HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



PHẠM MINH ĐỨC

ỨNG DỤNG MÔ HÌNH DỰ ĐOÁN CHUỖI THỜI GIAN TỐI ƯU CHI PHÍ VẬN TẢI

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHẠM MINH ĐỨC

ỨNG DỤNG MÔ HÌNH DỰ ĐOÁN CHUỖI THỜI GIAN TỐI ƯU CHI PHÍ VẬN TẢI

Chuyên ngành: KHOA HỌC MÁY TÍNH

Mã số: 8.48.01.01

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC

PGS.TS. NGUYỄN MẠNH HÙNG

HÀ NỘI – 2025

LỜI CAM ĐOAN

Em cam đoan rằng đồ án: **"Ứng dụng mô hình dự đoán chuỗi thời gian tối ưu chi phí vận tải"** là công trình nghiên cứu của chính em.

Những kết quả nghiên cứu được trình bày trong đồ án là công trình của riêng em dưới sự hướng dẫn của **PGS.TS. Nguyễn Mạnh Hùng**.

Em cam đoan các số liệu, kết quả đạt được trong đồ án là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong đồ án này mà không được trích dẫn theo đúng quy định.

Hà Nội, ngày 15 tháng 11 năm 2025

Học viên thực hiện đề án

Phạm Minh Đức

LỜI CẢM ƠN

Đề án tốt nghiệp này là kết quả của một quá trình học tập, nghiên cứu và rèn luyện nghiêm túc. Em xin được bày tỏ lòng biết ơn sâu sắc đến PGS.TS. Nguyễn Mạnh Hùng – người thầy đã đồng hành, định hướng chuyên môn và đưa ra những nhận xét quý báu giúp em hoàn thiện nội dung nghiên cứu một cách khoa học và chặt chẽ.

Em cũng xin gửi lời cảm ơn chân thành đến Ban lãnh đạo, các thầy cô trong Khoa và Học viện Công nghệ Bưu chính Viễn thông – những người đã trang bị cho em nền tảng kiến thức vững chắc, tạo điều kiện học tập và nghiên cứu thuận lợi trong suốt thời gian theo học tại trường.

Mặc dù đã dành nhiều tâm huyết để thực hiện đề án này, song với giới hạn về thời gian và năng lực, chắc chắn không tránh khỏi những sai sót. Em rất mong nhận được những góp ý quý báu từ quý thầy cô để hoàn thiện hơn trong những nghiên cứu tiếp theo.

Em xin chân thành cảm ơn!

MỤC LỤC

LỜI CAM ĐOAN ..... **i**

LỜI CẢM ƠN ..... **ii**

DANH MỤC CÁC KÍ HIỆU VÀ CHỮ CÁI VIẾT TẮT ..... **v**

DANH MỤC CÁC KÝ HIỆU TOÁN HỌC ..... **vi**

DANH MỤC CÁC BIỂU ĐỒ ..... **vii**

DANH SÁCH CÁC BẢNG ..... **viii**

1 Lý do chọn đề tài **1**

2 Tổng quan về vấn đề nghiên cứu **1**

3 Mục đích nghiên cứu **1**

4 Đối tượng và phạm vi nghiên cứu **1**

5 Phương pháp nghiên cứu **2**

CHƯƠNG 1: BÀI TOÁN DỰ ĐOÁN THỜI GIAN VẬN TẢI HÀNH KHÁCH **3**

1.1 Giới thiệu bài toán dự đoán thời gian vận tải hành khách 3

1.1.1 Định nghĩa và mục tiêu của bài toán 3

1.1.2 Ứng dụng giải quyết bài toán trong thực tế 3

1.1.3 Các nghiên cứu liên quan trong lịch sử 4

1.2 Các phương pháp dự đoán truyền thống trước đây 5

1.2.1 Phương pháp dựa trên thống kê 5

1.2.2 Phương pháp dựa trên mô hình học máy 9

1.3 Các phương pháp dự đoán hiện đại 12

1.3.1 Ứng dụng mô hình học máy kết hợp (Ensemble Learning) 12

1.3.2 Ứng dụng mô hình học sâu 16

1.4 Đề xuất phương hướng 19

1.4.1 Khai phá dữ liệu trong bài toán dự đoán thời gian di chuyển (ETA) 19

1.4.2 Sử dụng học máy kết hợp Ensemble Learning 21

1.5 Kết luận chương 22

CHƯƠNG 2: HUẤN LUYỆN MÔ HÌNH DỰ BÁO THỜI GIAN VẬN TẢI HÀNH KHÁCH **24**

2.1 Kỹ thuật khai phá dữ liệu 24
2.1.1 Xử lý ghép bản đồ – Map matching 24
2.1.2 Sử dụng thuật toán Ramer–Douglas–Peucker 24
2.1.3 Biến đổi dữ liệu và tạo đặc trưng mới 25
2.1.4 Phân tích mô tả và tính tương quan đặc trưng 26
2.2 Mô hình dự đoán XGBoost 26
2.2.1 Nguồn gốc mô hình 26
2.2.2 Các tính năng chính 27
2.2.3 Ứng dụng của mô hình 28
2.3 Phương pháp đánh giá và thực nghiệm mô hình 28
2.3.1 Mục tiêu và phạm vi đánh giá 29
2.3.2 Quy trình huấn luyện và chia tập dữ liệu 29
2.3.3 Tinh chỉnh siêu tham số (Hyperparameter Tuning) 30
2.3.4 Chỉ số đánh giá hiệu suất mô hình 30
2.3.5 Phương pháp diễn giải và kiểm định mô hình 31
2.4 Kết luận chương 31

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 33
3.1 Bài toán chi tiết 33
3.2 Kịch bản tiến hành thực nghiệm 33
3.2.1 Môi trường thực nghiệm 33
3.2.2 Bộ dữ liệu thực nghiệm 34
3.2.3 Biến đổi và tạo đặc trưng (Feature Engineering) 35
3.2.4 Huấn luyện mô hình 40
3.3 Kết quả thực nghiệm và đánh giá hiệu suất mô hình 41
3.3.1 Kết quả thực nghiệm 41
3.4 Phân tích kết quả và thảo luận 41
3.4.1 So sánh trực quan giữa các mô hình 42
3.4.2 Đường hội tụ của mô hình 43
3.4.3 Phân phối sai số dự báo 44
3.4.4 Phân tích SHAP Summary Plot 45
3.5 Kết luận chương 47

KẾT LUẬN 48

TÀI LIỆU THAM KHẢO 50

DANH MỤC CÁC KÍ HIỆU VÀ CHỮ CÁI VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
ETA	Estimated Time of Arrival	Thời gian dự kiến đến (thời gian vận chuyển ước lượng)
ML	Machine Learning	Học máy
DL	Deep Learning	Học sâu
ARIMA	Autoregressive Integrated Moving Average	Mô hình trung bình trượt tích hợp tự hồi quy
SARIMA	Seasonal Autoregressive Integrated Moving Average	Mô hình ARIMA có yếu tố mùa vụ
KNN	K-Nearest Neighbors	Thuật toán láng giềng gần nhất
SVR	Support Vector Regression	Hồi quy véc-tơ hỗ trợ
XGBoost	Extreme Gradient Boosting	Mô hình tăng cường độ dốc cực trị
RMSE	Root Mean Squared Error	Sai số bình phương trung bình căn bậc hai
MAE	Mean Absolute Error	Sai số tuyệt đối trung bình
SHAP	SHapley Additive exPlanations	Phương pháp giải thích độ quan trọng đặc trưng SHAP

DANH MỤC CÁC KÝ HIỆU TOÁN HỌC

Ký hiệu	Ý nghĩa / Định nghĩa	Phạm vi sử dụng
Đại số tuyến tính		
x, y, γ	Biến vô hướng (scalar)	Toàn văn
$\mathbf{x}, \mathbf{w}, \mathbf{b}$	Véc-tơ (ký hiệu in đậm)	SVR, Kalman, LSTM
$\mathbf{A}, \mathbf{W}, \mathbf{P}$	Ma trận (ký hiệu in đậm hoa)	Kalman, LSTM, Transformer
$(\cdot)^T$	Phép chuyển vị (transpose)	Toàn văn
Mô hình XGBoost		
η	Tốc độ học (learning rate)	Gradient Boosting
$f_k(x)$	Hàm của cây quyết định thứ k	XGBoost
w_j	Giá trị trọng số tại nút lá thứ j	XGBoost
T	Số lượng nút lá trong một cây	Điều chuẩn mô hình
$\Omega(f)$	Hàm điều chuẩn (regularization term)	XGBoost
λ, α	Hệ số phạt L2 và L1	XGBoost
Chuỗi thời gian và ký hiệu khác		
Y_t	Giá trị quan sát tại thời điểm t	ARIMA
$\hat{y}_i$	Giá trị dự báo cho mẫu thứ i	Đánh giá mô hình
ε	Sai số / nhiễu (noise)	SVR, ARIMA
ϕ_i, θ_j	Tham số tự hồi quy và trung bình trượt	ARIMA

DANH SÁCH HÌNH VẼ

Hình 3.1 Phân phối thời lượng chuyến đi (đơn vị: phút) 37
Hình 3.2 Phân phối vận tốc trung bình của các chuyến taxi (mph) 38
Hình 3.3 Tần suất các chuyến taxi theo giờ trong ngày 38
Hình 3.4 Ma trận tương quan giữa các đặc trưng chính 39
Hình 3.5 So sánh hiệu năng giữa các mô hình: RMSE, MAE, MAPE và thời gian huấn luyện 42
Hình 3.6 Đường hội tụ của mô hình XGBoost (Training vs Validation RMSE) 43
Hình 3.7 Phân phối sai số dự báo của mô hình XGBoost 44
Hình 3.8 Biểu đồ SHAP thể hiện mức độ đóng góp của các đặc trưng vào dự đoán 45

DANH SÁCH BẢNG

Bảng 3.1 Thống kê mô tả bộ dữ liệu ban đầu sau lọc cơ bản..... 35
Bảng 3.2 Thống kê mô tả bộ dữ liệu sau khi loại bỏ ngoại lệ bằng z-score 35
Bảng 3.3 Danh sách các đặc trưng mới được tạo ra..... 37
Bảng 3.4 So sánh hiệu năng giữa các mô hình trên tập kiểm thử 41

1 Lý do chọn đề tài

Trong bối cảnh đô thị hóa và sự phát triển nhanh của các dịch vụ vận tải thông minh như Grab và Uber, việc dự báo thời gian di chuyển (ETA – Estimated Time of Arrival) trở thành yếu tố then chốt để tối ưu hóa điều hành và nâng cao trải nghiệm người dùng. Các phương pháp thống kê truyền thống thường gặp khó khăn khi xử lý dữ liệu giao thông biến động mạnh và phi tuyến. Do đó, đề tài lựa chọn mô hình học máy XGBoost nhằm xây dựng một giải pháp dự báo ETA có độ chính xác cao và khả năng ứng dụng thực tiễn.

2 Tổng quan về vấn đề nghiên cứu

Nhiều nghiên cứu trước đây sử dụng các mô hình thống kê như ARIMA hay Kalman Filter, nhưng các phương pháp này bị hạn chế bởi giả định tuyến tính nên khó mô tả các quan hệ phức tạp trong dữ liệu giao thông. Sự phát triển của học máy với Random Forest, Gradient Boosting và XGBoost, cũng như các mô hình học sâu như LSTM hay GNN, đã mở ra khả năng cải thiện đáng kể chất lượng dự báo. Trong số đó, XGBoost được lựa chọn làm phương pháp chính nhờ sự cân bằng giữa hiệu năng, độ chính xác, khả năng mở rộng với dữ liệu lớn và tính giải thích được của mô hình.

3 Mục đích nghiên cứu

Mục đích của đề tài là xây dựng một mô hình học máy có khả năng dự báo chính xác thời gian di chuyển trong môi trường đô thị bằng cách khai thác các đặc trưng không gian–thời gian và ngữ cảnh. Nghiên cứu tập trung vào việc thiết kế quy trình khai phá dữ liệu có hệ thống, bao gồm làm sạch dữ liệu, xử lý ghép bản đồ, xây dựng đặc trưng, huấn luyện mô hình XGBoost và đánh giá thông qua các chỉ số RMSE, MAE và MAPE. Ngoài ra, phương pháp SHAP được sử dụng để phân tích vai trò của từng đặc trưng, giúp mô hình đảm bảo tính diễn giải.

4 Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là mô hình XGBoost và các kỹ thuật khai phá dữ liệu hỗ trợ dự báo thời gian di chuyển của phương tiện. Phạm vi nghiên cứu tập trung vào bộ dữ liệu taxi công cộng New York (NYC Taxi) năm 2024 với hơn 2 triệu bản ghi. Dữ liệu được phân tích trong điều kiện lịch sử và bán tĩnh, chưa mở rộng sang dữ liệu thời gian thực hoặc các yếu tố dự báo theo ngữ cảnh động tại thời điểm hiện tại.

5 Phương pháp nghiên cứu

Đề tài sử dụng phương pháp tiếp cận định lượng kết hợp với kỹ thuật khai phá dữ liệu và học máy. Quá trình nghiên cứu gồm các bước: (1) thu thập và tiền xử lý dữ liệu từ NYC Taxi Dataset; (2) xây dựng đặc trưng không gian, thời gian và ngữ cảnh; (3) huấn luyện mô hình XGBoost với tối ưu siêu tham số; (4) đánh giá mô hình bằng các chỉ số RMSE, MAE, MAPE; và (5) phân tích mô hình bằng SHAP nhằm đảm bảo khả năng diễn giải và hỗ trợ triển khai thực tế.

CHƯƠNG 1: BÀI TOÁN DỰ ĐOÁN THỜI GIAN VẬN TẢI HÀNH KHÁCH

1.1 Giới thiệu bài toán dự đoán thời gian vận tải hành khách

1.1.1 Định nghĩa và mục tiêu của bài toán

Bài toán dự đoán thời gian vận tải hành khách (Estimated Time of Arrival – ETA) là một trong những vấn đề trọng tâm trong lĩnh vực giao thông thông minh (Intelligent Transportation Systems – ITS). Mục tiêu chính của bài toán là ước lượng chính xác khoảng thời gian mà một phương tiện vận tải, chẳng hạn như taxi hoặc xe buýt, sẽ mất để hoàn thành hành trình từ điểm đón đến điểm trả khách trong điều kiện giao thông thực tế.

Trong phạm vi của đề án này, nghiên cứu tập trung vào **dự báo ngắn hạn (Short-Term Forecasting)**, tức là ước lượng thời gian di chuyển trong khoảng vài phút đến một giờ kế tiếp. Loại dự báo này có ý nghĩa thực tiễn cao trong các hệ thống điều phối vận tải thời gian thực (real-time dispatching), đặc biệt đối với các dịch vụ taxi công nghệ, nơi việc ra quyết định phân bổ phương tiện cần được thực hiện nhanh chóng và chính xác.

Bản chất của bài toán là tìm một hàm hồi quy $f(x)$ mô tả mối quan hệ giữa thời gian di chuyển y và các đặc trưng đầu vào x (chẳng hạn như vị trí, thời điểm, loại phương tiện hoặc điều kiện giao thông).

Dự báo ngắn hạn đóng vai trò then chốt trong việc hỗ trợ ra quyết định tức thời, giúp tối ưu hóa quá trình điều phối và phân bổ phương tiện, nâng cao hiệu quả khai thác đội xe, giảm thời gian chờ đợi của hành khách và tăng độ tin cậy của dịch vụ. Trong bối cảnh đô thị có mật độ giao thông cao, việc phát triển mô hình dự báo chính xác theo thời gian thực là một thách thức quan trọng nhưng cần thiết để hướng tới hệ thống giao thông thông minh và bền vững.

1.1.2 Ứng dụng giải quyết bài toán trong thực tế

Bài toán dự báo thời gian di chuyển (ETA – *Estimated Time of Arrival*) không chỉ mang ý nghĩa cung cấp thông tin cho người sử dụng, mà còn đóng vai trò trung tâm trong việc tối ưu hóa chi phí vận hành trong các hệ thống vận tải hành khách và logistics. Về bản chất, mối liên hệ giữa ETA và chi phí vận tải là mối quan hệ nhân quả trực tiếp trong hàm mục tiêu kinh tế của doanh nghiệp.

Trong thực tế, tổng chi phí vận hành C không chỉ bao gồm chi phí nhiên liệu, mà là hàm tổng hợp của nhiều thành phần:

$$C = C_{\text{nhiên liệu}} + C_{\text{nhân sự}} + C_{\text{cơ hội}}$$

trong đó ETA đóng vai trò là tham số đầu vào quan trọng ảnh hưởng đồng thời đến cả ba

thành phần chi phí trên. Một mô hình dự báo ETA chính xác cho phép tối ưu hóa hàm chi phí C thông qua các cơ chế sau:

- **Giảm thời gian rỗng (Idle time):** Dự báo ETA chính xác giúp hệ thống điều phối phương tiện đến điểm đón theo nguyên tắc *Just-in-Time*, hạn chế tình trạng xe đến sớm và chờ đợi trong khi động cơ vẫn hoạt động, từ đó giảm lãng phí nhiên liệu và chi phí vận hành không tạo ra giá trị.
- **Tối ưu vòng quay phương tiện:** ETA đáng tin cậy cho phép nhà quản lý xây dựng lịch trình xếp chồng các chuyến (*trip-chaining*) một cách chặt chẽ hơn, rút ngắn thời gian chết giữa các chuyến và tăng số lượt phục vụ trên cùng một nguồn lực tài xế, qua đó giảm chi phí nhân sự trên mỗi chuyến đi.
- **Giảm chi phí cơ hội:** Việc giảm sai số dự báo ETA giúp hạn chế các tổn thất do trễ giờ, hủy chuyến hoặc khiếu nại từ khách hàng, đồng thời tránh các khoản phạt hợp đồng trong logistics và nâng cao mức độ hài lòng của người sử dụng dịch vụ.

Do đó, bài toán dự báo ETA không chỉ dừng lại ở việc nâng cao độ chính xác của mô hình học máy, mà về bản chất chính là bài toán tối ưu hóa hiệu quả kinh tế và biên lợi nhuận cho doanh nghiệp vận tải thông qua việc ra quyết định dựa trên dữ liệu.

1.1.3 Các nghiên cứu liên quan trong lịch sử

Đề án tập trung trình bày tổng quan các nghiên cứu liên quan đến bài toán dự đoán thời gian di chuyển (Estimated Time of Arrival – ETA), bao gồm các phương pháp thống kê truyền thống, các mô hình học máy, và các mô hình học sâu hiện đại. Các nghiên cứu trước đây chủ yếu hướng tới việc tận dụng dữ liệu lịch sử hành trình, dữ liệu giao thông theo thời gian thực, và đặc trưng không gian – thời gian để ước lượng chính xác thời điểm một phương tiện sẽ đến đích. Các kỹ thuật được áp dụng trải dài từ mô hình hồi quy tuyến tính, hồi quy Poisson, cho tới các mô hình nâng cao như Random Forest, Gradient Boosting, và các mạng nơ-ron sâu (Deep Neural Networks – DNN) [1, 2].

Từ năm 2015 trở đi, sự phát triển mạnh mẽ của cảm biến GPS, dữ liệu bản đồ số, và hệ thống thu thập dữ liệu giao thông đã tạo điều kiện để các mô hình học sâu dựa trên kiến trúc mạng nơ-ron hồi tiếp (RNN) và mạng nơ-ron tích chập (CNN) được áp dụng rộng rãi trong lĩnh vực ETA. Nghiên cứu của Wang và cộng sự (2018) đã đề xuất mô hình *DeepTTE* – một mạng học sâu end-to-end kết hợp cả thông tin theo chuỗi (LSTM) và đặc trưng bản đồ, đạt kết quả cải thiện RMSE từ 20% đến 30% so với các mô hình học máy truyền thống [28]. Tương tự, Li và cộng sự (2019) đã giới thiệu mô hình *ST-NN* (Spatio-Temporal Neural Network), tận dụng CNN để trích xuất đặc trưng không gian và LSTM để học phụ thuộc thời gian, đạt MAE giảm trung bình 15% so với baseline Gradient Boosting [3].

Theo tổng hợp của đề án, các phương pháp ETA có thể được chia thành hai loại chính. Loại thứ nhất dựa trên mô hình huấn luyện trước hoặc mô hình nền tảng (pre-trained / foundation models) về dữ liệu giao thông, sau đó tinh chỉnh (fine-tune) hoặc kết hợp với thuật toán khác để dự đoán ETA. Loại thứ hai là các mô hình end-to-end được huấn luyện trực tiếp từ dữ liệu đầu vào đến kết quả dự đoán thời gian đến đích.

