

PHỤ LỤC

Phụ lục A

TẬP DỮ LIỆU YELPRESDATA.DB

1. Tổng quan tập dữ liệu

Tập dữ liệu *YelpResData.db* [3] bao gồm các đánh giá nhà hàng được thu thập từ Yelp, phản ánh quan điểm người dùng đối với chất lượng dịch vụ, món ăn và trải nghiệm tổng thể. Tập dữ liệu gồm ba bảng chính: review, restaurant và reviewer. Cấu trúc các bảng dữ liệu và phân bố điểm đánh giá của tập dữ liệu được trình bày tại Bảng A.1 - Bảng A.4.

2. Cấu trúc bảng restaurant review

Bảng A.1: Cấu trúc bảng đánh giá nhà hàng

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	date	varchar2(30)	Ngày đăng đánh giá
2	reviewID	varchar2(30)	Mã định danh đánh giá
3	reviewerID	varchar2(30)	Mã người đánh giá
4	reviewContent	varchar2(800)	Nội dung văn bản đánh giá
5	rating	INT	Điểm đánh giá (1–5)
6	usefulCount	INT	Số lượt đánh giá “hữu ích”
7	coolCount	INT	Số lượt đánh giá “thú vị”
8	funnyCount	INT	Số lượt đánh giá “hài hước”
9	flagged	varchar2(3)	Trạng thái gắn cờ
10	restaurantID	varchar2(30)	Mã nhà hàng

- Tổng số bản ghi: 788.471
- Khoảng thời gian: từ 12/10/2004 đến 22/10/2012

Bảng A.2: Phân bố điểm đánh giá nhà hàng

Điểm	Số lượng	Tỷ lệ (%)
1	45.642	5,79
2	71.672	9,09
3	155.427	19,71
4	290.538	36,85

5	225.192	28,56
---	---------	-------

3. Cấu trúc bảng *restaurant*

Bảng A.3: Cấu trúc bảng *restaurant*

Nhóm thông tin	Thuộc tính
Định danh	restaurantID, name, location
Đánh giá	reviewCount, rating, filReviewCount
Phân loại	categories
Thông tin dịch vụ	PriceRange, TakesReservations, Delivery, Takeout
Tiện ích	WiFi, Parking, OutdoorSeating, WheelchairAccessible
Môi trường	Ambience, NoiseLevel
Khác	Alcohol, GoodFor, HasTV, Caters
Liên hệ	address, phoneNumber, webSite
Chính sách	GoodforKids, GoodforGroups, Attire

- Số bản ghi bảng *restaurant*: 242.652 (số mã nhà hàng khác nhau xuất hiện trong bảng *review*: 250.078)

4. Cấu trúc bảng *restaurant reviewer*

Bảng A.4: Cấu trúc bảng *restaurant reviewer*

Thuộc tính	Mô tả
reviewerID	Mã người đánh giá
name	Tên người dùng
location	Địa điểm
yelpJoinDate	Thời điểm tham gia Yelp
friendCount	Số bạn bè
reviewCount	Số bài đánh giá
usefulCount, coolCount, funnyCount	Thống kê phản hồi
complimentCount	Số lời khen
tipCount	Số mẹo

- Số bản ghi bảng *reviewer*: 16.941 (số người đánh giá khác nhau xuất hiện trong bảng *review*: 35.430)

Phụ lục B

TẬP DỮ LIỆU YELPHOTELDATA.DB

1. Tổng quan tập dữ liệu

Tập dữ liệu *YelpHotelData.db* [3] phản ánh các đánh giá của người dùng đối với khách sạn, với cấu trúc tương tự tập nhà hàng, cho phép so sánh và kiểm chứng tính ổn định của phương pháp đề xuất trên các miền dữ liệu khác nhau. Cấu trúc các bảng dữ liệu và phân bố điểm đánh giá được trình bày tại Bảng B.1-Bảng B.4.

2. Cấu trúc bảng hotel review

Bảng B.1: Cấu trúc bảng đánh giá khách sạn

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	date	varchar2(30)	Ngày đánh giá
2	reviewID	varchar2(30)	Mã đánh giá
3	reviewerID	varchar2(30)	Mã người đánh giá
4	reviewContent	varchar2(800)	Nội dung đánh giá
5	rating	INT	Điểm đánh giá
6	usefulCount	INT	Số lượt hữu ích
7	coolCount	INT	Số lượt thú vị
8	funnyCount	INT	Số lượt hài hước
9	flagged	varchar2(3)	Trạng thái
10	hotelID	varchar2(30)	Mã khách sạn

- Tổng số bản ghi: 688.329
- Khoảng thời gian: 12/10/2004 – 21/10/2012

Bảng B.2: Phân bố điểm đánh giá khách sạn

Điểm	Số lượng	Tỷ lệ (%)
1	32.390	4,71
2	62.560	9,09
3	154.637	22,47
4	267.027	38,79
5	171.715	24,95

3. Cấu trúc bảng *hotel*

Bảng B.3: Cấu trúc bảng *hotel*

Nhóm	Thuộc tính
Định danh	hotelID, name, location
Đánh giá	reviewCount, rating, filReviewCount
Phân loại	categories
Dịch vụ	PriceRange, WiFi
Thanh toán	AcceptsCreditCards
Liên hệ	address, phoneNumber, webSite

- Số bản ghi bảng *hotel*: 283.086 (số mã khách sạn khác nhau xuất hiện trong bảng *review*: 283.291)

4. Cấu trúc bảng *reviewer*

Bảng B.4: Cấu trúc bảng *hotel reviewer*

Thuộc tính	Mô tả
reviewerID	Mã định danh duy nhất của người đánh giá
name	Tên người đánh giá
location	Địa điểm sinh sống của người đánh giá
yelpJoinDate	Thời điểm người dùng tham gia Yelp
friendCount	Số lượng bạn bè trên Yelp
reviewCount	Tổng số bài đánh giá đã đăng
firstCount	Số lượng đánh giá đầu tiên (first reviews)
usefulCount	Số lượt đánh giá được gắn nhãn “hữu ích”
coolCount	Số lượt đánh giá được gắn nhãn “thú vị”
funnyCount	Số lượt đánh giá được gắn nhãn “hài hước”
complimentCount	Tổng số lời khen nhận được
tipCount	Số lượng mẹo (tips) đã đăng
fanCount	Số lượng người theo dõi (fans)

- Số bản ghi bảng *reviewer*: 5.123 (số người đánh giá khác nhau xuất hiện trong bảng *review*: 5.132).

Phụ lục C

CHUYỂN ĐỔI DỮ LIỆU ĐÁNH GIÁ SANG CẤU TRÚC BỘ NĂM QUAN ĐIỂM

1. Chuyển đổi dữ liệu YelpResData.db

1.1 Cấu trúc dữ liệu đầu vào

Sau khi kết hợp các bảng review, restaurant và reviewer, mỗi bản ghi đánh giá được biểu diễn dưới dạng một dòng, trong đó mỗi dòng tương ứng với một đánh giá và mỗi cột biểu diễn một thuộc tính hoặc khía cạnh của nhà hàng tại Bảng C.1.

Xử lý giá trị thiếu (YelpResData): Trong quá trình kết hợp các bảng review, restaurant và reviewer, một số bản ghi có thể thiếu thông tin ở các cột khía cạnh. Chiến lược xử lý: (1) Bản ghi thiếu restaurantName hoặc reviewerName bị loại bỏ; (2) Các khía cạnh có giá trị null không được chuyển thành bộ năm; (3) Cột date thiếu được điền bằng ngày trung vị trong cùng nhà hàng. Sau xử lý, tổng số bộ năm hợp lệ là 14.791.500 mẫu.

Bảng C.1: Cấu trúc dữ liệu đầu vào – Restaurant

Thuộc tính	Mô tả	Vai trò
review_id	Mã định danh đánh giá	Định danh
reviewContent	Nội dung đánh giá	Nguồn trích xuất
restaurantName	Tên nhà hàng	Thực thể
reviewerName	Người đánh giá	Người giữ quan điểm
date	Ngày đánh giá	Thời gian
rating	Điểm đánh giá (1–5)	Tình cảm
GoodforKids	Phù hợp trẻ em	Khía cạnh
AcceptsCreditCards	Chấp nhận thẻ	Khía cạnh
Parking	Bãi đỗ xe	Khía cạnh
Attire	Trang phục	Khía cạnh
PriceRange	Mức giá	Khía cạnh
...	Các thuộc tính khác	Khía cạnh

1.2 Phương pháp chuyển đổi sang bộ năm quan điểm nhà hàng

Sử dụng hàm *pivot_longer()* trong gói *tidyr* của R để thực hiện phép chuẩn hóa dữ liệu từ dạng rộng sang dạng dài.

Đoạn mã chuyển đổi tại Hình C.1 có ý nghĩa như sau:

- Giữ nguyên các thuộc tính của quan điểm:
 - `restaurantName` → Thực thể (entity)
 - `reviewerName` → Người giữ quan điểm (holder)
 - `date` → Thời gian (time)
 - `rating` → dùng làm tình cảm tổng thể
- Biến đổi các cột khía cạnh thành các dòng quan điểm riêng biệt:
 - Tên cột → Khía cạnh (aspect)
 - Giá trị gán → Tình cảm (sentiment)

Sau chuyển đổi, mỗi bản ghi tương ứng với một quan điểm – khía cạnh cụ thể.

```
library(tidyr)

restaurant_aspect <- pivot_longer(
  restaurant,
  cols = -c(review_id, reviewContent, restaurantName,
            reviewerName, date, rating),
  names_to = "aspect",
  values_to = "sentiment"
)
```

Hình C.1: Đoạn mã chuyển đổi trên bộ dữ liệu `yelpResData`

1.3. Cấu trúc dữ liệu đầu ra

Dữ liệu sau chuyển đổi có dạng bộ năm quan điểm ở Bảng C.2.

Bảng C.2: Cấu trúc dữ liệu đầu ra – Restaurant

entity	aspect	holder	sentiment	time
Alinea	GoodforKids	Scott E.	5	09/22/2012
Alinea	AcceptsCreditCards	Scott E.	5	09/22/2012
Alinea	Parking	Scott E.	5	09/22/2012
Alinea	Attire	Scott E.	5	09/22/2012
...	...	...	...	...

2. Chuyển đổi dữ liệu `YelpHotelData.db`

2.1 Cấu trúc dữ liệu đầu vào

Xử lý giá trị thiếu (YelpHotelData): Áp dụng chiến lược tương tự YelpResData. Các bản ghi thiếu hotelName hoặc reviewerName bị loại bỏ; các khía cạnh null (WiFi, PriceRange, AcceptsCreditCards) bị bỏ qua khi pivot; cột date thiếu được điền bằng ngày trung vị trong cùng khách sạn. Sau xử lý, tổng số bộ năm quan điểm là 882.474 mẫu.

Bảng C.3: Cấu trúc dữ liệu đầu vào – Hotel

Thuộc tính	Mô tả	Vai trò
review_id	Mã đánh giá	Định danh
reviewContent	Nội dung đánh giá	Nguồn
hotelName	Tên khách sạn	Thực thể
reviewerName	Người đánh giá	Người giữ quan điểm
date	Ngày đánh giá	Thời gian
rating	Điểm đánh giá	Tình cảm
WiFi	Dịch vụ WiFi	Khía cạnh
PriceRange	Mức giá	Khía cạnh
Cleanliness	Vệ sinh	Khía cạnh
RoomService	Dịch vụ phòng	Khía cạnh
Location	Vị trí	Khía cạnh
...	Các khía cạnh khác	Khía cạnh

2.2 Phương pháp chuyển đổi sang bộ năm quan điểm khách sạn

Phương pháp chuyển đổi tương tự tập nhà hàng, sử dụng pivot_longer() để ánh xạ mỗi khía cạnh thành một quan điểm có đoạn mã ở Hình C.2.

```
hotel_aspect <- pivot_longer(
  hotel,
  cols = -c(review_id, reviewContent, hotelName,
             reviewerName, date, rating),
  names_to = "aspect",
  values_to = "sentiment"
)
```

Hình C.2: Đoạn mã chuyển đổi trên bộ dữ liệu yelpHotelData

2.3. Cấu trúc dữ liệu đầu ra quan điểm khách sạn

Dữ liệu sau chuyển đổi có dạng bộ năm quan điểm khách sạn ở Bảng C.4

Bảng C.4: Cấu trúc dữ liệu đầu ra – Hotel

entity	aspect	holder	sentiment	time
Tokyo Hotel	AcceptsCreditCards	Max C.	5	06/08/2011
Tokyo Hotel	PriceRange	Max C.	5	06/08/2011
Tokyo Hotel	WiFi	Max C.	5	06/08/2011
...	...	...	...	...

Phụ lục D
CHI TIẾT CÔNG CỤ VÀ THƯ VIỆN SỬ DỤNG

Bảng D.1: Danh sách chi tiết các thư viện R và công cụ sử dụng trong thực nghiệm

STT	Danh mục	Thư viện	Phiên bản	Mục đích sử dụng
MÔI TRƯỜNG PHÁT TRIỂN				
1	Môi trường	R	4.3.2	Ngôn ngữ lập trình chính cho toàn bộ thực nghiệm
2	Môi trường	RStudio	2023.12.1	Môi trường phát triển tích hợp để phát triển và gỡ lỗi mã nguồn
XỬ LÝ DỮ LIỆU				
3	Xử lý dữ liệu	dplyr	1.1.4	Thao tác và biến đổi dữ liệu dạng ngăn nắp
4	Xử lý dữ liệu	tidyr	1.3.0	Chuyển đổi cấu trúc dữ liệu (dạng rộng ↔ dạng dài)
5	Xử lý dữ liệu	data.table	1.14.10	Xử lý hiệu năng cao với tập dữ liệu lớn
6	Xử lý dữ liệu	readr	2.1.4	Đọc dữ liệu định dạng CSV và tập tin văn bản
XỬ LÝ VĂN BẢN				
7	Khai thác văn bản	tm	0.7-11	Khung làm việc xử lý văn bản: loại bỏ từ dừng, dấu câu, trích rút gốc từ
8	Khai thác văn bản	text2vec	0.6.3	Tạo ma trận TF-IDF và nhúng từ
9	Khai thác văn bản	tidytext	0.4.1	Phân tích văn bản theo phương pháp ngăn nắp
10	Khai thác văn bản	tokenizers	0.3.0	Tách từ cho văn bản tiếng Anh
11	Khai thác văn bản	SnowballC	0.7.1	Các thuật toán trích rút gốc từ (Porter, Snowball)
12	Khai thác văn bản	stringr	1.5.1	Xử lý và thao tác chuỗi ký tự
13	Khai thác văn bản	word2vec	0.3.4	Xây dựng vector nhúng từ Word2Vec
14	Khai thác văn bản	glove	0.2.0	Xây dựng vector nhúng từ GloVe
RỪNG PHÂN LOẠI (MÔ HÌNH ĐỀ XUẤT)				
15	Rừng ngẫu nhiên	randomForest	4.7-1.1	Cài đặt chính thức thuật toán [65]

16	Rừng ngẫu nhiên	ranger	0.16.0	Cài đặt Rừng ngẫu nhiên nhanh với xử lý song song
17	Rừng ngẫu nhiên	randomForest Explainer	0.10.1	Giải thích Rừng ngẫu nhiên: tầm quan trọng biến, tương tác
RỪNG MẬT ĐỘ (MÔ HÌNH ĐỀ XUẤT)				
18	Phân cụm	dbscan	1.1-12	Thuật toán DBSCAN cho phân cụm dựa trên mật độ
19	Phân cụm	fpc	2.2-11	Các phương pháp phân cụm linh hoạt và kiểm định cụm
20	Phân cụm	cluster	2.1.4	Phân tích cụm và phân cụm thứ bậc
21	Phân cụm	factoextra	1.0.7	Trực quan hóa kết quả phân cụm
22	Học quần thể	randomForest	4.7-1.1	Xây dựng Rừng ngẫu nhiên trong từng cụm
23	Học quần thể	rpart	4.1.23	Thuật toán CART cho cây quyết định
RỪNG ĐA TẬP (MÔ HÌNH ĐỀ XUẤT)				
24	Rừng đa tập	tree	1.0-43	Xây dựng cây quyết định không giám sát để học cấu trúc đa tập
25	Rừng đa tập	randomForest	4.7-1.1	Huấn luyện rừng ngẫu nhiên cho từng cụm trên không gian đa tập
26	Rừng đa tập	rpart	4.1.23	Phân hoạch đệ quy với entropy dựa trên ma trận hiệp phương sai
27	Rừng đa tập	dbscan	1.1-12	Phân cụm DBSCAN trên ma trận quan hệ đa tập
28	Rừng đa tập	caret	6.0-94	Huấn luyện và đánh giá mô hình với kiểm định chéo cho các mô hình chuyên biệt theo cụm
29	Rừng đa tập	tidymodels	1.1.1	Khung mô hình hóa gọn gàng cho quy trình làm việc và công thức
TĂNG CƯỜNG GRADIENT				
30	Tăng cường Gradient	gbm	2.1.9	Máy tăng cường Gradient (Friedman 2001)
31	Tăng cường Gradient	xgboost	1.7.7.1	Tăng cường Gradient cực đại với hỗ trợ GPU
32	Tăng cường Gradient	lightgbm	3.3.5	LightGBM (Microsoft) cho tập dữ liệu lớn

MÁY VECTOR HỖ TRỢ				
33	SVM	e1071	1.7-14	Gói bao bọc LIBSVM cho SVM với hàm nhân RBF
34	SVM	kernlab	0.9-32	Các phương pháp học máy dựa trên hàm nhân
HỌC SÂU (LSTM, BiLSTM-Attention)				
35	Học sâu	keras	2.13.0	Giao diện R của API Keras cho học sâu
36	Học sâu	tensorflow	2.13.0	Nền tảng TensorFlow cho keras
37	Học sâu	reticulate	1.35.0	Giao diện R-Python cho các khung làm việc học sâu
TRANSFORMER (BERT, RoBERTa)				
38	Transformer	reticulate	1.35.0	Cầu nối R-Python để sử dụng Hugging Face transformers
39	Transformer	transformers (Python)	4.35.0	Hugging Face Transformers: các mô hình tiền huấn luyện BERT, RoBERTa
40	Transformer	torch (Python)	2.1.0	Nền tảng PyTorch cho các mô hình Transformer
HUẤN LUYỆN VÀ ĐÁNH GIÁ				
41	Huấn luyện	caret	6.0-94	Giao diện thống nhất cho huấn luyện và điều chỉnh siêu tham số
42	Huấn luyện	mlr3	0.17.0	Khung làm việc học máy hiện đại trong R
43	Huấn luyện	tidymodels	1.1.1	Hệ sinh thái mô hình hóa ngăn nắp
44	Đánh giá	yardstick	1.3.0	Các độ đo hiệu năng: độ chính xác, độ chuẩn xác, độ thu hồi, F1
45	Đánh giá	MLmetrics	1.1.3	Các độ đo học máy bổ sung
46	Lấy mẫu lại	rsample	1.2.0	Lấy mẫu phân tầng và kiểm định chéo
47	Lấy mẫu lại	boot	1.3-28	Các phương pháp bootstrap để ước lượng độ không đảm bảo
GIẢI THÍCH MÔ HÌNH				
48	SHAP	treeshap	0.3.1	Thuật toán TreeSHAP cho các mô hình dựa trên cây

49	SHAP	fastshap	0.1.0	Giá trị SHAP xấp xỉ nhanh cho mọi mô hình
50	SHAP	shapviz	0.9.0	Trực quan hóa SHAP: biểu đồ tóm tắt, biểu đồ phụ thuộc
51	Độc lập mô hình	iml	0.11.2	Khung làm việc học máy khả diễn giải
52	Độc lập mô hình	DALEX	2.4.3	Giải thích học máy mô tả
53	Độc lập mô hình	DALEXtra	2.3.0	Các mở rộng cho DALEX
54	Độc lập mô hình	lime	0.5.3	Giải thích cục bộ khả diễn giải độc lập mô hình
55	Độc lập mô hình	breakDown	0.2.1	Giải thích phân bổ biến
TRỰC QUAN HÓA				
56	Trực quan hóa	ggplot2	3.4.4	Ngữ pháp đồ họa cho các biểu đồ chất lượng xuất bản
57	Trực quan hóa	patchwork	1.2.0	Kết hợp nhiều biểu đồ ggplot2 thành bố cục phức tạp
58	Trực quan hóa	cowplot	1.1.2	Tạo lưới biểu đồ và chủ đề
59	Trực quan hóa	ggiraph	0.8.7	Đồ họa ggplot2 tương tác
60	Trực quan hóa mô hình	vip	0.4.1	Biểu đồ tầm quan trọng biến
61	Trực quan hóa mô hình	pdp	0.8.1	Biểu đồ phụ thuộc từng phần
62	Trực quan hóa mô hình	ALEPlot	1.1	Biểu đồ hiệu ứng cục bộ tích lũy

Ghi chú: toàn bộ mô hình rừng đề xuất, mô hình so sánh và thực nghiệm bổ sung được triển khai bằng R; các mô hình học sâu LSTM, BiLSTM-Attention trong thực nghiệm so sánh thống nhất (Phụ lục J) được xây dựng bằng gói torch cho R, còn BERT, RoBERTa, PhoBERT được tinh chỉnh bằng thư viện transformers (Python, nền PyTorch) gọi từ kịch bản R. Các thực nghiệm chính tại Chương 2–4 dùng R 4.3.2; các thực nghiệm bổ sung tại Phụ lục F–J dùng R 4.3.3 và 4.5.2 với cùng các gói. Mô hình trích xuất bộ năm tại mục 1.6 được triển khai bằng Python (mục 1.6.4.2).

Phụ lục E

MÔ TẢ CÁC TẬP DỮ LIỆU SỬ DỤNG

Phụ lục này mô tả chi tiết các tập dữ liệu được sử dụng trong các thực nghiệm của luận án, bao gồm: nguồn gốc, kích thước, định dạng, phân bố nhãn và lý do lựa chọn. Sáu tập dữ liệu được sử dụng cho đa dạng các kịch bản thực nghiệm: Thông tin chi tiết của các tập dữ liệu được tổng hợp tại Bảng E.1 - Bảng E.12.

E.1. yelpHotelData và yelpResData

Hai tập dữ liệu yelpHotelData và yelpResData là dữ liệu đánh giá của người dùng trên nền tảng Yelp.com về khách sạn (Hotel) và nhà hàng (Restaurant) do Mukherjee và cộng sự thu thập và công bố [3]. Các tập dữ liệu này được sử dụng chủ yếu cho mô hình rừng mật độ (Chương 3) và rừng đa tập (Chương 4) vì có phân bố nhãn lệch về lớp positive ($p \approx 1,7-1,9$) và cấu trúc phân nhóm tự nhiên, đặc trưng phù hợp với thiết kế tích hợp DBSCAN của các mô hình.

