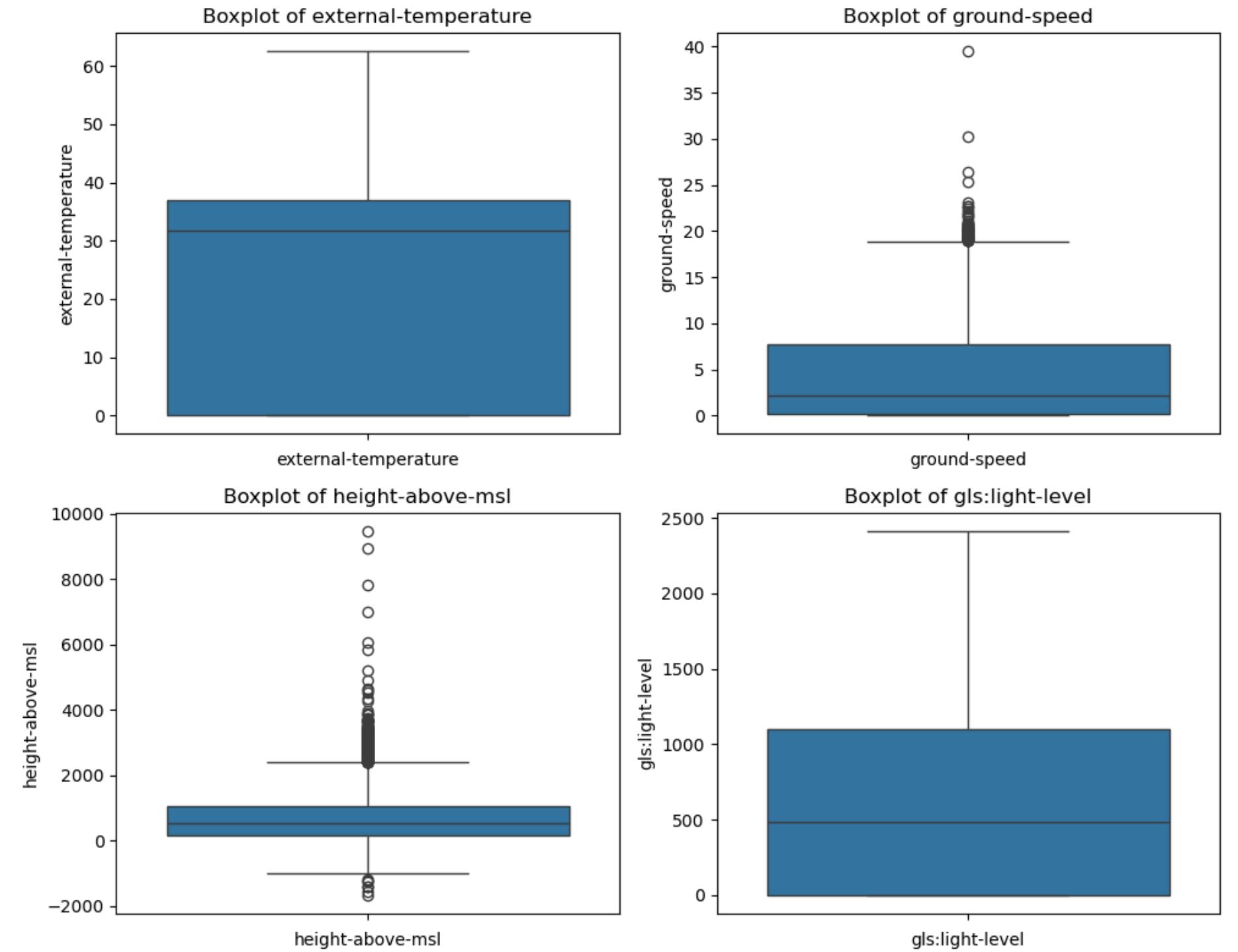
```
data['distance'] = np.sqrt(dx**2 + dy**2)

hour_of_day = data['timestamp'].dt.hour
data['time_of_day'] = np.where(hour_of_day < 12, 'morning',
                                np.where(hour_of_day < 18, 'afternoon', 'night'))

month_of_year = data['timestamp'].dt.month
# Xác định mùa theo tháng (Ví dụ: mùa xuân, hè, thu, đông)
data['season'] = np.where(month_of_year.isin([12, 1, 2]), 'winter',
                           np.where(month_of_year.isin([3, 4, 5]), 'spring',
                                   np.where(month_of_year.isin([6, 7, 8]), 'summer', 'fall')))
```

# PREPROCESSING

## 4. Features standard



# PREPROCESSING

## 4. Features standard

```
scaler = StandardScaler()
data['external-temperature'] = scaler.fit_transform(data[['external-temperature']])
data['gls:light-level'] = scaler.fit_transform(data[['gls:light-level']])
data['distance'] = scaler.fit_transform(data[['distance']])

robust_scaler = RobustScaler()
data['ground-speed'] = robust_scaler.fit_transform(data[['ground-speed']])
data['height-above-msl'] = robust_scaler.fit_transform(data[['height-above-msl']])

label = LabelEncoder()
data['time_of_day'] = label.fit_transform(data['time_of_day'])
data['season'] = label.fit_transform(data['season'])
```