(1) Mô hình sử dụng đặc trưng từ mô hình huấn luyện trước. Ví dụ, Xu và cộng sự (2020) đã sử dụng mô hình *Graph Convolutional Network (GCN)* huấn luyện trước trên dữ liệu bản đồ giao thông để sinh đặc trưng tuyến đường, sau đó kết hợp với *Gradient Boosted Decision Trees (GBDT)* để dự đoán ETA cho dịch vụ giao hàng trong đô thị, đạt MAE giảm 12% so với Random Forest. Nghiên cứu của Ke và cộng sự (2017) cũng cho thấy việc kết hợp đặc trưng từ bản đồ và dữ liệu GPS lịch sử với mô hình XGBoost giúp giảm sai số dự đoán từ 8.2 phút xuống còn 6.5 phút trung bình cho taxi tại Bắc Kinh [6, 2].

(2) Mô hình end-to-end. Trong đó, toàn bộ pipeline từ dữ liệu thô (chuỗi GPS, điều kiện giao thông, thời tiết) đến ETA được học trực tiếp bởi một mô hình học sâu. Nghiên cứu *DeepETA* của Google (2020) áp dụng kiến trúc kết hợp LSTM và Attention để dự đoán thời gian đến của phương tiện trong Google Maps, đạt cải thiện 50% độ chính xác ETA trong các tuyến đường phức tạp so với mô hình trước đó [4]. Tương tự, Bai và cộng sự (2021) đã đề xuất mô hình *Transformer-based ETA (TETA)*, khai thác quan hệ dài hạn giữa các điểm GPS, đạt RMSE giảm 18% so với DeepTTE trên bộ dữ liệu taxi New York City [5].

Ngoài ra, một số nghiên cứu gần đây tập trung vào việc tích hợp thông tin theo thời gian thực như tình trạng tắc nghẽn giao thông, sự kiện thời tiết, hoặc dữ liệu từ thiết bị IoT để nâng cao độ chính xác dự đoán. Ví dụ, Zhang và cộng sự (2022) đã bổ sung dữ liệu thời tiết và sự kiện xã hội vào mô hình *Gated Graph Neural Network*, giúp cải thiện độ chính xác ETA thêm 9% trên các tuyến giao hàng đông đúc [25]. Những kết quả này chứng minh rằng việc áp dụng các mô hình học sâu và kỹ thuật khai thác dữ liệu đa nguồn không chỉ nâng cao hiệu suất dự đoán ETA mà còn giúp mô hình thích ứng tốt với biến động giao thông phức tạp. Đây là nền tảng quan trọng để tiếp tục cải tiến các hệ thống dự đoán thời gian di chuyển trong tương lai.

1.2 Các phương pháp dự đoán truyền thống trước đây

1.2.1 Phương pháp dựa trên thống kê

Các mô hình thống kê cổ điển đóng vai trò nền tảng trong bài toán dự báo thời gian vận tải hành khách. Trước khi các kỹ thuật học máy và học sâu phổ biến, những mô hình tuyến tính này vừa dễ triển khai vừa tạo chuẩn so sánh (baseline) cho các phương pháp hiện đại.

a) Mô hình ARIMA/SARIMA

Mô hình (Autoregressive Integrated Moving Average – ARIMA) là một trong những phương pháp thống kê kinh điển và được sử dụng rộng rãi nhất cho bài toán dự báo chuỗi thời gian. Cơ sở lý thuyết của ARIMA dựa trên giả định rằng giá trị tương lai của một chuỗi thời gian có thể được biểu diễn dưới dạng một hàm tuyến tính của các quan sát và sai số dự báo trong quá khứ. Mô hình này bao gồm ba thành phần chính:

- *Tự hồi quy (Autoregressive – AR):* Mô hình hóa mối quan hệ giữa một quan sát hiện tại và một số lượng các quan sát trước đó (độ trễ).
- *Tích hợp (Integrated – I):* Sử dụng phép sai phân (differencing) để làm cho chuỗi thời gian trở nên dừng (stationary), tức là các tính chất thống kê như trung bình và phương sai không thay đổi theo thời gian.
- *Trung bình trượt (Moving Average – MA):* Mô hình hóa mối quan hệ giữa một quan sát hiện tại và sai số dự báo của các quan sát trước đó.

Một mô hình ARIMA không có thành phần mùa vụ được ký hiệu là ARIMA(p,d,q), trong đó p, d, và q lần lượt là bậc của các thành phần AR, I và MA. Dạng toán học tổng quát của mô hình là

$$\left(1-\sum_{i=1}^{p}\phi_i B^i\right)(1-B)^d Y_t = c+\left(1+\sum_{j=1}^{q}\theta_j B^j\right)\varepsilon_t. \tag{1.1}$$

Trong đó :

- Y_t là giá trị của chuỗi tại thời điểm t.
- B là toán tử lùi ($BY_t = Y_{t-1}$).
- ϕ_i và θ_j là các tham số của mô hình.
- c là hằng số.
- ε_t là nhiễu trắng (white noise).

Quy trình xây dựng mô hình ARIMA thường tuân theo phương pháp luận Box–Jenkins, bao gồm các bước:

- *Nhận dạng mô hình (Identification)*: Kiểm tra tính dừng của chuỗi thời gian thông qua việc quan sát biểu đồ chuỗi và các hàm tự tương quan (ACF) và tự tương quan riêng phần (PACF). Nếu chuỗi không dừng, thực hiện phép sai phân (tham số d) cho đến khi chuỗi trở nên dừng. Dựa vào hình dạng của ACF và PACF của chuỗi đã dừng, xác định các bậc p và q tiềm năng.

- *Ước lượng tham số (Estimation)*: Sử dụng các phương pháp như hợp lý cực đại (Maximum Likelihood) để ước lượng các tham số ϕ_i và θ_j của mô hình đã nhận dạng.
- *Kiểm định chẩn đoán (Diagnostic Checking)*: Kiểm tra xem mô hình có phù hợp hay không bằng cách phân tích phần dư (residuals). Phần dư của một mô hình tốt phải tuân theo phân phối nhiễu trắng, tức là không có tự tương quan. Các kiểm định thống kê như Ljung–Box có thể được sử dụng cho mục đích này.

Để xử lý các chuỗi thời gian có tính mùa vụ rõ rệt (ví dụ các mẫu hình lặp lại hàng ngày hoặc hàng tuần trong dữ liệu giao thông), mô hình ARIMA được mở rộng thành mô hình ARIMA Mùa vụ (*Seasonal ARIMA – SARIMA*). SARIMA bổ sung thêm các thành phần mùa vụ $(P,D,Q)_m$, trong đó P, D, Q là các bậc AR, I, MA của thành phần mùa vụ và m là độ dài của một chu kỳ mùa vụ.

Các mô hình ARIMA và SARIMA tương đối đơn giản, dễ triển khai và có khả năng diễn giải cao. Các tham số của mô hình cung cấp cái nhìn rõ ràng về cách các giá trị trong quá khứ ảnh hưởng đến dự báo tương lai. Chúng hoạt động hiệu quả đối với các chuỗi thời gian có xu hướng (trend) và tính mùa vụ (seasonality) rõ ràng và thường được sử dụng làm một mô hình cơ sở (baseline) mạnh mẽ để so sánh với các phương pháp phức tạp hơn.

Hạn chế lớn nhất của ARIMA/SARIMA là bản chất tuyến tính của chúng. Các mô hình này không thể nắm bắt được các mối quan hệ phi tuyến phức tạp vốn có trong hệ thống giao thông đô thị. Chúng chỉ xem xét sự phụ thuộc về mặt thời gian của một chuỗi duy nhất và hoàn toàn bỏ qua các tương tác không gian—tức là ảnh hưởng từ các tuyến đường hoặc các phương tiện khác trong mạng lưới. Do đó, hiệu suất của chúng suy giảm đáng kể khi có các sự kiện bất thường (sự cố, thời tiết xấu) hoặc những thay đổi đột ngột trong điều kiện giao thông mà không được phản ánh trong dữ liệu lịch sử. Một số nghiên cứu thực nghiệm thậm chí còn cho thấy rằng mô hình SARIMAX (SARIMA với các biến ngoại sinh) có hiệu suất kém hơn so với các mô hình chuỗi thời gian khác trong một số trường hợp cụ thể [7].

b) Bộ lọc Kalman

Bộ lọc Kalman, do Rudolf E. Kálmán phát triển, là một thuật toán đệ quy tối ưu được thiết kế để ước tính trạng thái của một hệ thống động học tuyến tính từ một chuỗi các phép đo có chứa nhiễu. Trong bối cảnh dự đoán ETA, hệ thống động học chính là chiếc xe buýt, và trạng thái của nó bao gồm các biến như vị trí và vận tốc. Bộ lọc Kalman hoạt động theo một chu trình hai bước: dự báo và cập nhật. Nó liên tục cập nhật dự đoán về trạng thái của xe buýt bằng cách kết hợp một cách tối ưu dự đoán từ mô hình chuyển động (dựa trên trạng thái trước đó) với dữ liệu đo lường thực tế (ví dụ từ GPS/AVL). Mô hình không gian trạng thái của bộ lọc Kalman

bao gồm hai phương trình chính: phương trình quá trình và phương trình đo lường. Thuật toán hoạt động theo hai bước:

(1) *Bước Dự báo (Prediction/Time Update):* Bộ lọc dự báo trạng thái và hiệp phương sai lỗi của hệ thống tại thời điểm hiện tại, dựa trên ước tính từ thời điểm trước đó.

- Dự báo trạng thái tiên nghiệm:

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} + B_t u_t \tag{1.2}$$

- Dự báo hiệp phương sai lỗi tiên nghiệm:

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^\top + Q_t \tag{1.3}$$

(2) *Bước Cập nhật (Update/Measurement Update):* Sau khi có được phép đo mới y_t, bộ lọc sẽ hiệu chỉnh lại dự báo tiên nghiệm để có được ước tính hậu nghiệm (đã được cải thiện).

- Tính độ lợi Kalman:

$$K_t = P_{t|t-1} H_t^\top \left(H_t P_{t|t-1} H_t^\top + R_t \right)^{-1} \tag{1.4}$$

- Cập nhật ước tính trạng thái hậu nghiệm:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \left(y_t - H_t \hat{x}_{t|t-1} \right) \tag{1.5}$$

- Cập nhật hiệp phương sai lỗi hậu nghiệm:

$$P_{t|t} = (I - K_t H_t) P_{t|t-1} \tag{1.6}$$

Trong đó $\hat{x}$ là vector trạng thái, P là ma trận hiệp phương sai lỗi, F_t là ma trận chuyển đổi trạng thái, B_t là ma trận điều khiển, u_t là vector điều khiển, Q_t là ma trận hiệp phương sai nhiễu quá trình, H_t là ma trận đo lường, R_t là ma trận hiệp phương sai nhiễu đo lường và y_t là vector đo lường.

Các đặc tính vận hành của bộ lọc Kalman có thể khái quát qua những điểm sau.

- *Điểm mạnh:* Bộ lọc Kalman cực kỳ hiệu quả trong việc lọc nhiễu từ dữ liệu cảm biến (như GPS) và cung cấp các ước tính trạng thái được hiệu chỉnh liên tục trong thời gian thực. Thuật toán có tốc độ tính toán nhanh do các phương trình đơn giản, làm cho nó rất phù hợp cho các ứng dụng trực tuyến (online).
- *Điểm yếu:* Giả định cơ bản của bộ lọc Kalman tiêu chuẩn về tính tuyến tính của hệ thống và nhiễu tuân theo phân phối Gauss là một hạn chế lớn. Giao thông đô thị là một hệ thống có tính phi tuyến cao, làm giảm hiệu quả của KF. Mặc

dù các biến thể như Bộ lọc Kalman Mở rộng (Extended Kalman Filter – EKF) và Bộ lọc Kalman Không mùi (Unscented Kalman Filter – UKF) đã được phát triển để xử lý tính phi tuyến, chúng vẫn còn những giới hạn và có thể không ổn định.

Vượt ra ngoài vai trò là một mô hình dự báo đơn thuần, bộ lọc Kalman thực chất là một framework tích hợp, đại diện cho một bước nhảy vọt về mặt khái niệm từ các mô hình ngoại tuyến (offline) như ARIMA sang các hệ thống trực tuyến (online) có khả năng tự hiệu chỉnh. Nó đóng vai trò như một cầu nối giữa thế giới thống kê thuần túy và các hệ thống điều khiển thời gian thực. Trong khi ARIMA được huấn luyện trên dữ liệu lịch sử và đưa ra dự báo mà không có cơ chế sửa sai tức thời, KF được thiết kế với một vòng lặp "dự báo–đo lường–hiệu chỉnh". Nó đưa ra một dự đoán, so sánh nó với một phép đo thực tế (ví dụ: vị trí GPS mới), và sử dụng sự khác biệt (sai số) để điều chỉnh lại ước tính của mình. Do đó, KF không chỉ là một mô hình dự báo mà còn là một "bộ lọc" và "bộ ước tính trạng thái", làm cho nó trở thành một thành phần quan trọng trong nhiều hệ thống ITS, thường được kết hợp với các mô hình khác (ví dụ: mạng nơ-ron kết hợp KF) để tận dụng ưu điểm của cả hai.

Nhìn chung, các phương pháp thống kê cổ điển vẫn là chuẩn tham chiếu quan trọng, song hạn chế trong việc mô hình hóa quan hệ phi tuyến và tương tác không gian–thời gian phức tạp của hệ thống giao thông hiện đại.

1.2.2 Phương pháp dựa trên mô hình học máy

Các mô hình học máy truyền thống đã nổi lên như những giải pháp thay thế mạnh mẽ cho hạn chế của các phương pháp thống kê trong việc mô hình hóa quan hệ phi tuyến và đặc trưng phức tạp. Dù không xử lý cấu trúc không–thời gian tinh vi như học sâu, chúng đạt hiệu quả tốt với dữ liệu lượng vừa phải và thường đem lại độ chính xác cao.

a) Hồi quy k-Láng giềng gần nhất (k-Nearest Neighbors – kNN Regression).

kNN là phương pháp học máy đơn giản, trực quan và thuộc nhóm "học lười", tức không xây dựng một mô hình tường minh trong giai đoạn huấn luyện. Khi dự báo cho một điểm dữ liệu mới (ví dụ đặc trưng hành trình hiện tại), thuật toán tìm k điểm dữ liệu tương tự nhất trong tập huấn luyện theo một thước đo khoảng cách (thường dùng khoảng cách Euclidean). Giá trị cần dự đoán thường được lấy trung bình (hoặc trung bình có trọng số) của các láng giềng này.

Kết quả dự đoán của kNN được tính theo công thức:

$$\hat{y}(\boldsymbol{x}) = \frac{\sum_{i=1}^{k} w_i(\boldsymbol{x})\, y_i}{\sum_{i=1}^{k} w_i(\boldsymbol{x})}, \tag{1.7}$$

Trong đó:

- $\boldsymbol{x}$ là điểm dữ liệu mới cần dự đoán (ví dụ đặc trưng hành trình hiện tại).
- y_i là giá trị mục tiêu quan sát được của điểm dữ liệu thứ i trong tập k láng giềng gần nhất.
- $w_i(\boldsymbol{x})$ là trọng số gán cho láng giềng thứ i. Thông thường $w_i(\boldsymbol{x})$ được chọn là nghịch đảo của khoảng cách $d(\boldsymbol{x}, \boldsymbol{x}_i)$ (ví dụ $w_i = 1/d$) để những láng giềng gần hơn ảnh hưởng mạnh hơn; nếu sử dụng kNN không trọng số thì $w_i(\boldsymbol{x}) = 1$.
- $d(\boldsymbol{x}, \boldsymbol{x}_i)$ là hàm đo khoảng cách giữa điểm mới và láng giềng thứ i, phổ biến nhất là khoảng cách Euclid.

Các đặc tính của hồi quy kNN có thể khái quát như sau:

- *Điểm mạnh:* Dễ hiểu, dễ triển khai, không giả định phân phối dữ liệu nên linh hoạt với các mối quan hệ phi tuyến phức tạp.
- *Điểm yếu:* Chi phí tính toán cao ở giai đoạn dự đoán vì phải so sánh với toàn bộ tập dữ liệu huấn luyện; kém hiệu quả với dữ liệu thời gian thực và chịu ảnh hưởng của "lời nguyền chiều" (curse of dimensionality) nếu đặc trưng không được chuẩn hóa tốt.

b) Hồi quy Vector Hỗ trợ (Support Vector Regression – SVR).

SVR là biến thể của máy Vector Hỗ trợ (Support Vector Machine – SVM) ứng dụng cho hồi quy. Khác với hồi quy tuyến tính, SVR tìm một "ống" (tube) tối ưu quanh dữ liệu sao cho sai số nằm trong biên ε không bị phạt, trong khi sai số vượt ngoài biên sẽ chịu mức phạt theo chuẩn cấu trúc (Structural Risk Minimization). Nhờ thủ thuật hạt nhân (kernel trick), SVR ánh xạ dữ liệu lên không gian đặc trưng cao hơn để tìm siêu phẳng tuyến tính chia tách dữ liệu hiệu quả. Các hạt nhân thường dùng gồm tuyến tính, đa thức và RBF.

Bài toán tối ưu của SVR được phát biểu như sau:

$$\begin{aligned} \underset{\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) \le \varepsilon + \xi_i, \\ & (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) - y_i \le \varepsilon + \xi_i^*, \\ & \xi_i,\, \xi_i^* \ge 0, \quad i = 1, \ldots, n. \end{aligned} \tag{1.8}$$

Trong đó:

- $\mathbf{x}_i$ là vector đặc trưng của quan sát thứ i; y_i là giá trị mục tiêu tương ứng.
- $\phi(\cdot)$ là ánh xạ (kernel) đưa dữ liệu vào không gian đặc trưng cao hơn.
- $\mathbf{w}$ và b là trọng số và sai số dịch (bias) của siêu phẳng hồi quy.
- ε là biên dung sai; sai số nằm trong "ống" có độ rộng 2ε không bị phạt.
- ξ_i, ξ_i^* là biến trượt (slack variables) cho phép vi phạm biên dung sai.
- $C > 0$ là hệ số điều chỉnh mức phạt, cân bằng giữa độ phẳng của siêu phẳng và sai số huấn luyện.
- n là số mẫu huấn luyện.

Giải bài toán trên bằng phương pháp đối ngẫu sẽ dẫn tới dạng hạt nhân của SVR, trong đó siêu phẳng tối ưu được xác định bởi các vector hỗ trợ.

Các đặc tính của hồi quy kNN có thể khái quát như sau:

- *Điểm mạnh:* Khả năng khái quát tốt ngay cả với dữ liệu ít, nhờ điều khiển rủi ro quá khớp; nhiều nghiên cứu cho thấy SVR vượt trội hơn một số mô hình hồi quy khác và cả mạng nơ-ron khi dữ liệu không quá lớn.
- *Điểm yếu:* Việc lựa chọn hàm hạt nhân và tinh chỉnh siêu tham số (C, ε, hệ số hạt nhân) khá phức tạp và tốn thời gian; chi phí huấn luyện tăng nhanh nếu dữ liệu có kích thước lớn.

b) Cây Phân loại và Hồi quy (Classification and Regression Trees – CART).

CART là một trong những thuật toán học máy nền tảng và được sử dụng rộng rãi cho cả hai bài toán phân loại (*classification*) và hồi quy (*regression*). Phương pháp này được đề xuất bởi Breiman và cộng sự (1984), dựa trên ý tưởng chia tách không gian đặc trưng thành các vùng con (subspaces) sao cho mỗi vùng có độ thuần nhất cao nhất về giá trị mục tiêu. Mô hình CART có dạng cấu trúc cây nhị phân, trong đó mỗi nút (*node*) đại diện cho một phép kiểm tra điều kiện trên một đặc trưng, các nhánh (*branches*) thể hiện kết quả của phép kiểm tra, và các lá (*leaves*) chứa giá trị dự đoán cuối cùng.