Bảng E.1: Kết quả thống kê tập dữ liệu yelpHotelData và yelpResData

Thuộc tính	yelpHotelData	yelpResData
Lĩnh vực	Khách sạn (Hotel)	Nhà hàng (Restaurant)
Tổng số đánh giá	688.329	788.471
Số khách sạn/nhà hàng	283.086 (bảng hotel)	242.652 (bảng restaurant)
Số người dùng (reviewers)	5.132 (trong bảng review)	35.430 (trong bảng review)
Tỷ lệ mất cân bằng $p(O)$	$\approx 1,7$ (560.769/321.705 bộ năm)	$\approx 1,9$ (9.650.537/5.140.963 bộ năm)
Định dạng nhãn	Đa lớp (1-5 sao) $\rightarrow$ 2 lớp	Đa lớp (1-5 sao) $\rightarrow$ 2 lớp
Quy ước nhãn	1-3 sao = Negative; 4-5 sao = Positive	1-3 sao = Negative; 4-5 sao = Positive
Loại bỏ	Không (3 sao gộp vào Negative)	Không (3 sao gộp vào Negative)

Mỗi đánh giá được chuyển đổi thành bộ năm quan điểm $x = (e, a, s, h, t)$ sau khi qua quy trình tiền xử lý (chi tiết tại Phụ lục C):

- (1) e (*entity*): tên của khách sạn/nhà hàng được đánh giá;
- (2) a (*aspect*): khía cạnh được đề cập (GoodforKids, Parking, PriceRange, WiFi, ...);
- (3) s (*sentiment*): cực tính cảm xúc (positive/negative);
- (4) h (*holder*): người viết đánh giá (reviewerName);
- (5) t (*time*): thời điểm đăng đánh giá.

E.2. Amazon Reviews

Tập dữ liệu Amazon Reviews được sử dụng từ phiên bản công bố của nghiên cứu [67], bao gồm hàng triệu đánh giá sản phẩm trên Amazon.com. Trong luận án, tập con Amazon Fine Food Reviews (568.454 đánh giá) được sử dụng để kiểm chứng tính tổng quát của mô hình rừng phân loại (Chương 2) trên dữ liệu thương mại điện tử với sự đa dạng về sản phẩm.

Bảng E.2: Kết quả thống kê tập dữ liệu Amazon Reviews

Thuộc tính	Giá trị
Nguồn	Stanford SNAP - Amazon Reviews [67]
Domain sử dụng	Fine Food (Amazon Food Reviews)
Tổng số đánh giá (đầy đủ)	568.454 (toàn bộ); 18.532 (Bài báo 1); 5.411 đánh giá $\rightarrow$ 8.020 bộ năm (F_{CF})
Số đặc trưng đánh giá	Rating (1-5 sao), Review text, Helpful votes, Verified purchase
Tỷ lệ mất cân bằng $p(O)$	$\approx 5,4$ (Mất cân bằng trung bình)
Quy ước nhãn	1-2 sao = Negative; 4-5 sao = Positive; 3 sao (loại bỏ)

E.3. TripAdvisor Reviews

Tập dữ liệu TripAdvisor được trích xuất từ các đánh giá về khách sạn trên nền tảng TripAdvisor. Tập dữ liệu này có cấu trúc đặc biệt với nhiều khía cạnh đánh giá riêng biệt (Value, Rooms, Location, Cleanliness, Check in/front desk, Service, Business service), cho phép kiểm chứng khả năng phân loại đa khía cạnh (multi-aspect ABSA) của mô hình rừng phân loại (Chương 2).

Bảng E.3: Kết quả thống kê tập dữ liệu TripAdvisor Reviews

Thuộc tính	Giá trị
Nguồn	TripAdvisor Hotels Dataset (Wang và cộng sự, KDD 2011 [2])
Tổng số đánh giá	108.891 (tập con sử dụng trong luận án, mục 2.7.1.2)
Số khách sạn	1.850
Số khía cạnh đánh giá riêng	7 (Value, Rooms, Location, Cleanliness, Check in/front desk, Service, Business service)
Định dạng đánh giá khía cạnh	1-5 sao trên mỗi khía cạnh
Số ngôn ngữ	Tiếng Anh

E.4. Tập dữ liệu VLSP-2018 (tiếng Việt)

Mục này mô tả chi tiết tập dữ liệu VLSP-2018 ABSA tiếng Việt, tập dữ liệu ABSA chuẩn cho tiếng Việt hiện được sử dụng để kiểm chứng khả năng tổng quát hóa của ba mô hình rừng đề xuất trên ngôn ngữ Việt. Mục này trình bày toàn bộ quy trình: nguồn gốc, thống kê tập dữ liệu, quy trình chú thích, cơ sở chú thích và khảo sát các tập dữ liệu ABSA tiếng Việt hiện có.

Tập dữ liệu VLSP-2018 ABSA được công bố tại [197] để phân tích tình cảm dựa trên khía cạnh (ABSA) tiếng Việt, nối tiếp tập dữ liệu VLSP-2016 chỉ thực hiện phân loại cực tính tổng thể. Tập dữ liệu cung cấp hai tập con đánh giá nhà hàng và

khách sạn tại Việt Nam, mỗi đánh giá được chú thích bộ ba [189] $\{Entity\#Attribute, Sentiment\}$ ở cấp tài liệu (document-level ABSA).

Bảng E.4: Thống kê tổng quan tập dữ liệu VLSP-2018 ABSA

Thuộc tính	VLSP-2018 Restaurant	VLSP-2018 Hotel
Lĩnh vực	Nhà hàng tại Việt Nam	Khách sạn tại Việt Nam
Số văn bản Train	2.961	3.000
Số văn bản Dev	1.290	2.000
Số văn bản Test	500	600
Tổng số văn bản	4.751	5.600
Số aspects (Train)	9.297	13.948
Số aspects (Dev)	3.443	7.111
Số aspects (Test)	2.419	2.584
Tổng aspects	15.159	23.643
Số cặp Entity#Attribute	12 cặp	34 cặp
Nhân tình cảm	3 lớp: positive, neutral, negative	3 lớp
Nguồn dữ liệu	Foody.vn	Agoda.com/ Booking.com

Bảng E.5: Phân bố cảm xúc trên các tập dữ liệu

Tập dữ liệu	Positive	Neutral	Negative	Tổng aspects
Restaurant Train	7.071 (76,1%)	1.556 (16,7%)	670 (7,2%)	9.297
Restaurant Dev	2.520 (73,2%)	692 (20,1%)	231 (6,7%)	3.443
Restaurant Test	1.669 (69,0%)	591 (24,4%)	159 (6,6%)	2.419
Hotel Train	10.229 (73,3%)	570 (4,1%)	3.149 (22,6%)	13.948
Hotel Dev	5.077 (71,4%)	226 (3,2%)	1.808 (25,4%)	7.111
Hotel Test	1.806 (69,9%)	133 (5,1%)	645 (24,9%)	2.584

Bảng E.6: Top 5 cặp Entity#Attribute phổ biến nhất đối với tập Restaurant Train

Hạng	Entity#Attribute		Số lượng	Tỷ lệ
1	FOOD#QUALITY		2.658	28,6%
2	FOOD#STYLE&OPTIONS		1.746	18,8%
3	FOOD#PRICES		1.401	15,1%
4	RESTAURANT#GENERAL		855	9,2%
5	SERVICE#GENERAL		795	8,6%

Bảng E.7: Top 5 cặp Entity#Attribute phổ biến nhất đối với tập Hotel Train

Hạng	Entity#Attribute	Số lượng	Tỷ lệ
1	SERVICE#GENERAL	1.912	13,7%
2	HOTEL#GENERAL	1.282	9,2%
3	LOCATION#GENERAL	1.196	8,6%
4	HOTEL#COMFORT	1.139	8,2%
5	ROOMS#DESIGN&FEATURES	916	6,6%

Mỗi mẫu trong tập VLSP-2018 có định dạng:

Dòng 1: Ký hiệu “#N” với N là số thứ tự mẫu.

Dòng 2: Văn bản đánh giá tiếng Việt.

Dòng 3: Danh sách các bộ ba {Entity#Attribute, sentiment} cách nhau bởi dấu phẩy.

Ví dụ: cho đánh giá “Hương vị thơm ngon, ăn cay rất thích, nêm nếm vừa miệng”, chú thích là {FOOD#QUALITY, positive}, {FOOD#STYLE&OPTIONS, positive}.

Quy trình chuyển bộ ba thành bộ năm (e, a, s, h, t)

VLSP-2018 cung cấp chú thích bộ ba {Entity#Attribute, Sentiment}. Để nhất quán với bài toán bộ năm $o = (e, a, s, h, t)$ được định nghĩa theo Bing Liu [1], luận án thực hiện bước ánh xạ mở rộng theo quy trình 5 bước:

Bảng E.8: Quy trình ánh xạ từ bộ ba VLSP-2018 sang bộ năm (e, a, s, h, t)

Thành phần	Nguồn trong VLSP-2018	Phương pháp trích xuất	Ví dụ cụ thể
e (Entity)	Phần Entity trong cặp Entity#Attribute	Tách chuỗi theo ký tự '#', lấy phần trước	FOOD#QUALITY → e = FOOD
a (Attribute)	Toàn bộ cặp Entity#Attribute	Lấy nguyên cặp, mã hóa label encoding	FOOD#QUALITY → a = 3
s (Sentiment)	Nhãn Sentiment gốc	positive → 1; negative → 0; neutral → loại bỏ	positive → s = 1
h (Holder)	Không chú thích tường minh	Gán theo từ khóa 3 cấp (xem Bảng E.9)	“gia đình mình” → h = 2
t (Time)	Không có trường ngày đăng bài	Suy từ từ khóa mùa/tháng (Bảng E.10); không có → 4 (không rõ)	“nghỉ hè” → t = 1 (Hạ)

Bước 1-Tách Entity#Attribute thành (e, a): Mỗi cặp được tách bằng ký tự '#': phần trước là e, toàn bộ cặp là a. Bảng ánh xạ $E \rightarrow \{0..4\}$ và $A \rightarrow \{0..11\}$ xây dựng từ tập Train, với E là tập hợp thực thể, A là tập hợp khía cạnh.

Bước 2 - Xử lý nhãn s: Nhãn gốc 3 lớp → 2 lớp: positive → 1; negative → 0; neutral → loại bỏ. Lý do: nhất quán với quy tắc loại mức trung lập áp dụng cho Amazon Food Reviews và TripAdvisor (Bảng 3.3); xem thêm cơ sở tâm lý học người tiêu dùng [168].

Bước 3 - Gán Holder h: VLSP-2018 không chú thích Holder tường minh. Gán theo hệ thống từ khóa 3 cấp (tại Bảng E.9). Mặc định h = 0 khi không có dấu hiệu.

Bước 4 - Trích xuất t: Từ từ khóa mùa/tháng trong câu đánh giá → phân mùa: tháng 1–3 = 0 (Xuân), 4–6 = 1 (Hạ), 7–9 = 2 (Thu), 10–12 = 3 (Đông); không có từ khóa → 4 (không rõ), do VLSP-2018 không cung cấp ngày đăng bài.

Bước 5 - Chuẩn hóa Z-score: Tính μ và σ từ tập Train, áp dụng cho Dev và Test. Kết quả: $\hat{O} \in \mathbb{R}^{n \times 4}$.

Bảng E.9: Tiêu chí chú thích h

Giá trị h	Loại holder	Từ khóa nhận dạng (Cấp 1)	Ý nghĩa	Xu hướng pos
0	khach_le	“tôi”, “mình”, không đề cập người đi cùng	Khách lẻ	65–70%
1	doi_ban	“đi với bạn”, “tụi mình”, “hai đứa”	Đôi bạn	76–82%
2	gia_dinh	“cả nhà”, “đưa con”, “vợ chồng”, “bọn trẻ”	Gia đình	80–85%
3	doanh_nhan	“tiếp khách”, “đổi tác”, “họp”, “công ty”	Doanh nhân	55–63%
4	phượt_thu	“du lịch”, “ghé qua”, “phượt”, “khám phá”	Phượt thủ	78–84%

Hệ thống quyết định 3 cấp cho h :

Cấp 1 - Từ khóa rõ ràng: “gia đình mình”, “hai đứa nhỏ” $\rightarrow h = 2$.

Cấp 2 - Dấu hiệu gián tiếp: “tiệc sinh nhật”, “họp mặt công ty” $\rightarrow$ suy luận.

Cấp 3 - Mặc định: không có dấu hiệu $\rightarrow h = 0$ (khach_le).

Bảng E.10: Tiêu chí chú thích t theo mùa

Giá trị t	Mùa	Khoảng thời gian	Đặc trưng hành vi tiêu dùng	Xu hướng pos
0	Xuân	Tháng 1–3 (Tết, lễ hội)	Tụ họp gia đình, tiệc tất niên, đầu năm nhiều kỳ vọng	72–76%
1	Hạ	Tháng 4–6 (nghỉ hè)	Cao điểm du lịch, nhiều khách nhất, trải nghiệm mới	78–84%
2	Thu	Tháng 7–9 (mưa nhiều miền Nam)	Trung bình, ít du lịch hơn, khách quen địa phương	70–74%
3	Đông	Tháng 10–12 (cuối năm)	Vắng khách nhất, tiêu chuẩn so sánh cao hơn	63–68%

Hệ thống quyết định 3 cấp cho t :

Cấp 1 - Từ khóa mùa: “hè”, “nghỉ hè”, “mùa đông”, “tết” $\rightarrow$ gán trực tiếp.

Cấp 2 - Từ khóa tháng: “tháng 5”, “tháng 12” → suy luận mùa.

Cấp 3 - Không có dấu hiệu: $t = 4$ (không rõ), do VLSP-2018 không cung cấp ngày đăng bài.

Bảng E.11: Độ đo đánh giá chú thích và ngưỡng mục tiêu

Thành phần	Loại biến	Độ đo IAA	Ngưỡng mục tiêu
h (Holder)	Danh nghĩa (5 lớp)	Cohen's Kappa (κ) [79]	$\kappa \geq 0,80$
t (Time/Mùa)	Thứ tự (4 mùa)	Weighted Kappa (κ_w) [198]	$\kappa_w \geq 0,75$
s (Sentiment)	Danh nghĩa (2 lớp)	Cohen's Kappa (κ) [79]	$\kappa \geq 0,80$
e#a (Entity#Attr)	Danh nghĩa (12/34 lớp)	Fleiss' Kappa [199]	$\kappa \geq 0,80$

E.5. Tổng hợp các tập dữ liệu

Bảng E.12: Bảng tổng hợp các tập dữ liệu dùng trong thực nghiệm

Tập dữ liệu	Lĩnh vực	Ngôn ngữ	Kích thước	$\rho(O)$	Sử dụng thực nghiệm
yelpHotelData	Khách sạn (Yelp)	Tiếng Anh	688.329 reviews	$\approx 1,7$	Chương 3, 4
yelpResData	Nhà hàng (Yelp)	Tiếng Anh	788.471 reviews	$\approx 1,9$	Chương 3, 4
Amazon	Thương mại điện tử	Tiếng Anh	568.454 reviews	$\approx 5,4$	Chương 2
TripAdvisor	Khách sạn	Tiếng Anh	108.891 reviews	≈ 4	Chương 2
VLSP-2018 Restaurant	Nhà hàng VN	Tiếng Việt	4.751 mẫu / 12 aspects	$\approx 10,5$	Phụ lục F

VLSP-2018 Hotel	Khách sạn VN	Tiếng Việt	5.600 mẫu / 34 aspects	$\approx 3,2$	Phụ lục F
--------------------	-----------------	------------	---------------------------	---------------	-----------

Bốn tập dữ liệu tiếng Anh và 2 tập tiếng Việt được sử dụng tổng cộng trong các kịch bản thực nghiệm của luận án, đảm bảo đa dạng về: (1) lĩnh vực (khách sạn, nhà hàng, sản phẩm thực phẩm); (2) ngôn ngữ (tiếng Anh, tiếng Việt); (3) mức độ mất cân bằng (từ $\rho \approx 1,7$ đến mất cân bằng nghiêm trọng $\rho \approx 10,5$); (4) cấp độ chú thích (cấp tài liệu, cấp câu, cấp khía cạnh). Sự đa dạng này cho phép đánh giá toàn diện tính tổng quát hóa của các mô hình rừng đề xuất trên các điều kiện thực tế khác nhau.

Phụ lục F

KẾT QUẢ THỰC NGHIỆM TRÊN VLSP-2018

Phụ lục trình bày kết quả thực nghiệm của ba mô hình rừng đề xuất (F_{CF} , F_{DF} , F_{MF}) và năm mô hình cơ sở (Logistic Regression, SVM tuyến tính, Naive Bayes, Random Forest, Gradient Boosting), cùng mô hình tham chiếu PhoBERT (mục F.6), trên dữ liệu gốc VLSP-2018 SA (Restaurant, Hotel) [197]. Toàn bộ thực nghiệm được thực hiện bằng kịch bản R, dùng chung các hàm phân cụm, huấn luyện và đánh giá với Phụ lục H và thực nghiệm so sánh thống nhất, nhằm bảo đảm quy trình đồng nhất giữa dữ liệu tiếng Anh và tiếng Việt.

F.1. Dữ liệu và quy trình xây dựng bộ năm quan điểm

Dữ liệu gốc gồm các tập dữ liệu train, dev và test (được định dạng CSV, mỗi cột ứng với một cặp Entity#Attribute, mã 0 = không đề cập, 1 = positive, 2 = negative, 3 = neutral). Mỗi cặp (đánh giá, khía cạnh) có nhãn được chuyển thành một bộ năm $o = (e, a, s, h, t)$ theo quy tắc sau:

- e: phần Entity của cặp Entity#Attribute (ví dụ FOOD); a: toàn bộ cặp Entity#Attribute (ví dụ FOOD#QUALITY);
- s: positive $\rightarrow 1$, negative $\rightarrow 0$; nhãn neutral được loại bỏ, cùng quy tắc với Amazon Food Reviews và TripAdvisor (Bảng 3.3);
- h: suy tự động bằng so khớp từ khóa trong câu đánh giá theo Bảng E.9 (gia đình, đôi bạn, doanh nhân, phượt thủ); không có dấu hiệu $\rightarrow h = 0$ (khách lẻ);
- t: suy tự động bằng so khớp từ khóa mùa/tháng trong câu theo Bảng E.10 (cấp 1–2); VLSP-2018 không cung cấp ngày đăng bài nên khi không có từ khóa $\rightarrow t = 4$ (không rõ).

Số liệu chi tiết Bảng F.1, Bảng F.2 cho thấy phần lớn bộ năm nhận giá trị mặc định: 92,9% (Restaurant) và 74,3% (Hotel) có $h =$ khách lẻ; 96,6% (Restaurant) và 97,2% (Hotel) có $t =$ không rõ. Như vậy thông tin thực sự phân biệt trong véc-tơ đặc trưng chủ yếu đến từ hai thành phần e, a, đây là giới hạn cốt lõi của việc áp dụng biểu diễn bộ năm cho VLSP-2018, vốn không chú thích Holder và Time.

F.2. Mã hóa đặc trưng và quy trình thực nghiệm

- Mã hóa: e, a, h, t được mã hóa target encoding (làm tròn Laplace, $m = 10$, chỉ ước lượng trên tập Train) và chuẩn hóa Z-score theo tập Train (Phụ lục G–J).

- Chia tập: dùng đúng các tập Train/Dev/Test chính thức của VLSP-2018; Dev dùng để chọn ngưỡng quyết định (tối đa Balanced Accuracy) cho mọi mô hình và làm tập xác thực cho bộ kết hợp Stacking của F_{MF} .

- F_{CF} : một rừng ngẫu nhiên 300 cây với Bootstrap cân bằng lớp (BalancedBoot). F_{DF} : DBSCAN Euclidean với eps thích ứng theo phân vị khoảng cách k-láng giềng, mỗi cụm một rừng BalancedBoot, kết hợp bằng trung bình xác suất. F_{MF} : ma trận quan hệ W từ rừng không giám sát (thật/giả) trên mẫu con 5.000 bộ năm, DBSCAN trên khoảng cách đa tập $dist_{\mathcal{M}} = \overline{1 - W}$ mỗi cụm một rừng BalancedBoot, kết hợp bằng Stacking (meta-learner rừng ngẫu nhiên trên tập Dev).

- Mô hình cơ sở: Logistic Regression [203] (glm, trọng số lớp nghịch tần suất), SVM tuyến tính [86] (e1071, hàm nhân tuyến tính, trọng số lớp), Naive Bayes [204] (e1071), Random Forest [65] (randomForest, 300 cây, Bootstrap mặc định) và Gradient Boosting [84] (gbm, 300 cây, độ sâu tương tác 3, hàm mất mát Bernoulli). Năm mô hình này dùng cùng véc-tơ bộ năm, cùng cách chia và cùng quy tắc chọn ngưỡng như ba mô hình rừng đề xuất.

- Về lựa chọn mô hình cơ sở: bộ mô hình cơ sở trên VLSP-2018 bao gồm đủ ba mô hình học máy dùng trên dữ liệu Yelp tại Chương 3, Chương 4 (Random Forest, Gradient Boosting, SVM), bổ sung hai mô hình tuyến tính, xác suất thông dụng cho phân loại văn bản (Logistic Regression, Naive Bayes). SVM tuyến tính là thuật toán SVM [86] với hàm nhân tuyến tính; SVM tại Chương 3, Chương 4 dùng cấu hình của các bài báo đã công bố (gói e1071, Bảng D.1), còn tại phụ lục này và trong thực nghiệm so sánh thống nhất (Phụ lục J, Bảng J.8) dùng hàm nhân tuyến tính vì phù hợp với véc-tơ đặc trưng ít chiều và có chi phí huấn luyện thấp. Nhóm mô hình học sâu, Transformer dùng văn bản gốc (LSTM, BiLSTM-Attention, BERT, RoBERTa trên Yelp) tương ứng với PhoBERT trên tiếng Việt (mục F.6), vì BERT, RoBERTa bản gốc được tiền huấn luyện cho tiếng Anh. Như vậy, trên cả hai ngôn ngữ, các mô hình dùng bộ năm được so sánh với nhau trong cùng điều kiện, còn các mô hình dùng văn bản gốc được trình bày như mốc tham chiếu.

- Độ đo: Accuracy, F1-macro, F1-weighted, F1-pos, F1-neg, Balanced Accuracy (BA), MCC, ROC-AUC; kiểm định McNemar từng cặp với hiệu chỉnh Holm. Thời gian là thời gian huấn luyện trên CPU.

- Khi một cụm chỉ gồm các điểm trùng nhau (mọi đặc trưng là hằng số), rừng ngẫu nhiên không thể tách nút; cụm đó được dự đoán bằng tỷ lệ positive của cụm (VLSP-2018 chỉ có khoảng 120 tổ hợp (e, a, h, t) khác nhau trên tập Restaurant).

F.3. Kết quả trên VLSP-2018 Restaurant ($p \approx 10,5$)

Số liệu chi tiết tại Bảng F.3, Bảng F.4 (trình bày ở cuối Phụ lục F).

F.4. Kết quả trên VLSP-2018 Hotel ($p \approx 3,2$)

Số liệu chi tiết tại Bảng F.5, Bảng F.6 (trình bày ở cuối Phụ lục F).