4

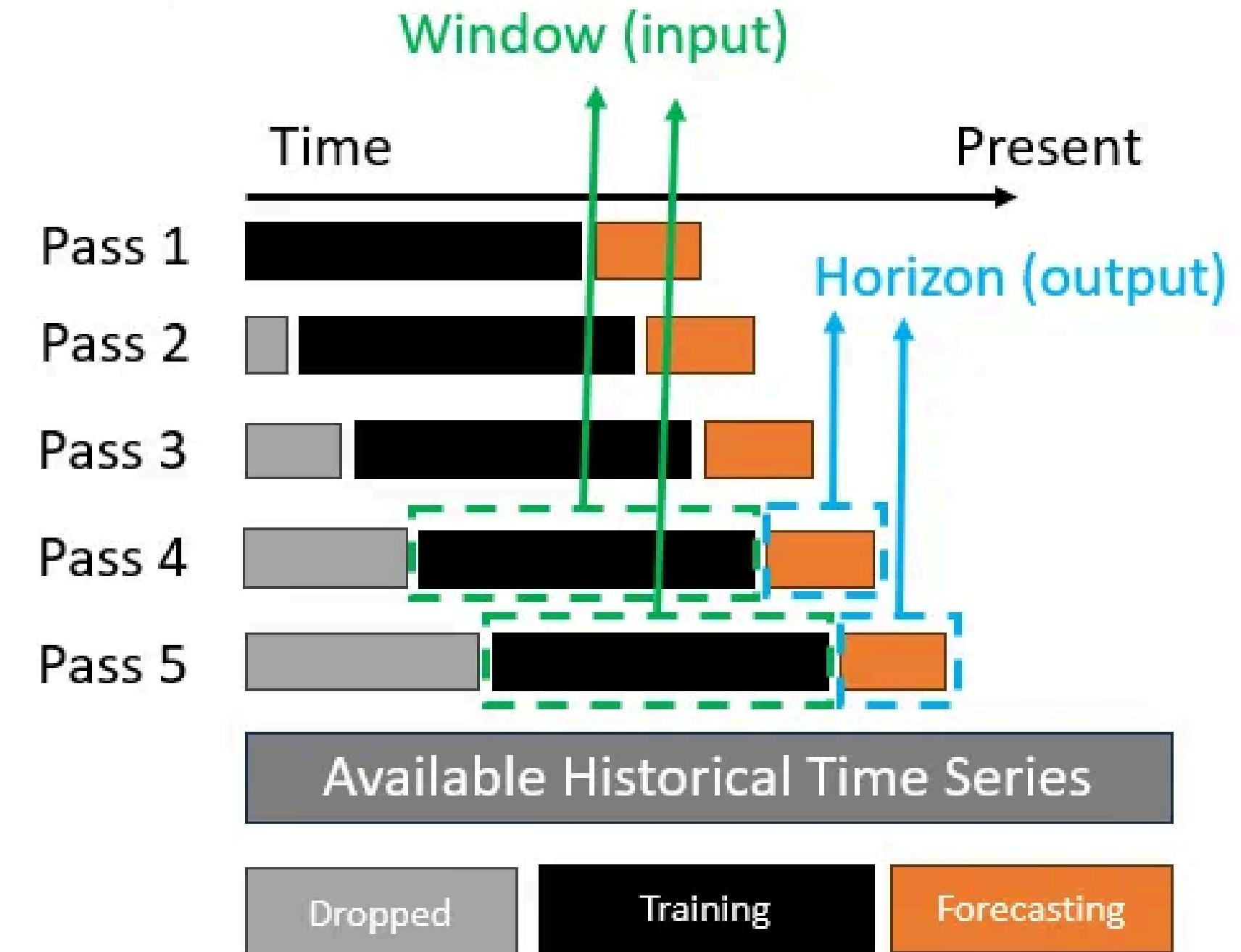
# MODELING

# MODELING

## 1. Create sliding window

- Input window = 48h trước
- Output horizon = 24h tới
- Step/Stride = 1
  - là khoảng dịch chuyển window sau mỗi lần tạo mẫu.