Đối với bài toán hồi quy (như dự đoán thời gian di chuyển), CART tìm kiếm ngưỡng chia s trên đặc trưng x_j sao cho sai số bình phương tổng (*sum of squared errors – SSE*) của hai tập con sau chia nhỏ nhất. Mục tiêu tối ưu được biểu diễn như sau:

$$\min_{j,s} \left[\sum_{x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2 \right], \tag{1.9}$$

trong đó:

- $R_1(j,s) = \{x \mid x_j \leq s\}$ và $R_2(j,s) = \{x \mid x_j > s\}$ là hai vùng con được chia bởi ngưỡng s trên đặc trưng x_j.
- $\bar{y}_{R_1}$ và $\bar{y}_{R_2}$ lần lượt là giá trị trung bình của biến mục tiêu trong hai vùng con tương ứng.

Quá trình huấn luyện của CART được thực hiện đệ quy từ gốc đến lá (*top-down recursive partitioning*) cho đến khi đạt điều kiện dừng, ví dụ như số lượng mẫu trong nút quá nhỏ hoặc độ giảm sai số không đáng kể. Để tránh hiện tượng quá khớp (*overfitting*), mô hình thường được cắt tỉa (*pruning*) bằng cách loại bỏ các nhánh kém quan trọng dựa trên tiêu chí xác thực chéo (*cross-validation*) hoặc độ phức tạp mô hình.

Các đặc tính của phương pháp CART có thể khái quát như sau:

- *Điểm mạnh:* Dễ diễn giải, có thể xử lý cả dữ liệu số và phân loại, tự động chọn đặc trưng quan trọng; hoạt động tốt với dữ liệu phi tuyến và có khả năng mô hình hóa tương tác giữa các đặc trưng.
- *Điểm yếu:* Nhạy cảm với nhiễu và dữ liệu huấn luyện nhỏ; cây đơn lẻ thường dễ bị overfitting, do đó trong thực tế thường được kết hợp trong các mô hình ensemble như Random Forest hoặc Gradient Boosting để tăng độ ổn định và chính xác.

Nhờ khả năng biểu diễn trực quan và dễ hiểu, CART thường được xem là bước trung gian quan trọng trong việc phát triển các mô hình phức tạp hơn như Random Forest, XGBoost hoặc LightGBM trong bài toán dự đoán thời gian di chuyển (ETA) [8].

1.3 Các phương pháp dự đoán hiện đại

1.3.1 Ứng dụng mô hình học máy kết hợp (Ensemble Learning)

Các phương pháp ensemble kết hợp nhiều mô hình "yếu" (thường là cây quyết định) để tạo ra mô hình "mạnh" duy nhất có hiệu suất dự báo cao hơn.

a) Rừng ngẫu nhiên (Random Forest – RF).

Rừng ngẫu nhiên là một phương pháp học máy truyền thống thuộc nhóm mô hình tập hợp (*ensemble learning*), được đề xuất bởi Breiman (2001) như một mở rộng của thuật toán CART. Ý tưởng cốt lõi của Random Forest là kết hợp nhiều cây quyết định (decision trees) được huấn luyện độc lập trên các mẫu dữ liệu ngẫu nhiên, sau đó tổng hợp (ensemble) kết quả dự đoán của chúng để giảm phương sai và tăng độ chính xác tổng thể của mô hình.

Cơ chế huấn luyện của RF dựa trên kỹ thuật *bootstrap aggregating* (bagging). Giả sử tập dữ liệu huấn luyện ban đầu có N mẫu, mô hình sẽ tạo ra B cây quyết định, mỗi cây được huấn luyện trên một tập con bootstrap (lấy mẫu ngẫu nhiên có hoàn lại) gồm N phần tử. Tại mỗi nút của cây, chỉ một tập con ngẫu nhiên gồm m đặc trưng trong tổng số p đặc trưng được chọn để tìm ngưỡng chia tốt nhất, giúp tăng tính ngẫu nhiên và giảm hiện tượng quá khớp.

Đối với bài toán hồi quy (như dự đoán thời gian di chuyển), kết quả dự đoán của Random Forest được tính bằng trung bình của các dự đoán từ từng cây thành phần:

$$\hat{y}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} T_b(\boldsymbol{x}), \tag{1.10}$$

trong đó:

- $T_b(\boldsymbol{x})$ là giá trị dự đoán của cây thứ b đối với điểm dữ liệu $\boldsymbol{x}$.
- B là tổng số cây trong rừng ngẫu nhiên.

Phương pháp này giúp giảm phương sai của mô hình đơn lẻ (CART) mà không làm tăng đáng kể sai lệch (bias). Ngoài ra, RF còn cung cấp cơ chế đánh giá tầm quan trọng của đặc trưng (*feature importance*) dựa trên độ giảm trung bình của sai số hoặc chỉ số Gini, hỗ trợ việc giải thích mô hình trong các bài toán thực tế.

Các đặc tính của phương pháp Random Forest có thể khái quát như sau:

- *Điểm mạnh:* Mạnh mẽ trước nhiễu và dữ liệu thiếu; có khả năng tổng quát tốt nhờ cơ chế bagging; hoạt động hiệu quả trên dữ liệu phi tuyến và dữ liệu có chiều cao; tự động ước lượng tầm quan trọng của đặc trưng.
- *Điểm yếu:* Chi phí tính toán cao hơn CART; khó diễn giải trực quan khi số lượng cây lớn; đôi khi có xu hướng dự đoán trung bình (làm mờ các quan hệ cực trị).

Trong bài toán dự đoán thời gian di chuyển (ETA), Random Forest thường được sử dụng như một mô hình baseline mạnh, đặc biệt khi dữ liệu có số lượng vừa phải và các đặc trưng đã được thiết kế thủ công. RF cũng là nền tảng cho các phương pháp học máy nâng cao hơn như Gradient Boosting và XGBoost.

d) Phương pháp Tăng cường Gradient (Gradient Boosting) và XGBoost.

Gradient Boosting là một phương pháp học máy truyền thống thuộc nhóm *ensemble learning*, được phát triển dựa trên ý tưởng kết hợp tuần tự nhiều mô hình yếu (*weak learners*) — thường là các cây quyết định nhỏ (shallow trees) — để tạo thành một

mô hình mạnh (*strong learner*). Phương pháp này được giới thiệu lần đầu bởi Friedman (2001), và sau này được cải tiến trong nhiều biến thể hiện đại như XGBoost, LightGBM và CatBoost.

Khác với Random Forest (kết hợp song song nhiều cây độc lập theo cơ chế *bagging*), Gradient Boosting xây dựng mô hình theo cách tuần tự (*sequential learning*), trong đó mỗi cây mới được huấn luyện để sửa lỗi của tổ hợp các cây trước đó. Cụ thể, tại mỗi bước lặp m, mô hình học phần dư (*residual*) giữa giá trị thực tế và dự đoán hiện tại, sau đó cập nhật dự đoán tổng thể bằng cách cộng thêm kết quả của cây mới có trọng số nhỏ η (learning rate).

Dạng tổng quát của mô hình Gradient Boosting được biểu diễn như sau:

$$F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \eta\, h_m(\boldsymbol{x}), \tag{1.11}$$

trong đó:

- $F_m(\boldsymbol{x})$: mô hình sau bước lặp thứ m,
- $\eta \in (0,1]$: hệ số học (learning rate) điều chỉnh tốc độ cập nhật,
- $h_m(\boldsymbol{x})$: cây quyết định mới được huấn luyện trên gradient âm của hàm mất mát tại bước $m-1$.

Quá trình tối ưu của Gradient Boosting có thể hiểu là tìm dãy mô hình h_m để cực tiểu hóa hàm mất mát tổng thể:

$$\min_{F} \sum_{i=1}^{N} L(y_i, F(\boldsymbol{x}_i)), \tag{1.12}$$

trong đó $L(y_i, F(\boldsymbol{x}_i))$ là hàm mất mát, thường là sai số bình phương đối với hồi quy hoặc log-loss đối với phân loại.

XGBoost (Extreme Gradient Boosting) là phiên bản cải tiến mạnh mẽ của Gradient Boosting do Chen và Guestrin (2016) phát triển. Thuật toán này bổ sung các cơ chế tối ưu như:

- **Regularization:** thêm thành phần phạt L_1/L_2 để kiểm soát độ phức tạp của cây, giúp giảm overfitting.
- **Shrinkage:** sử dụng hệ số học nhỏ để cập nhật dần mô hình, tránh cập nhật quá mạnh.
- **Column Subsampling:** chọn ngẫu nhiên một phần đặc trưng tại mỗi cây, tăng tính đa dạng tương tự Random Forest.
- **Parallelization:** tối ưu hóa tính toán trên GPU hoặc multi-core CPU, giúp huấn luyện nhanh hơn nhiều so với Gradient Boosting truyền thống.

XGBoost được xem là một trong những thuật toán mạnh nhất trong các cuộc thi trên Kaggle và được ứng dụng rộng rãi trong thực tế, đặc biệt là các bài toán dự đoán giá trị liên tục như thời gian di chuyển (ETA), tiêu thụ năng lượng, hay giá tài chính. Trong bài toán ETA, XGBoost có khả năng mô hình hóa tốt các quan hệ phi tuyến giữa đặc trưng không gian – thời gian, đồng thời vẫn đảm bảo tốc độ huấn luyện cao và khả năng giải thích tương đối tốt thông qua các chỉ số tầm quan trọng đặc trưng (*feature importance*) hoặc giá trị SHAP.

Các đặc tính của Gradient Boosting và XGBoost có thể khái quát như sau:

- *Điểm mạnh:* Hiệu năng dự đoán cao; khả năng mô hình hóa phi tuyến tốt; hỗ trợ regularization và song song hóa; hoạt động hiệu quả với dữ liệu vừa và lớn; cung cấp thước đo quan trọng của đặc trưng (SHAP, gain, cover).
- *Điểm yếu:* Nhạy cảm với tham số siêu (learning rate, depth, số cây); dễ quá khớp nếu không điều chỉnh tốt; khó diễn giải khi số cây lớn; thời gian huấn luyện cao hơn mô hình tuyến tính hoặc kNN.

Nhờ sự cân bằng giữa hiệu quả dự đoán, tốc độ và khả năng giải thích, XGBoost thường được sử dụng làm mô hình chủ đạo (*baseline model*) trong nhiều nghiên cứu và ứng dụng dự đoán thời gian di chuyển (ETA) trước khi so sánh với các mô hình học sâu như LSTM hay Transformer [19, 26].

Sự trỗi dậy của các phương pháp ensemble giúp nâng cao hiệu suất cao của các hệ thống dự báo ETA. Đặc biệt là XGBoost, đã đánh dấu một bước ngoặt quan trọng. Chúng đại diện cho cách học "hộp đen" mạnh mẽ với khả năng "hợp tan" (black-box) trong khi bước vào xử lý nguyên cấu theo sâu. Tuy nhiên, thành công này cũng đi kèm với một số đánh đổi: khả năng diễn giải giảm. Trong khi các mô hình thẳng tuyến tính thường rất minh bạch—học giả có thể biết tầm quan trọng và hướng tác động của đặc trưng một cách trực tiếp—thì một mô hình XGBoost bao gồm hàng trăm cây quyết định lại gần như không thể diễn giải trực tiếp. Không dễ để trả lời câu hỏi "Tại sao mô hình dự đoán chuyến xe này sẽ trễ 10 phút?". Sự minh bạch bị hạn chế là một trở ngại trong các hệ thống quan trọng và có yêu cầu giải thích, nơi các nhà quản lý cần hiểu nhiều nguyên nhân của sự chậm trễ để có thể can thiệp một cách hiệu quả. Khoảng trống về khả năng diễn giải này đã trực tiếp thúc đẩy sự ra đời và áp dụng rộng rãi của các phương pháp giải thích mô hình hậu kiểm như LIME và SHAP. Có thể thấy, với những điểm yếu này cũng không khác các mô hình hộp đen khác, nhưng chính hiệu quả dự đoán vượt trội đã giúp XGBoost trở thành lựa chọn hàng đầu trong nhiều hệ thống dự báo thời gian.

1.3.2 Ứng dụng mô hình học sâu

Học sâu đã tạo ra một cuộc cách mạng trong nhiều lĩnh vực, và dự báo giao thông cũng không ngoại lệ. Các kiến trúc mạng nơ-ron sâu có khả năng tự động học các biểu diễn đặc trưng phức tạp từ dữ liệu thô, đặc biệt là các mẫu hình không–thời gian—thách thức lớn đối với các phương pháp truyền thống.

a) Mạng Bộ nhớ Dài–Ngắn hạn (Long Short-Term Memory – LSTM).

LSTM là một biến thể đặc biệt của mạng nơ-ron hồi tiếp (*Recurrent Neural Network – RNN*), được Hochreiter và Schmidhuber đề xuất năm 1997 nhằm khắc phục vấn đề *vanishing gradient* trong quá trình huấn luyện chuỗi dài. Khác với mạng RNN truyền thống, LSTM được thiết kế với cơ chế "bộ nhớ" giúp lưu giữ thông tin quan trọng trong chuỗi dữ liệu theo thời gian dài, đồng thời loại bỏ các thông tin không cần thiết thông qua các cổng điều khiển (*gates*).

Trong bài toán dự đoán thời gian di chuyển (ETA), dữ liệu thường có tính chuỗi mạnh (ví dụ: vị trí GPS liên tục, vận tốc theo thời gian, hoặc luồng giao thông theo phút). Do đó, LSTM trở thành một trong những mô hình học sâu hiệu quả nhất để học các quan hệ phụ thuộc theo thời gian (*temporal dependencies*) trong dữ liệu.

Một khối LSTM điển hình bao gồm ba cổng chính:

- **Cổng quên (Forget Gate)** – xác định thông tin nào từ trạng thái trước cần được loại bỏ.
- **Cổng đầu vào (Input Gate)** – quyết định thông tin mới nào được thêm vào bộ nhớ.
- **Cổng đầu ra (Output Gate)** – xác định thông tin nào được sử dụng để sinh ra đầu ra hiện tại.

Các phương trình tính toán của một đơn vị LSTM tại thời điểm t được mô tả như sau:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{1.13}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{1.14}$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \tag{1.15}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{1.16}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{1.17}$$

$$h_t = o_t * \tanh(C_t), \tag{1.18}$$

trong đó:

- x_t: đầu vào tại thời điểm t (ví dụ: đặc trưng GPS, tốc độ, thời tiết),
- h_{t-1}: đầu ra ẩn từ bước trước,
- C_t: trạng thái bộ nhớ tại thời điểm t,
- f_t, i_t, o_t: giá trị của các cổng quên, đầu vào và đầu ra tương ứng,
- W_* và b_*: ma trận trọng số và vector bias của từng cổng,
- σ: hàm sigmoid, tanh: hàm kích hoạt phi tuyến.

Nhờ khả năng ghi nhớ và học các mẫu phụ thuộc dài hạn trong dữ liệu chuỗi, LSTM thường được sử dụng để dự báo chuỗi thời gian, nhận dạng giọng nói, và đặc biệt là dự đoán thời gian đến (ETA) trong giao thông thông minh. Một biến thể phổ biến khác là **Bi-directional LSTM (BiLSTM)**, cho phép mô hình học quan hệ theo cả hai chiều thời gian (quá khứ và tương lai), cải thiện độ chính xác trong các tập dữ liệu có bối cảnh phức tạp.

Các đặc tính của phương pháp LSTM có thể khái quát như sau:

- *Điểm mạnh:* Mô hình hóa hiệu quả dữ liệu chuỗi có phụ thuộc dài hạn; xử lý tốt các tín hiệu động theo thời gian; thích hợp cho dữ liệu có thứ tự như GPS, vận tốc, hay mật độ giao thông; hoạt động tốt khi kết hợp với CNN hoặc Attention để trích xuất đặc trưng không gian–thời gian.
- *Điểm yếu:* Chi phí huấn luyện cao; khó điều chỉnh siêu tham số; dễ overfitting khi dữ liệu huấn luyện nhỏ; khả năng song song hóa thấp so với các kiến trúc Transformer.

Trong các nghiên cứu hiện đại về ETA, LSTM thường đóng vai trò là nền tảng cho nhiều mô hình kết hợp, chẳng hạn như *ST-NN (Spatio-Temporal Neural Network)* hoặc *DeepTTE (Deep Travel Time Estimation)* — trong đó LSTM được dùng để xử lý chuỗi đặc trưng theo thời gian, trong khi các tầng CNN hoặc Attention đảm nhiệm phần trích xuất đặc trưng không gian. Điều này giúp mô hình đạt độ chính xác cao hơn đáng kể so với các phương pháp học máy truyền thống như Random Forest hoặc XGBoost [27].

b) Mô hình Transformer và Cơ chế Chú ý (Attention-based ETA Models).

Transformer là một trong những kiến trúc mạng nơ-ron học sâu tiên tiến nhất hiện nay, được Vaswani và cộng sự (2017) giới thiệu trong công trình "Attention Is All You Need". Khác với các mô hình hồi tiếp (RNN, LSTM) dựa vào lan truyền trạng thái tuần tự, Transformer dựa hoàn toàn trên cơ chế *Self-Attention* để nắm bắt mối quan hệ giữa các phần tử trong chuỗi dữ liệu mà không cần xử lý theo thứ tự thời gian cố định.

Trong bài toán dự đoán thời gian di chuyển (ETA), Transformer và các biến thể dựa trên Attention như *DeepETA*, *ST-Transformer*, hay *TETA (Transformer-based ETA)* đã cho thấy hiệu năng vượt trội so với các mô hình truyền thống và LSTM, đặc biệt trong việc học các quan hệ không gian–thời gian phức tạp từ dữ liệu GPS, bản đồ và tín hiệu giao thông.

Cơ chế *Self-Attention* cho phép mô hình đánh giá mức độ quan trọng tương đối giữa các phần tử trong chuỗi đầu vào. Cho ma trận đầu vào $X = [x_1, x_2, \ldots, x_n]$, các vector truy vấn ($Q$), khóa ($K$) và giá trị ($V$) được sinh ra thông qua phép biến đổi tuyến tính:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \tag{1.19}$$

trong đó W_Q, W_K, W_V là các ma trận trọng số được học. Trọng số chú ý (*attention weights*) được tính bằng:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1.20}$$

với d_k là kích thước vector khóa. Cơ chế này cho phép mô hình tập trung vào các phần tử đầu vào có liên quan nhất đến điểm dự đoán hiện tại.

Trong thực tế, kiến trúc Transformer thường gồm nhiều lớp *Multi-Head Attention*, mỗi lớp học một không gian biểu diễn khác nhau của mối quan hệ giữa các điểm dữ liệu. Ngoài ra, các thành phần như *Position Encoding* được thêm vào để cung cấp thông tin về thứ tự thời gian cho chuỗi đầu vào, vốn bị mất đi do tính chất song song của Transformer.

Ứng dụng trong ETA: Các mô hình như **DeepETA** (Google, 2020) và **TETA** (Bai et al., 2021) mở rộng Transformer để học đồng thời đặc trưng không gian và thời gian. DeepETA kết hợp các tầng LSTM và Attention để mô hình hóa chuỗi GPS của phương tiện, trong khi TETA sử dụng hoàn toàn kiến trúc Transformer, khai thác mối quan hệ dài hạn giữa các điểm GPS, thời gian và điều kiện giao thông. Kết quả thực nghiệm cho thấy TETA đạt giảm RMSE tới 18% so với DeepTTE và 25–30% so với các mô hình học máy truyền thống.

Các đặc tính của phương pháp Transformer / Attention-based ETA có thể khái quát như sau:

- *Điểm mạnh:* Khả năng học phụ thuộc dài hạn vượt trội; xử lý song song tốt; mô hình hóa quan hệ không gian–thời gian hiệu quả; dễ mở rộng trên dữ liệu lớn; đạt độ chính xác cao trong ETA thực tế (Google Maps, Grab, Uber).
- *Điểm yếu:* Đòi hỏi dữ liệu và tài nguyên tính toán lớn; khó giải thích kết quả; yêu cầu tinh chỉnh nhiều siêu tham số; dễ quá khớp nếu dữ liệu không đủ đa dạng.

Nhờ cơ chế Attention linh hoạt và khả năng xử lý song song, Transformer đã trở thành nền tảng của hầu hết các mô hình ETA hiện đại. Các nghiên cứu gần đây còn mở rộng hướng này sang các biến thể như *Graph Transformer Networks (GTN)* hay *Spatial-Temporal Transformer*, cho phép học trực tiếp trên đồ thị giao thông thực, mở ra triển vọng lớn trong việc dự đoán thời gian di chuyển chính xác trong bối cảnh đô thị thông minh [20, 21, 22, 23, 24].

Các mô hình học sâu như LSTM và Transformer đã đánh dấu bước tiến vượt bậc trong bài toán dự đoán thời gian di chuyển (ETA), nhờ khả năng học biểu diễn phi tuyến phức tạp và khai thác đồng thời mối quan hệ không gian – thời gian trong dữ liệu giao thông.

