

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN KHẮC ANH TÀI
LÊ TUẤN THÀNH**

**CẢI TIẾN HỆ THỐNG
KHUYẾN NGHỊ THỜI TRANG VỚI
LIGHTGBM VÀ TRANSFORMER:
HIỆN THỰC VÀ ĐÁNH GIÁ**

**KHÓA LUẬN TỐT NGHIỆP
KHOA HỌC MÁY TÍNH**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN KHẮC ANH TÀI - 52100306
LÊ TUẤN THÀNH - 52100312**

**CẢI TIẾN HỆ THỐNG
KHUYẾN NGHỊ THỜI TRANG VỚI
LIGHTGBM VÀ TRANSFORMER:
HIỆN THỰC VÀ ĐÁNH GIÁ**

**KHÓA LUẬN TỐT NGHIỆP
KHOA HỌC MÁY TÍNH**

Người hướng dẫn
TS. Trần Trung Tín

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Đầu tiên, chúng em được chân thành gửi lời cảm ơn đến các thầy cô giảng dạy trong trường ĐH Tôn Đức Thắng. Chúng em muốn gửi lời cảm ơn sâu sắc nhất đến những người đã truyền dạy cho chúng em những kiến thức quý báu, giúp chúng em phát triển và tiến bộ trong cuộc sống, đặc biệt là TS. Trần Trung Tín vì sự hỗ trợ và tạo điều kiện thuận lợi mà chúng em đã nhận được để thực hiện bài báo cáo Khoá luận tốt nghiệp. Chúng em xin chân thành cảm ơn TS. Trần Trung Tín đã dành thời gian và công sức để hướng dẫn, chỉ bảo chúng em trong suốt quá trình nghiên cứu và thực hiện bài báo cáo. Những chia sẻ, kinh nghiệm của thầy là yếu tố hết sức quan trọng để chúng em hoàn thành bài báo cáo này và đặc biệt hơn, chúng còn là những hành trang vô cùng đáng giá cho chúng em trên chặng đường sắp tới. Chúng em đã hoàn thành bài báo cáo của mình, tuy nhiên, do thiếu sót về kinh nghiệm và kiến thức, chúng em nhận thức rằng có những khía cạnh cần được cải thiện. Chúng em rất mong nhận được những ý kiến và đóng góp quý giá từ quý thầy cô để bài báo cáo được hoàn thiện hơn. Cuối cùng, chúng em xin kính chúc quý thầy cô dồi dào sức khỏe, luôn gặt hái được nhiều thành công trong công việc cũng như trong cuộc sống. Một lần nữa, chúng em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 17 tháng 07 năm 2025

Tác giả

(Ký tên và ghi rõ họ tên)

Nguyễn Khắc Anh Tài

Lê Tuấn Thành

LỜI CAM ĐOAN KHOÁ LUẬN TỐT NGHIỆP

Công trình được hoàn thành tại Trường Đại học Tôn Đức Thắng

Cán bộ hướng dẫn khoa học: TS. Trần Trung Tín

Khóa luận/Đồ án tốt nghiệp được bảo vệ tại **Hội đồng đánh giá Khóa luận/Đồ án tốt nghiệp của Trường Đại học Tôn Đức Thắng** vào ngày... /.../.....

Xác nhận của Chủ tịch Hội đồng đánh giá Khóa luận/Đồ án tốt nghiệp và Trưởng khoa quản lý chuyên ngành sau khi nhận Khóa luận/Đồ án tốt nghiệp đã được sửa chữa (nếu có).

CHỦ TỊCH HỘI ĐỒNG

TRƯỞNG KHOA

.....
.....

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Trần Trung Tín. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 17 tháng 07 năm 2025

Tác giả

(Ký tên và ghi rõ họ tên)

Nguyễn Khắc Anh Tài

Lê Tuấn Thành

TÓM TẮT

Trong bối cảnh thương mại điện tử phát triển nhanh chóng, hệ thống khuyến nghị trở thành thành phần thiết yếu nhằm cá nhân hóa trải nghiệm người dùng và tối ưu hóa hiệu quả kinh doanh. Đề tài đề xuất hai hướng cải tiến hệ thống khuyến nghị dựa trên hai kỹ thuật học máy tiên tiến là LightGBM và Transformer, được triển khai độc lập để xử lý các khía cạnh khác nhau của dữ liệu và hành vi người dùng.

LightGBM được áp dụng trong pipeline xếp hạng hai tầng. Tầng sinh ứng viên sử dụng các chiến lược dựa trên hành vi như lịch sử mua hàng và xu hướng tiêu dùng để tạo danh sách sản phẩm tiềm năng. Sau đó, tầng xếp hạng dùng LightGBM để học các mối quan hệ phi tuyến từ dữ liệu tương tác người dùng – sản phẩm, kết hợp với DNN nhằm tăng cường khả năng phân biệt các tương tác phức tạp.

Trong khi đó, Transformer đóng vai trò trung tâm trong mô hình Visual-TransGNN, tích hợp cùng Graph Neural Network (GNN) để tận dụng cấu trúc đồ thị tương tác và khắc phục hạn chế trong truyền thông tin đa bước. Transformer mở rộng vùng tiếp nhận và tập trung vào các node quan trọng nhờ attention đa đầu, từ đó cải thiện khả năng nắm bắt ngữ cảnh. Mô hình còn kết hợp đặc trưng hình ảnh sản phẩm trích xuất từ CNN và lan truyền theo độ tương đồng hình ảnh, phản ánh yếu tố thẩm mỹ trong hành vi tiêu dùng.

Hai mô hình được kiểm thử riêng biệt trên hai tập dữ liệu thực tế là Vibrent và H&M, qua đó cho thấy ưu điểm nổi bật của từng phương pháp. Mô hình LightGBM kết hợp DNN hiệu quả trong trích xuất đặc trưng và xếp hạng nhanh, trong khi Visual-TransGNN tận dụng tốt cấu trúc đồ thị và đặc trưng hình ảnh để cải thiện độ chính xác khuyến nghị. Với khả năng xử lý dữ liệu hành vi mạnh mẽ từ LightGBM và năng lực học ngữ cảnh sâu trên đồ thị của Transformer, hai hướng tiếp cận này mở ra triển vọng ứng dụng linh hoạt, cả độc lập lẫn kết hợp trong các hệ thống khuyến nghị thời trang cá nhân hóa ở môi trường thực tế.