F.5. Thảo luận

Trên VLSP-2018 Restaurant ($p \approx 10,5$), mọi mô hình chỉ đạt Balanced Accuracy 0,49–0,66 và ROC-AUC 0,50–0,69, tức khả năng phân biệt hai lớp rất hạn chế. F_{DF} đạt BA cao nhất (0,6596) và MCC cao nhất (0,1799); F_{CF} đạt MCC 0,1602, cao hơn năm mô hình cơ sở (MCC từ -0,0042 đến 0,1438). Random Forest và F_{MF} không học được lớp negative (BA 0,4988 và 0,4919, ROC-AUC 0,5032 và 0,5150): Random Forest dùng Bootstrap mặc định nên thiên về lớp đa số khi $p \approx 11$, còn bộ kết hợp Stacking của F_{MF} huấn luyện trên Dev với số mẫu negative nhỏ (231) cũng thiên hẳn về lớp positive; điều này cho thấy vai trò của cơ chế Bootstrap cân bằng lớp mà F_{CF} , F_{DF} sử dụng. Ở ngưỡng 0,5 (Bảng F.4), SVM tuyến tính, Random Forest và F_{DF} dự đoán gần như toàn bộ là positive (Accuracy 0,9103–0,9130, gần bằng tỷ lệ lớp đa số, $F1\text{-neg} = 0$), cho thấy Accuracy không phải độ đo phù hợp trên tập mất cân bằng mạnh này.

Trên VLSP-2018 Hotel ($p \approx 3,2$), các mô hình đạt BA 0,69–0,73 và ROC-AUC 0,71–0,77. Logistic Regression đạt BA cao nhất (0,7300), tiếp theo là Gradient Boosting (0,7270), F_{DF} (0,7239) và F_{MF} (0,7171); Random Forest đạt Accuracy cao nhất (0,7666) nhưng BA thấp nhất (0,6906) do thiên về lớp positive. F_{DF} và F_{MF} có Accuracy, F1-macro cao nhất trong ba mô hình rừng đề xuất (0,7503/0,7031 và 0,7556/0,7029). Kiểm định McNemar (hiệu chỉnh Holm trên 28 cặp) cho thấy F_{DF} và F_{MF} đúng nhiều hơn có ý nghĩa so với F_{CF} , SVM tuyến tính và Naive Bayes ($p < 0,001$), nhưng không khác biệt có ý nghĩa so với Logistic Regression, Random Forest và Gradient Boosting (p sau hiệu chỉnh từ 0,19 đến 1,0) cũng như giữa F_{DF} với F_{MF} ($p = 1,0$).

Kết quả trên dữ liệu gốc cho thấy: (i) trên VLSP-2018, cực tính cảm xúc được quyết định chủ yếu bởi nội dung câu đánh giá, trong khi véc-tơ bộ năm chỉ mang thông tin về khía cạnh (e, a) và hai thành phần h, t phần lớn nhận giá trị mặc định (Bảng F.2); vì vậy cả ba mô hình rừng lẫn năm mô hình cơ sở đều bị giới hạn ở mức phân biệt thấp; (ii) trong giới hạn đó, trên tập Hotel cơ chế phân cụm của F_{DF} và F_{MF} giúp cải thiện Accuracy, F1-macro so với F_{CF} (trên tập Restaurant, F_{DF} chỉ cải thiện Balanced Accuracy và MCC), phù hợp với nhận định của luận án về vai trò của phân cụm theo mật độ; (iii) trên tập Hotel, F_{DF} và F_{MF} không khác biệt có ý nghĩa thống kê so với Logistic Regression, Random Forest và Gradient Boosting (kiểm định McNemar–Holm), tức trên đặc trưng bộ năm của VLSP-2018, các mô hình rừng đề xuất đạt mức tương đương các mô hình học máy thông dụng. Kết quả này củng cố hạn chế đã nêu tại §5.1 (phụ thuộc vào chất lượng trích xuất bộ năm) và hướng phát triển tại §5.2 (kết hợp biểu diễn ngữ nghĩa PhoBERT với cấu trúc rừng).

F.6. Mô hình tham chiếu PhoBERT

PhoBERT (checkpoint vinai/phobert-base) được tinh chỉnh cho bài toán phân loại nhị phân trên cùng các cặp (đánh giá, khía cạnh), cùng nhãn và cùng tập Train/Dev/Test chính thức như Bảng F.1. Khác với tám mô hình tại Bảng F.3 - F.6 dùng véc-tơ bộ năm (e, a, h, t), đầu vào của PhoBERT là văn bản “khía cạnh + câu đánh giá” (ví dụ: “khía cạnh food quality . <câu đánh giá>”). Vì vậy PhoBERT được trình bày như mốc tham chiếu cho mức thông tin chứa trong văn bản gốc, không nằm trong cùng nhóm so sánh với các mô hình dùng bộ năm.

Cấu hình: độ dài chuỗi tối đa 128, kích thước batch 16, 2 epoch, tốc độ học 2×10^{-5} (AdamW), dừng sớm theo tổn thất trên Dev (patience = 1), hạt giống 2026; ngưỡng quyết định chọn trên Dev theo Balanced Accuracy như các mô hình khác. Huấn luyện trên CPU 8 luồng (PyTorch 2.2.0, không dùng GPU), bộ nhớ đỉnh của tiến trình khoảng 3,96 GB. Tổn thất (train/Dev) sau 2 epoch: Restaurant 0,2068/0,1890; Hotel 0,2687/0,3631.

Số liệu chi tiết tại Bảng F.7 (trình bày ở cuối Phụ lục F).

Khi dùng văn bản gốc, PhoBERT đạt Balanced Accuracy 0,7906 và ROC-AUC 0,8603 trên Restaurant, 0,8323 và 0,9160 trên Hotel. Kết quả này cho thấy phần lớn thông tin về cực tính cảm xúc của VLSP-2018 nằm trong nội dung câu đánh giá,

nhất quán với nhận định (i) tại mục F.5 về giới hạn của véc-tơ bộ năm khi hai thành phần h , t phần lớn nhận giá trị mặc định. Về chi phí, thời gian huấn luyện PhoBERT trên CPU là khoảng 5,5–6,5 giờ mỗi tập, trong khi các mô hình dùng bộ năm cần từ dưới 1 giây đến khoảng 5 phút (Bảng F.3- Bảng J.5). Hai đặc điểm này là cơ sở cho hướng phát triển tại §5.2: kết hợp biểu diễn ngữ nghĩa từ PhoBERT với cấu trúc rừng để giữ khả năng diễn giải và chi phí suy luận thấp của các mô hình đề xuất.

F.7. Mục đích, nguồn dữ liệu, tái lập và giới hạn

- *Mục đích*: phụ lục này kiểm tra khả năng áp dụng của quy trình bộ năm và ba mô hình rừng đề xuất trên một ngôn ngữ khác (tiếng Việt), với dữ liệu có chú thích khía cạnh công khai của cộng đồng. Đây là thực nghiệm bổ sung, không thay thế các kết quả trên dữ liệu tiếng Anh tại Chương 2, Chương 3, Chương 4 và không nhằm so sánh với bảng xếp hạng của VLSP-2018.

- *Nguồn của dữ liệu bộ năm*: từ sáu tệp gốc chính thức của VLSP-2018 SA [197] (Restaurant, Hotel $\times$ Train, Dev, Test). Mỗi dòng ứng với một cặp (đánh giá, khía cạnh) có nhãn positive hoặc negative, gồm các cột: mã đánh giá, tập chia (train/dev/test), e, a, s, h, t và văn bản đánh giá gốc; số dòng của từng tập khớp với cột “Số bộ năm dùng” tại Bảng F.1.

- *Cách xây dựng và cách suy ra h , t* : toàn bộ quá trình là tự động, không có bước gán nhãn thủ công. Ba thành phần e, a, s lấy trực tiếp từ chú thích chính thức của VLSP-2018 (tách cặp Entity#Attribute; nhãn positive, negative; loại nhãn neutral). Hai thành phần h, t được suy bằng so khớp từ khóa (biểu thức chính quy, không phân biệt chữ hoa, chữ thường) trên văn bản đánh giá theo bộ từ khóa của Bảng E.9, Bảng E.10, có bổ sung một số biến thể thường gặp (ví dụ “bố mẹ”, “ông bà”, “bạn bè”, “người yêu”, “đồng nghiệp”, “công tác”, “giảng sinh”, “trung thu”). Khi một câu chứa từ khóa của nhiều nhóm, h được gán theo thứ tự ưu tiên gia đình, doanh nhân, đôi bạn, phượt thủ; không khớp từ khóa nào thì $h = 0$ (khách lẻ) và $t = 4$ (không rõ). Do dùng so khớp từ khóa nên kết quả có nhiễu (ví dụ cụm “dịch vụ du lịch” khiến h nhận giá trị phượt thủ) và chưa có chú thích chuẩn để đo độ chính xác của bước suy diễn; các ngưỡng đồng thuận tại Bảng E.11 là ngưỡng mục tiêu cho chú thích thủ công trong nghiên cứu tiếp theo, chưa áp dụng trong phụ lục này.

- *Khả năng dùng chung cho mô hình cổ điển và hiện đại*: mỗi dòng giữ đồng thời bộ năm và văn bản gốc với cùng mã đánh giá và cùng tập chia chính thức, nên cùng một tập được dùng cho cả tám mô hình trên đặc trưng bộ năm (Bảng F.3–F.6) lẫn mô hình dùng văn bản PhoBERT (Bảng F.7, đầu vào “khía cạnh + câu đánh giá”). Nhờ vậy mọi mô hình được đánh giá trên đúng cùng các cặp (đánh giá, khía cạnh) và cùng tập Test; tập có thể dùng lại cho các mô hình khác (LSTM, BiLSTM-Attention hoặc mô hình ngôn ngữ lớn) mà không phải xây dựng lại dữ liệu.

- *Lý do PhoBERT không được so sánh trực tiếp*: PhoBERT dùng toàn bộ văn bản và tri thức tiền huấn luyện từ kho văn bản tiếng Việt lớn, trong khi tám mô hình tại Bảng F.3–F.6 chỉ dùng véc-tơ bốn đặc trưng (e, a, h, t); hai nhóm khác nhau về không gian đầu vào, về lượng thông tin được tiếp cận và về chi phí tính toán (khoảng 5,5–6,5 giờ CPU mỗi tập so với từ dưới 1 giây đến khoảng 5 phút). Vì vậy, giống cách trình bày BERT, RoBERTa tại Chương 3, Chương 4, PhoBERT được dùng làm mốc tham chiếu cho lượng thông tin chứa trong văn bản; phép so sánh có kiểm soát chỉ thực hiện giữa các mô hình cùng đầu vào bộ năm (Bảng F.3–F.6 và kiểm định McNemar).

- *Tái lập*: một kịch bản thực nghiệm trên phần mềm R (với hạt giống 2026, 300 cây cho các mô hình rừng) đọc sáu tập gốc, sinh hai tập bộ năm nói trên, huấn luyện tám mô hình dùng bộ năm và ghi các bảng kết quả, kết quả tại ngưỡng 0,5, thống kê thành phần h, t và kiểm định McNemar. PhoBERT được tinh chỉnh bằng thư viện transformers (nền tảng PyTorch), gọi từ kịch bản trên phần mềm R.

Bài toán tại phụ lục này là phân loại nhị phân trên từng cặp (đánh giá, khía cạnh) đã biết khía cạnh, khác định dạng gốc của VLSP-2018 Shared Task (phát hiện khía cạnh và cực tính) nên kết quả không so sánh trực tiếp với bảng xếp hạng chính thức của VLSP-2018.

BẢNG SỐ LIỆU CỦA PHỤ LỤC F

Bảng F.1: Số cặp (đánh giá, khía cạnh) và số bộ năm sau khi loại nhãn neutral

Tập	Tổng số cặp	Positive	Neutral	Negative	Số bộ năm dùng	ρ (pos/neg)
Restaurant – train	9.297	7.071	1.556	670	7.741	10,55
Restaurant – dev	3.443	2.520	692	231	2.751	10,91
Restaurant – test	2.419	1.669	591	159	1.828	10,50
Hotel – train	13.948	10.229	570	3.149	13.378	3,25
Hotel – dev	7.111	5.077	226	1.808	6.885	2,81
Hotel – test	2.584	1.806	133	645	2.451	2,80

Bảng F.2: Phân bố thành phần h và t suy từ từ khóa trên tập Train

Tập	h: khách lẻ	h: đôi bạn	h: gia đình	h: doanh nhân	h: phượt thủ	t: Xuân	t: Hạ	t: Thu	t: Đông	không rõ
Restaurant	7.192	213	182	70	84	83	158	5	14	7.481
Hotel	9.941	206	2.413	183	635	142	176	18	40	13.002

Bảng F.3: Kết quả trên tập Test VLSP-2018 Restaurant (1.828 bộ năm; ngưỡng tối ưu BA trên Dev)

Mô hình	Accuracy	F1-macro	F1-weighted	F1-pos	F1-neg	BA	MCC	ROC-AUC	Ngưỡng	Thời gian (s)
Logistic Regression [203]	0,4469	0,3908	0,5436	0,5757	0,2058	0,6175	0,1357	0,6713	0,70	0,0
SVM (tuyến tính) [86]	0,4551	0,3965	0,5519	0,5847	0,2083	0,6220	0,1404	0,6777	0,95	3,9

Mô hình	Accuracy	F1-macro	F1-weighted	F1-pos	F1-neg	BA	MCC	ROC-AUC	Ngưỡng	Thời gian (s)
Naive Bayes [204]	0,6668	0,5079	0,7389	0,7876	0,2281	0,6212	0,1438	0,6190	0,94	0,3
Random Forest [65]	0,8900	0,4900	0,8631	0,9417	0,0383	0,4988	-0,0042	0,5032	0,85	1,3
Gradient Boosting [84]	0,4759	0,4088	0,5734	0,6080	0,2096	0,6220	0,1391	0,6660	0,95	1,9
Rừng phân loại (F_{CF})	0,6406	0,5001	0,7190	0,7651	0,2352	0,6382	0,1602	0,6850	0,37	0,3
Rừng mật độ (F_{DF})	0,5602	0,4635	0,6516	0,6912	0,2357	0,6596	0,1799	0,6871	0,59	0,9
Rừng đa tạp (F_{MF})	0,8463	0,4913	0,8423	0,9162	0,0664	0,4919	-0,0171	0,5150	0,95	289,0

Bảng F.4: Đối chiếu tại ngưỡng cố định 0,5 – VLSP-2018 Restaurant

Mô hình	Accuracy	F1-macro	F1-pos	F1-neg	BA	MCC
Logistic Regression [203]	0,6422	0,5044	0,7658	0,2431	0,6504	0,1742
SVM (tuyến tính) [86]	0,9130	0,4773	0,9545	0,0000	0,5000	0,0000
Naive Bayes [204]	0,8217	0,5213	0,9005	0,1421	0,5268	0,0463
Random Forest [65]	0,9103	0,4765	0,9530	0,0000	0,4985	-0,0162
Gradient Boosting [84]	0,9054	0,4809	0,9503	0,0114	0,4987	-0,0082

Mô hình	Accuracy	F1-macro	F1-pos	F1-neg	BA	MCC
Rừng phân loại (F_{CF})	0,6247	0,4937	0,7513	0,2361	0,6437	0,1650
Rừng mật độ (F_{DF})	0,9130	0,4773	0,9545	0,0000	0,5000	0,0000
Rừng đa tập (F_{MF})	0,9054	0,4809	0,9503	0,0114	0,4987	-0,0082

Ghi chú: tại ngưỡng 0,5, rừng mật độ (F_{DF}) và SVM tuyến tính cùng dự đoán toàn bộ bộ năm là positive nên có chỉ số trùng nhau; rừng đa tập (F_{MF}) và Gradient Boosting cùng dự đoán 16 bộ năm là negative (1 dự đoán đúng) nhưng ở các bộ năm khác nhau (hai mô hình có 32 bộ năm dự đoán khác nhau), nên ma trận nhầm lẫn và các chỉ số trùng nhau. Kết quả đã được kiểm tra lại bằng cách chạy lại toàn bộ kịch bản với cùng hạt giống 2026.

Bảng F.5: Kết quả trên tập Test VLSP-2018 Hotel (2.451 bộ năm; ngưỡng tối ưu BA trên Dev)

Mô hình	Accuracy	F1-macro	F1-weighted	F1-pos	F1-neg	BA	MCC	ROC-AUC	Ngưỡng	Thời gian (s)
Logistic Regression [203]	0,7475	0,7041	0,7578	0,8174	0,5909	0,7300	0,4235	0,7666	0,43	0,0
SVM (tuyến tính) [68]	0,6879	0,6614	0,7063	0,7561	0,5666	0,7159	0,3818	0,7658	0,80	20,8
Naive Bayes [204]	0,6867	0,6594	0,7050	0,7557	0,5631	0,7126	0,3761	0,7567	0,86	0,7
Random Forest [65]	0,7666	0,6939	0,7646	0,8431	0,5446	0,6906	0,3881	0,7063	0,94	1,6

Mô hình	Accuracy	F1-macro	F1-weighted	F1-pos	F1-neg	BA	MCC	ROC-AUC	Ngưỡng	Thời gian (s)
Gradient Boosting [84]	0,7475	0,7028	0,7574	0,8180	0,5876	0,7270	0,4193	0,7663	0,69	1,3
Rừng phân loại (F _{CF})	0,6879	0,6612	0,7062	0,7563	0,5661	0,7154	0,3810	0,7544	0,67	1,0
Rừng mật độ (F _{DF})	0,7503	0,7031	0,7592	0,8215	0,5848	0,7239	0,4168	0,7503	0,53	1,6
Rừng đa tạp (F _{MF})	0,7556	0,7029	0,7622	0,8280	0,5779	0,7171	0,4112	0,7495	0,91	277,9

Bảng F.6: Đối chiếu tại ngưỡng cố định 0,5 – VLSP-2018 Hotel

Mô hình	Accuracy	F1-macro	F1-pos	F1-neg	BA	MCC
Logistic Regression [203]	0,7050	0,6704	0,7772	0,5637	0,7111	0,3777
SVM (tuyến tính) [86]	0,7870	0,6892	0,8636	0,5149	0,6721	0,3981
Naive Bayes [204]	0,7401	0,6740	0,8208	0,5271	0,6791	0,3489
Random Forest [65]	0,7891	0,7018	0,8631	0,5404	0,6869	0,4148
Gradient Boosting [84]	0,7895	0,7022	0,8634	0,5409	0,6872	0,4158
Rừng phân loại (F _{CF})	0,6928	0,6645	0,7619	0,5670	0,7158	0,3823
Rừng mật độ (F _{DF})	0,7882	0,7010	0,8625	0,5395	0,6864	0,4130

Mô hình	Accuracy	F1-macro	F1-pos	F1-neg	BA	MCC
Rừng đa tập (F _{MF})	0,7813	0,6658	0,8623	0,4693	0,6483	0,3668

Bảng F.7: Kết quả PhoBERT trên tập Test VLSP-2018 (Restaurant: 1.828 bộ năm; Hotel: 2.451 bộ năm)

Tập	Ngưỡng	Accuracy	F1-macro	F1-weighted	F1-pos	F1-neg	BA	MCC	ROC-AUC	Thời gian (s)
Restaurant	Tối ưu BA trên Dev	0,7735	0,6226	0,8197	0,8613	0,3839	0,7906	0,3646	0,8603	23.447,9
Restaurant	0,5	0,9190	0,6390	—	0,9570	0,3211	0,6029	—	0,8603	—
Hotel	Tối ưu BA trên Dev	0,8233	0,7944	0,8309	0,8716	0,7172	0,8323	0,6091	0,9160	19.747,1
Hotel	0,5	0,8633	0,8077	—	0,9111	0,7043	0,7847	—	0,9160	—

Ghi chú: ROC-AUC không phụ thuộc ngưỡng nên giống nhau ở hai dòng của cùng một tập; “—”: độ đo không được xuất ở lần chạy tại ngưỡng 0,5. Mốc “luôn dự đoán positive” có Accuracy 0,9130 (Restaurant) và 0,7368 (Hotel).

Phụ lục G

SO SÁNH BA MÔ HÌNH F_{CF} , F_{DF} , F_{MF} TRÊN CÙNG NĂM KỸ THUẬT XỬ LÝ MẤT CÂN BẰNG LỚP

Phụ lục này so sánh năm kỹ thuật xử lý mất cân bằng lớp trên cả ba mô hình đề xuất của luận án - F_{CF} (Chương 2), F_{DF} (Chương 3) và F_{MF} (Chương 4) nhằm làm rõ hai vấn đề: (i) “Ba mô hình F_{CF} , F_{DF} , F_{MF} được đánh giá lại trên cùng một tập dữ liệu, cùng tỉ lệ chia và cùng quy trình chia dữ liệu để việc so sánh được đồng nhất”; (ii) “Recall được duy trì ở mức 0,62–0,76 xuyên suốt ba mô hình. So sánh thêm với các kỹ thuật quen thuộc (class weights, balanced random forest, SMOTE, điều chỉnh ngưỡng quyết định) và báo cáo kèm F1-macro, PR-AUC hoặc MCC bên cạnh Accuracy.”

G.1. Thiết lập thực nghiệm

Cả ba mô hình được huấn luyện và đánh giá trên cùng một tập dữ liệu, cùng tỉ lệ chia (60% Train/ 20% Validation/ 20% Test, phân chia ngẫu nhiên với `set.seed(101)`) và cùng cỡ mẫu $N = 112.275$ (giống Phụ lục J), đáp ứng trực tiếp yêu cầu (i) nêu trên. Kiến trúc từng mô hình giữ nguyên như đã trình bày trong luận án:

- Mô hình F_{CF} : một rừng ngẫu nhiên duy nhất, huấn luyện trên toàn bộ tập Train (không phân cụm).

- Mô hình F_{DF} : DBSCAN Euclidean trên không gian đặc trưng của tập Train, sau đó K rừng ngẫu nhiên riêng biệt theo cụm, kết hợp bằng trung bình xác suất qua K rừng (density averaging).

- Mô hình F_{MF} : DBSCAN trên không gian đa tạp (khoảng cách suy ra từ một rừng không giám sát phân biệt dữ liệu thật/giả), sau đó K rừng riêng biệt theo cụm, kết hợp bằng Stacking qua một meta-learner huấn luyện trên Validation.

Riêng với F_{MF} , ma trận khoảng cách đa tạp W có kích thước $N \times N$; ở $N = 112.275$, ma trận này đòi hỏi xấp xỉ 100 GB bộ nhớ, không khả thi trên máy tính thông thường - đúng như hạn chế đã được luận án tự nêu tại mục 4.5.1 (hướng mở rộng quy mô), trong đó có đề xuất giải pháp lấy mẫu con (subsampling) trên tập huấn luyện trước khi xây W . Thực nghiệm này áp dụng cụ thể giải pháp đó: riêng bước phân cụm không gian đa tạp và huấn luyện K rừng cơ sở của F_{MF} dùng một tập con ngẫu nhiên phân tầng theo lớp gồm 5.000 dòng trích từ tập Train

($N_MF_SUBSAMPLE = 5.000$); tập Validation và Test vẫn dùng đầy đủ cho cả ba mô hình để việc đánh giá cuối cùng công bằng và có thể so sánh trực tiếp giữa F_{CF} , F_{DF} , F_{MF} .