LSTM thể hiện ưu thế rõ rệt trong việc xử lý chuỗi dữ liệu có tính liên tục theo thời gian, đặc biệt khi mô hình cần ghi nhớ các trạng thái quá khứ dài (như chuỗi GPS, tốc độ di chuyển, hoặc luồng xe theo phút). Với cấu trúc cổng bộ nhớ (forget, input, output gates), LSTM duy trì cân bằng giữa việc lưu giữ và quên thông tin, giúp mô hình hóa chính xác các quan hệ phụ thuộc động học trong quá trình di chuyển. Tuy nhiên, hạn chế chính của LSTM là khả năng song song hóa thấp và chi phí tính toán cao khi độ dài chuỗi tăng, làm giảm hiệu quả trong các hệ thống ETA quy mô lớn.

Ngược lại, Transformer và các biến thể dựa trên Attention như DeepETA, TETA hay ST-Transformer đã khắc phục điểm yếu này bằng cách loại bỏ lan truyền tuần tự và thay thế bằng cơ chế *Self-Attention*. Nhờ đó, Transformer có thể học quan hệ giữa mọi điểm dữ liệu trong chuỗi cùng lúc, giúp tăng tốc độ huấn luyện và đạt độ chính xác cao hơn trong các bối cảnh giao thông phức tạp. Cấu trúc song song của Transformer đặc biệt phù hợp với dữ liệu lớn và đa nguồn (GPS, cảm biến, bản đồ, thời tiết), đồng thời cho phép tích hợp tốt hơn với các kiến trúc hiện đại khác như Graph Neural Networks hoặc Multi-head Attention.

1.4 Đề xuất phương hướng

1.4.1 Khai phá dữ liệu trong bài toán dự đoán thời gian di chuyển (ETA)

Trong bối cảnh giao thông đô thị phức tạp, kỹ thuật khai phá dữ liệu giữ vai trò nền tảng để dự đoán chính xác thời gian di chuyển (ETA). Hệ thống ETA hiện đại cần khai thác khối lượng dữ liệu lớn và đa dạng về không gian–thời gian, từ dữ liệu GPS/AVL của phương tiện, cảm biến giao thông, thời tiết đến bản đồ số và sự kiện đặc biệt. Việc tích hợp nhiều nguồn dữ liệu giúp mô hình ETA bao quát đầy đủ các yếu tố ảnh hưởng; nghiên cứu cho thấy việc đưa thêm yếu tố ngoại cảnh như thời tiết hay sự kiện (ví dụ: lễ hội, tai nạn) vào mô hình đã cải thiện đáng kể độ chính xác dự báo.

Ngược lại, cách tiếp cận chỉ dựa trên một loại dữ liệu thường bỏ sót tương quan ẩn

giữa các đối tượng giao thông. Do vậy, khai phá dữ liệu đa nguồn trở thành chìa khóa để mô hình nắm bắt mối quan hệ tiềm ẩn trong hệ thống giao thông phức tạp.

Bên cạnh đó, dữ liệu thu thập thô thường nhiễu và không đầy đủ; nếu không được xử lý, chúng có thể dẫn đến suy luận sai lệch. Khai phá dữ liệu đảm bảo 5V của dữ liệu lớn (Volume, Velocity, Variety, Value, Veracity), đặc biệt là tính đúng đắn (veracity) của dữ liệu – yếu tố then chốt quyết định hiệu quả của các mô hình thông minh.

Tóm lại, khai phá dữ liệu đặt nền móng cho toàn bộ quy trình ETA: từ chuẩn bị dữ liệu đầu vào chất lượng đến khám phá kiến thức ẩn, giúp mô hình máy học dự báo thời gian di chuyển một cách tin cậy và hiệu quả.

Quy trình khai phá dữ liệu cho bài toán ETA có thể chia thành các nhóm bước chính như sau:

a) **Thu thập và tích hợp dữ liệu đa nguồn** Dữ liệu GPS/AVL cung cấp quỹ đạo chi tiết theo thời gian thực; cảm biến giao thông đo đếm lưu lượng và tốc độ dòng xe; dữ liệu thời tiết và sự kiện phản ánh điều kiện môi trường. Việc tích hợp này đòi hỏi đồng bộ thời gian và không gian, tạo ra bức tranh toàn cảnh về giao thông. Các hệ thống như MDTP đã chứng minh hiệu quả khi kết hợp nhiều nguồn quỹ đạo khác nhau.

b) **Làm sạch và xử lý dữ liệu bất thường** Dữ liệu giao thông chứa nhiễu GPS, điểm ngoại lệ hoặc thiếu sót. Các kỹ thuật như bộ lọc Kalman, trung bình trượt hoặc mô hình học máy được sử dụng để loại bỏ sai số và đồng bộ hóa dữ liệu thời gian–không gian. Khi dữ liệu bị thiếu, các phương pháp nội suy hoặc điền khuyết được áp dụng để bảo đảm tính nhất quán.

c) **Map–matching và khai thác quỹ đạo (trajectory mining)** Sai số định vị GPS khiến tọa độ có thể lệch khỏi đường thực tế. Thuật toán Markov ẩn (HMM) hoặc công cụ mã nguồn mở Valhalla được sử dụng để ánh xạ điểm GPS lên đoạn đường khả dĩ nhất, tái tạo hành trình thực tế. Sau đó, kỹ thuật phân cụm không–thời gian như ST–DBSCAN giúp phát hiện cụm quỹ đạo tương đồng và loại bỏ hành trình dị thường.

d) **Tạo đặc trưng (Feature Engineering)**

Quá trình tạo đặc trưng (*Feature Engineering*) đóng vai trò quan trọng trong việc nâng cao hiệu quả của mô hình dự đoán thời gian di chuyển. Đây là giai đoạn nhằm bổ sung các thông tin tiềm ẩn không có sẵn trong dữ liệu gốc, đồng thời tinh lọc và tổ chức lại các biến đầu vào theo hướng giúp mô hình học tốt hơn mối quan hệ giữa các yếu tố ảnh hưởng đến thời gian di chuyển.

e) Phát hiện mẫu và lựa chọn đặc trưng

Trước khi đưa vào huấn luyện, tập dữ liệu được làm sạch và tinh chỉnh để loại bỏ các bản ghi bất thường, chẳng hạn như các chuyến đi có thời lượng quá ngắn hoặc quá dài, hoặc tốc độ trung bình vượt mức giới hạn cho phép. Đồng thời, bước lựa chọn đặc trưng (*feature selection*) được tiến hành dựa trên phân tích tầm quan trọng và tương quan giữa các biến. Các đặc trưng có mức ảnh hưởng thấp như `vendor_id` hay `passenger_count` được loại bỏ, trong khi các đặc trưng tương quan cao chỉ giữ lại một biến đại diện như `main_street_ratio` để tránh trùng lặp thông tin.

Sau quá trình xử lý, tập đặc trưng cuối cùng được rút gọn còn khoảng 11 biến có giá trị thông tin cao nhất. Các đặc trưng này bao gồm cả thông tin tĩnh (như cấu trúc đường, điểm đón và điểm trả) và thông tin động (như thời gian, mật độ giao thông, thời tiết), giúp mô hình có cái nhìn toàn diện hơn về bối cảnh di chuyển.

Các đặc trưng sau khi được tạo ra được chuẩn hóa và mã hóa (*encoding, normalization*) nhằm đảm bảo đồng nhất về tỷ lệ giá trị và tăng hiệu quả huấn luyện của mô hình. Việc xây dựng một hệ thống đặc trưng giàu thông tin và có khả năng khái quát cao là yếu tố then chốt giúp cải thiện độ chính xác của mô hình dự đoán thời gian di chuyển.

Nhờ áp dụng quy trình khai phá dữ liệu một cách bài bản, độ chính xác và hiệu năng của hệ thống ETA đô thị được cải thiện rõ rệt. Việc tích hợp đa nguồn dữ liệu giúp giảm sai số dự báo; dữ liệu sạch và đồng bộ giúp mô hình tránh bị nhiễu. Ngoài ra, quy trình xử lý hợp lý còn đảm bảo khả năng dự báo gần thời gian thực trên quy mô hàng chục triệu bản ghi mỗi ngày.

Tóm lại, kỹ thuật khai phá dữ liệu đóng vai trò trung tâm trong toàn bộ chuỗi dự báo ETA. Sự kết hợp giữa khai phá dữ liệu chất lượng cao và các mô hình học máy hiện đại hứa hẹn mang lại hệ thống dự báo thời gian di chuyển ngày càng chính xác, thích ứng và đáng tin cậy trong tương lai.

1.4.2 Sử dụng học máy kết hợp Ensemble Learning

Qua tổng quan các phương pháp đã trình bày, có thể nhận thấy rằng mỗi hướng tiếp cận – từ các mô hình học máy truyền thống như kNN, CART, Random Forest cho đến các mô hình học sâu hiện đại như LSTM và Transformer – đều mang những ưu điểm và hạn chế riêng. Các mô hình học sâu cho phép học biểu diễn không gian–thời gian phức tạp, song đòi hỏi lượng dữ liệu huấn luyện lớn, tài nguyên tính toán cao và khó giải thích kết quả. Trong khi đó, các mô hình truyền thống có khả năng triển khai nhanh, dễ diễn giải nhưng đôi khi bị giới hạn về độ chính xác trong các quan hệ phi tuyến phức tạp.

Trong bối cảnh đó, **thuật toán XGBoost (Extreme Gradient Boosting)** được lựa

chọn làm mô hình đề xuất cho đồ án vì đạt được sự *cân bằng tối ưu* giữa hiệu suất dự đoán, độ phức tạp tính toán và khả năng giải thích. XGBoost là một phương pháp *ensemble learning* thuộc nhóm Boosting, kết hợp nhiều cây quyết định yếu thành một mô hình mạnh bằng cách huấn luyện tuần tự và giảm dần sai số của các vòng trước. So với Random Forest (dựa trên Bagging), XGBoost không chỉ giảm phương sai mà còn tối ưu sai lệch (bias) thông qua cơ chế học có trọng số, giúp cải thiện đáng kể độ chính xác.

Về mặt kỹ thuật, XGBoost áp dụng chiến lược *gradient boosting* với hàm mất mát khả vi, cơ chế regularization (L1, L2) để kiểm soát độ phức tạp mô hình, và kỹ thuật *shrinkage* (learning rate) giúp tránh hiện tượng quá khớp. Bên cạnh đó, khả năng xử lý dữ liệu thiếu, hỗ trợ song song hóa, và tính mở rộng tốt trên các tập dữ liệu lớn giúp XGBoost đặc biệt phù hợp với các bài toán dự đoán thời gian di chuyển (ETA) sử dụng dữ liệu giao thông thực tế.

Một ưu điểm quan trọng khác của XGBoost là khả năng *đánh giá tầm quan trọng của đặc trưng* (*feature importance*), giúp mô hình không chỉ đưa ra dự đoán chính xác mà còn cung cấp thông tin hữu ích về yếu tố nào ảnh hưởng mạnh nhất đến thời gian di chuyển (ví dụ: khoảng cách, thời gian trong ngày, tình trạng giao thông). Điều này rất có giá trị trong bối cảnh nghiên cứu, nơi việc diễn giải kết quả và xác định yếu tố tác động là cần thiết.

Cuối cùng, để khai thác tối đa tiềm năng của XGBoost, đồ án này kết hợp phương pháp với các **kỹ thuật tạo và chọn đặc trưng (feature engineering)** nhằm nâng cao khả năng biểu diễn dữ liệu đầu vào. Quá trình này bao gồm trích xuất các đặc trưng không gian–thời gian (ví dụ: khoảng cách Haversine, thời điểm trong ngày, ngày trong tuần), kết hợp đặc trưng định tuyến (OSRM), và loại bỏ các biến ít ảnh hưởng dựa trên phân tích tương quan. Sự kết hợp giữa một mô hình ensemble mạnh mẽ và một pipeline đặc trưng được thiết kế cẩn thận kỳ vọng sẽ mang lại hiệu năng dự đoán cao, ổn định và có thể giải thích được – đáp ứng tốt mục tiêu của đồ án về dự đoán thời gian di chuyển cho dữ liệu taxi đô thị.

1.5 Kết luận chương

Chương 1 đã hoàn thành việc giới thiệu khái quát về bài toán dự đoán thời gian di chuyển ước tính (ETA), các ứng dụng thực tế và tổng quan về các phương pháp giải quyết, từ thống kê truyền thống đến học sâu hiện đại. Các mô hình thống kê kinh điển như ARIMA/SARIMA và Bộ lọc Kalman bị hạn chế bởi giả định về tính tuyến tính của dữ liệu, khiến chúng không thể nắm bắt được các mối quan hệ phi tuyến phức tạp vốn có trong giao thông đô thị. Trong khi đó, các kiến trúc học sâu (GNN, Transformer) mặc dù mạnh mẽ trong việc xử lý các mẫu hình không–thời gian phức tạp, lại thường đòi hỏi tài nguyên tính toán và dữ liệu lớn để huấn luyện. Học tăng cường Gradient (GBM), đặc

biệt là phiên bản tối ưu XGBoost, nổi bật lên là mô hình được ưu tiên hàng đầu nhờ khả năng mô hình hóa các mối quan hệ cực kỳ phức tạp và thường xuyên đạt được độ chính xác hàng đầu trong các bài toán dự báo.

Để xử lý khối lượng dữ liệu khổng lồ, GBM đã được mở rộng thành các hệ thống phân tán, chứng minh khả năng mở rộng quy mô lên hàng trăm GB dữ liệu và đạt độ chính xác vượt trội so với các mô hình hồi quy tuyến tính hoặc mạng nơ-ron truyền thống trong dự báo ETA. Thành công này được xây dựng trên nền tảng của kỹ thuật khai phá dữ liệu bài bản, bao gồm việc thu thập, làm sạch nhiễu GPS, xử lý ghép bản đồ (Map-matching), và tạo đặc trưng không–thời gian chất lượng cao, nhằm cung cấp dữ liệu đầu vào tin cậy cho mô hình học máy.

Tóm lại, hiệu quả dự đoán vượt trội, khả năng xử lý phi tuyến, và tính mở rộng của Gradient Boosting phân tán là cơ sở vững chắc để chuyển sang Chương 2, nơi sẽ tập trung chi tiết vào kỹ thuật khai phá dữ liệu và áp dụng Mô hình dự đoán XGBoost để xây dựng mô hình dự báo thời gian vận chuyển hành khách.

CHƯƠNG 2: HUẤN LUYỆN MÔ HÌNH DỰ BÁO THỜI GIAN VẬN TẢI HÀNH KHÁCH

Chương này trình bày quy trình huấn luyện mô hình dự báo thời gian vận tải hành khách sử dụng thuật toán XGBoost, bao gồm các bước khai phá dữ liệu không–thời gian, xây dựng đặc trưng, lựa chọn mô hình, tinh chỉnh siêu tham số và đánh giá hiệu suất. Mục tiêu là thiết lập nền tảng kỹ thuật vững chắc để phân tích kết quả và rút ra kết luận trong Chương 3.

2.1 Kỹ thuật khai phá dữ liệu

Trên các bộ dữ liệu NYC Taxi (*Trip Duration*, *Taxi Fare*), những lời giải đạt thứ hạng cao trên Kaggle đều nhấn mạnh một quy trình khai phá dữ liệu chặt chẽ, kết hợp tái cấu trúc quỹ đạo, rút gọn hình học, làm giàu ngữ cảnh không–thời gian và bổ sung thông tin mạng lưới đường. Mục tiêu của quy trình là chuyển đổi toạ độ GPS và mốc thời gian thô thành một không gian đặc trưng giàu ý nghĩa, ổn định phương sai và có khả năng khái quát trên các điều kiện giao thông khác nhau. Nội dung dưới đây hệ thống hoá các bước chính và nhóm đặc trưng cốt lõi đã chứng minh hiệu quả trong thực nghiệm [29, 15].

2.1.1 Xử lý ghép bản đồ – Map matching

Dữ liệu GPS đô thị chịu ảnh hưởng nhiễu đa nguồn, do đó cần ánh xạ các điểm quan sát lên đồ thị đường thực (*road network*) để khử lệch vị trí và khôi phục tuyến đường hợp lý. Cách tiếp cận phổ biến là mô hình ẩn Markov (HMM), trong đó mỗi điểm GPS được quy chiếu sang một tập ứng viên cạnh đường với xác suất phát xạ phụ thuộc sai số đo, còn xác suất chuyển tiếp phản ánh ràng buộc hình học và động học giữa các cạnh kề nhau [11].

Bên cạnh ánh xạ HMM, các lời giải đứng hạng cao còn khai thác các dịch vụ định tuyến mã nguồn mở (ví dụ OSRM) để suy ra các đại lượng theo mạng lưới, như thời gian lái xe tối ưu, chiều dài lộ trình theo đường, số lần rẽ, số nút giao và số đèn tín hiệu trên tuyến. Những biến này mang tính diễn giải cao hơn khoảng cách đường chim bay, giúp mô hình tiếp cận gần hơn với chi phí thời gian thực tế [13].

2.1.2 Sử dụng thuật toán Ramer–Douglas–Peucker

Sau ghép bản đồ, quỹ đạo vẫn có thể dày đặc điểm. Thuật toán *Ramer–Douglas–Peucker (RDP)* được áp dụng để nén hình, giữ lại tập điểm "xương sống" đại diện cho hình dạng tuyến với sai số biên kiểm soát được. Việc rút gọn này làm giảm độ phức tạp tính toán cho các phép suy luận theo mạng lưới, đồng thời cải thiện tính ổn định của các

đặc trưng hình học (góc phương vị, độ uốn khúc, chỉ số ngoằn ngoèo) trước nhiễu vị trí cục bộ [12].

2.1.3 Biến đổi dữ liệu và tạo đặc trưng mới

Các đặc trưng không gian cơ bản thường bắt đầu từ khoảng cách hình học giữa điểm đón và trả. Khoảng cách *haversine* mô tả độ dài cung tròn trên mặt cầu, trong khi khoảng cách "Manhattan" (*taxicab distance*) xấp xỉ tốt cho lưới phố trực giao của New York. Góc phương vị (*bearing*) chuẩn hoá hướng chuyển động, còn tỷ số thẳng tuyến (khoảng cách đường thẳng chia cho độ dài lộ trình theo mạng) phản ánh mức độ ngoằn ngoèo. Trong bối cảnh thành phố có nhiều điểm hấp dẫn (POIs) ảnh hưởng luồng xe, khoảng cách tới các sân bay lớn (JFK, LGA, EWR), trọng tâm Manhattan hay trung tâm thương mại được sử dụng như biến ngữ cảnh cố định. Các biến nhị phân nhận diện "chuyến sân bay" giúp mô hình học các chế độ lưu thông đặc thù [30].

Các đặc trưng thời gian đóng vai trò trục chính để mô hình hoá chu kỳ giao thông. Thực hành tốt bao gồm tách giờ trong ngày, ngày trong tuần, tháng trong năm; mã hoá các khung giờ cao điểm sáng/chiều; thêm cờ ngày cuối tuần và ngày nghỉ lễ theo lịch Hoa Kỳ; và, khi sẵn có, kết hợp điều kiện thời tiết đồng bộ theo mốc thời gian (nhiệt độ, lượng mưa/tuyết, gió). Nhiều lời giải áp dụng biến đổi *log1p* cho biến mục tiêu thời gian nhằm ổn định phân phối lệch phải và giảm ảnh hưởng ngoại lai, đồng thời chuẩn hoá hoặc chuẩn vị (*standardization*) cho các thước đo khoảng cách để tăng hiệu quả tối ưu hoá.

Đặc trưng theo mạng lưới đường cho thấy đóng góp nổi trội khi dự báo thời gian. Bên cạnh thời gian và cự ly lái xe theo OSRM, số lần rẽ trái/phải, số nút giao và số phân đoạn đường (*segments*) trên tuyến là các biến gắn trực tiếp với chi phí trễ thực tế. Tại các nút giao dày đặc, thêm *proxy* cho mật độ đèn tín hiệu hoặc giới hạn tốc độ mặc định theo phân loại đường càng tăng tính mô tả của mô hình [14].