Sliding Window



# MODELING

## 2. Split train/val/test set

18/11/2023 - 30/11/2024 →

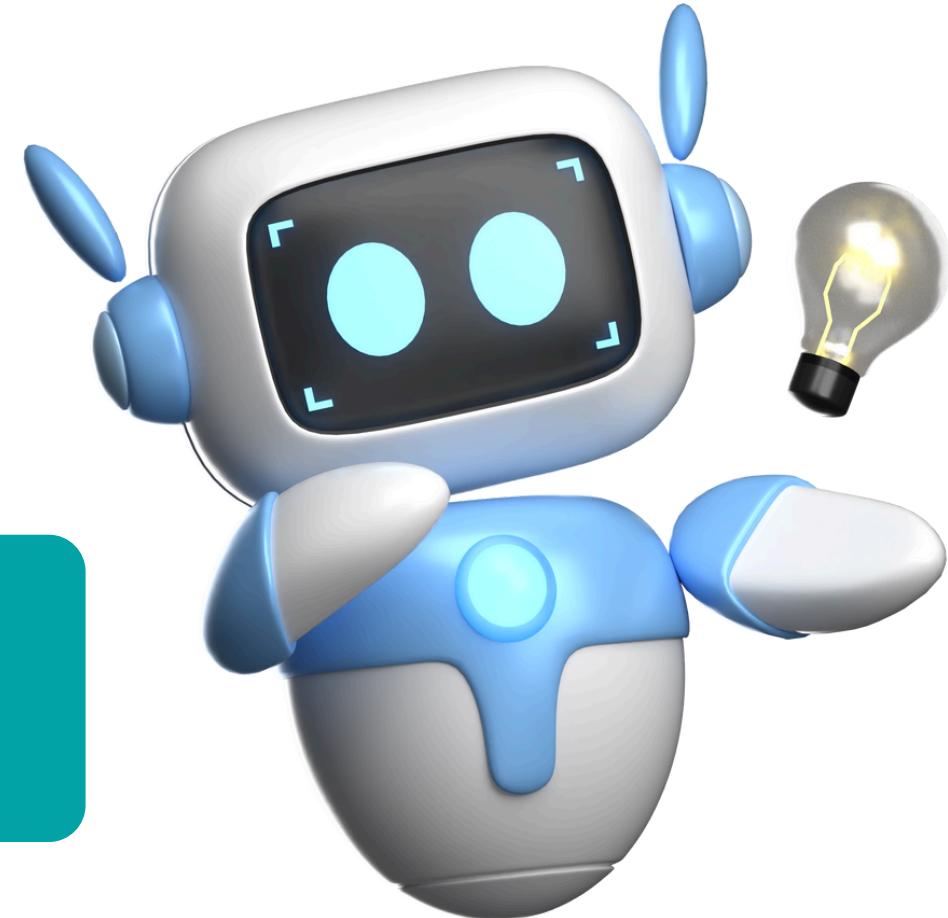
TRAIN

1/12/2024 - 31/01/2025 →

VALIDATION

1/02/2025- 31/03/2025 →

TEST



# MODELING

## 3. Model Training

- LinearRegression
- RandomForest
- XGBoost
- KNN
- MLP
- LSTM



# METRIC

# Metric

- MAE
- R<sup>2</sup>
- ADE
- FDE

# Metric

- MAE (sai số tuyệt đối trung bình): đo trực tiếp mức lệch trung bình giữa vị trí dự đoán và thực tế (nên tính theo khoảng cách địa lý – Haversine – để ra mét/km). Dễ hiểu, ít nhạy với outlier, hợp để theo dõi chất lượng chung.
- $R^2$  (hệ số xác định): cho biết mô hình tốt hơn baseline (giữ nguyên/điểm trung bình) bao nhiêu phần biến thiên. Hữu ích khi so sánh model/feature; có thể tính theo từng trực lat/lon hoặc trên khoảng cách.

=> MAE/ $R^2$  đánh giá “sức mạnh hồi quy tổng quát”

# MAE

Công thức toán học

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

**R<sup>2</sup>**

## Công thức toán học

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

# Metric

- ADE (Average Displacement Error): đo trung bình toàn bộ quỹ đạo 24h, phản ánh chất lượng dự báo ở mọi mốc thời gian, phù hợp khi cần theo dõi liên tục đường đi (trajectory). Bắt được lỗi tích lũy theo thời gian.
- FDE (Final Displacement Error): đo sai số tại mốc cuối 24h, quan trọng cho ra quyết định triển khai (điểm đến cần chính xác). Bổ sung cho ADE: mô hình có thể ADE tốt nhưng lệch đích – FDE sẽ lộ rõ.

=> ADE/FDE đánh giá độ chính xác không gian theo thời gian của quỹ đạo

# ADE

## Công thức toán học

$$\text{ADE} = \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T \| \mathbf{p}_{i,t} - \hat{\mathbf{p}}_{i,t} \|_2$$

Với chuẩn Euclid:  $\| \mathbf{a} - \mathbf{b} \|_2 = \sqrt{\sum_{k=1}^d (a_k - b_k)^2}$ .  
(Ở 2D:  $\sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}$ .)

# FDE

## Công thức toán học

$$\text{FDE} = \frac{1}{N} \sum_{i=1}^N \| \mathbf{p}_{i,T} - \hat{\mathbf{p}}_{i,T} \|_2$$