Năm biến thể xử lý mất cân bằng lớp được áp dụng khi huấn luyện mỗi rừng cơ sở (một rừng duy nhất với F_{CF} ; từng rừng theo cụm với F_{DF}/F_{MF}): Baseline (bootstrap mặc định, ngưỡng cố định 0,5), ThresholdOnly (bootstrap mặc định, ngưỡng hiệu chỉnh trên Validation theo Balanced Accuracy), ClassWeight, BalancedBoot (bootstrap cân bằng lớp qua sampsize/strata) và SMOTE (oversampling SMOTE bằng gói smotefamily, $K=5$ láng giềng, trên tập huấn luyện của mỗi rừng cơ sở). Riêng với F_{MF} , năm biến thể này chỉ áp dụng cho K rừng cơ sở; meta-learner của Stacking giữ nguyên mặc định ở mọi biến thể, nhằm tách bạch rõ đóng góp của việc xử lý mất cân bằng ở tầng rừng cơ sở, tránh nhầm lẫn với hiệu ứng của Stacking. Việc hiệu chỉnh ngưỡng luôn áp dụng trên xác suất đầu ra cuối cùng của từng mô hình (sau trung bình K rừng với F_{CF}/F_{DF} , sau meta-learner với F_{MF}), đảm bảo nhất quán giữa ba mô hình. Độ đo báo cáo cho mỗi tổ hợp (mô hình, biến thể): Accuracy, Precision, Recall, Specificity, F1, F1-macro, Balanced Accuracy, MCC và PR-AUC (diện tích dưới đường cong Precision-Recall, xấp xỉ hình thang trên lưới ngưỡng 0,01 - 0,99 của tập Test, không phụ thuộc ngưỡng được chọn). Toàn bộ tiền xử lý, mã hóa đặc trưng và phân cụm DBSCAN được giữ nguyên giữa các biến thể, nên khác biệt giữa các hàng trong Bảng G.1, Bảng G.2 chỉ đến từ cách xử lý mất cân bằng lớp.

Đây là thực nghiệm bổ sung, không thay thế số liệu ba mô hình đã báo cáo chính thức trong luận án (Chương 2, Chương 3 và Chương 4) hay tại Phụ lục J.

G.2. Kết quả trên tập Hotel

Số liệu chi tiết tại Bảng G.1 (trình bày ở cuối Phụ lục G).

G.3. Kết quả trên tập Restaurant

Số liệu chi tiết tại Bảng G.2 (trình bày ở cuối Phụ lục G).

Lưu ý về độ đo PR-AUC: chỉ số này được tính bằng quy tắc hình thang trên lưới ngưỡng 0,01 - 0,99. Với F_{CF} (một rừng duy nhất), nhiều quan sát có xác suất dự đoán bằng 1, nên ngay tại ngưỡng 0,99 Recall vẫn còn cao và đoạn đường Precision–Recall ứng với Recall thấp không được tính vào diện tích; đây nhiều khả năng là lý do PR-AUC của F_{CF} (và một phần F_{MF}) thấp hơn tỷ lệ lớp positive dù Balanced Accuracy tương đương F_{DF} . Vì vậy PR-AUC chỉ dùng để so sánh các biến thể của

cùng một mô hình; so sánh giữa các mô hình dựa trên Balanced Accuracy, F1-macro và MCC.

G.4. Thảo luận

Kết quả với Baseline (không xử lý mất cân bằng) tái hiện nhất quán ở cả ba mô hình đề xuất trên cả hai tập dữ liệu: Recall rất cao (0,7719-0,9837) nhưng Specificity rất thấp (0,0746- 0,4950), kéo theo Balanced Accuracy thấp hơn đa số các biến thể có xử lý mất cân bằng (mức giảm rõ rệt nhất ở F_{DF} trên Hotel). Điều này cho thấy khi không xử lý mất cân bằng, cả ba mô hình đều thiên mạnh về lớp đa số. Khoảng Recall 0,62 - 0,76 trong số liệu đã công bố (Chương 3, 4) là một điểm vận hành giữa Recall và Specificity; trong các thực nghiệm chạy lại, cơ chế cân bằng lớp (Balanced Bootstrap kết hợp hiệu chỉnh ngưỡng, mục J.1-J.6) đưa Recall và Specificity về mức cân bằng hơn: nếu bỏ hoàn toàn cơ chế cân bằng (biến thể Baseline), Recall tăng lên 0,77-0,98 nhưng Specificity sụt giảm nghiêm trọng.

Trên cả hai tập dữ liệu, BalancedBoot đạt hoặc gần đạt Balanced Accuracy cao nhất ở đa số tổ hợp mô hình: cả ba mô hình trên Hotel, và F_{CF} trên Restaurant. Với F_{DF} và F_{MF} trên Restaurant, ThresholdOnly và ClassWeight nhỉnh hơn BalancedBoot nhưng chênh lệch nhỏ (0,0010 điểm Balanced Accuracy với F_{DF} và 0,0084 điểm với F_{MF}) - xấp xỉ biên độ biến động giữa các lần chạy randomForest. Nhìn chung, không có biến thể nào trong bốn kỹ thuật xử lý mất cân bằng (ThresholdOnly, ClassWeight, BalancedBoot, SMOTE) vượt trội rõ rệt và nhất quán so với ba biến thể còn lại ở cả ba mô hình, củng cố thêm cho lựa chọn Balanced Bootstrap kết hợp hiệu chỉnh ngưỡng được dùng trong các lần chạy lại Kịch bản 1 tại Phụ lục J (mục J.1–J.6).

So sánh giữa ba mô hình ở cùng điều kiện dữ liệu cho thấy F_{MF} có các độ đo tổng hợp (Balanced Accuracy, F1-macro, MCC) thấp hơn F_{CF} và F_{DF} ở mọi biến thể có xử lý mất cân bằng (riêng biến thể Baseline trên Hotel, F_{MF} cao hơn F_{DF} do F_{DF} -Baseline thiên lệch mạnh về lớp đa số; riêng PR-AUC của F_{MF} thấp hơn F_{DF} nhưng cao hơn F_{CF}), trên cả hai tập dữ liệu. Đây là hệ quả trực tiếp và đã được dự liệu trước của việc phải lấy mẫu con xuống $N = 5.000$ cho bước phân cụm không gian đa tạp và huấn luyện K rừng cơ sở của F_{MF} (do giới hạn bộ nhớ $O(N^2)$ của ma trận W), khiến F_{MF} trong thực nghiệm này học trên lượng dữ liệu gán nhãn ít hơn đáng kể so với F_{CF}/F_{DF} (vốn dùng đủ $N = 112.275$). Do đó, chênh lệch này không nên diễn giải là

F_{MF} kém ưu việt hơn về mặt kiến trúc, mà phản ánh đúng giới hạn khả năng mở rộng quy mô đã nêu tại mục 4.5.1 của luận án; khi có hạ tầng tính toán đủ lớn để xây W ở quy mô đầy đủ (hoặc dùng biểu diễn W thưa/cục bộ theo cụm như luận án đề xuất), khoảng cách hiệu năng này được kỳ vọng thu hẹp lại.

Việc hiệu chỉnh ngưỡng trên Validation (ThresholdOnly trở lên) mang lại cải thiện Balanced Accuracy so với Baseline ở cả ba mô hình (rõ rệt nhất ở F_{DF} trên Hotel; F1-macro không tăng nhất quán), xác nhận rằng phần lớn lợi ích quan sát được đến từ việc xử lý mất cân bằng lớp một cách có hệ thống, phù hợp với đóng góp mà luận án đã công bố, chứ không phải ngẫu nhiên theo từng mô hình riêng lẻ.

Kết quả của F_{DF} với biến thể BalancedBoot (tập Hotel: Accuracy 0,6527/BA 0,6355; tập Restaurant: Accuracy 0,6145/BA 0,6083) gần khớp với số liệu Kịch bản 1 cải tiến tại Phụ lục J (tập Hotel: Accuracy 0,6558/BA 0,6389; tập Restaurant: Accuracy 0,6170/BA 0,6084); chênh lệch dưới 0,5 điểm phần trăm là biến động ngẫu nhiên giữa hai lần chạy độc lập của randomForest (cùng phương pháp, cùng dữ liệu), xác nhận tính lặp lại của kết quả. Trên Restaurant, PR-AUC của năm biến thể với F_{DF} gần như không đổi (0,7488–0,7543) trong khi Accuracy, Recall, Specificity thay đổi rõ, cho thấy lợi ích của các kỹ thuật xử lý mất cân bằng chủ yếu đến từ dịch chuyển điểm vận hành (ngưỡng); trên Hotel, PR-AUC tách biệt hơn (0,7709–0,7799) và BalancedBoot cao nhất, cho thấy cơ chế lấy mẫu cân bằng cải thiện cả năng lực phân biệt lớp.

G.5. Kết luận

Thực nghiệm bổ sung này làm rõ đồng thời hai vấn đề nêu ở đầu phụ lục: (i) ba mô hình F_{CF} , F_{DF} , F_{MF} đã được đánh giá lại trên cùng một tập dữ liệu, cùng tỉ lệ chia 60/20/20 và cùng quy trình chia dữ liệu (riêng F_{MF} có thêm bước lấy mẫu con bắt buộc do giới hạn bộ nhớ, đã công khai và giải thích rõ trong mục G.1); (ii) khi không xử lý mất cân bằng (Baseline), cả ba mô hình đều thiên về lớp đa số với Specificity rất thấp, còn bốn kỹ thuật thay thế thông dụng (ThresholdOnly, ClassWeight, BalancedBoot, SMOTE) đều cải thiện rõ rệt Specificity và nhìn chung cải thiện Balanced Accuracy so với không xử lý (Baseline) ở cả ba mô hình (F1-macro và MCC không cải thiện nhất quán), mà không có kỹ thuật nào vượt trội tuyệt đối so với Balanced Bootstrap được dùng trong các lần chạy lại tại Phụ lục J.

BẢNG SỐ LIỆU CỦA PHỤ LỤC G

Bảng G.1: So sánh 3 mô hình $\times$ 5 biến thể xử lý mất cân bằng lớp — Hotel

Mô hình	Biến thể	Accuracy	Precision	Recall	Specificity	F1	F1-macro	BalancedAccuracy	MCC	PR-AUC
F _{CF}	Baseline	0,6724	0,7315	0,7719	0,4950	0,7512	0,6359	0,6334	0,2736	0,5374
F _{CF}	ThresholdOnly	0,6411	0,7575	0,6469	0,6308	0,6979	0,6280	0,6389	0,2677	0,5374
F _{CF}	ClassWeight	0,6444	0,7535	0,6612	0,6143	0,7043	0,6291	0,6378	0,2665	0,5366
F _{CF}	BalancedBoot	0,6433	0,7562	0,6542	0,6240	0,7015	0,6293	0,6391	0,2685	0,5496
F _{CF}	SMOTE	0,6451	0,7416	0,6844	0,5749	0,7119	0,6249	0,6297	0,2531	0,5363
F _{DF}	Baseline	0,6570	0,6546	0,9837	0,0746	0,7861	0,4607	0,5292	0,1479	0,7761
F _{DF}	ThresholdOnly	0,6427	0,7501	0,6631	0,6061	0,7040	0,6267	0,6346	0,2607	0,7761
F _{DF}	ClassWeight	0,6172	0,7572	0,5925	0,6613	0,6648	0,6093	0,6269	0,2435	0,7709
F _{DF}	BalancedBoot	0,6527	0,7448	0,6966	0,5744	0,7199	0,6315	0,6355	0,2654	0,7799
F _{DF}	SMOTE	0,6291	0,7499	0,6318	0,6244	0,6858	0,6166	0,6281	0,2466	0,7730
F _{MF}	Baseline	0,6547	0,6940	0,8245	0,3520	0,7537	0,5882	0,5882	0,1985	0,5786
F _{MF}	ThresholdOnly	0,6200	0,7161	0,6741	0,5236	0,6945	0,5960	0,5989	0,1939	0,5786
F _{MF}	ClassWeight	0,6171	0,7154	0,6680	0,5262	0,6909	0,5939	0,5971	0,1901	0,5750
F _{MF}	BalancedBoot	0,6204	0,7174	0,6723	0,5278	0,6941	0,5969	0,6000	0,1960	0,6021
F _{MF}	SMOTE	0,6100	0,7078	0,6662	0,5097	0,6864	0,5854	0,5880	0,1725	0,5963

Ghi chú: Các hàng tô đậm (BalancedBoot) là biến thể đạt Balanced Accuracy cao nhất trong 5 biến thể, nhất quán ở cả ba mô hình trên tập Hotel.

Bảng G.2: So sánh 3 mô hình $\times$ 5 biến thể xử lý mất cân bằng lớp — Restaurant

Mô hình	Biến thể	Accuracy	Precision	Recall	Specificity	F1	F1-macro	BalancedAccuracy	MCC	PR-AUC
F _{CF}	Baseline	0,6544	0,7150	0,7815	0,4163	0,7468	0,6014	0,5989	0,2082	0,5802
F _{CF}	ThresholdOnly	0,6025	0,7427	0,5973	0,6124	0,6621	0,5898	0,6048	0,2000	0,5802
F _{CF}	ClassWeight	0,6192	0,7324	0,6555	0,5513	0,6918	0,5969	0,6034	0,1998	0,5744
F _{CF}	BalancedBoot	0,6148	0,7395	0,6317	0,5831	0,6814	0,5972	0,6074	0,2059	0,5939
F _{CF}	SMOTE	0,6110	0,7328	0,6350	0,5662	0,6804	0,5918	0,6006	0,1933	0,5802
F _{DF}	Baseline	0,6604	0,7123	0,8039	0,3916	0,7553	0,6003	0,5978	0,2112	0,7542
F _{DF}	ThresholdOnly	0,6179	0,7404	0,6375	0,5811	0,6851	0,5997	0,6093	0,2099	0,7542
F _{DF}	ClassWeight	0,6089	0,7441	0,6099	0,6071	0,6703	0,5948	0,6085	0,2072	0,7488
F _{DF}	BalancedBoot	0,6145	0,7408	0,6288	0,5878	0,6802	0,5975	0,6083	0,2075	0,7543
F _{DF}	SMOTE	0,6149	0,7400	0,6310	0,5847	0,6812	0,5975	0,6078	0,2068	0,7526
F _{MF}	Baseline	0,6453	0,6881	0,8342	0,2915	0,7541	0,5590	0,5628	0,1471	0,6719
F _{MF}	ThresholdOnly	0,5934	0,7098	0,6366	0,5124	0,6712	0,5692	0,5745	0,1440	0,6719
F _{MF}	ClassWeight	0,6072	0,7060	0,6813	0,4684	0,6934	0,5735	0,5748	0,1476	0,6436
F _{MF}	BalancedBoot	0,5830	0,7044	0,6211	0,5116	0,6602	0,5604	0,5664	0,1279	0,6748
F _{MF}	SMOTE	0,5923	0,7077	0,6383	0,5061	0,6712	0,5674	0,5722	0,1398	0,6759

Ghi chú: Các hàng tô đậm là biến thể đạt Balanced Accuracy cao nhất theo từng mô hình trên tập Restaurant: BalancedBoot (F_{CF}), ThresholdOnly (F_{DF}) và ClassWeight (F_{MF}), khác biệt giữa các biến thể tốt nhất khá nhỏ (dưới 0,003 điểm Balanced Accuracy so với biến thể đứng thứ hai).

Phụ lục H

ABLATION KIẾN TRÚC (2×2) VÀ KIỂM ĐỊNH Ý NGHĨA THỐNG KÊ GIỮA BA MÔ HÌNH F_{CF} , F_{DF} , F_{MF}

Phụ lục này bổ sung hai nội dung mà Phụ lục G, Phụ lục J chưa giải quyết bằng thực nghiệm mới: (i) kiểm định ý nghĩa thống kê (McNemar, Wilcoxon) cho khác biệt hiệu năng giữa các mô hình, thay vì chỉ so sánh điểm số trung bình; (ii) thí nghiệm loại trừ (ablation) tách bạch đóng góp riêng của từng thành phần kiến trúc - không gian phân cụm (Euclidean so với đa tạp) và phương pháp kết hợp K rừng (trung bình so với Stacking) đã bị nhập chung trong F_{DF} và F_{MF} khiến chưa thể quy kết rõ khác biệt hiệu năng đến từ thành phần nào.

H.1. Thiết kế ablation 2×2

Mô hình F_{DF} (Chương 3) kết hợp đồng thời hai lựa chọn kiến trúc: phân cụm trên không gian Euclidean và kết hợp K rừng bằng trung bình xác suất (density averaging). F_{MF} (Chương 4) kết hợp đồng thời hai lựa chọn còn lại: phân cụm trên không gian đa tạp (qua ma trận quan hệ W) và kết hợp bằng Stacking. Do hai thành phần luôn đi cùng nhau trong thiết kế gốc, chênh lệch hiệu năng quan sát được giữa F_{DF} và F_{MF} không thể quy kết riêng cho không gian phân cụm hay cho phương pháp kết hợp. Để tách bạch, thực nghiệm bổ sung hai ô còn thiếu của bảng 2×2:

- *Ablation_Euclid_Stacking*: phân cụm Euclidean (tương tự như F_{DF}) nhưng kết hợp K rừng bằng Stacking (tương tự như F_{MF}).
- *Ablation_Manifold_Avg*: phân cụm trên không gian đa tạp W (tương tự như F_{MF}) nhưng kết hợp K rừng bằng trung bình xác suất (tương tự như F_{DF}).

	Kết hợp: TRUNG BÌNH	Kết hợp: STACKING
Phân cụm: EUCLIDEAN	F_{DF} (đã có, Chương 3)	<i>Ablation_Euclid_Stacking</i> (mới)
Phân cụm: ĐA TẬP (W)	<i>Ablation_Manifold_Avg</i> (mới)	F_{MF} (đã có, Chương 4)

Hình H.1: Thiết kế ablation 2×2 theo không gian phân cụm và cách kết hợp K rừng

Mô hình F_{CF} (không phân cụm, Chương 2) đóng vai trò mốc tham chiếu “không có cụm” cho cả bốn ô trên. Với mỗi cặp so sánh có chung một thành phần (ví dụ F_{DF} và *Ablation_Euclid_Stacking* cùng phân cụm Euclidean, chỉ khác cách kết

hợp), chênh lệch hiệu năng quan sát được có thể quy trực tiếp cho đúng một thành phần kiến trúc khác biệt duy nhất.

Toàn bộ 5 biến thể (F_{CF} , F_{DF} , F_{MF} , *Ablation_Euclid_Stacking*, *Ablation_Manifold_Avg*) dùng chung một kỹ thuật xử lý mất cân bằng lớp cố định là Balanced Bootstrap (kết quả tốt nhất/gần tốt nhất theo Phụ lục G) nhằm cô lập hoàn toàn hiệu ứng kiến trúc, không lẫn với hiệu ứng xử lý mất cân bằng lớp. Toàn bộ quy trình (lấy mẫu $N = 112.275$, chia Train/Validation/Test theo tỉ lệ 60/20/20, hai bước phân cụm, huấn luyện cả 5 biến thể) được lặp lại 5 lần (5 lần chia dữ liệu) với 5 cặp hạt giống ngẫu nhiên độc lập (lấy mẫu: 2026–2030; chia tập: 101–105), báo cáo trung bình $\pm$ độ lệch chuẩn qua 5 lần lặp cho từng độ đo.

H.2. Hai phương pháp kiểm định ý nghĩa thống kê

Kiểm định McNemar được áp dụng trên dự đoán cặp đôi (paired) của cùng một tập Test cố định (lần lặp #1), theo bảng tương ứng 2×2 giữa đúng/sai của hai mô hình trên từng quan sát, phù hợp khi so sánh hai bộ phân loại trên cùng một tập kiểm thử [208]. Gọi b là số quan sát mô hình A dự đoán đúng còn mô hình B dự đoán sai, c là số quan sát A sai còn B đúng; các quan sát cả hai cùng đúng hoặc cùng sai không ảnh hưởng đến kiểm định. Thống kê kiểm định (có hiệu chỉnh liên tục, như hàm `mcnemar.test` của R) là $\chi^2 = (|b - c| - 1)^2 / (b + c)$, tuân theo phân phối chi bình phương 1 bậc tự do dưới giả thuyết H_0 : hai mô hình có cùng tỉ lệ dự đoán đúng (khi đó $b \approx c$); $p\text{-value} = P(\chi^2_1 > \chi^2)$. Vì $\text{Accuracy}(A) - \text{Accuracy}(B) = (b - c)/n$, kiểm định này đánh giá chênh lệch Accuracy của hai mô hình trên cùng tập kiểm tra.

Kiểm định Wilcoxon signed-rank (kiểm định hạng có dấu Wilcoxon, ghép cặp, hai phía) [207] là kiểm định phi tham số, không giả định chênh lệch tuân theo phân phối chuẩn, được áp dụng trên 5 giá trị Balanced Accuracy ứng với 5 lần chia dữ liệu ngẫu nhiên (5 hạt giống khác nhau) của mỗi mô hình. Do các lần chia cùng lấy từ một tập dữ liệu nên không hoàn toàn độc lập và có thể làm kiểm định lạc quan hơn [208]; vì vậy kết quả Wilcoxon được dùng để đánh giá tính nhất quán chiều khác biệt, hỗ trợ cho kiểm định McNemar. Với mỗi cặp mô hình A, B, chênh lệch Balanced Accuracy ở từng lần lặp được xếp hạng theo giá trị tuyệt đối; thống kê V (do hàm `wilcox.test` của phần mềm R trả về) là tổng hạng của các lần lặp mà A cao hơn B. Với $n = 5$, V nhận giá trị từ 0 đến 15: $V = 15$ khi A cao hơn B ở cả 5/5 lần lặp, $V = 0$ khi B cao hơn A ở cả 5/5 lần lặp, còn V gần giá trị kỳ vọng 7,5 khi thắng thua xen kẽ, không có chiều khác biệt nhất quán. Ở hai giá trị cực trị ($V = 0$ hoặc $V = 15$), $p\text{-value}$

chính xác hai phía là 0,0625 ($2/2^5$), là giá trị nhỏ nhất có thể đạt được khi $n = 5$; giá trị 0,05906 tại Bảng H.5–H.6 là p-value xấp xỉ chuẩn (có hiệu chỉnh liên tục) do hàm `wilcox.test` của phần mềm R trả về khi không dùng phép tính chính xác. Do đó ngưỡng $\alpha = 0,05$ quy ước không thể đạt được kể cả khi một mô hình cao hơn ở cả 5/5 lần lặp; đây là hạn chế do cỡ mẫu nhỏ ($N_REPEATS = 5$), không phải bằng chứng cho thấy hai mô hình không khác nhau.

H.3. Bảng tổng hợp mean $\pm$ SD qua 5 lần lặp

Số liệu chi tiết tại Bảng H.1, Bảng H.2 (trình bày ở cuối Phụ lục H).

Lưu ý về PR-AUC: chỉ số này được tính bằng quy tắc hình thang trên lưới ngưỡng 0,01–0,99. Với F_{CF} (một rừng duy nhất), nhiều quan sát có xác suất dự đoán bằng 1, nên ngay tại ngưỡng 0,99 Recall vẫn còn cao và đoạn đường Precision–Recall ứng với Recall thấp không được tính vào diện tích; đây nhiều khả năng là lý do PR-AUC của F_{CF} (và một phần F_{MF}) thấp hơn tỷ lệ lớp positive dù Balanced Accuracy tương đương F_{DF} . Vì vậy PR-AUC chỉ dùng để so sánh các biến thể của cùng một mô hình; so sánh giữa các mô hình dựa trên Balanced Accuracy, F1-macro và MCC.