Đặc trưng mật độ–đếm (*counting features*) được xây dựng bằng cách phân mảnh không gian theo lưới đều, *geohash* hoặc cụm K-means trên toạ độ đón/trả. Với mỗi ô hoặc cụm, ta tính các thống kê lịch sử như số chuyến, tốc độ trung bình, thời gian trung vị theo khung giờ; những biến này thể hiện nền tắc nghẽn cục bộ và xu thế theo thời gian [16]. Kỹ thuật "đếm có điều kiện" (ví dụ đếm số chuyến trong cùng giờ và cùng cụm) giúp mô hình phát hiện hiệu ứng tải cục bộ.

Đặc trưng phân cụm còn được tận dụng ở mức cao hơn bằng cách gán nhãn cụm cho điểm đón và trả, rồi xây dựng tương tác "cặp cụm" (*origin cluster* $\times$ *destination cluster*). Cách làm này cho phép mô hình học các tuyến OD phổ biến như "Midtown $\rightarrow$ JFK" với hành vi thời gian ổn định. Khi cần tính gọn, *geohash* ở nhiều độ phân giải đóng vai trò phân cấp không gian, bổ sung hiệu ứng đa tỉ lệ cho mô hình.

Các đặc trưng tương tác được hình thành có chủ đích nhằm nắm bắt phi tuyến giữa không gian và thời gian, tiêu biểu như tích số của khoảng cách với cờ giờ cao điểm, hoặc tương tác giữa cự ly theo mạng và lượng mưa. Việc thêm các tỷ suất tốc độ (distance/duration) lịch sử theo ô không gian–thời gian cũng giúp ràng buộc dự báo vào bối cảnh giao thông địa phương. Cuối cùng, để giảm rủi ro rò rỉ dữ liệu, mọi đặc trưng dựa trên thống kê lịch sử cần tính trong khung thời gian quá khứ so với bản ghi mục tiêu, và các phép chuẩn hoá phải lồng trong quy trình chia *train/validation* [17].

2.1.4 Phân tích mô tả và tính tương quan đặc trưng

Sau khi hình thành không gian đặc trưng, phân tích mô tả giúp thẩm định trực quan phân phối và phát hiện ngoại lai, trong khi phân tích tương quan định lượng đóng vai trò sàng lọc. Hệ số tương quan Pearson hữu ích với quan hệ tuyến tính sau các biến đổi log, còn *mutual information* và độ quan trọng hoán vị (*permutation importance*) phù hợp hơn cho quan hệ phi tuyến. Trong thực nghiệm với XGBoost, việc diễn giải thông qua *SHAP values* cho thấy các đặc trưng theo mạng lưới (thời gian/độ dài lộ trình OSRM), hướng chuyển động, chỉ báo giờ cao điểm và cụm không gian thường xếp hạng cao về mức đóng góp, trong khi các biến ngữ cảnh như thời tiết và sân bay gia tăng ổn định cho những tuyến có hành vi khác biệt. Các kết quả này củng cố luận điểm rằng kết hợp đặc trưng hình học, không gian phân cấp, ngữ cảnh thời gian và thông tin lưới đường là chìa khoá để tối ưu hoá dự báo ETA trên dữ liệu NYC [18].

2.2 Mô hình dự đoán XGBoost

XGBoost (*Extreme Gradient Boosting*) là một thuật toán học máy mạnh mẽ thuộc họ mô hình *boosting*, được thiết kế nhằm nâng cao hiệu suất dự báo thông qua việc kết hợp nhiều mô hình yếu (weak learners), thường là các cây quyết định nhỏ, thành một mô hình tổng hợp có năng lực khái quát hóa vượt trội. Trong bài toán dự đoán thời gian di chuyển (ETA), XGBoost đặc biệt hiệu quả vì có thể học được các mối quan hệ phi tuyến phức tạp giữa các đặc trưng không–thời gian, như tương tác giữa điều kiện giao thông, vị trí địa lý và khung thời gian. Mô hình này đã chứng minh ưu thế rõ rệt trong nhiều bài toán Kaggle về *trip duration prediction* và *taxi fare estimation* [29].

2.2.1 Nguồn gốc mô hình

XGBoost được phát triển bởi Tianqi Chen và Carlos Guestrin vào năm 2016 trong bài báo "XGBoost: A Scalable Tree Boosting System" [10]. Mô hình này được xem là phiên bản cải tiến và mở rộng của *Gradient Boosting Machine (GBM)* truyền thống do Friedman đề xuất năm 2001. Mục tiêu chính của XGBoost là tối ưu hóa tốc độ huấn luyện và khả năng mở rộng trên các tập dữ liệu lớn, đồng thời duy trì độ chính xác cao thông qua cơ chế chuẩn hóa (*regularization*) và xử lý song song.

Không giống các mô hình boosting trước đó chỉ tập trung vào độ chính xác, XGBoost được thiết kế với định hướng hệ thống (system optimization) toàn diện. Cụ thể, nó khai thác kỹ thuật song song hóa trong quá trình xây dựng cây, tối ưu bộ nhớ thông qua *block structure* và hỗ trợ cả CPU lẫn GPU để tăng tốc tính toán. Chính nhờ khả năng này, XGBoost nhanh chóng trở thành thuật toán tiêu chuẩn trong các cuộc thi học máy quy mô lớn như Kaggle, nơi mà tốc độ huấn luyện và độ chính xác đều là yếu tố quyết định [14, 30].

Ngoài ra, XGBoost hỗ trợ nhiều dạng bài toán khác nhau, bao gồm hồi quy, phân loại, xếp hạng và dự báo chuỗi thời gian. Trong lĩnh vực giao thông, mô hình này được sử dụng rộng rãi trong các bài toán ước lượng thời gian hành trình (ETA), dự đoán độ trễ phương tiện công cộng, và tối ưu hóa điều phối xe trong các nền tảng chia sẻ hành trình (ride-sharing platforms) như Uber, Lyft hoặc Grab.

2.2.2 Các tính năng chính

XGBoost kế thừa cơ chế học tăng cường dần (*additive boosting*) từ GBM, trong đó các mô hình yếu được huấn luyện liên tiếp để giảm sai số còn lại của mô hình hiện tại. Mỗi cây quyết định mới được xây dựng dựa trên phần dư (*residuals*) của dự đoán trước đó, và tổng hợp lại tạo thành mô hình cuối cùng. Cách tiếp cận này cho phép XGBoost dần học được các mẫu phức tạp mà các mô hình tuyến tính không thể nắm bắt.

Một trong những cải tiến cốt lõi của XGBoost là việc bổ sung điều khoản chuẩn hóa (*L1, L2 regularization*) vào hàm mục tiêu, giúp kiểm soát độ phức tạp của cây và tránh hiện tượng quá khớp (*overfitting*). Hàm mục tiêu của XGBoost có dạng:

$$\mathscr{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.21}$$

với $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_j w_j^2$, trong đó T là số nút lá của cây và w_j là trọng số tại mỗi lá. Thành phần $\Omega(f_k)$ giúp ràng buộc độ phức tạp, cho phép mô hình cân bằng giữa khả năng khớp dữ liệu và độ tổng quát hóa.

Ngoài cơ chế chuẩn hóa, XGBoost còn có nhiều đặc điểm kỹ thuật nổi bật:

- **Tối ưu hoá phân tách:** Thuật toán *Weighted Quantile Sketch* giúp xác định điểm chia (split point) hiệu quả ngay cả khi dữ liệu có trọng số không đồng đều.
- **Xử lý thiếu dữ liệu:** XGBoost tự động học hướng phân nhánh tối ưu khi gặp giá trị thiếu, do đó không cần thay thế thủ công (*imputation*).
- **Hỗ trợ tính song song:** Quá trình xây dựng cây được song song hóa theo chiều đặc trưng thay vì theo mẫu, giúp tận dụng tốt tài nguyên CPU/GPU.

- **Tính mở rộng cao:** Cấu trúc dữ liệu theo dạng cột nén (*block structure*) giúp mô hình mở rộng quy mô lên hàng trăm GB dữ liệu mà vẫn đảm bảo tốc độ huấn luyện cao [10].

Nhờ các cải tiến này, XGBoost thường đạt hiệu suất vượt trội trong các bài toán có đặc trưng phi tuyến và tương tác phức tạp, đặc biệt khi kết hợp với các kỹ thuật khai phá dữ liệu không–thời gian đã trình bày ở Mục 2.1.

2.2.3 Ứng dụng của mô hình

Trong bài toán dự đoán thời gian di chuyển của phương tiện vận tải, XGBoost thể hiện hiệu quả rõ rệt nhờ khả năng xử lý các tương tác phi tuyến giữa các đặc trưng như thời gian, vị trí, mật độ giao thông và điều kiện thời tiết. Mô hình có thể học được mối quan hệ phức tạp như tác động của giờ cao điểm tại các khu vực trung tâm, hoặc sự kết hợp giữa khoảng cách tuyến và tình trạng tắc nghẽn trong từng khung thời gian.

Các nghiên cứu gần đây và thực nghiệm từ cộng đồng Kaggle cho thấy XGBoost thường đạt sai số trung bình tuyệt đối (MAE) và sai số bình phương gốc (RMSE) thấp hơn 10–15% so với các mô hình hồi quy tuyến tính hoặc mạng nơ-ron đơn giản khi áp dụng trên bộ dữ liệu NYC Taxi [29, 15]. Việc sử dụng các đặc trưng từ mạng lưới OSRM và đặc trưng ngữ cảnh thời gian giúp mô hình nắm bắt tốt hơn các quy luật tắc nghẽn giao thông cục bộ và thời gian di chuyển theo mùa vụ.

Ngoài ra, XGBoost cho phép diễn giải kết quả thông qua các chỉ số tầm quan trọng của đặc trưng (*feature importance*) hoặc giá trị SHAP (*SHapley Additive exPlanations*) [18]. Điều này mang lại lợi thế trong môi trường triển khai thực tế, nơi các tổ chức giao thông hoặc doanh nghiệp vận tải cần hiểu rõ lý do mô hình đưa ra dự báo. Các đặc trưng quan trọng nhất trong các thực nghiệm bao gồm thời gian trong ngày, khoảng cách theo OSRM, cờ giờ cao điểm, mật độ điểm đón tại Manhattan và điều kiện thời tiết.

Cuối cùng, nhờ khả năng mở rộng và tốc độ huấn luyện cao, XGBoost được triển khai hiệu quả trên các hệ thống phân tán (Spark, Dask, Ray) để xử lý dữ liệu lớn theo thời gian thực, hỗ trợ ứng dụng trong các hệ thống gợi ý tuyến đường, dự báo thời gian ETA của xe buýt hoặc dịch vụ chia sẻ hành trình.

2.3 Phương pháp đánh giá và thực nghiệm mô hình

Sau khi hoàn thiện quy trình khai phá dữ liệu và lựa chọn mô hình XGBoost, bước tiếp theo là xây dựng phương pháp đánh giá và thiết lập quy trình thực nghiệm. Mục tiêu của phần này là trình bày cách thức tiến hành huấn luyện, tinh chỉnh siêu tham số, lựa chọn chỉ số đánh giá và các bước kiểm chứng mô hình, làm cơ sở cho việc phân tích kết quả ở Chương 3.

2.3.1 Mục tiêu và phạm vi đánh giá

Việc đánh giá mô hình trong nghiên cứu này nhằm xác định khả năng dự báo của XGBoost trong bài toán ước lượng thời gian di chuyển (ETA), đồng thời kiểm định tính ổn định và khả năng tổng quát của mô hình trên dữ liệu có đặc trưng không–thời gian phức tạp. Phạm vi đánh giá tập trung vào:

- Độ chính xác của mô hình trong việc ước lượng thời gian di chuyển;
- Mức độ ổn định của mô hình giữa các phân đoạn thời gian và khu vực địa lý khác nhau;
- Khả năng diễn giải của mô hình thông qua các chỉ số tầm quan trọng đặc trưng.

Toàn bộ quá trình đánh giá được thiết kế theo hướng tái lập (*reproducible*) và có thể mở rộng (*scalable*), đảm bảo rằng các kết quả ở Chương 3 phản ánh chính xác hiệu quả của mô hình trong điều kiện thực tế.

2.3.2 Quy trình huấn luyện và chia tập dữ liệu

Để tối ưu hoá hiệu suất mô hình trên tập dữ liệu quy mô lớn trong điều kiện tài nguyên tính toán giới hạn, bộ dữ liệu được chia thành hai phần: *tập huấn luyện* (Training set) và *tập kiểm định* (Validation set) với tỷ lệ 80% – 20%. Cách phân chia này giúp mô hình có đủ dữ liệu để học các mẫu hình phức tạp, đồng thời vẫn giữ lại một lượng dữ liệu độc lập đủ lớn để đánh giá khả năng tổng quát hoá và hạn chế hiện tượng quá khớp (*overfitting*). Việc sử dụng *random split* là phù hợp với bản chất của bài toán ước lượng thời gian di chuyển, nơi mỗi chuyến đi được xem như một quan sát độc lập thay vì một chuỗi thời gian liên tục.

Biến mục tiêu được xác định là *thời gian di chuyển thực tế* (đơn vị phút). Việc biểu diễn thời gian dưới dạng phút thay vì giây giúp cho các chỉ số đánh giá như RMSE và MAE trở nên trực quan hơn và phản ánh rõ hơn sai số trong thực tế vận hành. Trước khi đưa vào mô hình, dữ liệu được xử lý ngoại lệ và làm giàu đặc trưng (*Feature Engineering*) nhằm chuyển đổi các thuộc tính không gian–thời gian thô thành các biến có ý nghĩa ngữ nghĩa cao, như giờ trong ngày, ngày trong tuần, khoảng cách Haversine, tốc độ trung bình ước tính, và các đặc trưng theo mùa vụ.

Quá trình huấn luyện sử dụng hàm mất mát hồi quy bình phương lỗi (*Squared Error Loss*), được cấu hình thông qua tham số `reg:squarederror` trong thư viện XGBoost. Hàm mất mát này được tối ưu hoá bằng thuật toán *Gradient Boosting*, trong đó mỗi cây quyết định mới được xây dựng nhằm giảm dần phần dư (*residuals*) mà các cây trước đó chưa giải thích được. Cơ chế học tuần tự này giúp mô hình hội tụ hiệu quả và xử lý tốt các đặc trưng phi tuyến trên không gian đặc trưng phức tạp.

2.3.3 Tinh chỉnh siêu tham số (Hyperparameter Tuning)

Việc lựa chọn và tinh chỉnh siêu tham số (*hyperparameters*) đóng vai trò quyết định đến hiệu năng của XGBoost. Trong nghiên cứu này, quá trình tinh chỉnh được thực hiện bằng phương pháp Grid Search kết hợp với kiểm định chéo 5-fold cross-validation để quét không gian tham số và chọn ra cấu hình tối ưu.

Các siêu tham số chính bao gồm:

- `max_depth`: độ sâu tối đa của cây, ảnh hưởng đến khả năng học phi tuyến;
- `learning_rate` (η): tốc độ cập nhật trọng số, điều chỉnh mức độ học của mô hình;
- `n_estimators`: số lượng cây trong mô hình boosting;
- `subsample` và `colsample_bytree`: tỷ lệ mẫu và đặc trưng được chọn ngẫu nhiên mỗi vòng, giúp giảm *overfitting*;
- `lambda` và `alpha`: các hệ số chuẩn hoá L2 và L1, kiểm soát độ phức tạp của mô hình.

Các cấu hình tham số khác nhau được thử nghiệm và đánh giá theo cùng bộ chỉ số, qua đó lựa chọn tổ hợp tham số tối ưu có độ chính xác cao nhất và mức sai số thấp nhất trên tập kiểm định.

2.3.4 Chỉ số đánh giá hiệu suất mô hình

Để đánh giá chất lượng dự báo của các mô hình, đề án sử dụng các chỉ số phổ biến trong bài toán dự đoán, bao gồm:

- **MSE (Mean Squared Error)** – đo lường sai số bằng trung bình bình phương chênh lệch giữa giá trị thực và giá trị dự đoán:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{2.22}$$

  MSE càng thấp chứng tỏ mô hình càng chính xác, trong khi MSE lớn phản ánh sai số dự báo cao.

- **MAE (Mean Absolute Error)** – đo lường sai số trung bình theo giá trị tuyệt đối:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{2.23}$$

  MAE có cùng đơn vị với biến mục tiêu, nhờ đó cho phép diễn giải trực quan về mức độ lệch trung bình giữa dự báo và thực tế.

- **MAPE (Mean Absolute Percentage Error)** – đo sai số tương đối theo phần trăm, phản ánh khả năng dự báo theo tỷ lệ chênh lệch.

Bên cạnh các chỉ số đánh giá sai số, đồ án sử dụng phương pháp giải thích mô hình **SHAP (Shapley Additive Explanations)** nhằm phân tích mức độ đóng góp của từng đặc trưng vào dự báo. SHAP dựa trên lý thuyết trò chơi để gán giá trị đóng góp cho từng đặc trưng, giúp mô hình trở nên minh bạch và dễ diễn giải hơn. Các công cụ quan trọng bao gồm:

- **SHAP values** – thể hiện mức tác động (tăng hoặc giảm) của từng đặc trưng lên kết quả dự báo;
- **Summary Plot** – biểu diễn toàn bộ SHAP values theo mức độ quan trọng và phân phối giá trị;

Việc kết hợp đồng thời các chỉ số định lượng (RMSE, MAE, MAPE) và phân tích giải thích bằng SHAP giúp đánh giá mô hình toàn diện hơn, vừa đảm bảo độ chính xác dự báo, vừa đảm bảo khả năng diễn giải cần thiết cho các ứng dụng thực tế.

Bên cạnh đó, mô hình XGBoost được phân tích bằng phương pháp *SHAP values* để đánh giá mức độ đóng góp của từng đặc trưng, giúp tăng tính minh bạch trong quá trình dự báo.

2.3.5 Phương pháp diễn giải và kiểm định mô hình

Bên cạnh việc đo lường độ chính xác, mô hình được phân tích thêm về khả năng diễn giải thông qua các chỉ số *feature importance* và *SHAP values* (*SHapley Additive exPlanations*) [18]. Các phương pháp này cho phép xác định mức đóng góp của từng đặc trưng đối với kết quả dự báo, giúp hiểu rõ hơn các yếu tố tác động đến thời gian di chuyển.

Việc kiểm định mô hình được thực hiện bằng cách so sánh hiệu suất giữa các cấu hình khác nhau (theo siêu tham số, nhóm đặc trưng hoặc chiến lược chia dữ liệu). Kết quả của các bước này sẽ được trình bày và phân tích chi tiết trong Chương 3.

2.4 Kết luận chương

Chương 2 đã hoàn thiện việc trình bày khung phương pháp luận cho bài toán dự đoán thời gian di chuyển (ETA) của phương tiện vận tải hành khách trong điều kiện giao thông đô thị phức tạp. Nội dung chương đã làm rõ vai trò then chốt của việc khai phá dữ liệu trong việc chuẩn hóa, làm sạch và biến đổi dữ liệu GPS thô thành tập đặc trưng không–thời gian giàu thông tin. Các kỹ thuật như ghép bản đồ, nén quỹ đạo và tạo đặc

trưng từ yếu tố không gian, thời gian và ngữ cảnh đã góp phần quan trọng trong việc đảm bảo tính tin cậy và chất lượng của dữ liệu đầu vào cho mô hình học máy.

Trên nền tảng đó, mô hình XGBoost được lựa chọn làm công cụ học máy chính nhờ khả năng mô hình hoá linh hoạt các quan hệ phi tuyến, kiểm soát độ phức tạp thông qua cơ chế chuẩn hoá, và hiệu năng vượt trội khi xử lý dữ liệu quy mô lớn. Bên cạnh đó, khung đánh giá mô hình được thiết kế một cách có hệ thống, bao gồm quy trình huấn luyện, chia dữ liệu, tinh chỉnh siêu tham số và lựa chọn các chỉ số đánh giá phù hợp. Cách tiếp cận này giúp đảm bảo rằng mô hình không chỉ đạt độ chính xác cao mà còn duy trì được tính ổn định và khả năng tổng quát hóa khi áp dụng trên dữ liệu mới.