ABSTRACT

The rapid expansion of e-commerce has turned recommender systems into a core component for enhancing user experience and improving business performance. This study presents two distinct approaches for enhancing recommendation quality by leveraging two advanced machine learning techniques—LightGBM and Transformer. Each technique is implemented in a separate model, tailored to handle different aspects of data and user behavior.

The first model adopts a two-stage ranking pipeline. The candidate generation stage utilizes behavior-based strategies, such as purchase history and consumption trends, to build personalized product lists. In the subsequent stage, LightGBM captures non-linear relationships in user–item interactions, while DNN complements it by modeling more complex feature dependencies. Together, they form a robust ranking mechanism capable of fast and effective prediction.

The second model, Visual-TransGNN, integrates Transformer modules within a Graph Neural Network (GNN) architecture. This design addresses the limitations of traditional GNNs in long-range message passing and contextual understanding. Transformer layers, equipped with multi-head attention, expand the receptive field and highlight salient nodes within the interaction graph. Meanwhile, product image features extracted via CNNs are propagated through a visual similarity matrix, enhancing the model’s sensitivity to aesthetic preferences in user behavior.

Both models undergo independent evaluation on real-world datasets, Vibrent and H&M. The LightGBM+DNN pipeline excels at feature representation and efficient ranking, while Visual-TransGNN demonstrates strong performance in capturing structural and visual information. The two approaches, one grounded in behavioral modeling and the other in contextual reasoning demonstrate promising potential for

standalone or combined deployment in large-scale personalized fashion recommendation systems.

MỤC LỤC

DANH MỤC HÌNH VẼ	xi
DANH MỤC BẢNG BIỂU	xii
DANH MỤC CÁC CHỮ VIẾT TẮT	1
CHƯƠNG 1 - MỞ ĐẦU	2
1.1 Lý do chọn đề tài.....	2
1.2 Mục tiêu thực hiện đề tài.....	3
1.3 Đối tượng nghiên cứu	4
1.4 Phạm vi nghiên cứu.....	6
CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT	8
2.1 Các hệ thống khuyến nghị.....	8
2.1.1 <i>Content-based Filtering</i>	8
2.1.2 <i>Collaborative Filtering</i>	8
2.1.3 <i>Hybrid Recommender System</i>	9
2.1.4 <i>Tổng quan về các hệ thống khuyến nghị trên</i>	10
2.2 Tổng quan về Graph Neural Networks (GNN).....	13
2.3 Tổng quan về LightGBM.....	15
2.3.1 <i>Nguyên lý hoạt động</i>	16
2.3.2 <i>Ưu điểm của LightGBM</i>	16
2.3.3 <i>Ứng dụng trong hệ thống khuyến nghị</i>	17
2.4 Transformer trong biểu diễn đồ thị	18

2.4.1 Cơ chế <i>attention</i> trong <i>Transformer</i>	18
2.4.2 Khả năng mở rộng <i>Transformer</i> lên dữ liệu đồ thị.....	18
2.4.3 Một số biến thể <i>Transformer</i> cho dữ liệu đồ thị	19
2.4.4 Ưu điểm khi áp dụng trong hệ thống khuyến nghị.....	19
2.5 Dự đoán tương tác và tối ưu hóa mô hình với BPR Loss	20
2.5.1 Hàm dự đoán tương tác	20
2.5.2 Hàm mất mát BPR.....	21
2.6 Các độ đo đánh giá hiệu suất mô hình	22
CHƯƠNG 3 - KIẾN TRÚC MÔ HÌNH ĐỀ XUẤT VÀ PHƯƠNG PHÁP	
NGHIÊN CỨU.....	24
3.1 Light Gradient Boosting Machine (LightGBM)	24
3.1.1 Thuật toán dựa trên <i>histogram</i>	24
3.1.2 Chiến lược tăng trưởng cây: <i>Leaf-wise</i> thay vì <i>Level-wise</i>	25
3.1.3 Kỹ thuật <i>GOSS</i> và <i>EFB</i> đặc trưng	25
3.2 Deep Neural Network (DNN)	26
3.2.1 Kiến trúc của DNN.....	27
3.2.2 Quy trình huấn luyện.....	28
3.3 Visual-TransGNN (Visual Transformer Graph Neural Network)	28
3.3.1 Tổng quan kiến trúc <i>Visual-TransGNN</i>	29
3.3.2 Phương pháp đo độ tương đồng hình ảnh	32
3.3.3 <i>Attention Sampling Module</i>	33
3.3.4 <i>Positional Encoding Module</i>	34

3.3.5 <i>Visual TransGNN Module</i>	35
3.4 Các phương pháp sinh ứng viên nền tảng trong hệ thống khuyến nghị ...	37
3.4.1 <i>Dựa trên lịch sử giao dịch</i>	37
3.4.2 <i>Dựa trên nhóm người dùng</i>	38
3.4.3 <i>Dựa trên lọc cộng tác</i>	38
CHƯƠNG 4 - THỰC NGHIỆM	39
4.1 Các tập dữ liệu được sử dụng.....	39
4.1.1 <i>Vibrent Clothes Rental</i>	39
4.1.2 <i>H&M Personalized Fashion Recommendations</i>	41
4.2 Triển khai hệ thống khuyến nghị đa mô hình với LightGBM và DNN....	42
4.2.1 <i>Cấu trúc thư mục mô hình LightGBM</i>	42
4.2.2 <i>Thu thập các ứng viên</i>	43
4.2.3 <i>Huấn luyện mô hình</i>	46
4.2.4 <i>Kết hợp hai mô hình LightGBM 1 và LightGBM 2</i>	47
4.2.5 <i>Kết quả thực nghiệm</i>	49
4.3 Triển khai mô hình Visual-TransGNN	50
4.3.1 <i>Cấu trúc thư mục mô hình Visual-TransGNN</i>	50
4.3.2 <i>Attention Sampling Module</i>	53
4.3.3 <i>Positional Encoding Module</i>	54
4.3.4 <i>Visual-TransGNN Module</i>	55
4.3.5 <i>Đánh giá mức độ ảnh hưởng của từng thành phần trong mô hình Visual-TransGNN</i>	59

CHƯƠNG 5 – KẾT QUẢ THỰC NGHIỆM.....	61
5.1 Các mô hình nền tảng để so sánh.....	61
5.2 Kết quả và đánh giá.....	62
5.2.1 <i>Đánh giá hiệu suất mô hình với topK=5</i>	64
5.2.2 <i>Đánh giá hiệu suất mô hình với topK=20</i>	64
5.2.3 <i>Đánh giá tổng quan</i>	65
CHƯƠNG 6 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI.....	66
6.1 Kết luận	66
6.2 Hướng phát triển trong tương lai	67
TÀI LIỆU THAM KHẢO	70