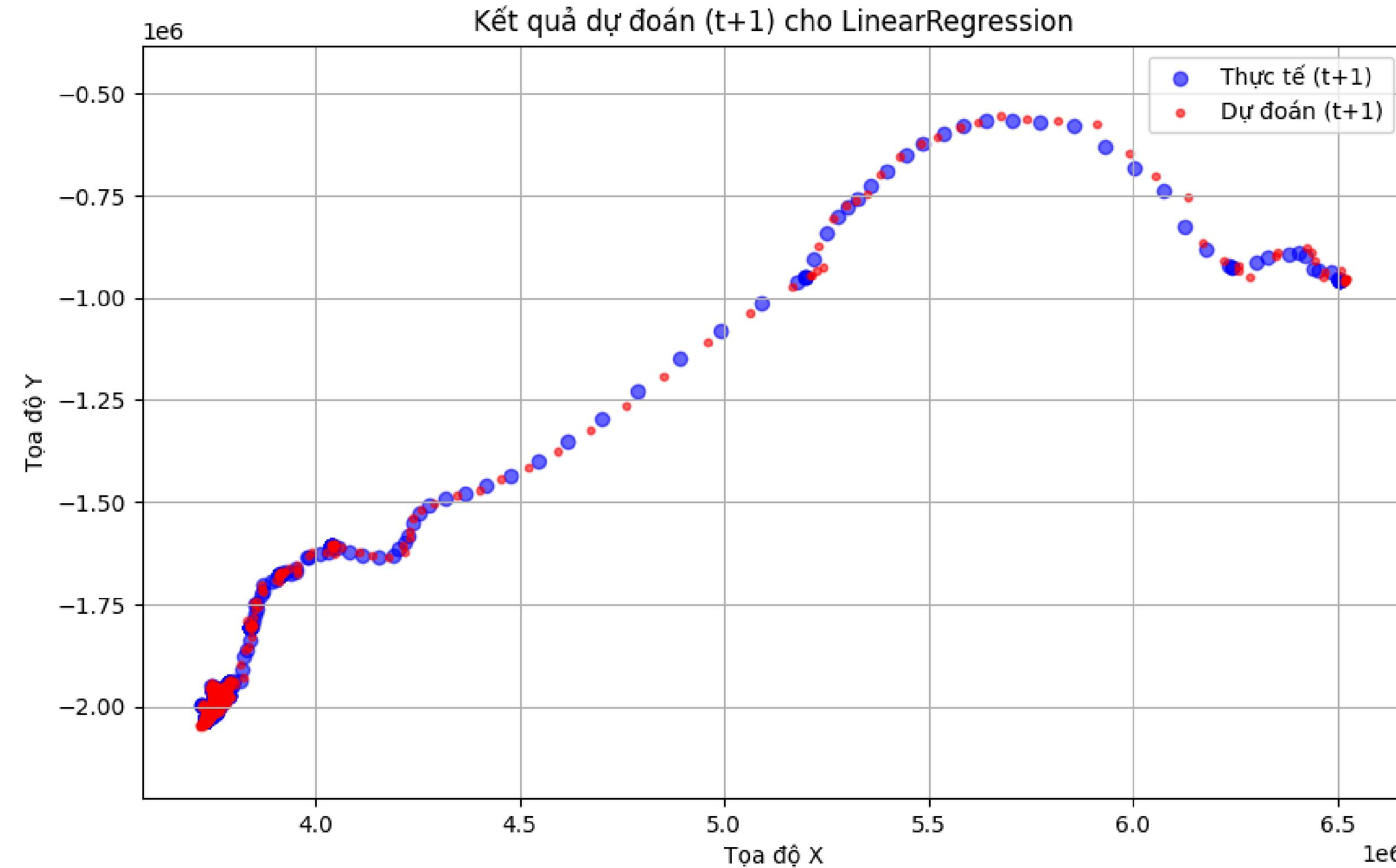


# RESULT

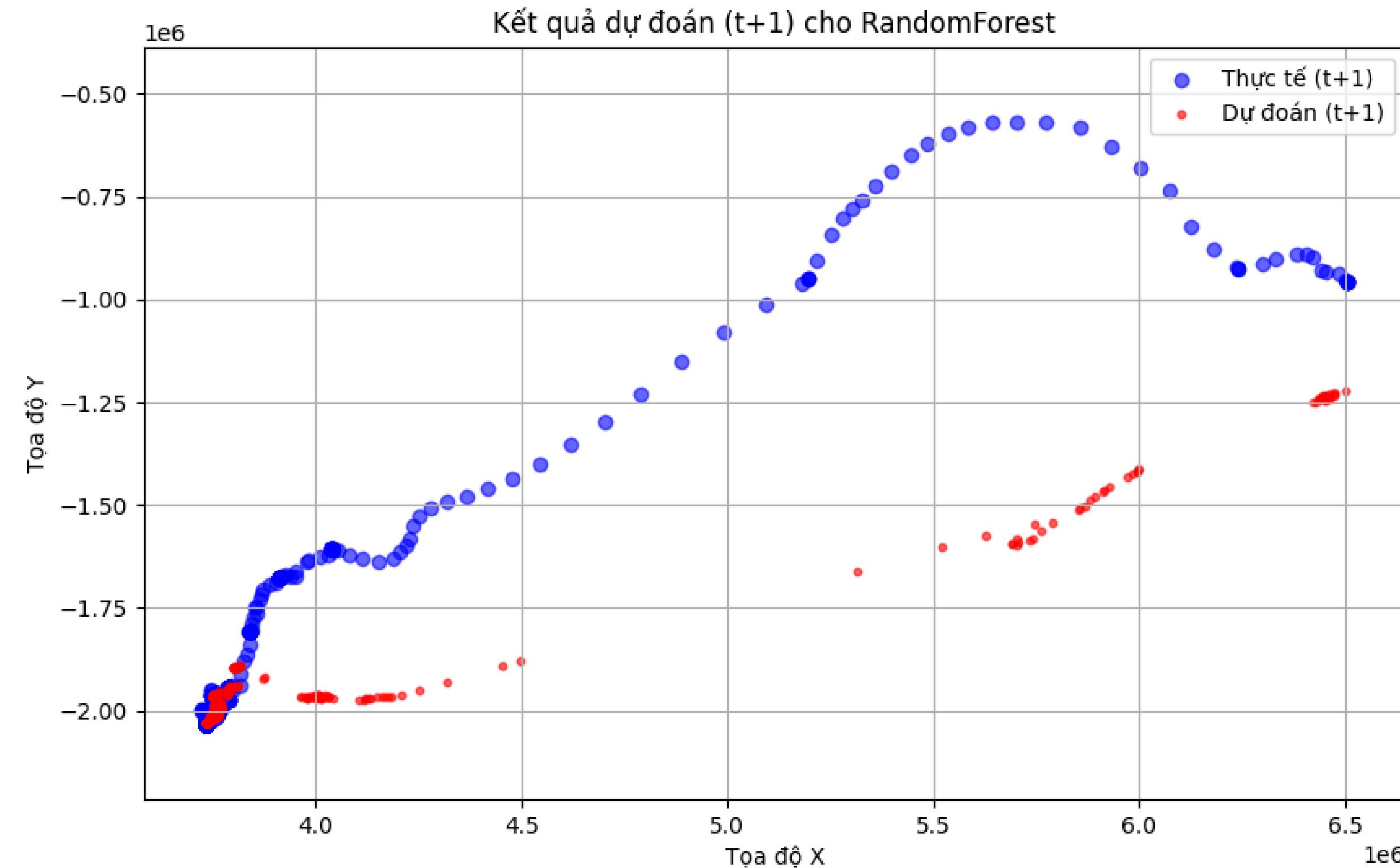
# RESULT

	MAE	R2	ADE	FDE
<b>LinearRegression</b>	<b>39892.31</b>	<b>0.92</b>	<b>60555.17</b>	106329.98
<b>RandomForest</b>	51475.48	0.72	78703.24	101384.48
<b>XGBoost</b>	51544.27	0.67	81189.57	101106.49
<b>KNN</b>	69693.11	0.66	108816.56	117361.30
<b>MLP</b>	63131.78	0.82	98639.18	113682.41
<b>LSTM</b>	52497.29	0.82	82091.74	<b>97908.78</b>