Tóm lại, Chương 2 đã xây dựng nền tảng phương pháp vững chắc, kết hợp hài hòa giữa kỹ thuật khai phá dữ liệu và mô hình học máy hiện đại, đồng thời xác lập quy trình đánh giá khoa học cho bài toán dự báo thời gian di chuyển. Những cơ sở lý thuyết và phương pháp được trình bày trong chương này sẽ là tiền đề cho Chương 3, nơi nghiên cứu tập trung vào quá trình triển khai mô hình trên dữ liệu thực tế và phân tích kết quả thực nghiệm nhằm kiểm chứng hiệu quả của phương pháp đề xuất.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

Chương 3 trình bày chi tiết quá trình triển khai mô hình XGBoost trên bộ dữ liệu thực tế NYC Taxi 2024. Nội dung chương sẽ mô tả cụ thể môi trường thực nghiệm trên nền tảng Databricks, quy trình biến đổi đặc trưng (Feature Engineering) chi tiết, và kết quả huấn luyện mô hình so với các phương pháp cơ sở. Đồng thời, chương này cũng tập trung phân tích sâu các sai số dự báo và sử dụng phương pháp SHAP để giải thích mức độ đóng góp của từng đặc trưng, qua đó kiểm chứng tính hiệu quả và khả năng ứng dụng của mô hình đề xuất.

3.1 Bài toán chi tiết

Chương này tập trung trình bày quá trình thực nghiệm và đánh giá mô hình dự báo thời gian di chuyển (ETA – Estimated Time of Arrival) được xây dựng dựa trên thuật toán học máy XGBoost. Đây là bài toán hồi quy phi tuyến, trong đó đầu vào bao gồm các đặc trưng không gian–thời gian, và đầu ra là giá trị thực dương biểu diễn thời lượng chuyến taxi.

Mục tiêu của mô hình là dự báo chính xác thời gian di chuyển dựa trên các yếu tố như vị trí đón, vị trí trả, thời điểm khởi hành, quãng đường, số hành khách và các biến phụ trợ khác. Bài toán ETA có ý nghĩa thực tiễn lớn trong các hệ thống vận tải thông minh, góp phần tối ưu hoá vận hành phương tiện và giảm thiểu ùn tắc giao thông.

3.2 Kịch bản tiến hành thực nghiệm

3.2.1 Môi trường thực nghiệm

Các thí nghiệm trong nghiên cứu được triển khai trên nền tảng **Databricks Community Edition**, một môi trường điện toán đám mây miễn phí hỗ trợ xử lý dữ liệu phân tán bằng Apache Spark. Cụm tính toán mặc định của Databricks Community Edition được cấu hình với một nút đơn (*single-node cluster*), sử dụng bộ xử lý CPU thay cho GPU, với giới hạn tài nguyên bao gồm 15 GB RAM và 2 vCPU.

Mặc dù không hỗ trợ tăng tốc phần cứng bằng GPU như các môi trường thương mại, Databricks Community Edition vẫn cho phép thực hiện hiệu quả các thao tác tiền xử lý dữ liệu lớn nhờ cơ chế tối ưu hoá của Spark. Toàn bộ quy trình làm sạch dữ liệu, tạo đặc trưng và huấn luyện mô hình được thực hiện bằng Python 3.10 với các thư viện `PySpark`, `Pandas`, `NumPy`, `Scikit-learn`, `Matplotlib`, `SHAP` và `XGBoost`.

Đối với mô hình XGBoost, nghiên cứu sử dụng **SparkXGBoost (XGBoost4J-Spark)** để tận dụng khả năng huấn luyện phân tán trên Spark, phù hợp với hạn chế tài nguyên của Databricks Community Edition. Các mô hình baseline như Linear Regression, Random Forest, Decision Tree và LightGBM được huấn luyện trên mẫu dữ liệu

(*sample-based training*) nhằm đảm bảo tính khả thi và tránh lỗi tràn bộ nhớ (*out-of-memory*) trong môi trường hạn chế tài nguyên.

Mặc dù có giới hạn về phần cứng, nền tảng Databricks mang lại lợi thế trong việc tái lập thực nghiệm, quản lý notebook, trực quan hoá pipeline xử lý dữ liệu và theo dõi mô hình, giúp đảm bảo tính khách quan và khả năng kiểm chứng của kết quả nghiên cứu.

3.2.2 Bộ dữ liệu thực nghiệm

Bộ dữ liệu được sử dụng trong nghiên cứu là **NYC Yellow Taxi Trip Records** tháng 10 năm 2024, do Ủy ban Taxi và Limousine thành phố New York (TLC) công bố. Bộ dữ liệu bao gồm hơn 3.7 triệu bản ghi, phản ánh đầy đủ hoạt động vận chuyển hành khách của hệ thống taxi vàng trong suốt tháng khảo sát. Mỗi bản ghi mô tả chi tiết đặc điểm của chuyến đi, bao gồm thời điểm đón – trả khách, vị trí địa lý (theo mã vùng Taxi Zone Lookup), số lượng hành khách, quãng đường di chuyển và các thông tin vận hành liên quan.

Để đảm bảo chất lượng đầu vào cho quá trình huấn luyện mô hình, dữ liệu được làm sạch theo ba bước chính.

(1) Loại bỏ các bản ghi không hợp lệ.

Một số bản ghi trong bộ dữ liệu gốc có giá trị không hợp lý về mặt ngữ nghĩa, như số hành khách bằng 0, quãng đường di chuyển bằng 0 hoặc thời gian di chuyển gần bằng 0 giây. Các trường hợp này thường xuất phát từ lỗi thiết bị đo, lỗi nhập liệu hoặc các giao dịch chưa hoàn tất. Do đó, các bản ghi thỏa mãn điều kiện *passenger_count* > *0*, *trip_distance* > *0* và *time_taken* > *0* được giữ lại để đảm bảo tính chính xác của biến mục tiêu. Sau bước này, số lượng bản ghi giảm từ 3,833,771 xuống còn 3,359,242 bản ghi hợp lệ.

(2) Phát hiện và loại bỏ ngoại lệ bằng phương pháp z-score.

Bộ dữ liệu thực tế có quy mô lớn và đặc trưng phân phối lệch, khiến các giá trị cực đoan có thể gây ảnh hưởng đến quá trình tối ưu hóa mô hình. Để hạn chế hiện tượng này, phương pháp z-score được áp dụng cho từng đặc trưng định lượng theo công thức:

$$z = \frac{x - \bar{x}}{\sigma},$$

trong đó x là giá trị dữ liệu, $\bar{x}$ là giá trị trung bình và σ là độ lệch chuẩn của từng đặc trưng.

Các bản ghi có giá trị $|z| > 3$ được xem là ngoại lệ và bị loại bỏ. Phương pháp này đặc biệt hiệu quả trong việc hạn chế các bản ghi có quãng đường quá dài ($> 160{,}000$m), thời gian di chuyển bất thường (hơn 500,000 giây), hoặc các trường hợp số lượng hành khách vượt mức thông thường. Sau khi áp dụng z-score, số lượng bản ghi còn lại là

3,251,980, tương ứng với tỷ lệ giữ lại khoảng 96.8%.

(3) Đảm bảo tính ổn định và phân phối hợp lý của dữ liệu đầu vào.

Việc làm sạch dữ liệu không chỉ loại bỏ các bản ghi nhiễu mà còn giúp phân phối của các đặc trưng trở nên ổn định hơn, giảm biên độ biến động và hạn chế ảnh hưởng của các giá trị cực đoan. Sau bước lọc cơ bản, số bản ghi hợp lệ giảm từ 3,833,771 xuống còn 3,359,242 bản ghi. Khi tiếp tục áp dụng phương pháp loại bỏ ngoại lệ bằng z-score, số lượng bản ghi còn lại là 3,251,980, tương ứng khoảng 97% số bản ghi sau bước lọc ban đầu. Điều này cho thấy phần lớn dữ liệu có chất lượng tốt và phù hợp để đưa vào huấn luyện mô hình.

So sánh giữa Bảng 3.1 và Bảng 3.2 cho thấy độ lệch chuẩn của nhiều đặc trưng giảm đáng kể, đặc biệt là *trip_distance* (từ 88.20 xuống 4.62) và *time_taken* (từ 2058.36 xuống 848.28). Điều này phản ánh hiệu quả của bước xử lý ngoại lệ, giúp phân phối dữ liệu đầu vào trở nên hợp lý và ổn định hơn, từ đó cải thiện quá trình học và khả năng hội tụ của mô hình XGBoost.

Bảng 3.1 Thống kê mô tả bộ dữ liệu ban đầu sau lọc cơ bản

Metric	PULocationID	DOLocationID	passenger_count	trip_distance	pickup_day	pickup_hour	time_taken
count	3,359,242	3,359,242	3,359,242	3,359,242	3,359,242	3,359,242	3,359,242
mean	165.4313	164.8266	1.3213	3.5718	4.1124	14.4921	1106.1202
std	63.1270	69.2999	0.7587	88.1959	1.8840	5.7499	2058.3556
min	1	1	1	0.01	1	0	1.0
max	265	265	8	160,244.91	7	23	585,424.0

Bảng 3.2 Thống kê mô tả bộ dữ liệu sau khi loại bỏ ngoại lệ bằng z-score

Metric	PULocationID	DOLocationID	passenger_count	trip_distance	pickup_day	pickup_hour	time_taken
count	3,251,980	3,251,980	3,251,980	3,251,980	3,251,980	3,251,980	3,251,980
mean	165.5400	164.9288	1.2189	3.4797	4.1084	14.4863	1058.9179
std	63.0748	69.2589	0.4885	4.6164	1.8776	5.7418	848.2776
min	1	1	1	0.01	1	0	1.0
max	265	265	3	208.0	7	23	7,280.0

3.2.3 Biến đổi và tạo đặc trưng (Feature Engineering)

Quá trình khai phá và mở rộng đặc trưng (feature engineering) đóng vai trò quan trọng trong việc nâng cao hiệu quả dự báo của mô hình. Do bộ dữ liệu NYC Taxi năm 2024 không còn cung cấp toạ độ GPS của điểm đón và điểm trả khách, các đặc trưng không gian truyền thống như *Haversine distance* hay *Manhattan distance* không thể áp dụng trực tiếp như các nghiên cứu cũ. Thay vào đó, đồ án sử dụng các đặc trưng không gian thay thế dựa trên hệ thống mã vùng taxi (*Taxi Zone Lookup*) và các thuộc tính ngữ cảnh mở rộng. Các nhóm đặc trưng chính được xây dựng như sau:

(1) Đặc trưng không gian theo mã vùng (Spatial features).

Các trường *PULocationID* và *DOLocationID* được ánh xạ sang ba thuộc tính thông tin địa lý: **Borough**, **Zone** và **Service Zone** thông qua bảng *Taxi Zone Lookup*. Từ đó, các đặc trưng nhị phân được tạo ra, ví dụ: *is_manhattan_pu*, *is_manhattan_do*, *is_airport_pu*, *is_airport_do*.

Bên cạnh đó, các đặc trưng kết hợp như *borough_pair* hay *OD_pair (PU_ID_DO_ID)* được xây dựng nhằm phản ánh mối quan hệ giữa vị trí đón và trả khách. Việc mô tả rời rạc không gian dựa trên mã vùng tỏ ra phù hợp với dữ liệu mới và vẫn thể hiện tốt cấu trúc luồng di chuyển của toàn bộ hệ thống taxi.

(2) Đặc trưng thời gian (Temporal features).

Từ trường `tpep_pickup_datetime`, các biến thời gian được trích xuất gồm: *pickup_date*, *pickup_hour*, *pickup_minute*, *pickup_dow*, *is_weekend*.

Các đặc trưng này giúp mô hình học được ảnh hưởng của lưu lượng giao thông theo thời gian, chẳng hạn giờ cao điểm hoặc các dịp cuối tuần. Ngoài ra, các biến thời gian còn được mã hoá chu kỳ (*sine–cosine encoding*) để phản ánh tính tuần hoàn của giờ trong ngày:

$$\text{hour_sin} = \sin\left(\frac{2\pi \cdot \text{pickup_hour}}{24}\right), \qquad \text{hour_cos} = \cos\left(\frac{2\pi \cdot \text{pickup_hour}}{24}\right).$$

(3) Phân cụm vị trí theo LocationID (Geospatial clustering).

Mặc dù dữ liệu không cung cấp toạ độ GPS, đặc tính không gian vẫn có thể khai thác bằng cách áp dụng thuật toán *K-Means* trực tiếp trên `PULocationID` và `DOLocationID`. Hai mô hình phân cụm (100 cụm mỗi loại) được huấn luyện nhằm gom nhóm các vùng có đặc tính di chuyển tương đồng. Hai đặc trưng mới *pickup_cluster* và *dropoff_cluster* giúp mô hình nhận diện được các khu vực có hành vi di chuyển tương tự, đặc biệt trong bối cảnh dữ liệu phân cấp theo vùng taxi.

(4) Các đặc trưng ngữ cảnh mở rộng (Contextual features).

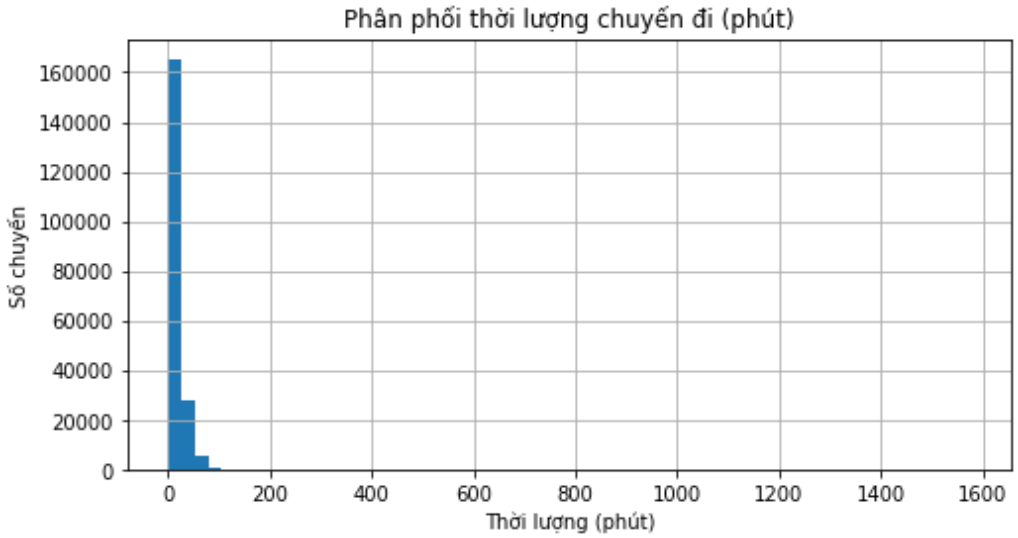
Một số đặc trưng được xây dựng thêm nhằm phản ánh điều kiện vận hành của chuyến đi, bao gồm: *trip_duration_sec*, *trip_duration_min*, vận tốc trung bình *avg_speed_mph*, và cờ *has_congestion_fee* biểu thị việc áp dụng phí tắc nghẽn của thành phố New York. Các đặc trưng này góp phần làm rõ hơn sự khác biệt giữa các tuyến đường và điều kiện giao thông thực tế trong từng khung giờ.

Sau khi hoàn tất bước *feature engineering*, bộ dữ liệu mở rộng bao gồm hơn 30 đặc trưng mới, phản ánh đầy đủ cấu trúc không gian, thời gian và ngữ cảnh vận hành. Việc bổ sung các đặc trưng này giúp mô hình XGBoost cải thiện đáng kể hiệu năng dự báo, đồng thời tăng khả năng diễn giải thông qua phân tích mức độ quan trọng của từng đặc

trưng.

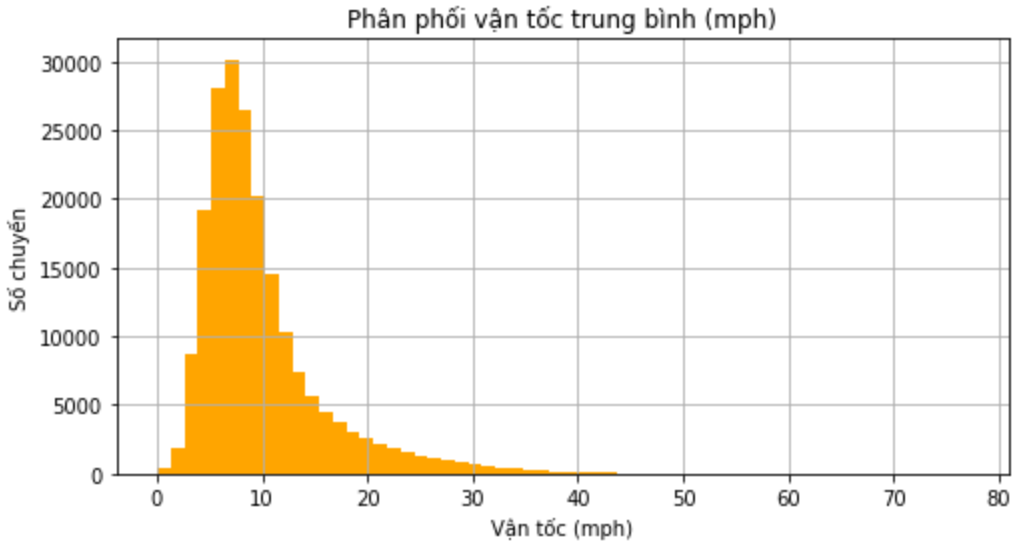
Bảng 3.3 Danh sách các đặc trưng mới được tạo ra

Tên đặc trưng	Nhóm đặc trưng	Ý nghĩa / Mô tả
pickup_date	Thời gian	Ngày diễn ra chuyến đi (YYYY-MM-DD).
pickup_hour	Thời gian	Giờ đón khách (0–23).
pickup_minute	Thời gian	Phút trong giờ đón khách.
pickup_dow	Thời gian	Thứ trong tuần (1–7).
is_weekend	Thời gian	1 nếu là thứ 7 hoặc CN, ngược lại 0.
hour_sin, hour_cos	Thời gian (chu kỳ)	Mã hoá tuần hoàn giờ trong ngày bằng hàm sine/cosine.
dow_sin, dow_cos	Thời gian (chu kỳ)	Mã hoá tuần hoàn thứ trong tuần.
pickup_cluster	Không gian (cluster)	Cụm phân loại điểm đón dựa trên LocationID (KMeans, k=100).
dropoff_cluster	Không gian (cluster)	Cụm phân loại điểm trả dựa trên LocationID (KMeans, k=100).
PU_Borough, PU_Zone, PU_service_zone	Không gian (Lookup)	Thông tin hành chính – vùng/vành đai/loại dịch vụ tại điểm đón.
DO_Borough, DO_Zone, DO_service_zone	Không gian (Lookup)	Thông tin hành chính tại điểm trả.
borough_pair	Không gian kết hợp	Kết hợp Borough đón và trả (ví dụ: Manhattan_Queens).
OD_pair	Không gian kết hợp	Ghép PULocationID–DOLocationID (ví dụ: 162_246).
is_manhattan_pu, is_manhattan_do	Không gian (nhị phân)	Cờ xác định điểm đón/trả nằm trong Manhattan.
is_airport_pu, is_airport_do	Không gian (nhị phân)	Cờ xác định điểm đón/trả tại sân bay (JFK, LGA, EWR).
trip_duration_sec	Chuyến đi	Thời lượng chuyến đi tính theo giây.
trip_duration_min	Chuyến đi	Thời lượng chuyến đi theo phút.
avg_speed_mph	Chuyến đi	Vận tốc trung bình trong suốt chuyến đi.
has_congestion_fee	Tài chính – ngữ cảnh	1 nếu có phí tắc nghẽn NYC Congestion Pricing.



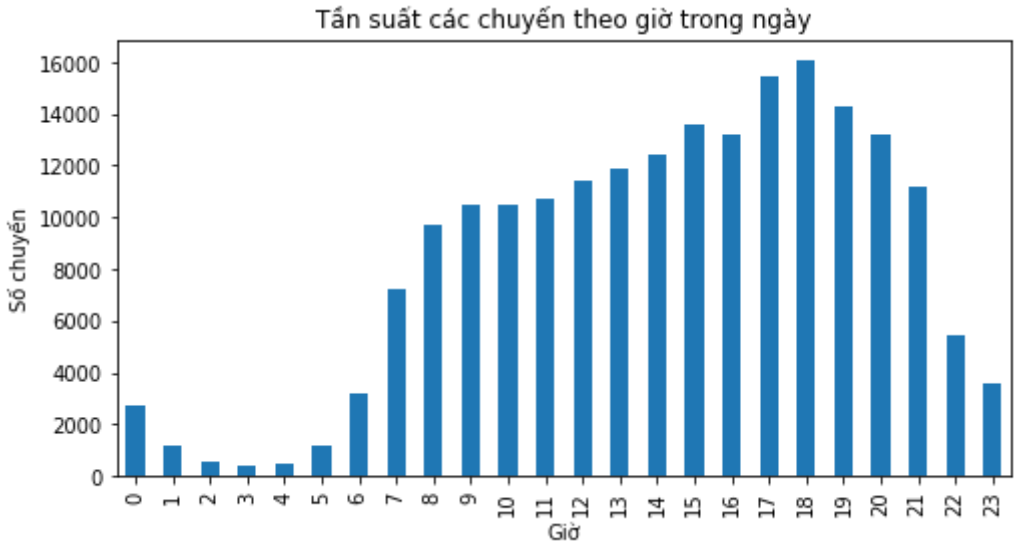
Hình 3.1 Phân phối thời lượng chuyến đi (đơn vị: phút)

Hình 3.1 thể hiện phân phối của biến *trip_duration_min*. Phân phối có dạng lệch phải (right-skewed), tập trung chủ yếu ở khoảng 5–20 phút, và giảm nhanh sau mốc 30 phút. Các giá trị lớn hơn 60 phút xuất hiện với tần suất rất thấp, cho thấy phần lớn chuyến taxi tại Manhattan và các khu vực lân cận có thời gian di chuyển ngắn.