H.4. Kiểm định McNemar (dự đoán cặp đôi, tập Test cố định)

Số liệu chi tiết tại Bảng H.3, Bảng H.4 (trình bày ở cuối Phụ lục H).

H.5. Kiểm định Wilcoxon signed-rank (5 lần lặp, Balanced Accuracy)

Số liệu chi tiết tại Bảng H.5, Bảng H.6 (trình bày ở cuối Phụ lục H)

H.6. Thảo luận

Kiểm định McNemar (cỡ mẫu bằng số quan sát trong tập Test, rất lớn) cho thấy khác biệt có ý nghĩa thống kê ($p < 0,05$, phần lớn $p < 10^{-10}$) ở 6/7 cặp so sánh trên Hotel và 7/7 cặp trên Restaurant, xác nhận rằng chênh lệch điểm số trung bình đã báo cáo trong luận án và các phụ lục trước (G, H, I) không phải do dao động ngẫu nhiên của một lần chia dữ liệu. Riêng cặp F_{DF} so sánh *Ablation_Euclid_Stacking* trên Hotel không đạt ý nghĩa ($\chi^2 = 0,0969$; $p = 0,7556$): khi giữ nguyên không gian phân cụm Euclidean, số quan sát chỉ F_{DF} dự đoán đúng và số quan sát chỉ bản Stacking dự đoán đúng gần bằng nhau, nên tổng số dự đoán đúng của hai mô hình không khác biệt có ý nghĩa. Điều này không có nghĩa hai mô hình mắc lỗi trên cùng các quan sát: Stacking dịch điểm cân bằng sang lớp positive (Bảng H.1: Recall tăng từ 0,6864 lên 0,7806, Specificity giảm từ 0,5857 xuống 0,4481), tức đổi thêm lỗi ở lớp negative lấy thêm dự đoán đúng ở lớp positive với số lượng gần tương đương. Kết luận “chưa đủ

bằng chứng” cũng không khẳng định hai mô hình tương đương: trên Restaurant, chính cặp này khác biệt có ý nghĩa Bảng H.4).

Ngược lại, mọi cặp so sánh có khác biệt về không gian phân cụm (Euclidean so với đa tạp, ở cùng một phương pháp kết hợp) đều đạt ý nghĩa thống kê rất mạnh qua McNemar (F_{DF} so sánh *Ablation_Manifold_Avg* và *Ablation_Euclid_Stacking* so sánh với F_{MF} đều có $p < 10^{-14}$ ở cả hai tập; cặp F_{DF} so sánh F_{MF} , khác đồng thời cả hai thành phần, cũng có $p < 10^{-23}$), cho thấy lựa chọn không gian phân cụm là thành phần kiến trúc có ảnh hưởng mạnh và nhất quán hơn đến tập dự đoán, so với lựa chọn phương pháp kết hợp K rừng.

Kiểm định Wilcoxon trên 5 lần chia dữ liệu cho kết quả cùng chiều với kiểm định McNemar: ở 6 cặp so sánh còn lại (trừ cặp $F_{CF} - F_{DF}$), một mô hình đạt Balanced Accuracy cao hơn ở cả 5/5 lần lặp tại cả Hotel và Restaurant, tương ứng với giá trị cực trị của thống kê V ($V = 0$ hoặc $V = 15$; xem H.2). Tuy p-value (0,05906) chưa đạt ngưỡng 0,05, đây là p-value nhỏ nhất có thể đạt được khi $n = 5$, nên kết quả này được xem là bằng chứng về tính nhất quán chiều khác biệt qua các lần chia dữ liệu độc lập, chưa phải bằng chứng có ý nghĩa thống kê. Riêng cặp $F_{CF} - F_{DF}$ không nhất quán chiều khác biệt giữa các lần lặp ($V = 7$ ở Hotel và $V = 5$ ở Restaurant, gần giá trị kỳ vọng 7,5), phù hợp với việc đây là cặp có chênh lệch điểm số nhỏ nhất trong toàn bộ các cặp so sánh (xem Bảng H.1–H.2).

H.7. Kết luận

Thực nghiệm ablation 2×2 kết hợp hai phép kiểm định thống kê độc lập (McNemar trên dự đoán cặp đôi, Wilcoxon trên nhiều lần lặp) cho phép kết luận: (i) theo kiểm định McNemar, phần lớn khác biệt hiệu năng giữa ba mô hình đề xuất của luận án có ý nghĩa thống kê, không phải dao động ngẫu nhiên, và kiểm định Wilcoxon cho thấy chiều khác biệt nhất quán qua 5 lần lặp (trừ cặp $F_{CF} - F_{DF}$); (ii) trong hai thành phần kiến trúc cấu thành F_{MF} (không gian phân cụm đa tạp và Stacking), thành phần không gian phân cụm đóng vai trò quyết định hơn đến sự khác biệt trong tập dự đoán, còn Stacking chủ yếu điều chỉnh lại điểm cân bằng Recall/Specificity. Đây là bằng chứng thực nghiệm trực tiếp cho việc kiểm định ý nghĩa thống kê và tách bạch đóng góp của các thành phần DBSCAN, W và Stacking.

BẢNG SỐ LIỆU CỦA PHỤ LỤC H

Bảng H.1: Ablation 2×2 — tập Hotel (mean ± SD, 5 lần lặp)

Mô hình	Accuracy	Precision	Recall	Specificity	F1	F1-macro	BalancedAcc.	MCC	PR-AUC
F _{CF} (không phân cụm)	0,6465 ± 0,0085	0,7454 ± 0,0069	0,6758 ± 0,0245	0,5953 ± 0,0223	0,7087 ± 0,0119	0,6294 ± 0,0058	0,6355 ± 0,0043	0,2642 ± 0,0091	0,5503 ± 0,0034
F _{DF} (Euclid + TB)	0,6499 ± 0,0082	0,7439 ± 0,0042	0,6864 ± 0,0208	0,5857 ± 0,0200	0,7139 ± 0,0113	0,6312 ± 0,0062	0,6360 ± 0,0056	0,2660 ± 0,0115	0,7776 ± 0,0047
F _{MF} (Đa tập + Stacking)	0,6147 ± 0,0062	0,7127 ± 0,0068	0,6615 ± 0,0066	0,5326 ± 0,0085	0,6862 ± 0,0062	0,5936 ± 0,0058	0,5971 ± 0,0061	0,1899 ± 0,0118	0,6302 ± 0,0410
Ablation: Euclid + Stacking	0,6598 ± 0,0053	0,7128 ± 0,0073	0,7806 ± 0,0277	0,4481 ± 0,0353	0,7448 ± 0,0090	0,6166 ± 0,0050	0,6144 ± 0,0056	0,2400 ± 0,0096	0,7547 ± 0,0030
Ablation: Đa tập + TB	0,6142 ± 0,0120	0,7399 ± 0,0081	0,6081 ± 0,0375	0,6247 ± 0,0353	0,6669 ± 0,0211	0,6036 ± 0,0080	0,6164 ± 0,0046	0,2247 ± 0,0089	0,7515 ± 0,0075

Ghi chú: Hàng tô đậm: F_{DF} (mô hình chính thức Chương 3) và Ablation_Manifold_Avg (ô ablation đối trọng cùng không gian đa tập nhưng kết hợp bằng trung bình).

Bảng H.2: Ablation 2×2 — tập Restaurant (mean ± SD, 5 lần lặp)

Mô hình	Accuracy	Precision	Recall	Specificity	F1	F1-macro	BalancedAcc.	MCC	PR-AUC
F _{CF} (không phân cụm)	0,6202 ± 0,0036	0,7334 ± 0,0057	0,6570 ± 0,0158	0,5510 ± 0,0215	0,6930 ± 0,0070	0,5974 ± 0,0024	0,6040 ± 0,0037	0,2010 ± 0,0062	0,5935 ± 0,0042

Mô hình	Accuracy	Precision	Recall	Specificity	F1	F1-macro	BalancedAcc.	MCC	PR-AUC
F _{DF} (Euclid + TB)	0,6165 ± 0,0120	0,7370 ± 0,0016	0,6411 ± 0,0318	0,5698 ± 0,0267	0,6854 ± 0,0183	0,5966 ± 0,0070	0,6055 ± 0,0040	0,2031 ± 0,0086	0,7573 ± 0,0035
F _{MF} (Đa tập + Stacking)	0,5937 ± 0,0059	0,7036 ± 0,0024	0,6523 ± 0,0139	0,4836 ± 0,0132	0,6769 ± 0,0078	0,5647 ± 0,0044	0,5679 ± 0,0040	0,1324 ± 0,0082	0,6714 ± 0,0470
Ablation: Euclid + Stacking	0,6364 ± 0,0216	0,7042 ± 0,0121	0,7668 ± 0,0868	0,3913 ± 0,1008	0,7317 ± 0,0345	0,5770 ± 0,0065	0,5790 ± 0,0071	0,1696 ± 0,0031	0,7356 ± 0,0040
Ablation: Đa tập + TB	0,6017 ± 0,0231	0,7252 ± 0,0088	0,6289 ± 0,0752	0,5497 ± 0,0775	0,6714 ± 0,0393	0,5795 ± 0,0096	0,5893 ± 0,0056	0,1733 ± 0,0100	0,7305 ± 0,0059

Ghi chú: Cấu hình, seed và quy trình lặp giống hệt Bảng H.1 chỉ khác tập dữ liệu (Restaurant, N = 112.275).

Bảng H.3: Kiểm định McNemar — Hotel

Cặp so sánh (McNemar, lần lặp #1)	chi ²	p-value	Kết luận ($\alpha=0,05$)
F _{CF} vs F _{DF}	26,1406	3,174e-07	Có ý nghĩa
F _{CF} vs F _{MF}	48,6990	2,984e-12	Có ý nghĩa
F _{DF} vs F _{MF}	114,8741	8,386e-27	Có ý nghĩa
F _{DF} vs Ablation_Euclid_Stacking	0,0969	0,7556	Chưa đủ bằng chứng
F _{MF} vs Ablation_Manifold_Avg	3,9467	0,04696	Có ý nghĩa (biên)
Ablation_Euclid_Stacking vs F _{MF}	111,8068	3,939e-26	Có ý nghĩa

Cặp so sánh (McNemar, lần lặp #1)	chi ²	p-value	Kết luận ($\alpha=0,05$)
F _{DF} vs Ablation_Manifold_Avg	98,9783	2,553e-23	Có ý nghĩa

Bảng H.4: Kiểm định McNemar — Restaurant

Cặp so sánh (McNemar, lần lặp #1)	chi ²	p-value	Kết luận ($\alpha=0,05$)
F _{CF} vs F _{DF}	10,6152	1,122e-03	Có ý nghĩa
F _{CF} vs F _{MF}	53,7380	2,291e-13	Có ý nghĩa
F _{DF} vs F _{MF}	102,4501	4,424e-24	Có ý nghĩa
F _{DF} vs Ablation_Euclid_Stacking	80,9070	2,366e-19	Có ý nghĩa
F _{MF} vs Ablation_Manifold_Avg	10,2399	1,374e-03	Có ý nghĩa
Ablation_Euclid_Stacking vs F _{MF}	315,5833	1,327e-70	Có ý nghĩa
F _{DF} vs Ablation_Manifold_Avg	63,3225	1,755e-15	Có ý nghĩa

Ghi chú: Hàng tô đậm (tập Hotel, F_{DF} so sánh Ablation_Euclid_Stacking): cặp so sánh duy nhất không đạt ý nghĩa thống kê trong cả hai bảng (ở Restaurant, cặp này có khác biệt có ý nghĩa, $p = 2,366e-19$), khi cố định không gian phân cụm ở Euclidean, số quan sát chỉ một trong hai mô hình dự đoán đúng ở hai chiều gần cân bằng, nên tổng số dự đoán đúng của F_{DF} và bản Stacking không khác biệt có ý nghĩa ở tập Hotel; Stacking chủ yếu dịch điểm cân bằng Recall/Specificity (Bảng H.1) chứ không làm thay đổi đáng kể độ chính xác tổng thể.

Bảng H.5: Kiểm định Wilcoxon — Hotel

Cặp so sánh (Wilcoxon, 5 lần lặp)	V	p-value	Kết luận ($\alpha=0,05$)
F_{CF} vs F_{DF}	7	1,0000	Chưa đủ bằng chứng
F_{CF} vs F_{MF}	15	0,05906	Chưa đủ (biên, min-p n=5)
F_{DF} vs F_{MF}	15	0,05906	Chưa đủ (biên, min-p n=5)
F_{DF} vs Ablation_Euclid_Stacking	15	0,05906	Chưa đủ (biên, min-p n=5)
F_{MF} vs Ablation_Manifold_Avg	0	0,05906	Chưa đủ (biên, min-p n=5)
Ablation_Euclid_Stacking vs F_{MF}	15	0,05906	Chưa đủ (biên, min-p n=5)
F_{DF} vs Ablation_Manifold_Avg	15	0,05906	Chưa đủ (biên, min-p n=5)

Bảng H.6: Kiểm định Wilcoxon — Restaurant

Cặp so sánh (Wilcoxon, 5 lần lặp)	V	p-value	Kết luận ($\alpha=0,05$)
F_{CF} vs F_{DF}	5	0,5896	Chưa đủ bằng chứng
F_{CF} vs F_{MF}	15	0,05906	Chưa đủ (biên, min-p n=5)
F_{DF} vs F_{MF}	15	0,05906	Chưa đủ (biên, min-p n=5)
F_{DF} vs Ablation_Euclid_Stacking	15	0,05906	Chưa đủ (biên, min-p n=5)

Cặp so sánh (Wilcoxon, 5 lần lặp)	V	p-value	Kết luận ($\alpha=0,05$)
F_{MF} vs Ablation_Manifold_Avg	0	0,05906	Chưa đủ (biên, min-p n=5)
Ablation_Euclid_Stacking vs F_{MF}	15	0,05906	Chưa đủ (biên, min-p n=5)
F_{DF} vs Ablation_Manifold_Avg	15	0,05906	Chưa đủ (biên, min-p n=5)

Phụ lục I

THÍ NGHIỆM LOẠI TRỪ ĐẶC TRƯNG (h/e), ƯỚC LƯỢNG CHIỀU NỘI TẠI CỦA KHÔNG GIAN ĐA TẬP VÀ SO SÁNH CÁCH MÃ HÓA ĐẶC TRƯNG

Phụ lục này làm rõ hai nội dung: (1) mức độ đóng góp riêng của đặc trưng Holder (h - định danh người viết đánh giá) và Entity (e - định danh doanh nghiệp) trong bộ dữ liệu Phụ lục C, ngoài Aspect và Time; và (2) ước lượng trực tiếp chiều nội tại (intrinsic dimension) của không gian đặc trưng dùng để xây ma trận quan hệ đa tập W ở Chương 4, thay vì chỉ suy luận gián tiếp qua tham số DBSCAN như trước; ngoài ra, mục I.3 đánh giá cách mã hóa các đặc trưng định danh (so sánh mã hóa số thứ tự, target encoding và one-hot).

I.1. Ablation đặc trưng h/e

Số liệu chi tiết tại Bảng I.1, Bảng I.2 (trình bày ở cuối Phụ lục I).

So với biến thể đầy đủ, loại bỏ Entity (e) làm giảm Accuracy nhiều hơn loại bỏ Holder (h) trên Hotel ($-0,0652$ so với $-0,0288$ điểm), trong khi trên Restaurant, loại Holder làm giảm Accuracy nhiều hơn ($-0,0711$ so với $-0,0281$ điểm) nhưng loại Entity lại làm giảm Balanced Accuracy nhiều hơn ($-0,0369$ so với $-0,0283$ điểm). Nói cách khác, mức độ đóng góp tương đối của h và e là phụ thuộc bộ dữ liệu chứ không có một thành phần nào luôn chiếm ưu thế tuyệt đối ở cả hai tập; cả hai đều mang tín hiệu dự đoán bổ sung đáng kể ngoài Aspect và Time. Khi loại bỏ đồng thời cả h và e, Balanced Accuracy sụt về xấp xỉ mức ngẫu nhiên/đa số ($0,5119$ ở tập Hotel; $0,5006$ ở tập Restaurant, tức gần như $0,5$ tuyệt đối) và MCC gần bằng 0 ($0,0260$ và $0,0010$), khẳng định Aspect và Time đơn thuần gần như không đủ để phân biệt hai lớp quan điểm, cả Holder lẫn Entity đều là thành phần đặc trưng cần thiết, không chỉ mang tính “nhiều định danh” như có thể đặt vấn đề.

I.2. Ước lượng chiều nội tại của không gian đa tập

Không gian đặc trưng 4 chiều (Entity_te, Aspect_te, Holder_te, Time_id đã chuẩn hóa), đúng không gian dùng để xây ma trận quan hệ đa tập W ở Chương 4 được ước lượng chiều nội tại bằng hai phương pháp độc lập: MLE (Maximum Likelihood, Levina–Bickel [77]) và DANCo [209], trên 5 lần lấy mẫu con ngẫu nhiên (5.000 quan

sát/lần, $k = 10$ láng giềng gần nhất). Một xấp xỉ cục bộ của CA-PCA (Local-PCA: với mỗi điểm, PCA trên k láng giềng gần nhất, đếm số thành phần cần giữ để đạt $\geq 90\%$ phương sai, sau đó lấy trung vị/trung bình) được bổ sung tham khảo do CA-PCA nguyên bản không có sẵn trong các gói R hiện hành.

Số liệu chi tiết tại Bảng I.3, Bảng I.4 (trình bày ở cuối Phụ lục I).

Phương pháp MLE cho d' nhỏ hơn số chiều quan sát $D = 4$ một cách nhất quán qua toàn bộ 5 lần lấy mẫu độc lập ở cả hai tập dữ liệu; DANCo cho $d' = 3$ (tập Hotel) và $d' = 4 = D$ (tập Restaurant): trung bình $d' \approx 2,78$ (tập Hotel) và $d' \approx 3,66$ (tập Restaurant) khi gộp MLE và DANCo. Kết quả này là bằng chứng thực nghiệm trực tiếp (không chỉ suy luận gián tiếp qua tham số DBSCAN) ủng hộ giả thuyết đa tập trên tập Hotel, nơi cả hai phương pháp cho $d' < D$; trên tập Restaurant, phương pháp MLE cho $d' < D$ nhưng DANCo cho $d' = D = 4$, nên bằng chứng ở tập này chưa nhất quán giữa hai phương pháp. Do không gian quan sát chỉ có $D = 4$ chiều, điều kiện $d' < 5$ nêu ở Chương 4 luôn được thỏa; điều có ý nghĩa là d' nhỏ hơn D . Ước lượng Local-PCA (2,62 và 3,01) thấp hơn DANCo (và thấp hơn MLE trên tập Restaurant; trên tập Hotel xấp xỉ MLE là 2,56), phù hợp với đặc điểm đã biết của các phương pháp local-PCA/CA-PCA là có xu hướng đánh giá thấp chiều nội tại khi mật độ điểm không đều hoặc khi có nhiều điểm gần trùng lặp (do một số đặc trưng được mã hóa mục tiêu - target encoding, chỉ nhận rất ít giá trị phân biệt, ví dụ Aspect_te), nên được xem là tham khảo bổ sung, còn hai phương pháp MLE và DANCo là hai bằng chứng chính được khuyến nghị trích dẫn.

I.3. So sánh các cách mã hóa đặc trưng định danh

Nội dung này đánh giá cách mã hóa các đặc trưng định danh của bộ năm. Ba cách mã hóa được so sánh trên cùng mô hình rừng phân loại F_{CF} (randomForest 300 cây, Bootstrap cân bằng lớp qua phương pháp sampsize/strata, ngưỡng quyết định chọn trên tập xác thực theo Balanced Accuracy) và lặp với ba hạt giống 2026, 2027, 2028: (1) mã hóa số thứ tự (ordinal/label encoding), là cách dùng trong các bài báo đã công bố và công thức (3.2); (2) mã hóa mục tiêu- target encoding làm trơn Laplace ($m = 10$), chỉ ước lượng trên tập huấn luyện, là cách dùng tại Phụ lục G–J; (3) mã hóa one-hot, giữ đầy đủ với đặc trưng ít giá trị, còn với đặc trưng nhiều giá trị chỉ giữ 30 giá trị phổ biến nhất trên tập huấn luyện và gộp phần còn lại vào một cột “khác”. Giới

hạn này là cần thiết vì trên yelpHotelData, tập huấn luyện có 39.925 thực thể, 3 khía cạnh, 2.585 người đánh giá và 3.969 giá trị thời gian khác nhau, one-hot đầy đủ sẽ tạo hơn 46.000 cột. Dữ liệu gồm tập yelpHotelData (lấy mẫu $N = 112.275$ như Phụ lục G, chia 60/20/20 thành 67.365/22.455/22.455 bộ năm) và VLSP-2018 Restaurant, Hotel (bộ năm từ dữ liệu gốc, các tập Train/Dev/Test chính thức như Phụ lục F).

Số liệu chi tiết tại Bảng J.15 (trình bày ở cuối Phụ lục I).

Trên yelpHotelData, nơi các đặc trưng định danh có lực lượng rất lớn, target encoding cho kết quả tốt nhất ở cả bốn độ đo (Balanced Accuracy 0,6303, MCC 0,2517) và có thời gian huấn luyện thấp nhất; mã hóa số thứ tự cho kết quả thấp hơn (Balanced Accuracy 0,5765) do gán một thứ tự nhân tạo cho các giá trị định danh; one-hot rút gọn cho kết quả thấp nhất (Balanced Accuracy 0,5184) vì phần lớn thực thể và người đánh giá bị gộp vào cột “khác” nên mất gần hết thông tin, đồng thời thời gian huấn luyện tăng khoảng 23–41 lần do véc-tơ đặc trưng có 96 cột. Trên VLSP-2018, nơi các đặc trưng có ít giá trị (Bảng F.1, F.2), ba cách mã hóa cho kết quả gần như nhau (chênh lệch Balanced Accuracy khoảng 0,03 trên tập Restaurant và dưới 0,001 trên tập Hotel); mã hóa one-hot kém ổn định trên tập Restaurant (độ lệch chuẩn Accuracy 0,1628) do ngưỡng chọn trên tập Dev dao động giữa các lần lặp. Kết quả cho thấy: (i) với dữ liệu có đặc trưng định danh lực lượng lớn như tập Yelp thì mã hóa target encoding là lựa chọn phù hợp và đã được áp dụng cho các thực nghiệm bổ sung tại Phụ lục G–J; (ii) số liệu của các bảng chính tại Chương 3, Chương 4 dùng mã hóa số thứ tự theo các bài báo đã công bố được giữ nguyên và cách mã hóa này là một hạn chế đã được nêu tại §5.1.

I.4. Thảo luận chung

Các thực nghiệm của Phụ lục I bổ khuyết trực tiếp hai câu hỏi còn để ngỏ về vai trò của h , e và về giả thuyết đa tạp (cùng với Phụ lục G, Phụ lục H, Phụ lục J). Kết quả không làm thay đổi kiến trúc hay số liệu chính thức đã báo cáo trong luận án, mà cung cấp thêm bằng chứng thực nghiệm định lượng cho hai giả định đã được luận án nêu nhưng chưa kiểm chứng trực tiếp bằng thực nghiệm mới: vai trò của định danh Holder/Entity trong Phụ lục C và giả thuyết đa tạp làm nền tảng cho Chương 4.