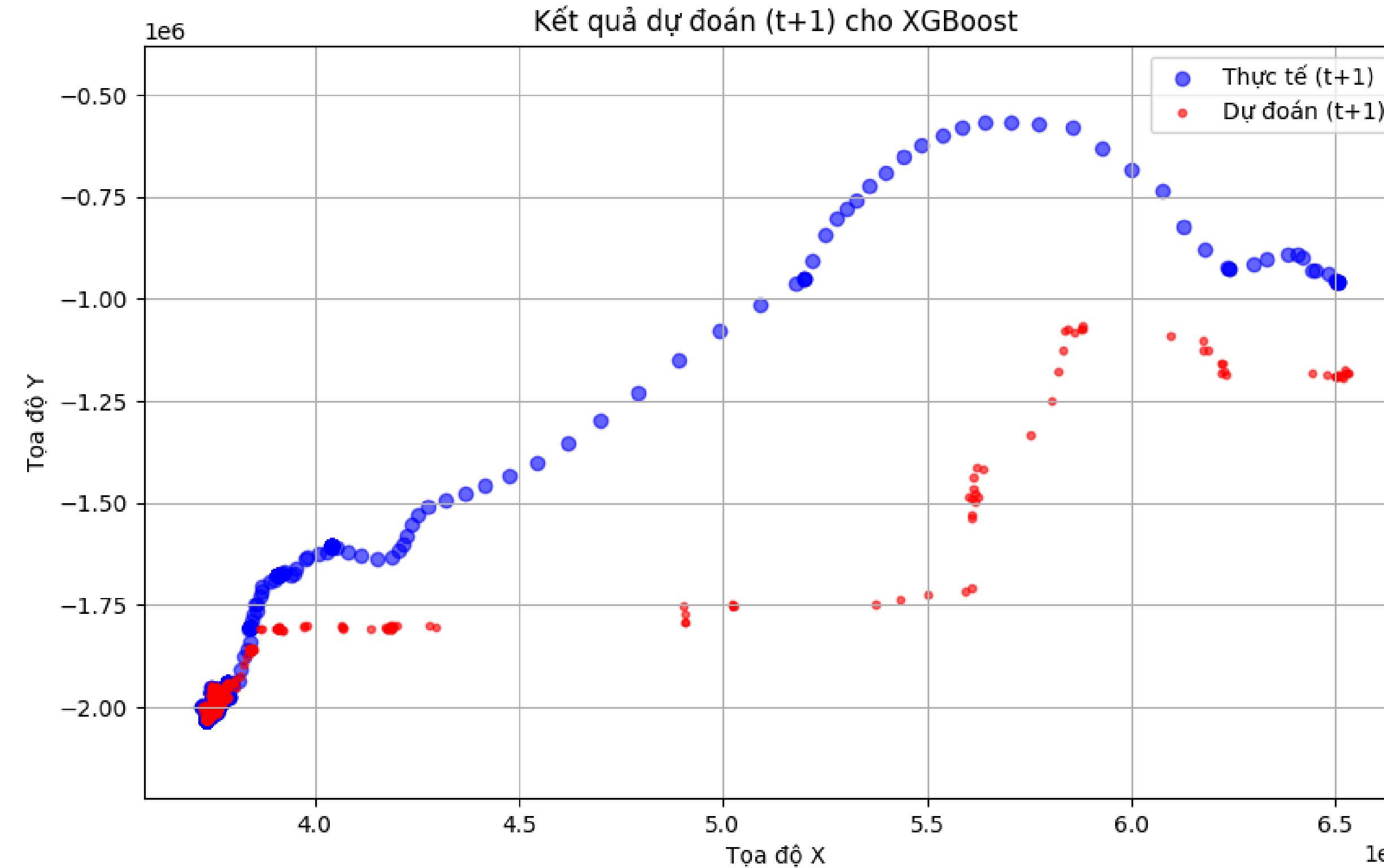
# LinearRegression



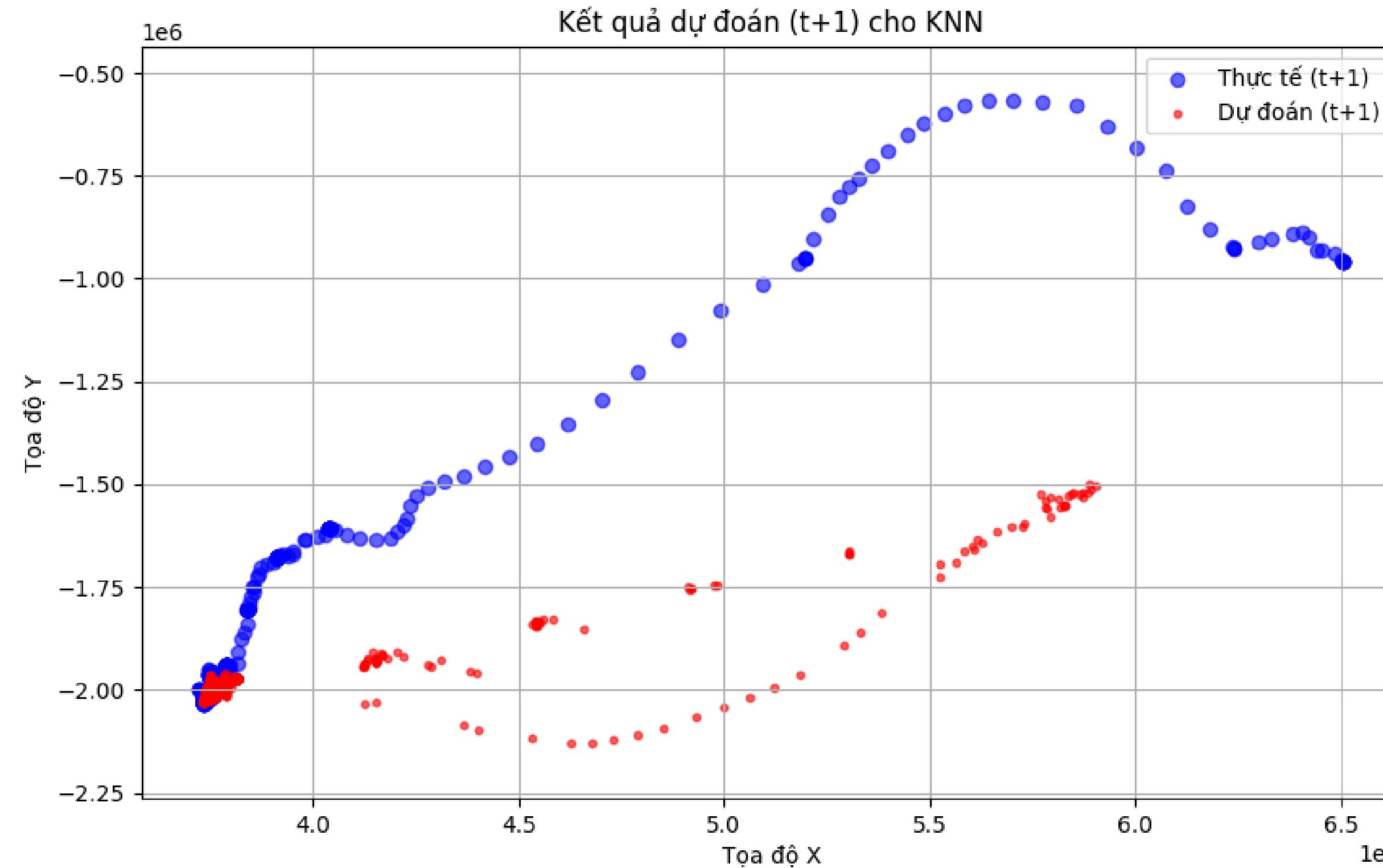
# RandomForest



# XGBoost



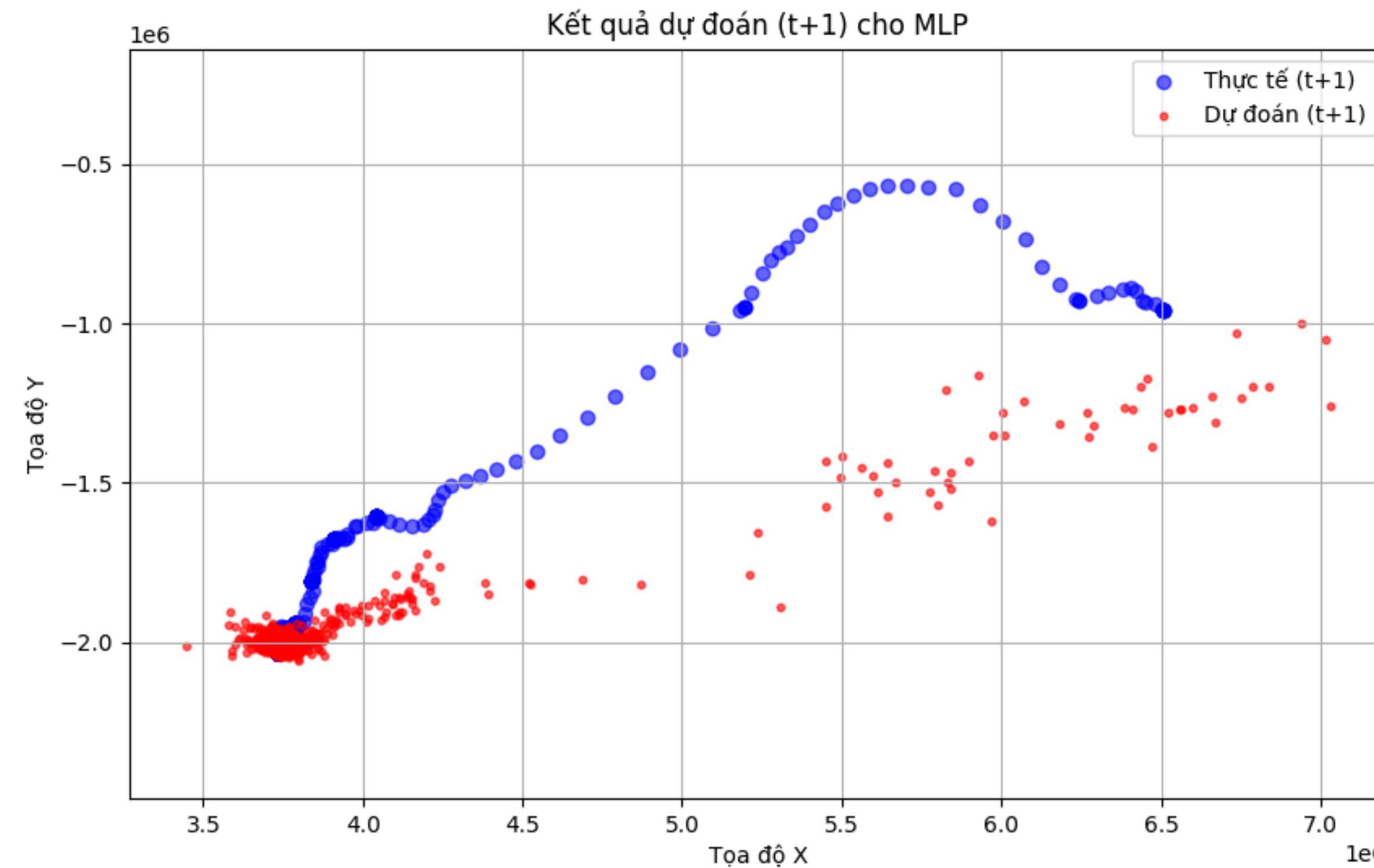
# KNN



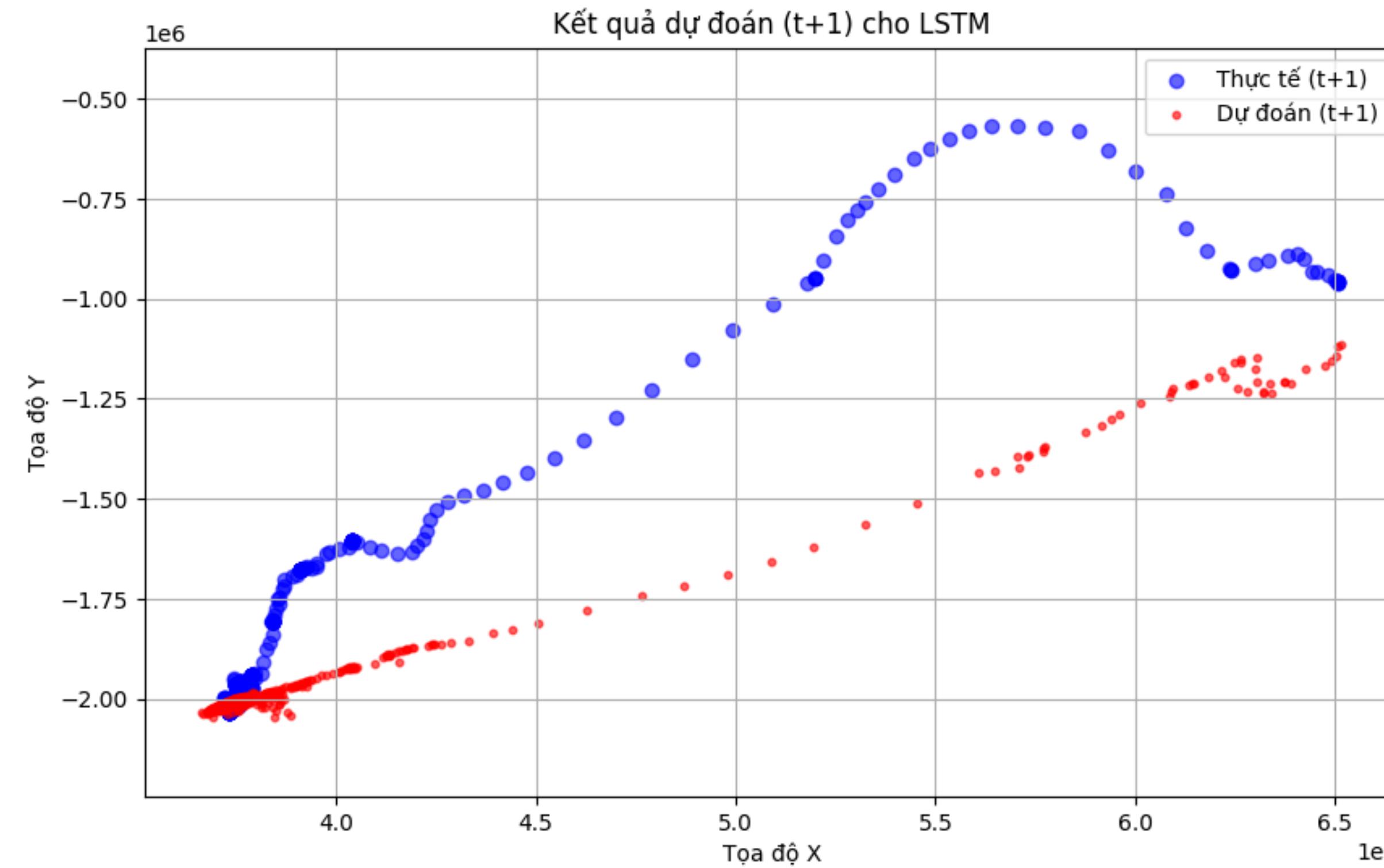
# RESULT

- Tổng thể, ta thấy Linear Regression phục vụ bài toán tốt hơn so với các mô hình còn lại. Mô hình tuyến tính (LR) ngoại suy tốt khi chuyển động 24h kế tiếp gần như “vận tốc + hướng” ổn định → MAE/ADE thấp,  $R^2$  cao.
- Trong khi đó, RandomForest/XGBoost/KNN dự đoán kiểu N điểm dữ liệu gần nhất nên đường đi bị “bậc thang”, kém ngoại suy trên tọa độ liên tục → MAE/ADE cao hơn.  
=> Linear Regression có khả năng ngoại suy tốt.