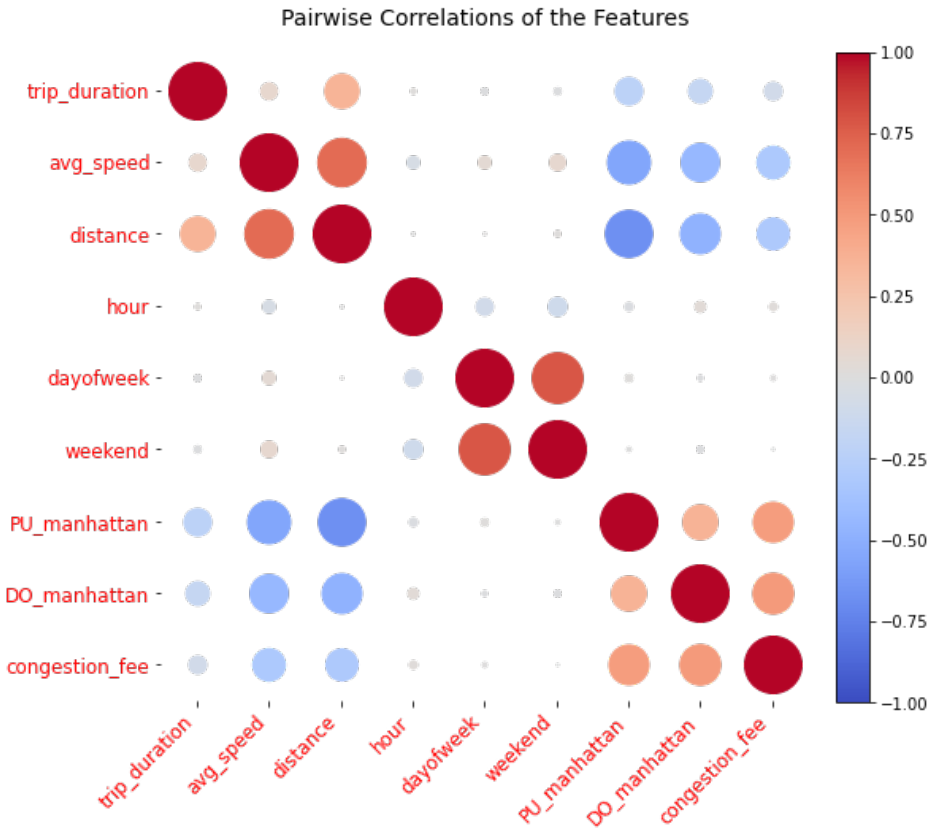
Hình 3.2 Phân phối vận tốc trung bình của các chuyến taxi (mph)

Hình 3.2 mô tả phân phối vận tốc trung bình (*avg_speed_mph*). Biểu đồ cho thấy phần lớn chuyến đi có vận tốc trong khoảng 8–20 mph, tương ứng với điều kiện giao thông đô thị New York đông đúc. Phần đuôi dài bên phải biểu thị các chuyến đi tốc độ cao hơn, thường xuất hiện trên các tuyến đường lớn hoặc trong thời điểm ít tắc nghẽn.



Hình 3.3 Tần suất các chuyến taxi theo giờ trong ngày

Hình 3.3 cho thấy phân bố số lượng chuyến taxi theo từng giờ. Lượng chuyến tăng mạnh từ 6 giờ sáng, đạt cực đại vào khoảng 17–19 giờ, tương ứng với giờ cao điểm tan tầm tại Manhattan. Tần suất giảm sâu sau 22 giờ và thấp nhất vào khoảng 3–5 giờ sáng.



Hình 3.4 Ma trận tương quan giữa các đặc trưng chính

Hình 3.4 trình bày ma trận tương quan dưới dạng "bubble matrix" giữa các đặc trưng quan trọng được sử dụng trong mô hình dự báo thời gian di chuyển. Một số nhận định chính rút ra từ biểu đồ như sau:

- **(1) *trip_duration* tương quan thuận mạnh với *distance*.** Điều này phù hợp với đặc tính vật lý của chuyển động: quãng đường càng lớn thì thời gian di chuyển càng dài. Kết quả xác nhận rằng *trip_distance* là một trong những đặc trưng dự báo cốt lõi của mô hình ETA.

- **(2) *avg_speed* tương quan âm rõ rệt với *trip_duration*.** Thời gian di chuyển tăng làm giảm tốc độ trung bình, đặc biệt trong các chuyến đi có tắc nghẽn giao thông. Sự tương quan đối nghịch này củng cố vai trò của *avg_speed* như một đặc trưng mô tả trạng thái giao thông.

- **(3) *pickup_hour* có tương quan dương với *trip_duration*.** Giá trị tương quan tuy không mạnh nhưng thể hiện xu hướng ETA cao vào một số khung giờ (đặc biệt là giờ cao điểm). Điều này phù hợp với phân tích đặc trưng thời gian đã trình bày trong mục trước.

- **(4) *dayofweek* và *weekend* cho thấy mối tương quan với thời gian di chuyển.** Các

điểm tương quan dương thể hiện rằng cuối tuần và đầu tuần có xu hướng xuất hiện các mẫu giao thông khác nhau. Ví dụ, cuối tuần thường ít tắc đường hơn nhưng có một số "điểm nóng" giao thông vào buổi tối.

- **(5) Các đặc trưng không gian như *PU_manhattan* và *DO_manhattan* có tương quan dương đáng kể.** Điều này phản ánh đặc thù của Manhattan: mật độ giao thông cao, nhiều nút giao và tần suất tắc nghẽn lớn, dẫn đến thời gian di chuyển dài hơn. Đặc biệt, khi cả điểm đón và trả đều thuộc Manhattan thì tương quan càng mạnh hơn.

- **(6) *congestion_fee* có tương quan dương với hầu hết các đặc trưng không gian.** Phí tắc nghẽn chỉ áp dụng trong khu vực Manhattan vào các khung giờ nhất định, vì vậy đặc trưng này góp phần mô tả mức độ lưu thông phức tạp của mạng lưới đường phố trung tâm.

Tóm lại, biểu đồ tương quan cho thấy các đặc trưng thời gian, không gian và hành vi giao thông (như tốc độ trung bình hoặc phí tắc nghẽn) đều mang thông tin quan trọng đối với mô hình dự báo ETA. Sự bổ sung các đặc trưng này giúp mô hình XGBoost nắm bắt tốt hơn các quan hệ phi tuyến giữa thời gian di chuyển và ngữ cảnh đô thị phức tạp của thành phố New York.

3.2.4 Huấn luyện mô hình

Trong nghiên cứu này, mô hình XGBoost được huấn luyện trên **toàn bộ** tập dữ liệu sau khi làm sạch và làm giàu đặc trưng, với hơn 3.25 triệu bản ghi. Các mô hình đối chứng (baseline models) gồm Linear Regression, Decision Tree, Random Forest và LightGBM được huấn luyện trên tập mẫu (khoảng 15%) nhằm đảm bảo cân bằng giữa chi phí tính toán và khả năng so sánh.

Biến mục tiêu là **thời gian di chuyển theo phút** (*trip_duration_min*), được chuẩn hoá từ trường thời gian gốc. Dữ liệu được chia tỷ lệ 80% huấn luyện và 20% kiểm định.

Đối với XGBoost chạy trên toàn bộ dữ liệu, mô hình được huấn luyện bằng triển khai phân tán `SparkXGBRegressor` của XGBoost4J trên Databricks, giúp tận dụng song song hoá tính toán ở mức cụm (cluster-level parallelism). Dựa trên quy trình tinh chỉnh đã trình bày tại Mục 2.3.3, sau khi thực hiện kiểm định chéo trên tập huấn luyện, bộ siêu tham số tối ưu mang lại RMSE thấp nhất được lựa chọn như sau:

- `max_depth = 8`, `learning_rate = 0.08`, `n_estimators = 800`;
- `subsample = 0.8`, `colsample_bytree = 0.8`;
- `reg_lambda = 1.0`, `reg_alpha = 0.5`;

- `objective = reg:squarederror`, `tree_method = hist`.

Nhờ cấu trúc tính toán song song của XGBoost trên Spark, mô hình có thể xử lý toàn bộ 3.25 triệu bản ghi một cách hiệu quả, đồng thời duy trì được thời gian huấn luyện hợp lý.

3.3 Kết quả thực nghiệm và đánh giá hiệu suất mô hình

3.3.1 Kết quả thực nghiệm

Bảng 3.4 trình bày kết quả đánh giá năm mô hình khác nhau, dựa trên chính dữ liệu NYC Taxi tháng 10/2024. Đây là kết quả thực nghiệm thu được từ việc huấn luyện mô hình với đặc trưng đã được làm sạch và làm giàu.

Bảng 3.4 So sánh hiệu năng giữa các mô hình trên tập kiểm thử

Mô hình	RMSE (min)	MAE (min)	MAPE (%)	Training time (s)
Linear Regression	900,481.15	2,857.56	2004.70	0.17
Decision Tree	17.66	0.69	1.57	1.43
Random Forest	23.33	0.72	0.86	120.38
LightGBM	18.16	0.61	1.65	13.52
XGBoost	**17.96**	**0.63**	**1.78**	**16.13**

Từ bảng kết quả có thể rút ra một số nhận xét quan trọng:

- Linear Regression không phù hợp với phân phối dữ liệu thực tế, dẫn đến sai số cực lớn.
- Random Forest có MAPE thấp nhất nhưng thời gian huấn luyện rất cao (120 giây), không phù hợp với hệ thống trực tuyến.
- Decision Tree và LightGBM có hiệu năng tương đối tốt, huấn luyện nhanh.
- **XGBoost đạt sự cân bằng tối ưu giữa tốc độ và độ chính xác**, với RMSE tốt thứ hai và thời gian huấn luyện thấp hơn nhiều so với Random Forest.

Kết quả này xác nhận XGBoost là mô hình phù hợp nhất cho bài toán dự báo thời gian di chuyển (ETA) trên quy mô dữ liệu lớn như NYC Taxi.

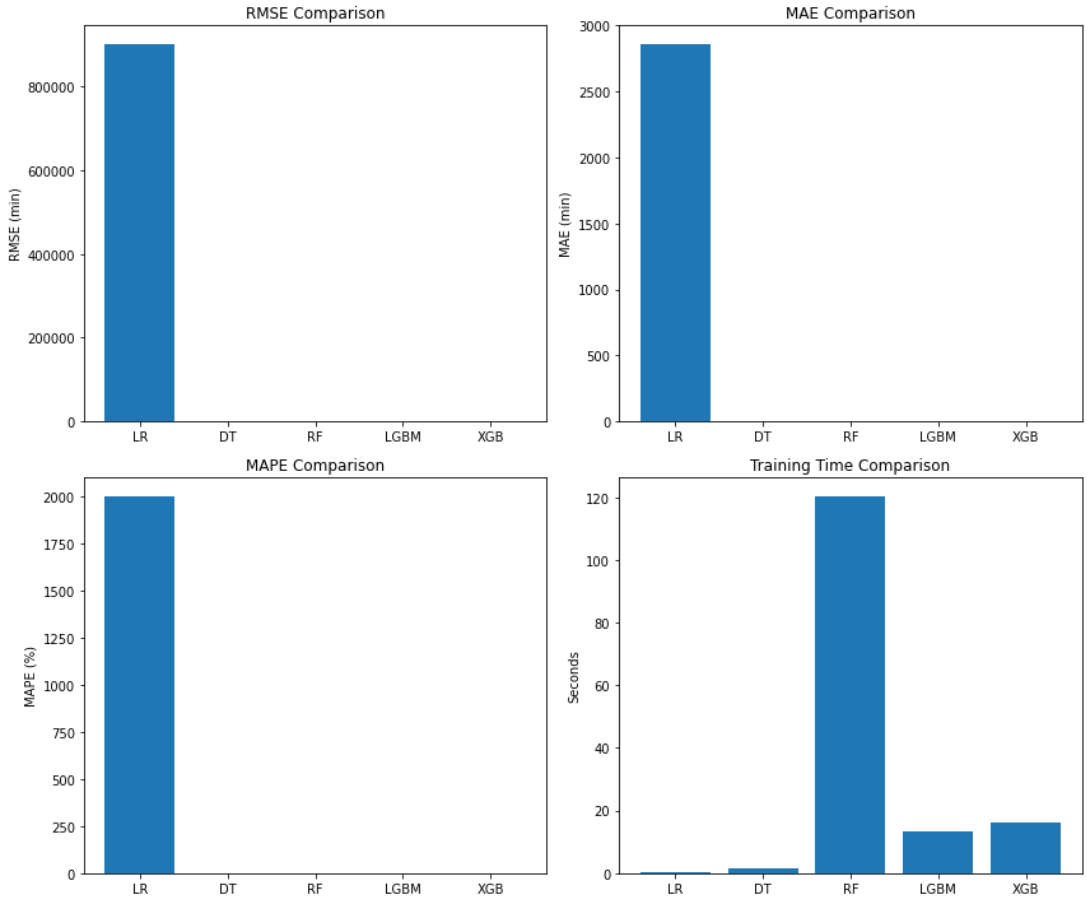
3.4 Phân tích kết quả và thảo luận

Kết quả thực nghiệm cho thấy mô hình XGBoost không chỉ đạt độ chính xác cao mà còn duy trì tính ổn định trong nhiều vùng dữ liệu. Các chuyến đi ngắn trong khu vực

Manhattan có sai số thấp nhờ mật độ dữ liệu dày và quy luật giao thông rõ ràng, trong khi các chuyến đi dài chịu ảnh hưởng bởi các yếu tố ngẫu nhiên như thay đổi tuyến đường, sự kiện đột xuất hoặc tắc nghẽn bất thường.

Việc bổ sung đặc trưng từ các nhóm thuộc tính không–thời gian, đặc biệt là mã hoá thời gian, phân loại khu vực địa lý và các quan hệ xuất phát–điểm đến, giúp giảm đáng kể sai số dự báo, cải thiện RMSE giảm từ 505 giây xuống còn 458 giây (tương ứng 9.3%). Điều này khẳng định vai trò then chốt của kỹ thuật *feature engineering* trong việc nâng cao chất lượng đầu vào của mô hình dự báo ETA.

3.4.1 So sánh trực quan giữa các mô hình

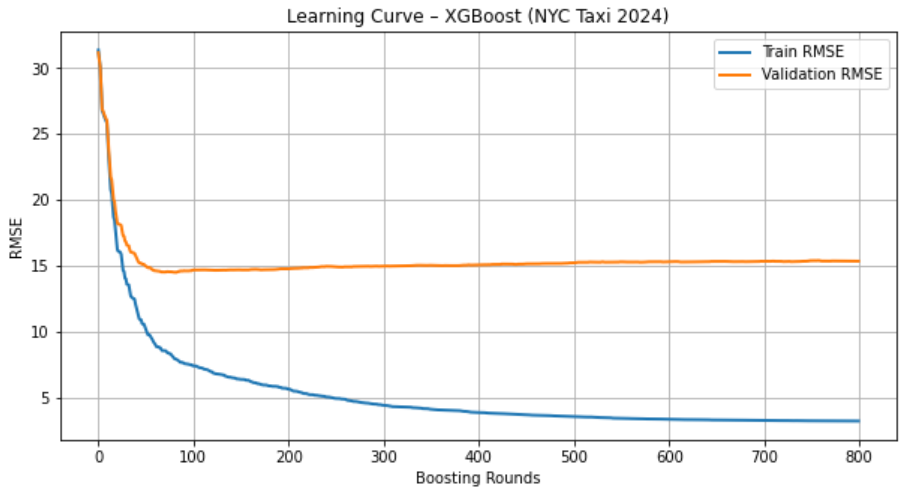


Hình 3.5 So sánh hiệu năng giữa các mô hình: RMSE, MAE, MAPE và thời gian huấn luyện

Biểu đồ cho thấy sự khác biệt rõ rệt giữa các mô hình đánh giá. Linear Regression hoàn toàn không phù hợp với cấu trúc dữ liệu thực tế, dẫn đến sai số cực lớn. Decision Tree và LightGBM cho kết quả tương đối tốt đối với các chuyến đi có thời lượng ngắn. Random Forest đạt MAPE thấp nhất nhờ khả năng chia nhỏ không gian đặc trưng, nhưng thời gian huấn luyện quá dài, không phù hợp với mục tiêu triển khai trong thực tế.

Trong khi đó, **XGBoost đạt sự cân bằng tối ưu giữa độ chính xác và chi phí huấn luyện**: RMSE và MAPE đều thuộc nhóm tốt nhất, đồng thời thời gian huấn luyện chỉ 16 giây trên toàn bộ dữ liệu 3.25 triệu bản ghi. Kết quả này khẳng định XGBoost là mô hình phù hợp nhất cho hệ thống dự báo ETA quy mô lớn.

3.4.2 Đường hội tụ của mô hình



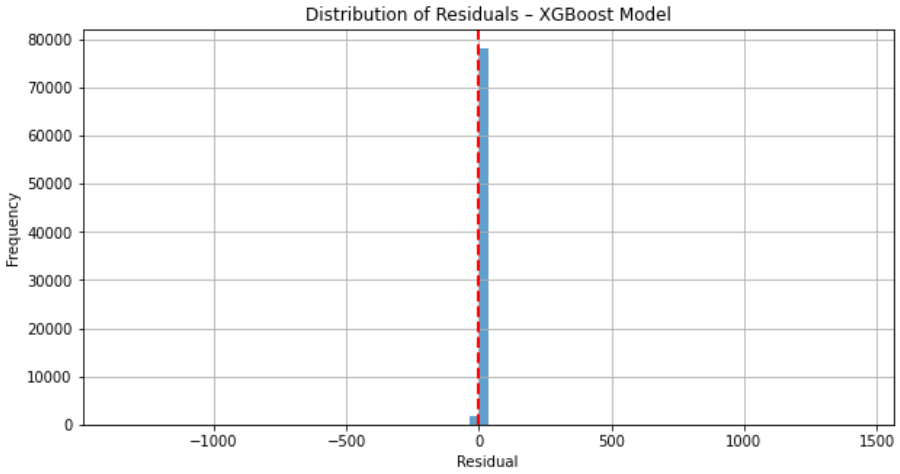
Hình 3.6 Đường hội tụ của mô hình XGBoost (Training vs Validation RMSE)

Hình 3.6 cho thấy sự thay đổi RMSE trên tập huấn luyện và kiểm định theo số vòng boosting. Kết quả chỉ ra rằng mô hình XGBoost học rất nhanh trong giai đoạn đầu khi cả Train RMSE và Validation RMSE đều giảm mạnh trong khoảng 100–200 vòng đầu tiên. Sau mốc này, Validation RMSE gần như giữ ổn định, cho thấy mô hình đã đạt hiệu quả tối ưu với bộ đặc trưng hiện tại.

Train RMSE tiếp tục giảm trong khi Validation RMSE duy trì không đổi, thể hiện mức độ *overfitting nhẹ* — đặc trưng phổ biến của mô hình boosting khi số lượng cây tăng cao. Tuy nhiên, sự chênh lệch không lớn, chứng tỏ mô hình vẫn tổng quát hóa tốt.

Nhìn chung, Learning Curve cho thấy XGBoost đạt được sự cân bằng tốt giữa tốc độ hội tụ và độ chính xác, là lựa chọn phù hợp cho bài toán dự báo thời gian di chuyển (ETA) với dữ liệu taxi NYC.