I.5. Kết luận

Kết quả phụ lục cho thấy: (i) cả hai đặc trưng định danh Holder và Entity trong Phụ lục C đều đóng góp tín hiệu dự đoán thực chất, mức độ đóng góp tương đối phụ thuộc bộ dữ liệu và (ii) chiều nội tại của không gian đặc trưng dùng xây dựng đa tạp ở Chương 4 nhỏ hơn số chiều quan sát, được xác nhận bằng phương pháp MLE ở cả hai tập và bằng DANCo trên tập Hotel, củng cố tính hợp lý của giả thuyết đa tạp mà mô hình F_{MF} dựa vào; (iii) với đặc trưng định danh có lực lượng lớn, mã hóa target encoding cho kết quả tốt hơn mã hóa số thứ tự và one-hot, còn với đặc trưng ít giá trị (VLSP-2018) ba cách mã hóa cho kết quả tương đương.

BẢNG SỐ LIỆU CỦA PHỤ LỤC I

Bảng I.1: Ablation đặc trưng — Hotel (mean $\pm$ SD, 3 lần lặp, một rừng ngẫu nhiên duy nhất, BalancedBoot)

Biến thể đặc trưng	Số DT	Accuracy	Precision	Recall	Specificity	F1	F1-macro	BalancedAcc.	MCC
Đầy đủ (e, a, h, t)	4	0,6449 $\pm$ 0,0067	0,7493 $\pm$ 0,0101	0,6643 $\pm$ 0,0241	0,6113 $\pm$ 0,0265	0,7040 $\pm$ 0,0106	0,6300 $\pm$ 0,0041	0,6378 $\pm$ 0,0038	0,2676 $\pm$ 0,0069
Loại bỏ Holder (h)	3	0,6161 $\pm$ 0,0159	0,7128 $\pm$ 0,0065	0,6644 $\pm$ 0,0577	0,5308 $\pm$ 0,0592	0,6867 $\pm$ 0,0294	0,5937 $\pm$ 0,0054	0,5976 $\pm$ 0,0031	0,1922 $\pm$ 0,0068
Loại bỏ Entity (e)	3	0,5797 $\pm$ 0,0165	0,7163 $\pm$ 0,0073	0,5623 $\pm$ 0,0566	0,6093 $\pm$ 0,0577	0,6288 $\pm$ 0,0328	0,5708 $\pm$ 0,0099	0,5858 $\pm$ 0,0067	0,1658 $\pm$ 0,0131
Chỉ Aspect + Time	2	0,4524 $\pm$ 0,0235	0,6570 $\pm$ 0,0080	0,2935 $\pm$ 0,0853	0,7303 $\pm$ 0,0851	0,4002 $\pm$ 0,0841	0,4459 $\pm$ 0,0333	0,5119 $\pm$ 0,0021	0,0260 $\pm$ 0,0055

Bảng I.2: Ablation đặc trưng — Restaurant (mean $\pm$ SD, 3 lần lặp)

Biến thể đặc trưng	Số DT	Accuracy	Precision	Recall	Specificity	F1	F1-macro	BalancedAcc.	MCC
Đầy đủ (e, a, h, t)	4	0,6202 $\pm$ 0,0036	0,7354 $\pm$ 0,0048	0,6533 $\pm$ 0,0130	0,5580 $\pm$ 0,0134	0,6919 $\pm$ 0,0052	0,5984 $\pm$ 0,0021	0,6057 $\pm$ 0,0007	0,2039 $\pm$ 0,0016
Loại bỏ Holder (h)	3	0,5491 $\pm$ 0,0143	0,7343 $\pm$ 0,0063	0,4852 $\pm$ 0,0365	0,6695 $\pm$ 0,0313	0,5837 $\pm$ 0,0252	0,5456 $\pm$ 0,0123	0,5774 $\pm$ 0,0049	0,1490 $\pm$ 0,0087
Loại bỏ Entity (e)	3	0,5921 $\pm$ 0,0050	0,7052 $\pm$ 0,0064	0,6452 $\pm$ 0,0223	0,4925 $\pm$ 0,0264	0,6736 $\pm$ 0,0092	0,5647 $\pm$ 0,0016	0,5688 $\pm$ 0,0024	0,1337 $\pm$ 0,0035
Chỉ Aspect + Time	2	0,4509 $\pm$ 0,0260	0,6529 $\pm$ 0,0039	0,3387 $\pm$ 0,0725	0,6626 $\pm$ 0,0646	0,4431 $\pm$ 0,0645	0,4492 $\pm$ 0,0271	0,5006 $\pm$ 0,0010	0,0010 $\pm$ 0,0096

Ghi chú: e = Entity_te, a = Aspect_te, h = Holder_te, t = Time_id (đã mã hóa/chuẩn hóa như Chương 2). Biến thể “Chi Aspect + Time” loại bỏ đồng thời cả h và e, dùng làm mốc dưới để đánh giá mức độ mô hình phụ thuộc vào định danh thực thể/người viết.

Bảng I.3: Ước lượng chiều nội tại d' — Hotel (D = 4 chiều quan sát)

Lần lặp subsample	MLE	DANCo	Local-PCA (trung vị)	Local-PCA (trung bình)
1	2,5771	3	3	2,6230
2	2,5547	3	3	2,6210
3	2,5766	3	3	2,6246
4	2,5411	3	3	2,5980
5	2,5697	3	3	2,6172
Trung bình	2,5638	3,0000	3	2,6168

Bảng I.4: Ước lượng chiều nội tại d' — Restaurant (D = 4 chiều quan sát)

Lần lặp subsample	MLE	DANCo	Local-PCA (trung vị)	Local-PCA (trung bình)
1	3,3411	4	3	3,0182
2	3,2904	4	3	2,9934
3	3,3295	4	3	3,0102
4	3,3166	4	3	3,0028
5	3,3229	4	3	3,0056
Trung bình	3,3201	4,0000	3	3,0060

Bảng I.5: So sánh ba cách mã hóa đặc trưng định danh với F_{CF} trên tập kiểm tra (trung bình $\pm$ độ lệch chuẩn qua 3 hạt giống)

Tập dữ liệu	Cách mã hóa	Số cột	Accuracy	F1-macro	Balanced Accuracy	MCC	Thời gian (s)
yelpHotelData	Số thứ tự	4	$0,5500 \pm 0,0159$	$0,5479 \pm 0,0139$	$0,5765 \pm 0,0018$	$0,1501 \pm 0,0008$	19,5
yelpHotelData	Target encoding	4	$0,6276 \pm 0,0048$	$0,6181 \pm 0,0030$	$0,6303 \pm 0,0004$	$0,2517 \pm 0,0012$	11,0
yelpHotelData	One-hot (rút gọn)	96	$0,4128 \pm 0,0005$	$0,3701 \pm 0,0009$	$0,5184 \pm 0,0002$	$0,0572 \pm 0,0006$	452,7
VLSP-2018 Restaurant	Số thứ tự	4	$0,6331 \pm 0,0033$	$0,4994 \pm 0,0002$	$0,6492 \pm 0,0069$	$0,1720 \pm 0,0074$	0,3
VLSP-2018 Restaurant	Target encoding	4	$0,6435 \pm 0,0026$	$0,5018 \pm 0,0014$	$0,6388 \pm 0,0013$	$0,1612 \pm 0,0015$	0,2
VLSP-2018 Restaurant	One-hot	28	$0,5035 \pm 0,1628$	$0,4187 \pm 0,0905$	$0,6181 \pm 0,0127$	$0,1409 \pm 0,0170$	0,6
VLSP-2018 Hotel	Số thứ tự	4	$0,6897 \pm 0,0072$	$0,6624 \pm 0,0054$	$0,7158 \pm 0,0020$	$0,3819 \pm 0,0044$	0,8
VLSP-2018 Hotel	Target encoding	4	$0,6879 \pm 0,0008$	$0,6613 \pm 0,0006$	$0,7158 \pm 0,0003$	$0,3815 \pm 0,0005$	0,8
VLSP-2018 Hotel	One-hot	48	$0,6897 \pm 0,0006$	$0,6627 \pm 0,0005$	$0,7166 \pm 0,0004$	$0,3832 \pm 0,0008$	4,0

Ghi chú: Thời gian là thời gian huấn luyện trung bình trên CPU (giây); “Số cột” là số chiều của véc-tơ đặc trưng sau mã hóa bốn thành phần e, a, h, t.

Phụ lục J

KẾT QUẢ THỰC NGHIỆM BỔ SUNG

Phụ lục này tập hợp các thực nghiệm bổ sung nhằm đối chiếu với số liệu đã công bố, gồm ba phần: (1) mục J.1–J.6 thực nghiệm Kịch bản 1 của mô hình rừng mật độ với cơ chế cân bằng lớp và tối ưu ngưỡng quyết định; (2) mục J.7–J.10 thực nghiệm ba mô hình rừng trên cỡ mẫu đầy đủ và so sánh thống nhất 10 mô hình theo cùng một quy trình; (3) mục J.11–J.12 phân tích ảnh hưởng của cách chia dữ liệu đến chỉ số đánh giá và lặp lại phép so sánh thống nhất với 5 hạt giống $\times$ 10.000 đánh giá trên yelpHotelData và yelpResData, kèm bản cải tiến của mô hình rừng đề xuất.

Mục J.1–J.6 trình bày một thực nghiệm bổ sung cho Kịch bản 1 (mục 3.7.3.1 của luận án) trên hai tập dữ liệu Hotel và Restaurant, nhằm kiểm tra độ vững của phương pháp trước hiện tượng mất cân bằng lớp và cải tiến các mô hình đề xuất. Toàn bộ nội dung, phương pháp luận và số liệu Kịch bản 1 đã trình bày ở Chương 3, phụ lục bổ sung với hai cải tiến kỹ thuật nhằm kiểm soát mất cân bằng lớp.

J.1. Mục đích và bối cảnh

Kịch bản 1 sử dụng thuật toán DBSCAN để phân cụm tập Train không nhãn thành K cụm, huấn luyện K rừng ngẫu nhiên giám sát trên K cụm này, sau đó kết hợp kết quả dự đoán của K rừng theo phương pháp trung bình mật độ xác suất (density averaging) như mô tả tại mục 3.7.3.1 của luận án và hai cải tiến bổ sung là:

(1) Bootstrap cân bằng lớp trong từng rừng con, thông qua tham số `samplesize/strata` của hàm `randomForest()`, mỗi cây được huấn luyện trên mẫu bootstrap có số lượng bằng nhau giữa hai lớp.

(2) Tối ưu ngưỡng quyết định trên một tập Validation độc lập (không dùng để huấn luyện hay đánh giá cuối), bằng tìm kiếm lưới trong khoảng 0,05–0,95 để tối đa hóa $\text{Balanced Accuracy} = 0,5 \times (\text{Recall} + \text{Specificity})$, thay vì áp ngưỡng cố định 0,5.

J.2. Thiết lập thực nghiệm

Cỡ mẫu: $N = 112.275$ bộ năm quan điểm cho cả hai tập Hotel và Restaurant, giống với cỡ mẫu đã sử dụng trong luận án, nhằm đảm bảo khả năng đối chiếu. Tách dữ liệu theo tỷ lệ 60% Train / 20% Validation / 20% Test thay vì 80/20 nhằm có một tập Validation độc lập phục vụ việc tối ưu ngưỡng mà không rò rỉ thông tin từ tập

Test. Các đặc trưng Entity/Aspect/Holder được mã hóa bằng target encoding (làm tròn Laplace, $m = 10$) và tham số mã hóa được ước lượng trên tập Train và áp dụng lại cho Validation/Test.

Cơ chế kết hợp K rừng: trung bình xác suất dự đoán lớp “positive” của K rừng (density averaging), đúng theo phương pháp chính thức của Kịch bản 1 tại mục 3.7.3.1 của luận án, không thay đổi so với công trình đã công bố.

J.3. Kết quả trên tập Test

Số liệu chi tiết tại Bảng J.1, Bảng J.2, Bảng J.3(trình bày ở cuối Phụ lục J).

J.4. Đối chiếu với Bài báo số 4, Bài báo số 5 và Bài báo số 6

Số liệu chi tiết tại Bảng J.4, Bảng J.5, Bảng J.6(trình bày ở cuối Phụ lục J).

J.5. Thảo luận về khác biệt số liệu

Điểm đáng chú ý nhất trong đối chiếu trên là chỉ số F1 của mục J.1–J.6 trên tập Hotel (0,7223) rất gần với F1 của Bài báo số 4 (0,72), trong khi Accuracy chênh lệch rất lớn (0,6558 so với 0,92). Vì F1 phản ánh trực tiếp năng lực phân loại trên lớp “positive” (điều hòa giữa Precision và Recall), sự tương đồng này cho thấy năng lực phân loại thực sự của Kịch bản 1 là tương đương giữa hai lần thực nghiệm; phần chênh lệch lớn về Accuracy chủ yếu bắt nguồn từ cách xử lý mất cân bằng lớp và lựa chọn ngưỡng quyết định, chứ không phải từ một mô hình yếu hơn.

Ba yếu tố kỹ thuật giải thích phần lớn khác biệt còn lại giữa các công bố và mục J.1–J.6 như sau:

Thứ nhất, cơ chế kết hợp K rừng khác nhau (trung bình mật độ xác suất trong luận án và mục J.1–J.6, majority vote trong Bài báo số 4, stacking trong Bài báo số 6) tạo ra phân phối xác suất dự đoán khác nhau, dẫn đến độ nhạy khác nhau khi áp một ngưỡng quyết định cố định.

Thứ hai, Bài báo số 4, Bài báo số 5 và Bài báo số 6 đều không áp dụng cơ chế cân bằng bootstrap và không báo cáo Balanced Accuracy hay Specificity, nên không thể loại trừ khả năng Accuracy cao của các công bố này phần nào phản ánh xu hướng mô hình thiên về lớp đa số. Ngoài ra, Bài báo số 4 báo cáo Recall 0,60 (Hotel) và 0,59 (Restaurant) cùng Accuracy 0,91–0,92; với lớp thiểu số chiếm khoảng 35–37% mẫu, Recall khoảng 0,6 ở bất kỳ lớp nào chỉ cho phép Accuracy tối đa khoảng 0,85, nên hai chỉ số này khó đồng thời đúng trên cùng một tập kiểm tra và cần được đối

chiều thận trọng (xem mục J.11 về ảnh hưởng của cách chia dữ liệu). Hiện tượng thiên lệch tương tự cũng từng được ghi nhận với chính Kịch bản 1 trước khi áp dụng hai cải tiến của Phụ lục này (Balanced Accuracy ban đầu chỉ đạt 0,4978–0,6081 khi dùng bootstrap mặc định và ngưỡng 0,5), cũng như với mô hình cơ sở SVM tuyến tính (và F_{DF}) tại ngưỡng 0,5 trên tập VLSP-2018 Restaurant tại Phụ lục F (Bảng F.4, $F1\text{-neg} = 0,0000$ khi tỉ lệ mất cân bằng $\rho \approx 10,5$).

Thứ ba, tỷ lệ tách dữ liệu khác nhau (80/20 ở Bài báo số 4 so với 60/20/20 ở mục J.1–J.6) làm giảm tỷ lệ dữ liệu huấn luyện từ 80% xuống 60% tổng số mẫu (tức khoảng một phần tư lượng dữ liệu huấn luyện) để dành cho tập Validation phục vụ tối ưu ngưỡng; đây có thể là một phần nguyên nhân phụ, nhưng không phải nguyên nhân chính vì F1 giữa hai lần thực nghiệm gần như không đổi.

J.6. Kết luận và khuyến nghị

Nội dung Kịch bản 1 trình bày tại Chương 3 (mục 3.7.3.1) của luận án, cũng như số liệu đã công bố tại Bài báo số 4 và Bài báo số 5, được giữ nguyên như đã công bố; Phụ lục này không thay thế mà chỉ bổ sung một phép đối chứng có kiểm soát mất cân bằng lớp.

Kết quả đối chứng cho thấy: khi kiểm soát mất cân bằng lớp bằng bootstrap cân bằng và tối ưu ngưỡng trên tập Validation độc lập, Kịch bản 1 vẫn giữ được năng lực phân loại tương đương với số liệu đã công bố ($F1 \approx$ Bài báo số 4), đồng thời đạt được chỉ số Balanced Accuracy trung thực hơn (0,6389 trên Hotel, 0,6084 trên Restaurant), phản ánh sát hơn hiệu năng thực tế của mô hình trên cả hai lớp positive và negative.

Các bảng kết quả chính của luận án (Bảng 3.5–3.9, Bảng 4.2–4.4) trình bày số liệu đã công bố tại Bài báo 5 và Bài báo 6, được giữ nguyên để bảo đảm tính truy xuất nguồn. Trong quá trình hoàn thiện luận án, nghiên cứu sinh đã chạy lại ba mô hình rừng đề xuất và các mô hình tham chiếu theo một quy trình thống nhất (cùng dữ liệu, cùng cách chia, cùng cơ chế chọn ngưỡng và cùng bộ độ đo). Mục J.7–J.10 tổng hợp các kết quả chạy lại đó để người đọc đối chiếu.

J.7. Kết quả chạy lại trên cỡ mẫu đầy đủ ($N = 112.275$, Phụ lục H và mục J.1–J.6)

Số liệu chi tiết tại Bảng J.7 (trình bày ở cuối Phụ lục J).

J.8. So sánh thống nhất 10 mô hình (yelpHotelData, 3.000 đánh giá, 1 lần chia)

Thực nghiệm dùng 3.000 đánh giá lấy ngẫu nhiên (hạt giống 2026), chia theo đánh giá thành 60/20/20; tập kiểm tra gồm 600 đánh giá (63,7% positive). Ngưỡng quyết định của mọi mô hình được chọn trên tập xác thực theo Balanced Accuracy (BA). Cách 1: mỗi mô hình dùng đầu vào tự nhiên của nó (mô hình rừng và học máy dùng bộ năm; LSTM, BiLSTM-Attention, BERT, RoBERTa dùng văn bản đánh giá gốc). Cách 2: mọi mô hình chỉ dùng bộ năm (e, a, h, t); các mô hình văn bản đọc bộ năm dưới dạng chuỗi. Bảng J.8 được giữ nguyên để đối chiếu; kết quả lặp lại với 5 hạt giống và 10.000 đánh giá được trình bày tại mục J.12 (Bảng J.11, Bảng J.12).

Số liệu chi tiết tại Bảng J.8, Bảng J.9 (trình bày ở cuối Phụ lục J).

J.9. Các yếu tố tạo nên khác biệt so với số liệu công bố

- Cơ chế kết hợp K rừng khác nhau giữa các công bố (biểu quyết đa số ở Bài báo 4, Stacking ở Bài báo 6) và các lần chạy lại (trung bình xác suất cho rừng mật độ, Stacking cho rừng đa tập) - xem Bảng J.4.

- Xử lý mất cân bằng lớp và ngưỡng quyết định: các lần chạy lại dùng Bootstrap cân bằng lớp và chọn ngưỡng theo Balanced Accuracy trên tập xác thực; các công bố dùng ngưỡng 0,5 và không cân bằng lớp. Khi dữ liệu có khoảng 63–65% mẫu positive, Accuracy chịu ảnh hưởng mạnh của ngưỡng.

- Cách chia dữ liệu: các công bố chia ngẫu nhiên theo dòng bộ năm (80/20 hoặc 60/20/20); thực nghiệm J.8 chia theo đánh giá gốc để các dòng của cùng một đánh giá không xuất hiện đồng thời ở tập huấn luyện và tập kiểm tra.

- Quy tắc nhãn: các lần chạy lại gộp mức 3★ vào negative (theo mã nguồn, Bảng 3.3); mô tả nhãn trong Bài báo 4 và Bài báo 6 khác nhau (xem ghi chú dưới Bảng đối chiếu công trình – chương – nội dung sử dụng).

- Cỡ mẫu và lấy mẫu con: rừng đa tập cần lấy mẫu con 5.000 dòng để xây ma trận W do giới hạn bộ nhớ (Phụ lục G).

J.10. Nhận xét

Kết quả chạy lại theo quy trình thống nhất cho các giá trị thấp hơn số liệu đã công bố. Khi mọi mô hình chỉ sử dụng thông tin của bộ năm (Cách 2), Balanced Accuracy của cả mô hình rừng, mô hình học máy, mô hình học sâu và mô hình Transformer đều nằm trong khoảng 0,49–0,56, cho thấy lượng thông tin về cực tính

cảm xúc chứa trong bốn thành phần (e, a, h, t) là hạn chế; khi được sử dụng văn bản gốc (Cách 1), BERT và RoBERTa đạt 0,719 và 0,783. Điều này phù hợp với nhận định tại §5.1 rằng hiệu năng của các mô hình rừng phụ thuộc vào chất lượng và nội dung của bước trích xuất bộ năm, và củng cố hướng phát triển tại §5.2: kết hợp biểu diễn ngữ nghĩa của văn bản với cấu trúc rừng để giữ khả năng diễn giải và chi phí tính toán thấp của các mô hình đề xuất. Mục J.11 (Bảng J.10) làm rõ nguyên nhân chênh lệch so với số liệu đã công bố; mục J.12 (Bảng J.11–J.17) củng cố nhận xét trên bằng nhiều hạt giống với cỡ mẫu lớn hơn, theo cả Cách 1 (Bảng J.11) và Cách 2 (Bảng J.14 - Bảng J.17).

J.11. Ảnh hưởng của cách chia dữ liệu đến chỉ số đánh giá

Để làm rõ nguyên nhân chênh lệch giữa số liệu đã công bố (Bảng 3.7–3.9, Bảng 4.4) và kết quả chạy lại tại mục J.7–J.8, nghiên cứu sinh thực hiện một kiểm tra có kiểm soát trên yelpHotelData. Trên tập dữ liệu này, mỗi đánh giá được tách thành ba dòng bộ năm có cùng thực thể e, người đánh giá h, thời gian t và cùng nhãn s, chỉ khác khía cạnh a (ba khía cạnh AcceptsCreditCards, PriceRange, WiFi lặp lại ở mọi đánh giá). Khi chia ngẫu nhiên theo dòng bộ năm như các bài báo, một phần dòng của cùng một đánh giá có thể đồng thời nằm ở tập huấn luyện và tập kiểm tra; khi đó mô hình có thể dự đoán đúng nhờ ghi nhớ bộ (e, h, t) đã thấy thay vì khái quát hóa. Ngược lại, khi chia theo đánh giá gốc, mọi dòng của một đánh giá cùng nằm trong một tập, nên tập kiểm tra chỉ gồm các đánh giá mô hình chưa từng thấy.

Cấu hình kiểm tra: Random Forest 300 cây, mã hóa số thứ tự như các bài báo, ngưỡng 0,5, chia 80/20, cỡ mẫu 112.275 bộ năm như luận án, hạt giống 2026; chỉ thay đổi cách lấy mẫu và cách chia dữ liệu. Kết quả được trình bày tại Bảng J.10.