DANH MỤC HÌNH VẼ

Hình 2.1 Tổng quan về GNN	13
Hình 2.2 Nguyên lý hoạt động của LightGBM.....	15
Hình 3.1 Quy trình làm việc của mô hình đề xuất.....	24
Hình 3.2 So sánh kiến trúc giữa Neural Networks và Deep Neural Networks.....	26
Hình 3.3 Tổng quan kiến trúc Visual-TransGNN.....	29
Hình 4.1 Cấu trúc thư mục mô hình LightBGM.....	42
Hình 4.2 Cấu trúc thư mục mô hình Visual-TransGNN	52

DANH MỤC BẢNG BIỂU

Bảng 2.1 So sánh Content-based Filtering, Collaborative Filtering và Hybrid Recommender System.....	11
Bảng 4.1 Hiệu suất mô hình LightGBM 1	49
Bảng 4.2 Hiệu suất mô hình LightGBM 2	49
Bảng 4.3 Hiệu suất mô hình kết hợp giữa LightGBM 1 và LightGBM 2	50
Bảng 4.4 So sánh ảnh hưởng các thành phần của Visual-TransGNN trên tập dữ liệu Vibrent.....	59
Bảng 5.1 So sánh hiệu suất các mô hình với topK=5.....	62
Bảng 5.2 Bảng So sánh hiệu suất các mô hình với topK=20.....	63

DANH MỤC CÁC CHỮ VIẾT TẮT

BPR	Bayesian Personalized Ranking
CF	Collaborative Filtering
CNN	Convolutional Neural Networks
GCN	Graph Convolution Networks
HRS	Hybrid Recommend System
LightGBM	Light Gradient-Boosting Machine
MLP	Multi-layer Perceptron
NGCF	Neural Graph Collaborative Filtering
PCA	Principal Component Analysis

CHƯƠNG 1 - MỞ ĐẦU

1.1 Lý do chọn đề tài

Trong thời đại số, thương mại điện tử không ngừng phát triển với sự cạnh tranh khốc liệt giữa các nền tảng bán lẻ trực tuyến. Để gia tăng khả năng giữ chân khách hàng và tối ưu hóa doanh thu, các hệ thống gợi ý sản phẩm đóng vai trò then chốt trong việc cá nhân hóa trải nghiệm người dùng. Tuy nhiên, trong lĩnh vực thời trang, nơi mà cảm quan thẩm mỹ như màu sắc, chất liệu, kiểu dáng đóng vai trò quan trọng, các phương pháp khuyến nghị truyền thống như lọc cộng tác (Collaborative Filtering) hoặc gợi ý dựa trên nội dung (Content-based Filtering) thường gặp khó khăn trong việc phản ánh chính xác sở thích cá nhân của người dùng (Ricci, Rokach, & Shapira, 2011).

Gần đây, sự phát triển mạnh mẽ của các mô hình học sâu đã mở ra hướng tiếp cận mới trong xây dựng hệ thống khuyến nghị. Các mô hình như LightGBM (Ke et al., 2017) và Deep Neural Network – DNN (He et al., 2017) cho phép học được các đặc trưng phi tuyến từ dữ liệu tương tác và thông tin người dùng/sản phẩm, từ đó nâng cao hiệu quả khuyến nghị. Bên cạnh đó, các kỹ thuật mô hình hóa đồ thị như GNN (Graph Neural Network) giúp khai thác tốt cấu trúc tương tác giữa người dùng và sản phẩm, trong khi kiến trúc Transformer lại nổi bật với khả năng mở rộng trường tiếp nhận và tập trung vào các thành phần quan trọng trong dữ liệu (Vaswani et al., 2017).

Tuy vậy, mỗi phương pháp vẫn còn những hạn chế riêng: GNN bị giới hạn bởi chiều sâu lan truyền và dễ bị nhiễu từ các kết nối không mang tính đại diện (Wu et al., 2020); LightGBM và DNN thiếu khả năng biểu diễn mối quan hệ có cấu trúc; và các phương pháp truyền thống khó tích hợp đặc trưng thẩm mỹ từ hình ảnh. Việc tận dụng ưu điểm của các phương pháp hiện đại, đồng thời khắc phục nhược điểm của từng hướng tiếp cận, đang trở thành một xu hướng nghiên cứu đầy tiềm năng.

Từ thực tiễn nêu trên, đề tài lựa chọn khai thác hai hướng tiếp cận độc lập sử dụng LightGBM và Transformer – đại diện cho hai trường phái học máy hiện đại: một dựa trên boosting trees với khả năng xử lý phi tuyến mạnh mẽ, và một dựa trên cơ chế attention giúp mở rộng ngữ cảnh qua cấu trúc dữ liệu. Cụ thể, đề tài hiện thực hóa hai mô hình: mô hình LightGBM+DNN theo pipeline xếp hạng hai tầng, và mô hình Visual-TransGNN tích hợp Transformer trong GNN để khai thác cả thông tin cấu trúc và hình ảnh. Hai mô hình được triển khai, đánh giá riêng biệt trên dữ liệu thực tế, từ đó đưa ra cái nhìn thực chứng về hiệu quả và tiềm năng ứng dụng của từng hướng tiếp cận trong các hệ thống khuyến nghị thời trang hiện đại.

1.2 Mục tiêu thực hiện đề tài

Mục tiêu của đề tài là nghiên cứu và phát triển các hướng tiếp cận dựa trên hai kỹ thuật học máy hiện đại là LightGBM và Transformer, nhằm cải tiến hiệu quả của hệ thống khuyến nghị sản phẩm thời trang cá nhân hóa. Thay vì xây dựng một kiến trúc tích hợp, đề tài tập trung hiện thực hóa và đánh giá hai mô hình độc lập — mỗi mô hình khai thác tối đa điểm mạnh của một kỹ thuật, qua đó giải quyết các khía cạnh khác nhau trong dữ liệu và hành vi người dùng.

Cụ thể, đề tài đặt ra các mục tiêu sau:

- Phân tích tổng quan các hướng tiếp cận trong bài toán khuyến nghị, từ các kỹ thuật truyền thống (Collaborative Filtering, Content-based Filtering) đến các phương pháp hiện đại như DNN, GNN, LightGBM và Transformer.
- Xây dựng mô hình xếp hạng hai tầng sử dụng LightGBM kết hợp với DNN để học các đặc trưng phi tuyến từ dữ liệu hành vi người dùng – sản phẩm, nâng cao khả năng phân biệt và xếp hạng các lựa chọn khuyến nghị.