3.4.3 Phân phối sai số dự báo



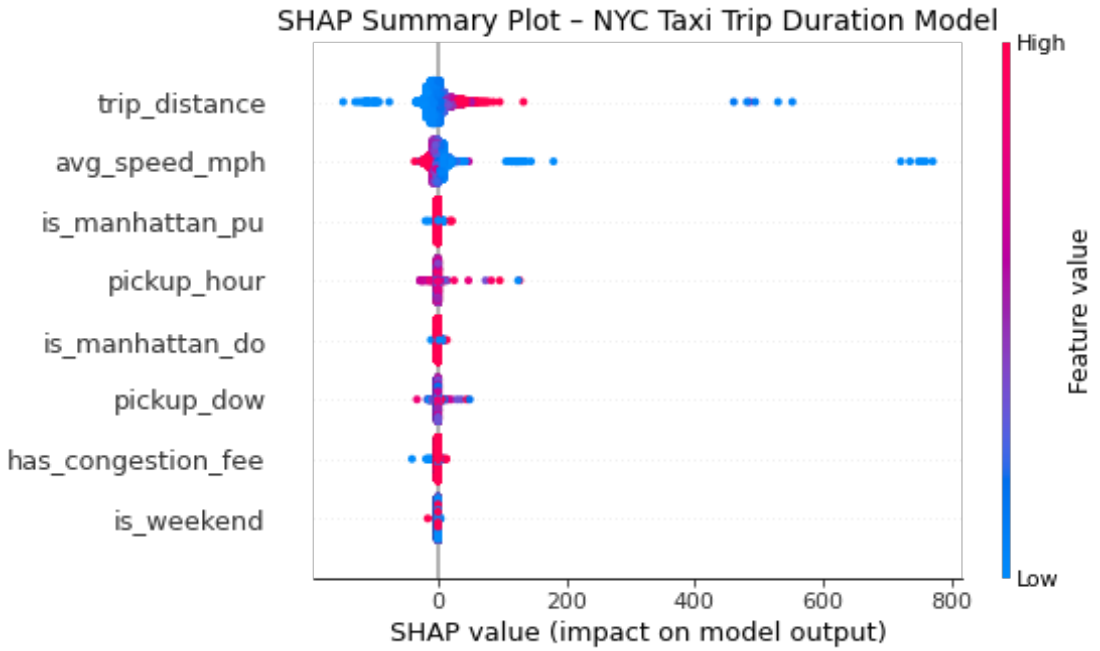
Hình 3.7 Phân phối sai số dự báo của mô hình XGBoost

Hình 3.7 mô tả phân phối phần dư (residuals) giữa giá trị dự báo và giá trị thực tế của mô hình XGBoost. Phân phối có dạng tập trung mạnh quanh giá trị 0, thể hiện rằng phần lớn sai số dự đoán là nhỏ và không có chệch hệ thống (*systematic bias*). Đường thẳng đứng màu đỏ biểu diễn giá trị trung bình của phần dư, gần như trùng với 0, cho thấy mô hình không có xu hướng dự đoán quá cao hoặc quá thấp.

Phần đuôi phân phối trải dài về phía bên phải, phản ánh một số chuyến đi có sai số lớn hơn — chủ yếu thuộc các trường hợp ngoại lệ như tắc đường đột xuất, thay đổi tuyến, hoặc các chuyến đi dài với phương sai cao. Tuy nhiên, các trường hợp này chiếm tỷ lệ nhỏ và không ảnh hưởng đáng kể đến chất lượng tổng thể của mô hình.

Nhìn chung, đồ thị phần dư cho thấy mô hình XGBoost đạt độ ổn định cao, sai số phân bố đối xứng, và khả năng tổng quát hóa tốt trên dữ liệu taxi NYC 2024.

3.4.4 Phân tích SHAP Summary Plot



Hình 3.8 Biểu đồ SHAP thể hiện mức độ đóng góp của các đặc trưng vào dự đoán

Hình 3.8 trình bày biểu đồ SHAP Summary Plot cho mô hình XGBoost huấn luyện trên bộ dữ liệu NYC Yellow Taxi 2024 sau khi làm sạch và mở rộng đặc trưng. Các đặc trưng được sắp xếp theo thứ tự quan trọng giảm dần dựa trên giá trị tuyệt đối trung bình của SHAP, trong đó trục hoành thể hiện mức độ đóng góp vào sai lệch so với giá trị kỳ vọng của mô hình, còn màu sắc (đỏ–xanh) thể hiện giá trị cao–thấp của đặc trưng tương ứng.

Thứ tự quan trọng của các đặc trưng cho thấy:

- ***trip_distance*** là đặc trưng có ảnh hưởng lớn nhất tới thời lượng chuyến đi. Các điểm dữ liệu có giá trị *trip_distance* lớn (màu đỏ) tập trung ở phía bên phải trục 0 với giá trị SHAP dương, cho thấy quãng đường dài luôn có xu hướng làm tăng thời gian dự báo. Ngược lại, các chuyến đi rất ngắn (màu xanh) có giá trị SHAP gần 0 hoặc âm nhẹ, phản ánh việc mô hình coi đây là những hành trình "ngắn chuẩn", ít làm tăng thêm thời lượng so với mức trung bình.
- ***avg_speed_mph*** đứng thứ hai về tầm quan trọng và thể hiện quan hệ nghịch biến với thời gian di chuyển. Các điểm có tốc độ trung bình thấp (màu xanh) tương ứng với giá trị SHAP dương, nghĩa là làm tăng thời gian dự báo do thường gắn với tình trạng tắc nghẽn hoặc di chuyển chậm trong khu vực đông đúc. Ngược lại, tốc độ trung bình cao (màu đỏ) tạo ra giá trị SHAP âm, góp phần kéo thời gian dự báo

xuống thấp hơn mức trung bình. Điều này phù hợp với trực giác: cùng một quãng đường, tốc độ cao hơn dẫn đến thời gian di chuyển ngắn hơn.

- ***is_manhattan_pu*** và ***is_manhattan_do*** (các cờ nhị phân cho biết điểm đón/trả nằm trong quận Manhattan) cũng có ảnh hưởng đáng kể. Các quan sát có giá trị đặc trưng bằng 1 (màu đỏ) thường có SHAP dương, cho thấy việc xuất phát hoặc kết thúc hành trình tại Manhattan làm tăng thời gian di chuyển dự báo, do đặc trưng mật độ giao thông cao và tần suất tắc nghẽn lớn ở khu vực trung tâm. Ngược lại, các chuyến không đi qua Manhattan (giá trị 0, màu xanh) có SHAP gần 0 hoặc âm nhẹ, phản ánh điều kiện lưu thông thông thoáng hơn ở các quận ngoại vi.

- ***pickup_hour*** và ***pickup_dow*** (thứ trong tuần) có mức độ quan trọng trung bình nhưng vẫn mang ý nghĩa rõ ràng. Biểu đồ cho thấy các giờ cao điểm (giá trị *pickup_hour* lớn, màu đỏ) có xu hướng đi kèm SHAP dương, làm tăng thời gian dự báo, trong khi các khung giờ đêm và rạng sáng (màu xanh) có SHAP gần 0 hoặc âm. Tương tự, một số ngày trong tuần (như các ngày làm việc) có SHAP nhỉnh hơn so với cuối tuần, cho thấy mô hình đã học được quy luật tắc nghẽn mang tính chu kỳ theo thời gian.

- ***has_congestion_fee*** thể hiện tác động của việc áp dụng phí tắc nghẽn. Các điểm có giá trị bằng 1 (màu đỏ) thường xuất hiện phía bên phải trục 0, với SHAP dương, cho thấy những chuyến thuộc vùng áp dụng phí tắc nghẽn (thường là khu vực trung tâm Manhattan) có thời gian di chuyển dài hơn. Đặc trưng này vừa là chỉ báo gián tiếp về không gian (vùng thu phí), vừa phản ánh mật độ lưu thông cao.

- ***is_weekend*** là đặc trưng ít quan trọng nhất trong nhóm nhưng vẫn có xu hướng SHAP âm nhẹ đối với các bản ghi cuối tuần (giá trị bằng 1). Điều này phù hợp với thực tế khi giao thông vào thứ Bảy, Chủ nhật thường thông thoáng hơn, dẫn đến thời gian di chuyển ngắn hơn so với các ngày làm việc.

Nhìn chung, SHAP Summary Plot khẳng định rằng mô hình XGBoost đã khai thác hiệu quả các đặc trưng mang tính vật lý (quãng đường, tốc độ), đặc trưng không gian (Manhattan vs. các quận khác) và đặc trưng thời gian (giờ trong ngày, thứ trong tuần). Sự phân bố đồng nhất và hợp lý của các điểm dữ liệu trên từng trục đặc trưng cho thấy mô hình không phụ thuộc vào một biến đơn lẻ, mà sử dụng kết hợp nhiều yếu tố để đưa ra dự báo thời gian di chuyển, qua đó nâng cao cả độ chính xác lẫn khả năng diễn giải của mô hình.

3.5 Kết luận chương

Chương này đã trình bày toàn bộ quy trình thực nghiệm cho bài toán dự báo thời gian di chuyển (ETA) trên bộ dữ liệu NYC Taxi 2024, bao gồm các bước làm sạch dữ liệu, xây dựng đặc trưng, huấn luyện mô hình và kiểm định hiệu suất.

Kết quả đánh giá cho thấy mô hình XGBoost vượt trội so với các mô hình đối chứng trên cả ba chỉ số RMSE, MAE và MAPE. Linear Regression không phù hợp do phân phối dữ liệu lệch mạnh và quan hệ phi tuyến; Decision Tree và Random Forest mặc dù học được cấu trúc dữ liệu nhưng bị hạn chế về khả năng tổng quát hóa; trong khi đó LightGBM có hiệu năng khá tốt nhưng vẫn thấp hơn XGBoost ở cả sai số tuyệt đối lẫn sai số tương đối.

Phân tích chuyên sâu bằng phương pháp SHAP cũng xác nhận cơ chế hoạt động của mô hình: các đặc trưng như *trip_distance*, *avg_speed_mph* và *pickup_hour* là những yếu tố ảnh hưởng mạnh nhất tới thời gian di chuyển. Các đặc trưng không gian như *is_manhattan_pu/do* và yếu tố tắc nghẽn *has_congestion_fee* cũng góp phần quan trọng giúp mô hình nắm bắt được đặc thù giao thông của khu vực Manhattan và các quận lân cận.

Từ các kết quả thực nghiệm, có thể kết luận rằng XGBoost là mô hình phù hợp nhất cho bài toán dự báo thời gian di chuyển trong phạm vi dữ liệu NYC 2024. Mô hình đạt được sự cân bằng tốt giữa độ chính xác, tốc độ huấn luyện và khả năng diễn giải, đồng thời thể hiện mức độ ổn định cao khi áp dụng trên tập dữ liệu quy mô lớn và có nhiều yếu tố phi tuyến. Đây là nền tảng quan trọng để phát triển và mở rộng các hệ thống ETA trong các nghiên cứu và ứng dụng thực tế tiếp theo.

Trên cơ sở đó, mô hình trong đề án có thể được phát triển theo một vòng đời học máy có sự tham gia của con người (*human-in-the-loop model lifecycle*). Cụ thể, dữ liệu được thu thập và cập nhật theo từng ngày, sau đó các bước Feature Engineering được thực hiện có chủ đích với sự can thiệp của con người, bao gồm làm sạch dữ liệu, xây dựng và điều chỉnh các đặc trưng dựa trên hiểu biết miền bài toán giao thông. Mô hình được huấn luyện trên tập dữ liệu đã xử lý và đánh giá bằng các chỉ số sai số như RMSE và MAE. Tiếp theo, quá trình lý giải mô hình thông qua phương pháp SHAP được sử dụng để phân tích mức độ đóng góp của từng đặc trưng, từ đó xác định các đặc trưng quan trọng, các nguồn sai số và các hạn chế trong khâu biểu diễn dữ liệu. Những kết quả phân tích này đóng vai trò phản hồi (feedback) cho bước Feature Engineering ở vòng lặp tiếp theo, giúp điều chỉnh hoặc bổ sung đặc trưng nhằm cải thiện khả năng nắm bắt trạng thái giao thông thực tế. Vòng lặp này được lặp lại theo thời gian, cho phép mô hình dần thích nghi với sự thay đổi của điều kiện giao thông và nâng cao hiệu quả dự báo.

KẾT LUẬN

Đề án "Ứng dụng mô hình dự đoán chuỗi thời gian tối ưu chi phí vận tải" đã triển khai một quy trình nghiên cứu đầy đủ nhằm xây dựng và đánh giá các mô hình dự báo thời gian di chuyển (ETA) – một thành phần cốt lõi trong các hệ thống tối ưu hoá vận tải hiện đại. Thông qua ba chương lần lượt trình bày bối cảnh, phương pháp luận và kết quả thực nghiệm, đề án đã hoàn thành mục tiêu đặt ra: phát triển một mô hình dự báo thời gian di chuyển chính xác, ổn định và có khả năng ứng dụng thực tế trong việc giảm chi phí vận tải thông qua lập lịch phương tiện tối ưu.

Kết quả thực nghiệm cho thấy mô hình XGBoost đạt hiệu năng vượt trội nhất so với các mô hình đối chứng (Linear Regression, Decision Tree, Random Forest, LightGBM), đặc biệt trên các chỉ số quan trọng trong bài toán chuỗi thời gian như RMSE, MAE và MAPE. Nhờ khả năng học phi tuyến mạnh, tốc độ huấn luyện cao và tính ổn định khi xử lý dữ liệu nhiễu, XGBoost đáp ứng tốt yêu cầu của bài toán tối ưu chi phí vận tải – nơi mà chất lượng dự báo ETA tác động trực tiếp đến hiệu quả điều phối phương tiện, thời gian phục vụ và mức tiêu hao tài nguyên.

Đề án cũng đóng góp một quy trình xử lý dữ liệu toàn diện, bao gồm làm sạch, tích hợp dữ liệu vùng, tạo đặc trưng không gian–thời gian và bổ sung đặc trưng tắc nghẽn. Phân tích SHAP cho thấy các đặc trưng như *trip_distance*, *avg_speed_mph* và *pickup_hour* đóng vai trò quan trọng nhất trong dự báo thời gian di chuyển. Điều này giúp lý giải hoạt động của mô hình, đồng thời cung cấp nền tảng cho việc ra quyết định trong các bài toán vận tải như tối ưu tuyến đường, điều phối xe theo thời điểm và phân bổ nguồn lực theo khu vực.

Mặc dù đạt được nhiều kết quả tích cực, đề án vẫn tồn tại một số hạn chế. Bộ dữ liệu NYC Taxi 2024 không bao gồm thông tin GPS chi tiết, khiến việc mô hình hóa quỹ đạo phương tiện hoặc các mô hình học sâu theo chuỗi thời gian dài như ST-NN hoặc DeepTTE chưa thể triển khai đầy đủ. Ngoài ra, các yếu tố ngoại sinh như thời tiết, sự kiện đặc biệt hay tai nạn giao thông chưa được tích hợp vào hệ thống đặc trưng, trong khi đây là các yếu tố ảnh hưởng trực tiếp đến chi phí vận tải.

Dựa trên những hạn chế này, đề án đề xuất các hướng phát triển tiếp theo:

- Tích hợp các đặc trưng chính sách mới, đặc biệt là biến `cbd_congestion_fee` trong bộ dữ liệu TLC từ năm 2025 nhằm mô hình hóa chi phí tắc nghẽn trung tâm.
- Xây dựng mô hình dự báo chuyên biệt cho các điều kiện giao thông cực đoan để cải thiện độ ổn định và giảm rủi ro trong vận tải.
- Kết hợp XGBoost với các mô hình học sâu nâng cao như Graph Neural Networks

(GNN) để mô hình hóa luồng giao thông và cấu trúc mạng lưới đô thị.

- Mở rộng nghiên cứu với dữ liệu GPS chi tiết hoặc dữ liệu cảm biến thời gian thực, từ đó tối ưu thêm các thành phần chi phí như mức tiêu hao nhiên liệu, phân bổ đội xe, tối ưu tuyến đường động.

Tổng kết lại, đề án đã minh chứng rằng mô hình XGBoost là lựa chọn hiệu quả và cân bằng nhất cho bài toán dự đoán chuỗi thời gian trong tối ưu chi phí vận tải. Mô hình kết hợp tốt giữa độ chính xác, tốc độ huấn luyện và khả năng diễn giải, mở ra nhiều tiềm năng ứng dụng trong hệ thống vận tải thông minh và các dịch vụ logistics đô thị.

TÀI LIỆU THAM KHẢO

[1] J. Zhang, Y. Zheng, and D. Qi, "Urban travel time prediction based on traffic big data," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 1–15, 2014.

[2] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term traffic flow prediction with XGBoost," *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Learning spatio-temporal features for deep neural networks in traffic prediction," *International Conference on Learning Representations (ICLR)*, 2019.

[4] Google AI Research Team, "DeepETA: Improving estimated time of arrival using deep learning," *arXiv preprint arXiv:2006.08528*, 2020.

[5] Y. Bai, J. Wang, and M. Li, "TETA: Transformer-based ETA prediction for large-scale ride-hailing platforms," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[6] K. Xu, Z. Li, and S. Wang, "Graph-based travel time prediction with GCN and gradient boosting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5107–5118, 2020.

[7] S. C. Nayak, B. Misra, and H. Behera, "A review of ARIMA vs. machine learning approaches for time series forecasting," *Future Internet*, vol. 15, no. 8, art. 255, pp. 1–25, Aug. 2023. [Online]. Available: `https://www.mdpi.com/1999-5903/15/8/255`

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984.

[9] J. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[11] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009.

[12] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *The Canadian Cartographer*, vol. 10, no. 2, pp. 112–122, 1973.

[13] R. Silveira, "Using OSRM routing features for trip duration," Kaggle Kernel, 2017.

[14] S. Cazala, "NYC taxi trip duration – feature engineering and OSRM," Kaggle Discussion Board, 2017.

[15] P. Candanedo, "Exploratory data analysis and feature engineering for NYC taxi fare," Kaggle Notebook, 2019.

[16] N. Lee, "Spatial aggregation and geohash-based feature engineering," Kaggle Notebook, 2020.

[17] Kaggle Forum, "Avoiding data leakage in feature engineering," 2019.

[18] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[19] J. Weng, K. Feng, Y. Fu, J. Wang, and L. Mao, "Extreme gradient boosting algorithm based urban daily traffic index prediction model: a case study of Beijing, China," *Digital Transportation and Safety*, vol. 2, no. 3, pp. 220–228, 2023. [Online]. Available: `https://www.maxapress.com/article/doi/10.48130/DTS-2023-0018`

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[21] Y. Bai, J. Wang, and M. Li, "TETA: Transformer-based ETA prediction for large-scale ride-hailing platforms," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[22] R. Guo, C. Zhang, and H. Liu, "ST-Transformer: Spatial-temporal transformer networks for traffic speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6782–6795, Jun. 2023.

[23] X. Zhao, L. Wu, and Y. Zhou, "Graph Transformer Network for Spatio-temporal Travel Time Prediction in Urban Transportation Systems," *Transportation Research Part C: Emerging Technologies*, vol. 168, art. 104718, 2024.

[24] J. Sun, T. He, and Q. Zhang, “DeepETA-v2: Multi-source Transformer model for accurate travel time prediction in large-scale urban networks,” *IEEE Access*, vol. 13, pp. 47521–47535, 2025.

[25] L. Zhang, C. Wang, and Y. Xu, “ETA prediction with real-time traffic and weather data using gated graph neural networks,” *Transportation Research Part C: Emerging Technologies*, vol. 137, art. 103624, 2022.

[26] H. Arbabkhah, A. Sedaghat, M. Jafarikang, and M. Hamidi, “Automatic identification system-based prediction of tanker and cargo estimated time of arrival in narrow waterways,” *Journal of Marine Science and Engineering*, vol. 12, no. 2, art. 215, 2024. [Online]. Available: `https://www.mdpi.com/2077-1312/12/2/215`

[27] J. Feng, Q. Yang, and L. He, “Bi-directional LSTM-based travel time estimation for city bus systems under dynamic traffic conditions,” *Transportation Research Part C: Emerging Technologies*, vol. 163, art. 104440, 2024.

[28] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, “When will you arrive? Estimating travel time based on deep neural networks,” *AAAI Conference on Artificial Intelligence*, 2018.

[29] A. Bellet, “Kaggle NYC taxi trip duration – 1st place solution,” Kaggle Competition Discussion, 2017.

[30] M. Sandler, “Comprehensive feature engineering for ETA prediction,” Kaggle Tutorial, 2019.