Số liệu chi tiết tại Bảng J.10 (trình bày ở cuối Phụ lục J).

Với cùng dữ liệu và cùng mô hình, Accuracy thay đổi từ 0,611 đến 0,946 chỉ do cách chia dữ liệu (Bảng J.10): chỉ số càng cao khi tỷ lệ dòng kiểm tra có dòng cùng đánh giá trong tập huấn luyện càng lớn (19,1% và 96,0%), trong khi chia theo đánh giá gốc cho kết quả ổn định quanh 0,61 bất kể cách lấy mẫu. Như vậy, các chỉ số đã công bố (Accuracy 0,75–0,82) cần được hiểu là kết quả theo quy trình chia theo dòng của các bài báo; quy trình chia theo đánh giá gốc dùng tại mục J.8 và J.12 cho chỉ số

thấp hơn nhưng phản ánh sát hơn khả năng dự đoán cho đánh giá mới, và là cách chia được khuyến nghị cho các nghiên cứu tiếp theo trên dữ liệu bộ năm (§5.1).

J.12. So sánh thống nhất mở rộng: 5 hạt giống $\times$ 10.000 đánh giá

Để có kết luận chắc chắn hơn so với thực nghiệm một lần chia trên 3.000 đánh giá tại mục J.8 (Bảng J.8, giữ nguyên để đối chiếu), phép so sánh thống nhất được lặp lại với cỡ mẫu 10.000 đánh giá yelpHotelData và 5 hạt giống (2026–2030). Mọi điều kiện khác giữ nguyên như mục J.8: chia theo đánh giá gốc 60/20/20, ngưỡng quyết định chọn trên tập xác thực theo Balanced Accuracy, cấu hình mô hình như Bảng J.9. Các mô hình dùng bộ năm được huấn luyện bằng R; LSTM, BiLSTM-Attention bằng gói torch cho R; BERT, RoBERTa được tinh chỉnh trên GPU NVIDIA L4 (Google Colab), mỗi mô hình khoảng 2 phút cho một hạt giống. Kết quả trung bình $\pm$ độ lệch chuẩn qua 5 hạt giống được trình bày tại Bảng J.11; chênh lệch ghép cặp theo hạt giống so với Random Forest tại Bảng J.12, Bảng J.11 là kết quả theo Cách 1 (các mô hình văn bản đọc văn bản gốc); kết quả theo Cách 2 (mọi mô hình chỉ dùng bộ năm) được trình bày tại Bảng J.14.

Số liệu chi tiết Bảng J.11, Bảng J.12 (trình bày ở cuối Phụ lục J).

Bảng J.11 cho thấy: (i) khi dùng văn bản gốc, RoBERTa (Balanced Accuracy 0,807) và BERT (0,755) cao hơn rõ rệt các mô hình chỉ dùng bộ năm; (ii) trong nhóm dùng bộ năm, các mô hình đạt 0,534–0,568; F_{DF} (0,561) và F_{CF} (0,557) cao hơn Random Forest (0,554) ở 4/5 hạt giống, nhưng chênh lệch chưa có ý nghĩa thống kê (Bảng J.12); F_{MF} thấp hơn Random Forest có ý nghĩa ($-0,020$), do bước phân cụm đa tập và các rừng thành phần chỉ học trên mẫu con 5.000 dòng. Độ lệch chuẩn nhỏ (0,004–0,013) cho thấy kết luận ổn định giữa các lần lặp, khác với Bảng J.8 (3.000 đánh giá, một lần chia). Ở hạt giống 2030, BERT không hội tụ (tồn thất đứng ở khoảng 0,658 qua cả ba epoch, mô hình dự đoán toàn bộ là positive), là hiện tượng tinh chỉnh không ổn định đã được ghi nhận với BERT. LSTM chỉ đạt 0,518 do cài đặt dùng trạng thái ẩn cuối trong khi chuỗi được đệm ở cuối, nên kết quả này không đại diện cho năng lực của kiến trúc LSTM; BiLSTM-Attention không chịu ảnh hưởng này nhờ cơ chế chú ý trên toàn chuỗi.

Song song, nghiên cứu sinh thử nghiệm ba điều chỉnh cho các mô hình rừng đề xuất trên cùng 5 hạt giống $\times$ 10.000 đánh giá, chỉ dùng bộ năm (Bảng J.13): (1)

target encoding ngoài fold, đặc trưng tần suất của thực thể và người đánh giá, cùng kích thước nút lá tối thiểu lớn hơn để tránh học thuộc các giá trị mã hóa hiếm (F_{CF+}); (2) định tuyến mỗi mẫu tới các rừng của cụm lân cận theo mật độ k láng giềng, kết hợp với một rừng toàn cục (F_{DF+}); (3) mở rộng nhãn cụm đa tập từ 5.000 điểm mốc cho toàn bộ tập huấn luyện bằng độ gần đa tập, để các rừng thành phần học trên đủ dữ liệu (F_{MF+}). Khi mọi mô hình dùng cùng bộ đặc trưng mới (nhóm B), ba bản điều chỉnh cao hơn Random Forest khoảng 0,011–0,013 Balanced Accuracy (F_{CF+} có ý nghĩa thống kê) và F_{MF+} thu hẹp khoảng cách của F_{MF} (0,547 so với 0,523). Ở cỡ mẫu bằng luận án (37.425 đánh giá, tương đương 112.275 bộ năm, một hạt giống), các bản điều chỉnh đạt Balanced Accuracy 0,585–0,586, cao hơn Random Forest (0,578) và tương đương Gradient Boosting (0,586), Logistic Regression (0,590). Đây là kết quả thăm dò, được ghi nhận như hướng phát triển của các mô hình đề xuất.

Số liệu chi tiết tại Bảng J.13 (trình bày ở cuối Phụ lục J).

Bảng J.14 lặp lại phép so sánh theo Cách 2: mọi mô hình chỉ dùng bộ năm (e, a, h, t), các mô hình văn bản đọc bộ năm dưới dạng chuỗi, với cùng 5 hạt giống $\times$ 10.000 đánh giá và cùng cách chia như Bảng J.11 Khi không còn văn bản gốc, ưu thế của các mô hình văn bản không còn: Balanced Accuracy của BiLSTM-Attention giảm từ 0,701 xuống 0,556, BERT từ 0,755 xuống 0,538, RoBERTa từ 0,807 xuống 0,523; tổn thất trên tập xác thực của BERT, RoBERTa chỉ dao động quanh 0,66 qua ba epoch, và ở hạt giống 2026, 2029 RoBERTa dự đoán toàn bộ là positive. Trên cùng thông tin bộ năm, F_{DF} (0,561) và F_{CF} (0,557) đạt Balanced Accuracy tương đương BiLSTM-Attention (0,556) và cao hơn BERT (0,538), RoBERTa (0,523), LSTM (0,504), trong khi thời gian huấn luyện chỉ vài giây so với khoảng 1,5–6 phút của các mô hình học sâu và Transformer; RoBERTa và LSTM thấp hơn Random Forest có ý nghĩa thống kê. Như vậy, chênh lệch giữa mô hình rừng và Transformer tại Bảng J.11 đến từ thông tin trong văn bản gốc chứ không phải từ năng lực mô hình hóa trên cùng đầu vào: khi cùng dùng bộ năm, các mô hình rừng đề xuất (trừ F_{MF}) đạt mức tương đương hoặc cao hơn các mô hình học sâu và Transformer với chi phí tính toán thấp hơn nhiều.

Số liệu chi tiết tại Bảng J.14 (trình bày ở cuối Phụ lục J).

Để kiểm tra tính nhất quán trên miền dữ liệu thứ hai, phép so sánh theo Cách 1 được lặp lại trên yelpResData với cùng quy trình: 10.000 đánh giá, 5 hạt giống (2026–2030), chia theo đánh giá 60/20/20, ngưỡng quyết định chọn trên tập xác thực theo Balanced Accuracy, cấu hình mô hình như Bảng J.9; kết quả tại Bảng J.15. Xu hướng giống yelpHotelData: các mô hình đọc văn bản gốc đạt Balanced Accuracy cao nhất (RoBERTa 0,823; BERT 0,769; BiLSTM-Attention 0,735), các mô hình dùng bộ năm đạt 0,514–0,553. Trong nhóm dùng bộ năm, F_{DF} (0,543) tương đương Random Forest (0,543), F_{CF} (0,541) thấp hơn không đáng kể (chênh lệch $-0,002$, khoảng tin cậy chứa 0); SVM và Gradient Boosting cao hơn Random Forest khoảng 0,008–0,010 có ý nghĩa thống kê, còn F_{MF} thấp hơn có ý nghĩa ($-0,029$). Trên yelpResData, BERT hội tụ ở cả 5 hạt giống và LSTM đạt 0,542, ngang các mô hình dùng bộ năm. Độ lệch chuẩn nhỏ (0,007–0,017) cho thấy kết luận ổn định giữa các lần lặp và giữa hai tập dữ liệu: khoảng cách giữa mô hình đọc văn bản và mô hình dùng bộ năm đến từ lượng thông tin của đầu vào, phù hợp với phân tích theo Cách 2 tại Bảng J.14.

Số liệu chi tiết tại Bảng J.14 (trình bày ở cuối Phụ lục J).

Bảng J.16 lặp lại Cách 2 trên yelpResData với cùng mẫu, cùng cách chia và cùng 5 hạt giống như Bảng J.15. Do mỗi đánh giá nhà hàng sinh khoảng 19 dòng bộ năm (tập huấn luyện khoảng 114.000 dòng), LSTM, BiLSTM-Attention, BERT, RoBERTa được huấn luyện trên 30.000 dòng bộ năm lấy ngẫu nhiên từ tập huấn luyện. Kết quả nhất quán với yelpHotelData (Bảng J.14): khi chỉ có bộ năm, BiLSTM-Attention (0,542) chỉ ngang Random Forest (0,543) và F_{DF} (0,543); BERT (0,510), RoBERTa (0,502) và LSTM (0,502) gần như không học được – tồn thất trên tập xác thực của BERT, RoBERTa dừng quanh 0,65 qua ba epoch và ở 4/5 hạt giống hai mô hình này dự đoán toàn bộ là positive – và đều thấp hơn Random Forest có ý nghĩa thống kê. Như vậy, trên cả hai tập dữ liệu, ưu thế của các mô hình học sâu và Transformer tại Bảng J.11, Bảng J.15 đến từ văn bản gốc; trên cùng đầu vào bộ năm, các mô hình rừng đề xuất F_{CF} , F_{DF} đạt mức tương đương hoặc cao hơn; thời gian huấn luyện của F_{CF} (11,9 giây) thấp hơn nhiều, còn của F_{DF} trên tập này (221,9 giây) cùng bậc với BiLSTM-Attention (167,6 giây) và thấp hơn BERT, RoBERTa.

Số liệu chi tiết tại Bảng J.16 (trình bày ở cuối Phụ lục J).

Thử nghiệm các bản điều chỉnh của mô hình rừng (như Bảng J.13) được lặp lại trên yelpResData với 5 hạt giống $\times$ 10.000 đánh giá (Bảng J.17). Trên tập này, các bản điều chỉnh không mang lại cải thiện: F_{CF+} , F_{DF+} , F_{MF+} đạt 0,537–0,539, tương đương Random Forest cùng bộ đặc trưng (0,537; chênh lệch +0,001 đến +0,002, khoảng tin cậy chứa 0); trọng số kết hợp a chọn trên tập xác thực chỉ từ 0 đến 0,2, tức F_{DF+} , F_{MF+} gần như trùng với rừng toàn cục F_{CF+} . Với bộ đặc trưng điều chỉnh, Logistic Regression và Gradient Boosting cao hơn Random Forest khoảng 0,02–0,03 có ý nghĩa thống kê. Ở nhóm đặc trưng như Bảng J.8, F_{CF} tương đương Random Forest (chênh lệch $-0,004$, khoảng tin cậy chứa 0), F_{DF} thấp hơn 0,020 (khoảng tin cậy sát 0) và F_{MF} thấp hơn có ý nghĩa ($-0,020$). Vì vậy, các điều chỉnh tại Bảng J.13 chỉ được ghi nhận như kết quả thăm dò trên yelpHotelData, chưa khái quát cho yelpResData.

Số liệu chi tiết tại Bảng J.17 (trình bày ở cuối Phụ lục J).

Tổng hợp mục J.11–J.12: theo quy trình chia theo đánh giá gốc, trên cả yelpHotelData và yelpResData (Bảng J.11, Bảng J.15, Bảng J.17,), F_{CF} và F_{DF} đạt Balanced Accuracy tương đương Random Forest (chênh lệch không có ý nghĩa thống kê) và thấp hơn SVM tuyến tính khoảng 0,01; F_{MF} thấp hơn Random Forest khoảng 0,02–0,03 có ý nghĩa thống kê do bước phân cụm đa tập chỉ học trên mẫu con. Các mô hình rừng đề xuất giữ ưu thế về khả năng diễn giải; giới hạn chung của mọi mô hình dùng bộ năm nằm ở lượng thông tin của bốn thành phần (e , a , h , t), phù hợp với hướng kết hợp biểu diễn ngữ nghĩa văn bản với cấu trúc rừng tại §5.2. Theo Cách 2 (Bảng J.14, Bảng J.16), khi mọi mô hình cùng chỉ dùng bộ năm, F_{DF} và F_{CF} đạt Balanced Accuracy tương đương BiLSTM-Attention và cao hơn về trung bình so với LSTM, BERT, RoBERTa trên cả hai tập dữ liệu; thời gian huấn luyện của F_{CF} thấp hơn nhiều so với các mô hình học sâu, còn của F_{DF} thấp hơn nhiều trên yelpHotelData và cùng bậc với BiLSTM-Attention trên yelpResData. Các bản điều chỉnh tại Bảng J.13 cải thiện trên yelpHotelData nhưng chưa cải thiện trên yelpResData (Bảng J.17).

BẢNG SỐ LIỆU CỦA PHỤ LỤC J

Bảng J.1: Kết quả Kịch bản 1 (có cân bằng bootstrap và tối ưu ngưỡng) trên Hotel và Restaurant

Chỉ số	Hotel	Restaurant
Accuracy	0,6558	0,6170
Precision	0,7475	0,7397
Recall (Sensitivity)	0,6987	0,6366
Specificity	0,5791	0,5802
F1	0,7223	0,6843
Balanced Accuracy	0,6389	0,6084
Ngưỡng tối ưu (trên tập Validation)	(*)	0,53
Cỡ mẫu N	112.275	112.275
Tỷ lệ Train / Validation / Test	60% / 20% / 20%	60% / 20% / 20%

Ghi chú: (*) Với cả hai tập dữ liệu, ngưỡng quyết định được chọn tự động bằng tìm kiếm lưới trong khoảng 0,05–0,95 trên tập Validation để tối đa hóa Balanced Accuracy (mục J.1); giá trị ngưỡng cụ thể của tập Hotel không được lưu lại trong nhật ký chạy (log) gốc, giá trị của tập Restaurant là 0,53.

Bảng J.2: Ma trận nhầm lẫn trên tập Test — Hotel

Thực tế \ Dự đoán	Negative	Positive
Negative	4.673	3.396

Thực tế \ Dự đoán	Negative	Positive
Positive	4.334	10.052

Bảng J.3: Ma trận nhầm lẫn trên tập Test — Restaurant

Thực tế \ Dự đoán	Negative	Positive
Negative	4.534	3.280
Positive	5.321	9.320

Bảng J.4: So sánh thiết lập thực nghiệm giữa các công bố và mục J.1–J.6

Nguồn	Bộ kết hợp K rừng	Cân bằng bootstrap	Tối ưu ngưỡng	Tách dữ liệu	Báo cáo Balanced Accuracy
Bài báo số 4 (2023)	Majority vote	Không	Không (0,5)	80 / 20	Không
Bài báo số 5 (2024)	Không nêu rõ	Không nêu rõ	Không nêu rõ	Không nêu rõ	Không
Bài báo số 6 (2024, đối chứng)	Stacking	Không	Không	60 / 20 / 20	Không
Mục J.1–J.6 (2026)	Trung bình mật độ xác suất (đúng §3.7.3.1)	Có	Có	60 / 20 / 20	Có

Bảng J.5: So sánh chỉ số trên tập Hotel

Nguồn	Accuracy	Precision	Recall	F1	Balanced Accuracy
Bài báo số 4 (test)	0,92	0,91	0,60	0,72	—
Bài báo số 5 (test)	0,75	0,71	0,62	0,67	—
Bài báo số 6 (test, đối chứng)	0,78	0,74	0,62	0,67	—
Mục J.1–J.6 (test)	0,6558	0,7475	0,6987	0,7223	0,6389

Bảng J.6: So sánh chỉ số trên tập Restaurant

Nguồn	Accuracy	Precision	Recall	F1	Balanced Accuracy
Bài báo số 4 (test)	0,91	0,89	0,59	0,71	—
Bài báo số 5 (test)	0,80	0,72	0,67	0,69	—
Bài báo số 6 (test, đối chứng)	0,82	0,75	0,67	0,71	—
Mục J.1–J.6 (test)	0,6170	0,7397	0,6366	0,6843	0,6084

Bảng J.7: Ba mô hình rừng đề xuất khi chạy lại (trung bình $\pm$ độ lệch chuẩn qua 5 lần lặp, Phụ lục H)

Tập dữ liệu	Mô hình	Accuracy	Balanced Accuracy	F1-macro	Số liệu công bố (Accuracy)
yelpHotelData	Rừng phân loại (F_{CF})	$0,6465 \pm 0,0085$	$0,6355 \pm 0,0043$	$0,6294 \pm 0,0058$	—
yelpHotelData	Rừng mật độ (F_{DF})	$0,6499 \pm 0,0082$	$0,6360 \pm 0,0056$	$0,6312 \pm 0,0062$	0,75 (Bài báo 5)
yelpHotelData	Rừng đa tập (F_{MF})	$0,6147 \pm 0,0062$	$0,5971 \pm 0,0061$	$0,5936 \pm 0,0058$	0,78 (Bài báo 6)
yelpResData	Rừng phân loại (F_{CF})	$0,6202 \pm 0,0036$	$0,6040 \pm 0,0037$	$0,5974 \pm 0,0024$	—
yelpResData	Rừng mật độ (F_{DF})	$0,6165 \pm 0,0120$	$0,6055 \pm 0,0040$	$0,5966 \pm 0,0070$	0,80 (Bài báo 5)
yelpResData	Rừng đa tập (F_{MF})	$0,5937 \pm 0,0059$	$0,5679 \pm 0,0040$	$0,5647 \pm 0,0044$	0,82 (Bài báo 6)

Bảng J.8: Balanced Accuracy trên tập kiểm tra theo hai cách đưa đầu vào

Mô hình	Nhóm	BA – Cách 1	BA – Cách 2 (KTC 95%)	ROC-AUC – Cách 2	Thời gian huấn luyện (s)
Rừng phân loại (F_{CF})	Đề xuất	0,544	0,544 (0,515–0,572)	0,545	0,4
Rừng mật độ (F_{DF})	Đề xuất	0,511	0,511 (0,472–0,548)	0,543	0,8
Rừng đa tập (F_{MF})	Đề xuất	0,526	0,526 (0,485–0,569)	0,536	811,3
Random Forest	Học máy	0,547	0,547 (0,519–0,575)	0,544	0,5
Gradient Boosting	Học máy	0,561	0,561 (0,526–0,595)	0,554	0,6

Mô hình	Nhóm	BA – Cách 1	BA – Cách 2 (KTC 95%)	ROC-AUC – Cách 2	Thời gian huấn luyện (s)
SVM (tuyến tính)	Học máy	0,545	0,545 (0,518–0,573)	0,565	0,2
LSTM	Học sâu	0,506	0,503 (0,495–0,513)	0,525	22,2
BiLSTM-Attention	Học sâu	0,634	0,488 (0,447–0,529)	0,488	38,3
BERT (tinh chỉnh)	Transformer	0,719	0,512 (0,470–0,553)	0,525	8.312,2
RoBERTa (tinh chỉnh)	Transformer	0,783	0,500 (0,500–0,500)	0,523	2.903,5

Ghi chú: ba mô hình rừng và ba mô hình học máy dùng cùng đầu vào bộ năm ở cả hai cách nên BA không đổi; thời gian huấn luyện lấy từ lần chạy Cách 2 trên cùng một máy CPU. Đây là một lần chia trên mẫu 3.000 đánh giá, dùng để so sánh tương đối giữa các mô hình trong cùng điều kiện.

Bảng J.9: Cấu hình huấn luyện của 10 mô hình trong thực nghiệm so sánh thống nhất (Bảng J.8)

Mô hình	Đầu vào	Cấu hình chính	Huấn luyện / dừng sớm
Rừng phân loại (F_{CF})	Bộ năm (e, a, h, t): target encoding ($m = 10$), chuẩn hóa Z-score	300 cây, Bootstrap cân bằng lớp (BalancedBoot)	Không lặp theo epoch
Rừng mật độ (F_{DF})	Bộ năm (e, a, h, t): target encoding ($m = 10$), chuẩn hóa Z-score	DBSCAN Euclidean (minPts = 5, eps theo phân vị khoảng cách k-láng giềng); mỗi cụm một rừng 300 cây BalancedBoot; trung bình xác suất	Không lặp theo epoch

Mô hình	Đầu vào	Cấu hình chính	Huấn luyện / dừng sớm
Rừng đa tập (F_{MF})	Bộ năm (e, a, h, t): target encoding (m = 10), chuẩn hóa Z-score	Ma trận W từ rừng không giám sát 300 cây trên mẫu con 5.000 dòng; DBSCAN trên $\sqrt{(1 - W)}$ (minPts = 5); mỗi cụm một rừng BalancedBoot; Stacking (meta-learner rừng 300 cây trên tập xác thực)	Không lặp theo epoch
Random Forest	Bộ năm (e, a, h, t): target encoding (m = 10), chuẩn hóa Z-score	randomForest, 300 cây, tham số mặc định	Không lặp theo epoch
Gradient Boosting	Bộ năm (e, a, h, t): target encoding (m = 10), chuẩn hóa Z-score	gbm, 300 cây, độ sâu tương tác 3, phân phối Bernoulli	Không lặp theo epoch
SVM (tuyến tính)	Bộ năm (e, a, h, t): target encoding (m = 10), chuẩn hóa Z-score	e1071::svm, nhân tuyến tính, ước lượng xác suất	Không lặp theo epoch
LSTM	Văn bản đánh giá gốc	Từ điển 20.000 từ, nhúng 128 chiều, 64 đơn vị ẩn, độ dài tối đa 128	Adam, lr = 10^{-3} , batch 32, tối đa 10 epoch
BiLSTM-Attention	Văn bản đánh giá gốc	Như LSTM, LSTM hai chiều kèm lớp chú ý (attention)	Adam, lr = 10^{-3} , batch 32, tối đa 10 epoch

Mô hình	Đầu vào	Cấu hình chính	Huấn luyện / dừng sớm
BERT (tinh chỉnh)	Văn bản đánh giá gốc	bert-base-multilingual-cased, độ dài tối đa 128	AdamW, lr = 2×10^{-5} , batch 16, tối đa 3 epoch
RoBERTa (tinh chỉnh)	Văn bản đánh giá gốc	roberta-base, độ dài tối đa 128	Như BERT

Ghi chú: mọi mô hình dùng cùng dữ liệu (3.000 đánh giá yelpHotelData, hạt giống 2026), cùng cách chia theo đánh giá 60/20/20 và chọn ngưỡng quyết định trên tập xác thực theo Balanced Accuracy; toàn bộ thực nghiệm chạy trên CPU, không dùng GPU (LSTM, BiLSTM-Attention bằng gói torch cho R; BERT, RoBERTa bằng PyTorch/Transformers). Thời gian huấn luyện tại Bảng J.8.