- Thiết kế mô hình Visual-TransGNN, trong đó Transformer được tích hợp vào kiến trúc GNN để tăng khả năng học ngữ cảnh từ đồ thị tương tác và mở rộng trường tiếp nhận thông tin, đồng thời khắc phục các hạn chế lan truyền trong GNN truyền thống.
- Tích hợp đặc trưng hình ảnh sản phẩm, được trích xuất bằng CNN, vào mô hình Visual-TransGNN nhằm phản ánh yếu tố thẩm mỹ trong hành vi tiêu dùng, thông qua lan truyền theo ma trận tương đồng hình ảnh.
- Thực nghiệm và đánh giá độc lập hiệu suất của từng mô hình trên các tập dữ liệu thực tế (Vibrent và H&M), sử dụng các chỉ số phổ biến như Recall@K, NDCG@K, HR@K, MRR@K; từ đó rút ra so sánh và phân tích tiềm năng ứng dụng của từng hướng tiếp cận trong môi trường thương mại điện tử quy mô lớn.

1.3 Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là hệ thống khuyến nghị sản phẩm thời trang cá nhân hóa, với trọng tâm là xây dựng và hiện thực hóa các mô hình khuyến nghị hiện đại, có khả năng xử lý đa dạng các nguồn dữ liệu như hành vi người dùng, đặc trưng hình ảnh của sản phẩm và cấu trúc tương tác người dùng – sản phẩm dưới dạng đồ thị. Trong đó, hai kỹ thuật chính được khai thác và đánh giá là LightGBM và Transformer, mỗi kỹ thuật được triển khai trong một mô hình riêng biệt phù hợp với từng loại dữ liệu và mục tiêu xử lý cụ thể.

Đề tài tập trung làm rõ và triển khai các thành phần cốt lõi sau trong quá trình nghiên cứu:

- Mô hình khuyến nghị dựa trên Graph Neural Network (GNN) nhằm biểu diễn cấu trúc tương tác người dùng – sản phẩm, qua đó lan truyền thông tin để học được embedding phản ánh ngữ nghĩa sâu sắc.
- Tích hợp Transformer vào kiến trúc GNN nhằm mở rộng vùng tiếp nhận thông tin, đồng thời xác định các node quan trọng trong quá trình lan truyền qua cơ chế multi-head attention.
- Áp dụng Positional Encoding để bổ sung thông tin vị trí tương đối giữa các node trên đồ thị, cải thiện hiệu quả biểu diễn trong mô hình GNN-Transformer kết hợp.
- Xây dựng pipeline xếp hạng hai tầng sử dụng LightGBM để khai thác mối quan hệ phi tuyến trong dữ liệu hành vi người dùng, kết hợp với Deep Neural Network (DNN) để tăng cường khả năng phân biệt và tối ưu hóa kết quả khuyến nghị.
- Khai thác đặc trưng hình ảnh sản phẩm thông qua các mô hình học sâu như CNN, từ đó bổ sung yếu tố thẩm mỹ vào quá trình biểu diễn và lan truyền thông tin trong đồ thị tương tác.
- Sử dụng cơ chế Attention Sampling để chọn lọc node quan trọng, giảm nhiễu và tối ưu hiệu quả huấn luyện trong các mô hình dựa trên GNN và Transformer.

Khác với các nghiên cứu tập trung vào một mô hình duy nhất, đối tượng nghiên cứu trong đề tài này là một hệ thống gồm hai mô hình độc lập, được thiết kế theo hai hướng tiếp cận học máy hiện đại. Mỗi mô hình xử lý một khía cạnh đặc trưng của dữ liệu, từ đó làm rõ khả năng ứng dụng riêng lẻ hoặc kết hợp của LightGBM và

Transformer trong việc nâng cao chất lượng khuyến nghị thời trang cá nhân hóa trên các nền tảng thương mại điện tử.

1.4 Phạm vi nghiên cứu

Đề tài tập trung vào việc thiết kế, triển khai và đánh giá hai mô hình khuyến nghị độc lập nhằm cải thiện chất lượng gợi ý sản phẩm thời trang cá nhân hóa trong thương mại điện tử. Hai mô hình được phát triển dựa trên hai kỹ thuật học máy hiện đại là LightGBM và Transformer, trong đó mỗi mô hình xử lý một khía cạnh riêng biệt của dữ liệu: hành vi tương tác và ngữ cảnh đồ thị.

Về phạm vi dữ liệu, nghiên cứu sử dụng thông tin tương tác người dùng – sản phẩm ở dạng feedback ẩn (implicit feedback), cùng với đặc trưng hình ảnh của sản phẩm được trích xuất từ các mô hình học sâu như CNN. Dữ liệu văn bản như mô tả sản phẩm hoặc đánh giá từ người dùng không được đưa vào xử lý nhằm giữ sự tập trung vào phân tích hành vi và tín hiệu thị giác.

Trên phương diện mô hình hóa, đề tài không triển khai một hệ thống khuyến nghị tích hợp đầu cuối hay giao diện ứng dụng, mà giới hạn trong việc xây dựng và kiểm thử hai mô hình khuyến nghị độc lập:

- Mô hình LightGBM+DNN, theo kiến trúc xếp hạng hai tầng, tập trung vào học các đặc trưng phi tuyến từ hành vi người dùng.
- Mô hình Visual-TransGNN, tích hợp Transformer trong GNN, nhằm mở rộng khả năng tiếp nhận thông tin từ đồ thị tương tác và khai thác đồng thời đặc trưng hình ảnh sản phẩm.

Đề tài chỉ thực hiện đánh giá hiệu quả mô hình trên hai tập dữ liệu thực tế là Vibrent (kaborg15, 2023) và H&M Personalized Fashion Recommendations (H&M Group, 2022) từ Kaggle, thông qua các chỉ số phổ biến như Recall@K, NDCG@K,

HR@K và MRR@K. Việc triển khai thực tế, tối ưu hóa hệ thống, tích hợp giao diện người dùng hay đánh giá hiệu năng trong môi trường sản xuất không nằm trong phạm vi nghiên cứu của đề tài này.

CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT

2.1 Các hệ thống khuyến nghị

2.1.1 *Content-based Filtering*

Content-based filtering (Gợi ý dựa trên nội dung) là phương pháp đề xuất sản phẩm bằng cách phân tích đặc trưng của các mục tiêu mà người dùng đã quan tâm trước đó, từ đó tìm ra những sản phẩm tương đồng. Ý tưởng cốt lõi là nếu người dùng thích một sản phẩm, thì nhiều khả năng họ cũng sẽ thích những sản phẩm có đặc điểm tương tự.

Phương pháp này có một số lợi thế nhất định, chẳng hạn như không phụ thuộc vào dữ liệu hành vi của người dùng, dễ triển khai và có khả năng đưa ra gợi ý ngay cả khi sản phẩm mới vừa được thêm vào hệ thống. Hơn nữa, vì dựa trên đặc trưng nội dung, nên lý do một sản phẩm được đề xuất thường khá rõ ràng và dễ giải thích cho người dùng.

Tuy nhiên, phương pháp này cũng tồn tại nhiều hạn chế. Hệ thống thường chỉ gợi ý những sản phẩm tương tự với lịch sử tương tác, nên thiếu khả năng khám phá các sở thích mới. Ngoài ra, việc xây dựng mô hình nội dung hiệu quả đòi hỏi dữ liệu chi tiết và công sức đáng kể. Nếu sản phẩm không có mô tả nội dung đầy đủ, hoặc nội dung thay đổi nhanh như trong ngành thời trang, hệ thống có thể không theo kịp.

2.1.2 *Collaborative Filtering*

Collaborative Filtering (Lọc cộng tác) đưa ra gợi ý bằng cách khai thác hành vi tương tác của tập người dùng. Dựa trên giả định rằng những người dùng có sở thích tương tự trong quá khứ thì cũng có xu hướng thích những sản phẩm giống nhau trong tương lai, hệ thống có thể dự đoán các mục tiêu mà người dùng chưa từng tương tác.

Có nhiều biến thể của phương pháp này, như lọc cộng tác dựa trên người dùng, sản phẩm, hoặc sử dụng kỹ thuật phân rã ma trận và mạng nơ-ron sâu để học biểu diễn ẩn. Một trong những điểm mạnh của phương pháp này là khả năng khám phá sở thích tiềm ẩn và không yêu cầu thông tin mô tả chi tiết về sản phẩm. Khi áp dụng trên quy mô lớn, lọc cộng tác có thể mang lại hiệu quả đáng kể.

Tuy vậy, phương pháp này lại gặp nhiều khó khăn trong việc xử lý trường hợp cold start, đặc biệt là khi người dùng hoặc sản phẩm chưa có đủ dữ liệu tương tác. Ngoài ra, sự thừa thớt của ma trận tương tác cũng ảnh hưởng đến độ chính xác của dự đoán. Trong một số trường hợp, hệ thống còn có xu hướng gợi ý lặp lại hoặc quá chuyên sâu, làm giảm tính đa dạng của kết quả đề xuất.

2.1.3 Hybrid Recommender System

Hybrid Recommender System (Hệ thống khuyến nghị kết hợp) được xây dựng nhằm khắc phục những hạn chế của từng phương pháp riêng lẻ bằng cách tích hợp nhiều chiến lược gợi ý khác nhau, phổ biến nhất là kết hợp giữa gợi ý dựa trên nội dung và lọc cộng tác. Ngoài ra, một số hệ thống còn sử dụng thông tin tri thức do người dùng cung cấp như sở thích hoặc điều kiện lựa chọn sản phẩm.

Việc kết hợp các nguồn dữ liệu và kỹ thuật khác nhau giúp hệ thống trở nên linh hoạt hơn, tăng độ chính xác và cá nhân hóa các đề xuất hiệu quả hơn. Đặc biệt, hybrid system có thể xử lý tốt hơn các vấn đề cold start và mang lại khả năng tùy biến theo ngữ cảnh sử dụng.

Tuy nhiên, đi kèm với lợi ích đó là sự phức tạp trong thiết kế, triển khai và bảo trì. Các mô hình kết hợp thường đòi hỏi lượng dữ liệu lớn, chi phí phát triển cao và cần sự tinh chỉnh kỹ lưỡng để đạt được hiệu quả tối ưu. Thêm vào đó, việc giải thích kết quả đề xuất cũng trở nên khó khăn hơn khi nhiều thành phần hoạt động song song trong hệ thống.

2.1.4 Tổng quan về các hệ thống khuyến nghị trên

Các phương pháp khuyến nghị phổ biến hiện nay đều có những ưu điểm và hạn chế riêng, phụ thuộc vào đặc điểm dữ liệu và mục tiêu hệ thống. Phương pháp gợi ý dựa trên nội dung (Content-based Filtering) mang lại khả năng cá nhân hóa tốt khi dựa vào đặc trưng của sản phẩm mà người dùng từng tương tác. Nhờ không phụ thuộc vào dữ liệu cộng đồng, phương pháp này tỏ ra phù hợp trong bối cảnh dữ liệu người dùng hạn chế hoặc đặc trưng sản phẩm rõ ràng. Tuy nhiên, nó dễ rơi vào tình trạng chỉ đề xuất những nội dung tương tự, làm giảm khả năng khám phá các sản phẩm mới.

Trong khi đó, gợi ý dựa trên cộng tác (Collaborative Filtering) tận dụng hành vi của tập người dùng để phát hiện các mẫu sở thích chung, từ đó mở rộng phạm vi đề xuất và khám phá hiệu quả hơn. Phương pháp này đặc biệt hữu ích trong việc khai thác xu hướng cộng đồng. Tuy nhiên, nó đòi hỏi dữ liệu tương tác phong phú và thường gặp khó khăn trong các tình huống cold start, khi người dùng hoặc sản phẩm còn mới và thiếu dữ liệu.

Để khắc phục các nhược điểm trên, hệ thống khuyến nghị kết hợp (Hybrid Recommender Systems) được phát triển nhằm tích hợp nhiều chiến lược khác nhau. Hệ thống này vừa đảm bảo khả năng cá nhân hóa, vừa nâng cao hiệu quả khám phá sản phẩm và tận dụng được xu hướng cộng đồng. Mặc dù vậy, tính hiệu quả của hybrid system đi kèm với yêu cầu thiết kế phức tạp hơn và đòi hỏi dữ liệu phong phú, đa dạng từ nhiều nguồn khác nhau.