# MLP



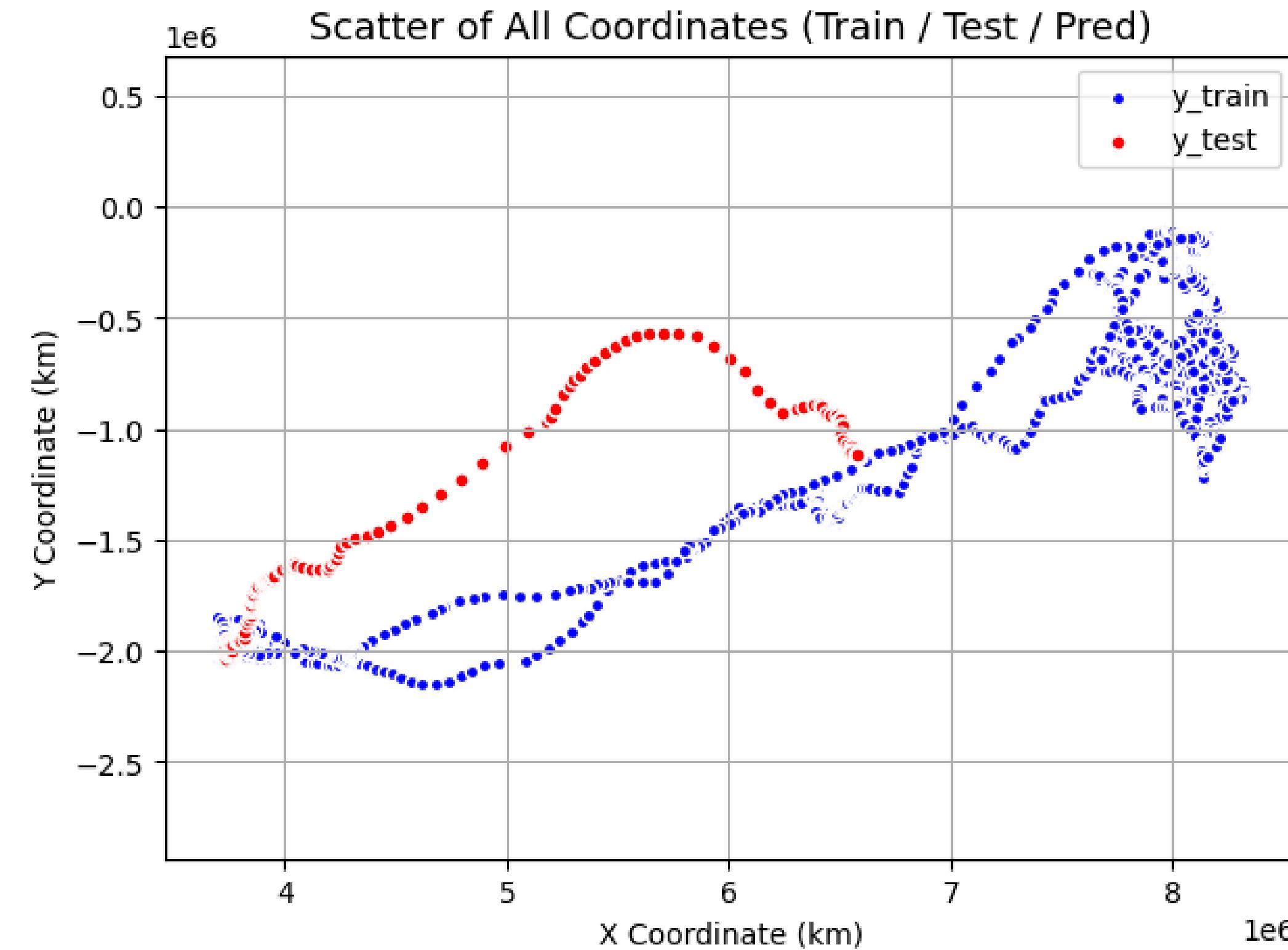
# LSTM



# RESULT

- Ngoài ra, mô hình dạng chuỗi như LSTM khai thác được quy luật/trật tự 48h, vì là mô hình có thiên kiến chuỗi (inductive bias theo thời gian) → hiểu thứ tự tốt hơn so với các mô hình khác nên FDE thường tốt nhất. ⇒ FDE thường trội.
- Tuy nhiên, LSTM ưu tiên mốc thời gian cuối cùng trong khoảng dự báo ⇒ FDE tốt, nhưng phải đánh đổi độ khớp toàn hành trình ⇒ ADE/MAE thấp hơn so với các mô hình khác.

# RESULT



# Summary

---

- Hiệu quả Mô hình (Model Performance)
- Ý nghĩa Thực tiễn (Practical Impact)
- Đặc tính Dữ liệu (Data Characteristics)

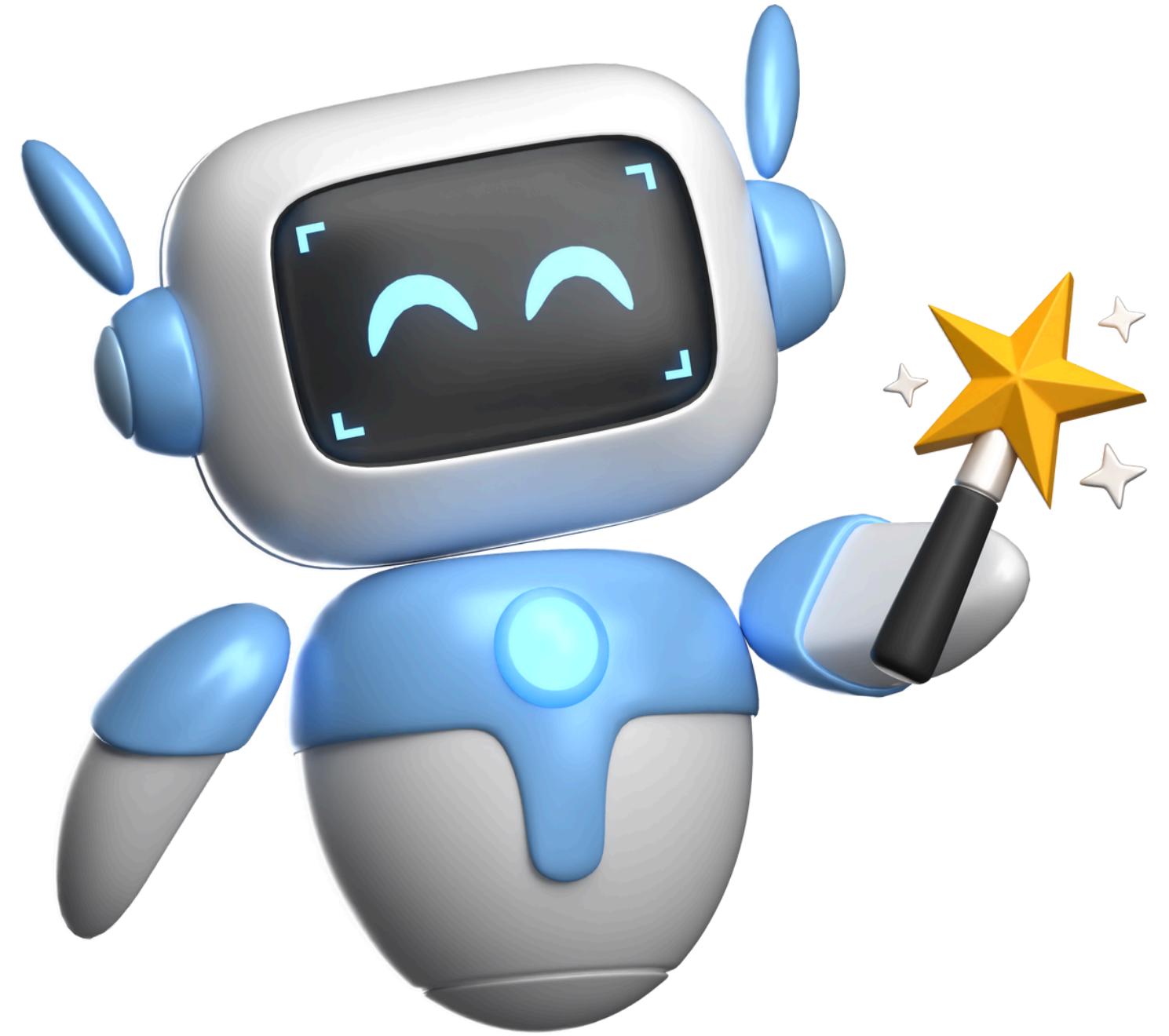




# DEMO

# DEMO

- Demo của nhóm được công bố ở đây.



**Thank you for  
listening !!!**

---

NHÓM 12 - CS313