Bảng J.10: Ảnh hưởng của cách lấy mẫu và cách chia dữ liệu đến chỉ số đánh giá (Random Forest, yelpHotelData, 112.275 bộ năm)

Cách lấy mẫu 112.275 bộ năm	Cách chia dữ liệu	Tỷ lệ dòng Test có dòng cùng đánh giá trong Train	Accuracy	Recall	F1	Balanced Accuracy
Ngẫu nhiên trong toàn tập	Theo dòng bộ năm (như bài báo)	19,1%	0,664	0,866	0,766	0,589
Ngẫu nhiên trong toàn tập	Theo đánh giá gốc	0%	0,611	0,833	0,732	0,527
112.275 dòng đầu tập	Theo dòng bộ năm (như bài báo)	96,0%	0,946	0,984	0,960	0,930
112.275 dòng đầu tập	Theo đánh giá gốc	0%	0,611	0,793	0,725	0,537

Ghi chú: “dòng cùng đánh giá” là dòng bộ năm có cùng (e, h, t) và cùng nhãn với một dòng thuộc tập huấn luyện. Tỷ lệ positive của mẫu là 63,9% (lấy ngẫu nhiên) và 65,1% (dòng đầu tập).

Bảng J.11: So sánh thống nhất 10 mô hình trên yelpHotelData (10.000 đánh giá, 5 hạt giống; trung bình $\pm$ độ lệch chuẩn trên tập kiểm tra)

Mô hình	Nhóm	Đầu vào	Balanced Accuracy	ROC-AUC	Accuracy	F1-macro	Thời gian huấn luyện (s)
Rừng phân loại (F_{CF})	Đề xuất	Bộ năm	$0,557 \pm 0,009$	0,572	0,584	0,551	2,1
Rừng mật độ (F_{DF})	Đề xuất	Bộ năm	$0,561 \pm 0,006$	0,586	0,592	0,558	3,9
Rừng đa tập (F_{MF})	Đề xuất	Bộ năm	$0,534 \pm 0,010$	0,544	0,556	0,531	943,6
Random Forest	Học máy	Bộ năm	$0,554 \pm 0,011$	0,572	0,569	0,543	2,5
Gradient Boosting	Học máy	Bộ năm	$0,558 \pm 0,006$	0,580	0,575	0,548	1,5
SVM (tuyến tính)	Học máy	Bộ năm	$0,568 \pm 0,009$	0,590	0,589	0,562	7,7
LSTM	Học sâu	Văn bản gốc	$0,518 \pm 0,008$	0,526	0,597	0,493	45,8
BiLSTM-Attention	Học sâu	Văn bản gốc	$0,701 \pm 0,004$	0,774	0,698	0,687	104,1
BERT (tinh chỉnh) (*)	Transformer	Văn bản gốc	$0,755 \pm 0,013$	0,836	0,752	0,742	130,8
RoBERTa (tinh chỉnh)	Transformer	Văn bản gốc	$0,807 \pm 0,007$	0,895	0,818	0,805	108,2

Ghi chú: (*) BERT tính trên 4 lần chạy hội tụ; ở hạt giống 2030, BERT không hội tụ (Balanced Accuracy 0,500), tính cả lần này BERT đạt $0,704 \pm 0,114$. Thời gian huấn luyện của các mô hình dùng bộ năm và LSTM, BiLSTM-Attention đo trên CPU 12 luồng; BERT, RoBERTa đo trên GPU NVIDIA L4.

Bảng J.12: Chênh lệch Balanced Accuracy so với Random Forest (ghép cặp theo hạt giống, số liệu Bảng J.11)

Mô hình	Chênh lệch trung bình	Khoảng tin cậy 95%	Số hạt giống cao hơn Random Forest
Rừng phân loại (F_{CF})	+0,0036	$[-0,0033; 0,0105]$	4/5
Rừng mật độ (F_{DF})	+0,0074	$[-0,0042; 0,0190]$	4/5

Mô hình	Chênh lệch trung bình	Khoảng tin cậy 95%	Số hạt giống cao hơn Random Forest
Rừng đa tạp (F_{MF})	-0,0198	$[-0,0239; -0,0157]$	0/5
Gradient Boosting	+0,0044	$[-0,0062; 0,0150]$	3/5
SVM (tuyến tính)	+0,0144	$[0,0076; 0,0212]$	5/5

Ghi chú: khoảng tin cậy 95% theo phân phối t (4 bậc tự do) của chênh lệch ghép cặp qua 5 hạt giống; khoảng chứa 0 nghĩa là chênh lệch chưa có ý nghĩa thống kê.

Bảng J.13: Các bản điều chỉnh của mô hình rừng đề xuất (chỉ dùng bộ năm; yelpHotelData, 10.000 đánh giá, 5 hạt giống)

Nhóm đặc trưng	Mô hình	Balanced Accuracy (TB $\pm$ SD)	ROC-AUC	Chênh lệch so với Random Forest cùng nhóm	Khoảng tin cậy 95%
A (như Bảng J.8)	Rừng phân loại (F_{CF})	$0,547 \pm 0,003$	0,565	-0,0029	$[-0,0063; 0,0005]$
A (như Bảng J.8)	Rừng mật độ (F_{DF})	$0,545 \pm 0,019$	0,564	-0,0052	$[-0,0279; 0,0174]$
A (như Bảng J.8)	Rừng đa tạp (F_{MF})	$0,523 \pm 0,012$	0,533	-0,0272	$[-0,0401; -0,0142]$
A (như Bảng J.8)	Random Forest	$0,550 \pm 0,004$	0,563	—	—
A (như Bảng J.8)	Gradient Boosting	$0,544 \pm 0,011$	0,568	-0,0059	$[-0,0191; 0,0073]$
A (như Bảng J.8)	SVM (tuyến tính)	$0,554 \pm 0,015$	0,577	+0,0038	$[-0,0147; 0,0223]$
A (như Bảng J.8)	Logistic Regression	$0,553 \pm 0,014$	0,576	+0,0035	$[-0,0134; 0,0204]$

Nhóm đặc trưng	Mô hình	Balanced Accuracy (TB $\pm$ SD)	ROC-AUC	Chênh lệch so với Random Forest cùng nhóm	Khoảng tin cậy 95%
B (điều chỉnh)	F _{CF} ⁺	0,548 $\pm$ 0,008	0,560	+0,0121	[0,0034; 0,0208]
B (điều chỉnh)	F _{DF} ⁺	0,549 $\pm$ 0,011	0,565	+0,0134	[−0,0004; 0,0271]
B (điều chỉnh)	F _{MF} ⁺	0,547 $\pm$ 0,011	0,557	+0,0106	[−0,0026; 0,0237]
B (điều chỉnh)	Random Forest	0,536 $\pm$ 0,006	0,558	—	—
B (điều chỉnh)	Gradient Boosting	0,539 $\pm$ 0,005	0,560	+0,0028	[−0,0074; 0,0130]
B (điều chỉnh)	SVM (tuyến tính)	0,515 $\pm$ 0,021	0,526	−0,0208	[−0,0528; 0,0112]
B (điều chỉnh)	Logistic Regression	0,551 $\pm$ 0,014	0,579	+0,0153	[−0,0003; 0,0309]

Ghi chú: nhóm A dùng đặc trưng như Bảng J.8 (target encoding ước lượng trên chính tập huấn luyện, thời gian theo ngày); nhóm B dùng target encoding ngoài fold (5 fold theo đánh giá), tần suất của thực thể, người đánh giá và năm; các rừng nhóm B dùng nút lá tối thiểu 150 dòng. Thục nghiệm này dùng bộ đặc trưng và quy trình riêng nên giá trị nhóm A khác nhẹ so với các dòng tương ứng tại Bảng J.11.

Bảng J.14: So sánh thống nhất 10 mô hình theo Cách 2 – mọi mô hình chỉ dùng bộ năm (yelpHotelData, 10.000 đánh giá, 5 hạt giống; trung bình $\pm$ độ lệch chuẩn trên tập kiểm tra)

Mô hình	Nhóm	Balanced Accuracy	ROC-AUC	Accuracy	F1-macro	Thời gian huấn luyện (s)	Chênh lệch BA so với Random Forest [KTC 95%]	BA theo Cách 1 (Bảng J.11)
Rừng phân loại (F_{CF})	Đề xuất	$0,557 \pm 0,009$	0,572	0,584	0,551	2,2	$+0,0036 [-0,0033; 0,0105]$	0,557
Rừng mật độ (F_{DF})	Đề xuất	$0,561 \pm 0,006$	0,586	0,592	0,558	3,8	$+0,0074 [-0,0042; 0,0190]$	0,561
Rừng đa tạp (F_{MF})	Đề xuất	$0,534 \pm 0,010$	0,544	0,556	0,531	1.044,3	$-0,0198 [-0,0239; -0,0157]$	0,534
Random Forest	Học máy	$0,554 \pm 0,011$	0,572	0,569	0,543	2,6	–	0,554
Gradient Boosting	Học máy	$0,558 \pm 0,006$	0,580	0,575	0,548	1,6	$+0,0044 [-0,0062; 0,0150]$	0,558
SVM (tuyến tính)	Học máy	$0,568 \pm 0,009$	0,590	0,589	0,562	7,5	$+0,0144 [0,0076; 0,0212]$	0,568
LSTM	Học sâu	$0,504 \pm 0,008$	0,505	0,454	0,369	62,0	$-0,0494 [-0,0623; -0,0364]$	0,518

Mô hình	Nhóm	Balanced Accuracy	ROC-AUC	Accuracy	F1-macro	Thời gian huấn luyện (s)	Chênh lệch BA so với Random Forest [KTC 95%]	BA theo Cách 1 (Bảng J.11)
BiLSTM-Attention	Học sâu	$0,556 \pm 0,012$	0,577	0,539	0,530	95,7	+0,0017 [-0,0085; 0,0119]	0,701
BERT (tinh chỉnh)	Transformer	$0,538 \pm 0,009$	0,556	0,534	0,517	375,7	-0,0157 [-0,0379; 0,0066]	0,755 (*)
RoBERTa (tinh chỉnh)	Transformer	$0,523 \pm 0,024$	0,543	0,581	0,473	307,1	-0,0304 [-0,0554; -0,0054]	0,807

Ghi chú: cùng mẫu 10.000 đánh giá, cùng cách chia theo đánh giá 60/20/20, cùng 5 hạt giống (2026–2030) và cùng quy tắc chọn ngưỡng như Bảng J.11; sáu mô hình dùng bộ năm vì vậy có Balanced Accuracy trùng với Bảng J.11 (thời gian huấn luyện đo ở phiên chạy khác). LSTM, BiLSTM-Attention, BERT, RoBERTa đọc mỗi dòng bộ năm dưới dạng chuỗi “hotel <e> aspect <a> reviewer <h> time <t>” (tối đa 32 token), dự đoán theo từng dòng rồi lấy trung bình xác suất về đánh giá như các mô hình dùng bộ năm. Ở hạt giống 2026 và 2029, RoBERTa dự đoán toàn bộ là positive (Balanced Accuracy 0,500); ở hạt giống 2030, LSTM cũng dự đoán toàn bộ là positive. (*) BERT theo Cách 1 tính trên 4 lần chạy hội tụ (Bảng J.11). Khoảng tin cậy 95% theo phân phối t (4 bậc tự do) của chênh lệch ghép cặp qua 5 hạt giống.

Bảng J.15: So sánh thống nhất 10 mô hình trên yelpResData theo Cách 1 (10.000 đánh giá, 5 hạt giống; trung bình $\pm$ độ lệch chuẩn trên tập kiểm tra)

Mô hình	Nhóm	Đầu vào	Balanced Accuracy	ROC-AUC	Accuracy	F1-macro	Thời gian huấn luyện (s)	Chênh lệch BA so với Random Forest [KTC 95%]	BA trên yelpHotelData (Bảng J.11)
Rừng phân loại (F_{CF})	Đề xuất	Bộ năm	$0,541 \pm 0,010$	0,557	0,518	0,514	11,9	$-0,0019 [-0,0049; 0,0011]$	0,557
Rừng mật độ (F_{DF})	Đề xuất	Bộ năm	$0,543 \pm 0,011$	0,562	0,534	0,516	221,9	$+0,0007 [-0,0100; 0,0114]$	0,561
Rừng đa tạp (F_{MF})	Đề xuất	Bộ năm	$0,514 \pm 0,017$	0,520	0,522	0,507	1.048,4	$-0,0288 [-0,0472; -0,0105]$	0,534
Random Forest	Học máy	Bộ năm	$0,543 \pm 0,009$	0,560	0,528	0,518	14,7	–	0,554
Gradient Boosting	Học máy	Bộ năm	$0,551 \pm 0,009$	0,568	0,573	0,547	8,4	$+0,0079 [0,0029; 0,0128]$	0,558
SVM (tuyến tính)	Học máy	Bộ năm	$0,553 \pm 0,010$	0,580	0,576	0,543	13,5	$+0,0098 [0,0007; 0,0190]$	0,568

Mô hình	Nhóm	Đầu vào	Balanced Accuracy	ROC-AUC	Accuracy	F1-macro	Thời gian huấn luyện (s)	Chênh lệch BA so với Random Forest [KTC 95%]	BA trên yelpHotelData (Bảng J.11)
LSTM	Học sâu	Văn bản gốc	$0,542 \pm 0,007$	0,555	0,595	0,537	45,1	$-0,0011 [-0,0195; 0,0174]$	0,518
BiLSTM-Attention	Học sâu	Văn bản gốc	$0,735 \pm 0,014$	0,812	0,733	0,721	107,3	$+0,1923 [0,1814; 0,2032]$	0,701
BERT (tinh chỉnh)	Transformer	Văn bản gốc	$0,769 \pm 0,013$	0,850	0,771	0,759	130,5	$+0,2259 [0,2118; 0,2401]$	0,755 (*)
RoBERTa (tinh chỉnh)	Transformer	Văn bản gốc	$0,823 \pm 0,015$	0,911	0,828	0,817	108,2	$+0,2798 [0,2664; 0,2932]$	0,807

Ghi chú: mẫu 10.000 đánh giá lấy ngẫu nhiên từ tập bộ năm yelpResData đã ghép văn bản đánh giá gốc; mỗi đánh giá nhà hàng sinh tối đa 19 dòng bộ năm (yelpHotelData: 3 dòng). Cùng quy trình như Bảng J.11: chia theo đánh giá 60/20/20, ngưỡng chọn trên tập xác thực theo Balanced Accuracy, xác suất các dòng bộ năm gộp về đánh giá bằng trung bình. Do số dòng bộ năm lớn hơn, mô hình SVM tuyến tính chỉ huấn luyện trên tối đa 30.000 dòng Train lấy ngẫu nhiên. Thời gian huấn luyện của các mô hình dùng bộ năm và LSTM, BiLSTM-Attention đo trên CPU 12 luồng; BERT, RoBERTa trên GPU NVIDIA L4. (*) BERT trên yelpHotelData tính trên 4 lần chạy hội tụ. Khoảng tin cậy 95% theo phân phối t (4 bậc tự do) của chênh lệch ghép cặp qua 5 hạt giống.

Bảng J.16: So sánh thống nhất 10 mô hình trên yelpResData theo Cách 2 – mọi mô hình chỉ dùng bộ năm (10.000 đánh giá, 5 hạt giống; trung bình $\pm$ độ lệch chuẩn trên tập kiểm tra)

Mô hình	Nhóm	Balanced Accuracy	ROC-AUC	Accuracy	F1-macro	Thời gian huấn luyện (s)	Chênh lệch BA so với Random Forest [KTC 95%]	BA theo Cách 1 (Bảng J.15)
Rừng phân loại (F_{CF}) (**)	Đề xuất	$0,541 \pm 0,010$	0,557	0,518	0,514	11,9	$-0,0019 [-0,0049; 0,0011]$	0,541
Rừng mật độ (F_{DF}) (**)	Đề xuất	$0,543 \pm 0,011$	0,562	0,534	0,516	221,9	$+0,0007 [-0,0100; 0,0114]$	0,543
Rừng đa tập (F_{MF}) (**)	Đề xuất	$0,514 \pm 0,017$	0,520	0,522	0,507	1.048,4	$-0,0288 [-0,0472; -0,0105]$	0,514
Random Forest (**)	Học máy	$0,543 \pm 0,009$	0,560	0,528	0,518	14,7	–	0,543
Gradient Boosting (**)	Học máy	$0,551 \pm 0,009$	0,568	0,573	0,547	8,4	$+0,0079 [0,0029; 0,0128]$	0,551
SVM (tuyến tính) (**)	Học máy	$0,553 \pm 0,010$	0,580	0,576	0,543	13,5	$+0,0098 [0,0007; 0,0190]$	0,553
LSTM	Học sâu	$0,502 \pm 0,003$	0,501	0,580	0,388	93,9	$-0,0410 [-0,0539; -0,0281]$	0,542

Mô hình	Nhóm	Balanced Accuracy	ROC-AUC	Accuracy	F1-macro	Thời gian huấn luyện (s)	Chênh lệch BA so với Random Forest [KTC 95%]	BA theo Cách 1 (Bảng J.15)
BiLSTM-Attention	Học sâu	0,542 ± 0,010	0,563	0,559	0,531	167,6	−0,0011 [−0,0102; 0,0080]	0,735
BERT (tĩnh chỉnh)	Transformer	0,510 ± 0,021	0,522	0,639	0,420	692,8	−0,0332 [−0,0547; −0,0117]	0,769
RoBERTa (tĩnh chỉnh)	Transformer	0,502 ± 0,004	0,502	0,591	0,378	580,2	−0,0409 [−0,0552; −0,0266]	0,823

Ghi chú: cùng mẫu 10.000 đánh giá, cùng cách chia 60/20/20 theo đánh giá, cùng 5 hạt giống và cùng quy tắc chọn ngưỡng như Bảng J.15 (**). Sáu mô hình dùng bộ năm có cùng đầu vào ở hai cách nên không chạy lại; số liệu lấy từ Bảng J.15 (trên yelpHotelData, hai cách cho kết quả trùng nhau, Bảng J.14. Mô hình LSTM, BiLSTM-Attention, BERT, RoBERTa đọc mỗi dòng bộ năm dưới dạng chuỗi “restaurant <e> aspect <a> reviewer <h> time <t>” (tối đa 32 token), được huấn luyện trên 30.000 dòng bộ năm lấy ngẫu nhiên từ tập huấn luyện (giữ tỉ lệ lớp), dự đoán trên toàn bộ dòng của tập xác thực và tập kiểm tra rồi lấy trung bình xác suất về đánh giá. BERT dự đoán toàn bộ là positive ở hạt giống 2026, 2027, 2029, 2030; RoBERTa ở hạt giống 2026–2029; LSTM ở hạt giống 2029. Thời gian huấn luyện của LSTM, BiLSTM-Attention đo trên CPU 12 luồng; BERT, RoBERTa trên GPU NVIDIA L4. Khoảng tin cậy 95% theo phân phối t (4 bậc tự do) của chênh lệch ghép cặp qua 5 hạt giống.

Bảng J.17: Các bản điều chỉnh của mô hình rừng đề xuất trên yelpResData (chỉ dùng bộ năm; 10.000 đánh giá, 5 hạt giống)

Nhóm đặc trưng	Mô hình	Balanced Accuracy (TB $\pm$ SD)	ROC-AUC	Chênh lệch so với Random Forest cùng nhóm	Khoảng tin cậy 95%	BA trên yelpHotelData (Bảng J.13)
A (như Bảng J.8)	Rừng phân loại (F_{CF})	$0,551 \pm 0,013$	0,567	-0,0044	[-0,0167; 0,0078]	0,547
A (như Bảng J.8)	Rừng mật độ (F_{DF})	$0,535 \pm 0,020$	0,547	-0,0202	[-0,0405; 0,0001]	0,545
A (như Bảng J.8)	Rừng đa tạp (F_{MF})	$0,535 \pm 0,005$	0,540	-0,0203	[-0,0300; -0,0105]	0,523
A (như Bảng J.8)	Random Forest	$0,555 \pm 0,007$	0,572	—	—	0,550
A (như Bảng J.8)	Gradient Boosting	$0,558 \pm 0,015$	0,575	+0,0030	[-0,0119; 0,0179]	0,544
A (như Bảng J.8)	SVM (tuyến tính)	$0,569 \pm 0,011$	0,592	+0,0138	[0,0048; 0,0229]	0,554
A (như Bảng J.8)	Logistic Regression	$0,566 \pm 0,010$	0,592	+0,0108	[0,0029; 0,0187]	0,553
B (điều chỉnh)	F_{CF}^+	$0,539 \pm 0,011$	0,560	+0,0022	[-0,0168; 0,0212]	0,548
B (điều chỉnh)	F_{DF}^+	$0,537 \pm 0,009$	0,560	+0,0005	[-0,0182; 0,0191]	0,549
B (điều chỉnh)	F_{MF}^+	$0,539 \pm 0,011$	0,560	+0,0019	[-0,0165; 0,0203]	0,547
B (điều chỉnh)	Random Forest	$0,537 \pm 0,013$	0,553	—	—	0,536
B (điều chỉnh)	Gradient Boosting	$0,557 \pm 0,014$	0,579	+0,0204	[0,0084; 0,0324]	0,539

Nhóm đặc trưng	Mô hình	Balanced Accuracy (TB $\pm$ SD)	ROC-AUC	Chênh lệch so với Random Forest cùng nhóm	Khoảng tin cậy 95%	BA trên yelpHotelData (Bảng J.13)
B (điều chỉnh)	SVM (tuyến tính)	0,540 $\pm$ 0,019	0,553	+0,0030	[−0,0219; 0,0278]	0,515
B (điều chỉnh)	Logistic Regression	0,566 $\pm$ 0,006	0,595	+0,0287	[0,0142; 0,0431]	0,551

Ghi chú: quy trình và đặc trưng như Bảng J.13; mẫu 10.000 đánh giá lấy từ toàn bộ tập bộ năm yelpResData (không cần văn bản gốc), nên khác mẫu của Bảng J.15–J.16 và giá trị nhóm A khác nhẹ so với các dòng tương ứng tại Bảng J.15. Khoảng tin cậy 95% theo phân phối t (4 bậc tự do) của chênh lệch ghép cặp qua 5 hạt giống (2026–2030). SVM tuyến tính huấn luyện trên tối đa 30.000 dòng Train. F_{DF+} , F_{MF+} chọn trọng số kết hợp α từ 0 đến 0,2 trên tập xác thực nên kết quả gần trùng với F_{CF+} . Ở hạt giống 2026 và 2030, DBSCAN chọn $\text{eps} = 0$ nên có cụm gồm toàn các điểm trùng nhau; rừng của cụm như vậy không thể tách nút nên được thay bằng dự đoán hằng số 0,5 (đầu ra của rừng cân bằng lớp khi không tách được nút).