Bảng 2.1 So sánh Content-based Filtering, Collaborative Filtering và Hybrid Recommender System

Tiêu chí	Content-based Filtering	Collaborative Filtering	Hybrid Recommender System
Dữ liệu đầu vào	Đặc trưng của sản phẩm (nội dung, metadata).	Hành vi của người dùng (đánh giá, tương tác).	Kết hợp cả đặc trưng sản phẩm và hành vi người dùng.
Nguyên lý hoạt động	Đề xuất sản phẩm có đặc điểm tương tự với những gì người dùng đã yêu thích.	Đề xuất dựa trên sự tương đồng giữa người dùng hoặc sản phẩm.	Tích hợp nhiều chiến lược đề xuất để tối ưu hiệu quả.
Ưu điểm	Cá nhân hóa cao, hoạt động tốt với người dùng mới, dễ giải thích lý do gợi ý.	Tận dụng xu hướng cộng đồng, khám phá sản phẩm mới hiệu quả, không cần phân tích nội dung.	Kết hợp ưu điểm của hai phương pháp, độ chính xác cao, khắc phục cold-start tốt.

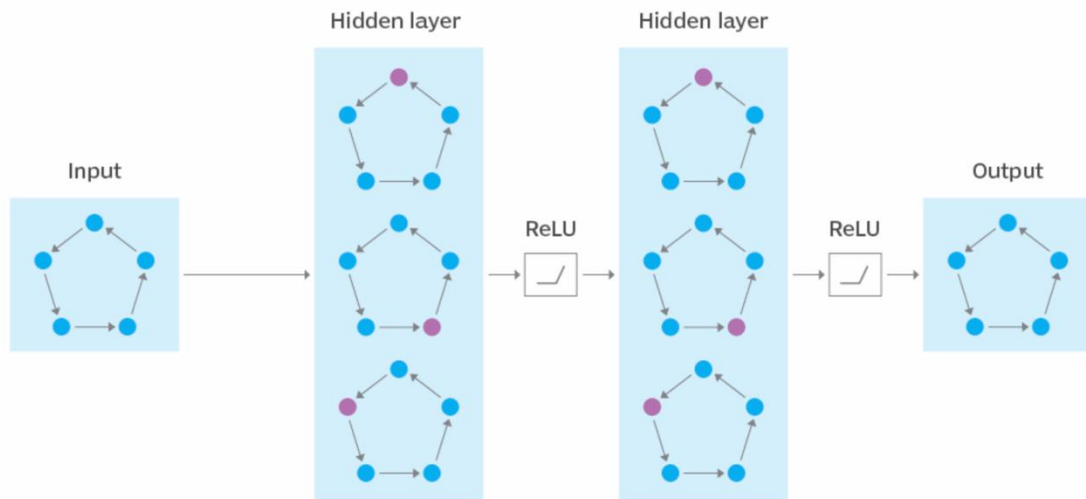
Nhược điểm	Bị giới hạn trong phạm vi sở thích cá nhân, khó khám phá sản phẩm mới, không xử lý tốt cold-start item.	Gặp khó khăn với cold-start, dữ liệu thừa thớt làm giảm hiệu quả, độ phức tạp tính toán tăng theo quy mô.	Thiết kế phức tạp, yêu cầu tài nguyên tính toán cao, cần chiến lược kết hợp hiệu quả.
Giải quyết cold-start	Chỉ giải quyết tốt với cold-start user.	Không giải quyết tốt cold-start user hoặc cold-start item.	Giải quyết tốt cả hai vấn đề cold-start user và cold-start item.
Khả năng khám phá sản phẩm mới	Thấp, vì chỉ gợi ý sản phẩm tương tự với sở thích trước đó.	Cao, vì tận dụng được xu hướng chung của cộng đồng.	Cao nhất, kết hợp cả hai cách tiếp cận để mở rộng khả năng gợi ý.

Nhìn chung, khi hệ thống làm việc với dữ liệu hạn chế nhưng đặc trưng sản phẩm rõ ràng và yêu cầu mức độ cá nhân hóa cao, Content-based Filtering là lựa chọn phù hợp. Ngược lại, trong các tình huống có sẵn dữ liệu phong phú từ hành vi người dùng và mục tiêu là khám phá sản phẩm mới một cách hiệu quả, Collaborative Filtering sẽ phát huy thế mạnh.

Đối với các nền tảng quy mô lớn như Amazon, Netflix hay Spotify, nơi đòi hỏi độ chính xác cao, khả năng cá nhân hóa sâu và tận dụng tối đa dữ liệu từ cả người dùng lẫn sản phẩm, thì Hybrid Recommender System là lựa chọn ưu việt. Phương pháp này không chỉ dung hòa các ưu điểm của hai hướng tiếp cận truyền thống mà còn cho phép mở rộng linh hoạt theo đặc thù từng ứng dụng cụ thể.

2.2 Tổng quan về Graph Neural Networks (GNN)

The structure of a graph neural network



(Nguồn: *Definition of Graph Neural Networks (GNNs)*. Adapted from *Graph Neural Networks (GNNs)* (n.d.), TechTarget. Retrieved July 20, 2025)

Hình 2.1 Tổng quan về GNN

Graph Neural Networks (GNN) là một họ các mô hình học sâu được thiết kế để xử lý dữ liệu có cấu trúc dạng đồ thị – trong đó các thực thể (node) được kết nối thông qua các quan hệ (cạnh). Không giống như dữ liệu Euclidean (ví dụ như chuỗi hay ảnh), dữ liệu đồ thị có tính không gian phi Euclidean, đòi hỏi các kiến trúc đặc thù để khai thác thông tin hiệu quả.

Một trong những mô hình tiêu biểu trong GNN là Graph Convolutional Network (GCN), được giới thiệu bởi Kipf & Welling (2017). GCN học biểu diễn (embedding) cho mỗi node bằng cách tổng hợp đặc trưng từ các node láng giềng thông qua phép tích

chập trên đồ thị. Cụ thể, biểu thức lan truyền đặc trưng tại lớp $l+1+1+1$ được định nghĩa như sau:

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (2.1)$$

Trong đó:

- $\tilde{A} = A + I$ là ma trận kề đã được thêm kết nối tự thân (self-loop).
- \tilde{D} là ma trận bậc của \tilde{A} .
- $H^{(l)}$ là ma trận biểu diễn tại lớp l (với $H^{(0)}$ là đầu vào ban đầu).
- $W^{(l)}$ là trọng số cần học.
- σ là hàm kích hoạt phi tuyến.

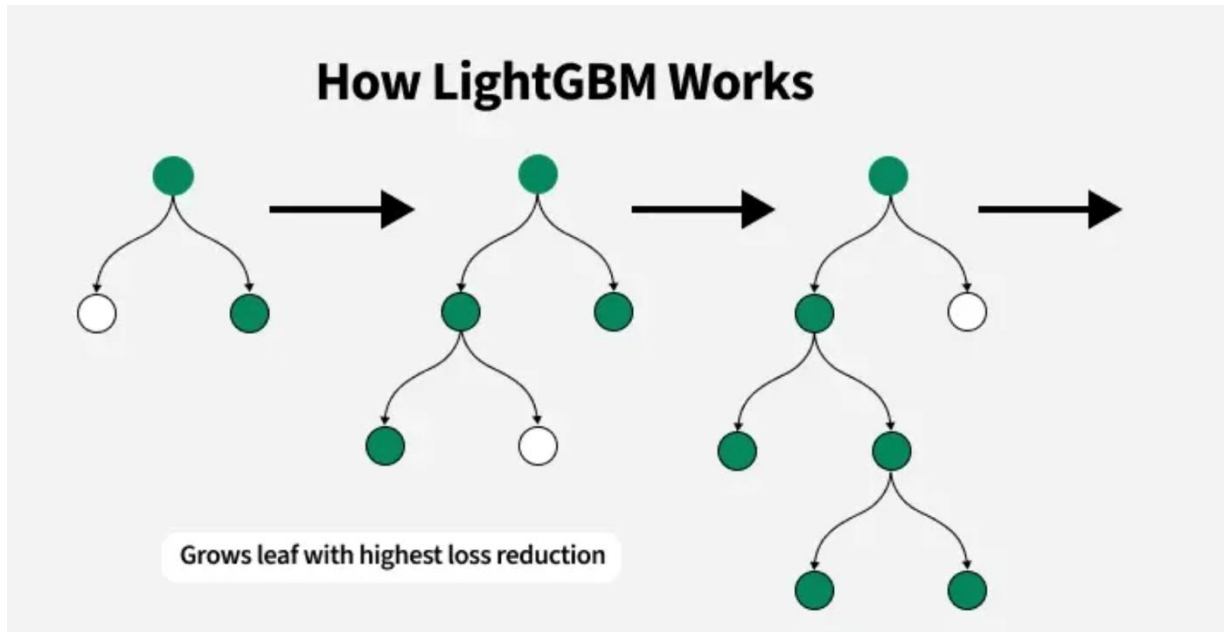
Ưu điểm nổi bật của GCN nằm ở khả năng khai thác mối quan hệ giữa các thực thể trong đồ thị thông qua lan truyền thông tin giữa các node. Tuy nhiên, GCN cũng tồn tại nhiều hạn chế:

- Dễ gặp hiện tượng over-smoothing khi mạng quá sâu, khiến các node trở nên đồng nhất,
- Không phân biệt được tầm quan trọng khác nhau của các node láng giềng trong quá trình lan truyền,
- Trường tiếp nhận mở rộng chậm do lan truyền thông tin đồng đều.

Chính vì những hạn chế này, các kiến trúc mới như Graph Attention Network (GAT), GraphSAGE, và đặc biệt là Transformer trên đồ thị (TransGNN) đã được đề xuất nhằm khắc phục các nhược điểm của GCN. Trong đó, cơ chế attention đóng vai trò quan trọng trong việc gán trọng số khác nhau cho từng node láng giềng, đồng thời cho phép

mô hình học được mối quan hệ phức tạp hơn – đây cũng là nền tảng cho hướng tiếp cận trong đề tài.

2.3 Tổng quan về LightGBM



(Nguồn: *LightGBM leaf-wise tree growth illustration. Adapted from GeeksforGeeks (2025), LightGBM (Light Gradient Boosting Machine). Retrieved July 20, 2025*)

Hình 2.2 Nguyên lý hoạt động của LightGBM

LightGBM (Light Gradient Boosting Machine) là một thuật toán boosting được phát triển bởi Microsoft, thuộc nhóm các mô hình tăng cường gradient (Gradient Boosting Decision Tree – GBDT). Khác với các thuật toán boosting truyền thống như XGBoost hay AdaBoost, LightGBM được thiết kế tối ưu hóa cho tốc độ huấn luyện, khả năng mở rộng trên tập dữ liệu lớn và tiêu thụ bộ nhớ thấp (Ke et al., 2017).

2.3.1 Nguyên lý hoạt động

LightGBM xây dựng một tập hợp các cây quyết định theo nguyên tắc boosting, tức là mỗi cây mới được huấn luyện nhằm sửa lỗi của cây trước đó. Trong quá trình xây dựng cây, LightGBM sử dụng các kỹ thuật chính để cải thiện hiệu suất như:

- Leaf-wise growth: Thay vì mở rộng cây theo từng tầng (level-wise), LightGBM chọn mở rộng lá có mức giảm lỗi lớn nhất. Điều này giúp mô hình đạt được độ chính xác cao hơn, mặc dù có thể gây overfitting nếu không kiểm soát độ sâu.
- Gradient-based One-Side Sampling (GOSS): LightGBM giữ lại các mẫu có gradient lớn (biểu hiện độ lỗi cao), và chỉ lấy ngẫu nhiên một phần mẫu có gradient nhỏ để tiết kiệm tài nguyên nhưng vẫn đảm bảo độ chính xác.
- Exclusive Feature Bundling (EFB): Các đặc trưng rời rạc không đồng thời xuất hiện được gom nhóm lại thành một đặc trưng duy nhất để giảm chiều dữ liệu đầu vào, tiết kiệm bộ nhớ.

2.3.2 Ưu điểm của *LightGBM*

Tốc độ huấn luyện nhanh và tiết kiệm bộ nhớ: LightGBM sử dụng phương pháp học dựa trên histogram (Histogram-based Learning), trong đó các giá trị đầu vào liên tục được gộp thành các khoảng (bin) rời rạc trước khi xây dựng cây. Kỹ thuật này giúp giảm đáng kể độ phức tạp tính toán, đồng thời tiết kiệm bộ nhớ so với việc sử dụng giá trị thực (float) trực tiếp. Ngoài ra, LightGBM còn hỗ trợ huấn luyện song song (parallel learning) và tăng tốc trên GPU, giúp cải thiện đáng kể thời gian huấn luyện cho các tập dữ liệu lớn.

Khả năng xử lý hiệu quả dữ liệu lớn và dữ liệu thưa (sparse): Trong nhiều bài toán thực tế, đặc biệt là bài toán khuyến nghị, dữ liệu thường có dạng ma trận thưa – ví dụ

như ma trận tương tác người dùng – sản phẩm. LightGBM có khả năng xử lý trực tiếp các giá trị thiếu hoặc đặc trưng rời rạc với tần suất thấp mà không cần tiền xử lý phức tạp, nhờ vào cơ chế xử lý dữ liệu rời rạc và kỹ thuật “exclusive feature bundling” để giảm số chiều của dữ liệu đầu vào.

Hiệu quả cao trong các bài toán phân loại, hồi quy và xếp hạng: LightGBM hỗ trợ nhiều mục tiêu học khác nhau như hồi quy (regression), phân loại nhị phân hoặc đa lớp (binary/multiclass classification), và đặc biệt là xếp hạng (ranking). Đối với bài toán khuyến nghị, mô hình có thể sử dụng hàm mất mát LambdaRank hoặc pairwise loss để học thứ tự ưu tiên giữa các mục tiêu, giúp tăng độ chính xác trong việc đề xuất top-k sản phẩm phù hợp.

Dễ dàng tích hợp vào pipeline học sâu có nhiều đặc trưng phi tuyến: LightGBM có thể được tích hợp linh hoạt vào giai đoạn cuối của một hệ thống học sâu, đóng vai trò là bộ xếp hạng dựa trên các đặc trưng đầu vào phong phú. Các đặc trưng này có thể đến từ nhiều nguồn khác nhau như embedding của ảnh (CNN), biểu diễn từ đồ thị (GNN) hoặc các hành vi người dùng đã được mã hóa bằng mạng nơ-ron sâu. Nhờ khả năng học phi tuyến mạnh mẽ và hỗ trợ đa dạng các kiểu đặc trưng, LightGBM hoạt động hiệu quả trong vai trò hậu xử lý (re-ranking) trong hệ thống khuyến nghị.

2.3.3 Ứng dụng trong hệ thống khuyến nghị

Trong hệ thống khuyến nghị, LightGBM thường được sử dụng ở giai đoạn cuối cùng (ranking layer) để xếp hạng các sản phẩm dựa trên các đặc trưng được trích xuất từ hành vi người dùng, đặc trưng sản phẩm (embedding), hoặc các chỉ số thống kê. Ví dụ:

- Tích hợp đầu ra từ các mô hình GNN, Transformer hoặc CNN.
- Xử lý tập dữ liệu lớn, nơi đặc trưng không tuyến tính rất đa dạng.
- Huấn luyện trên nhãn dạng implicit feedback (như lượt xem, mua hàng).

Nhờ khả năng học các quan hệ phi tuyến phức tạp và hiệu quả xử lý cao, LightGBM đã trở thành thành phần quan trọng trong nhiều hệ thống gợi ý hiện đại, đặc biệt trong thương mại điện tử, tìm kiếm sản phẩm và cá nhân hóa trải nghiệm người dùng.

2.4 Transformer trong biểu diễn đồ thị

Transformer (Vaswani et al. 2017) là một kiến trúc mạng nơ-ron sâu nhằm thay thế các mô hình tuần tự truyền thống như RNN hay CNN trong việc xử lý chuỗi. Điểm nổi bật của Transformer là cơ chế attention, cho phép mô hình học được mối quan hệ giữa các phần tử đầu vào mà không phụ thuộc vào vị trí kế cận, giúp tăng khả năng biểu diễn ngữ cảnh dài và giảm độ phức tạp khi xử lý song song.

2.4.1 Cơ chế attention trong Transformer

Cơ chế attention trong Transformer nhận vào ba thành phần chính: truy vấn (Q), khóa (K) và giá trị (V). Trọng số attention thể hiện mức độ liên quan giữa truy vấn và các phần tử khác trong tập đầu vào. Attention được tính bằng công thức:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (2.2)$$

Trong đó d là kích thước không gian của các vector truy vấn và khóa. Hàm *softmax* đảm bảo các trọng số được chuẩn hóa về tổng bằng 1.

2.4.2 Khả năng mở rộng Transformer lên dữ liệu đồ thị

Ban đầu được phát triển cho dữ liệu chuỗi, Transformer đã chứng minh khả năng vượt trội trong các lĩnh vực như xử lý ngôn ngữ tự nhiên (NLP) và thị giác máy tính (CV). Gần đây, các nghiên cứu đã mở rộng Transformer để áp dụng trên dữ liệu đồ thị phi Euclidean bằng cách xem mỗi node như một token, và mối quan hệ giữa node được

học thông qua attention thay cho lan truyền thông điệp (message passing) truyền thống trong GNN.

So với các mô hình GCN hoặc GAT, Transformer trên đồ thị có nhiều ưu điểm:

- Mở rộng trường tiếp nhận (receptive field) theo cơ chế attention toàn cục, không bị giới hạn bởi số lớp mạng.
- Trọng số attention linh hoạt giúp mô hình phân biệt các node láng giềng quan trọng, thay vì lan truyền đồng đều như GCN.
- Dễ dàng tích hợp với các tín hiệu cấu trúc như Positional Encoding hoặc ma trận kề.

2.4.3 Một số biến thể Transformer cho dữ liệu đồ thị

Việc mở rộng Transformer cho dữ liệu đồ thị đã dẫn đến sự ra đời của nhiều mô hình chuyên biệt:

- Graphormer: Sử dụng attention toàn cục giữa các node, kết hợp các biểu diễn cấu trúc như khoảng cách ngắn nhất và độ trung tâm để cải thiện positional encoding.
- GTN (Graph Transformer Network): Tận dụng multi-head attention để học trực tiếp mối quan hệ giữa các node mà không cần lan truyền truyền thống.
- TransGNN: Là mô hình kết hợp giữa Transformer và GCN, tận dụng attention để lựa chọn các node láng giềng có ảnh hưởng mạnh, đồng thời vẫn giữ khả năng khai thác cấu trúc đồ thị qua các tầng lan truyền.

2.4.4 Ưu điểm khi áp dụng trong hệ thống khuyến nghị

Khi được triển khai trong các hệ thống khuyến nghị, kiến trúc Transformer hoạt động trên dữ liệu đồ thị thể hiện nhiều lợi thế vượt trội. Trước hết, nhờ cơ chế attention

toàn cục, Transformer có khả năng nắm bắt được mối quan hệ phức tạp giữa người dùng và sản phẩm, không chỉ ở các kết nối trực tiếp mà còn ở các tầng tương tác sâu hơn. Điều này cho phép mô hình khai thác hiệu quả hơn các tín hiệu tiềm ẩn trong hành vi người dùng, từ đó nâng cao khả năng cá nhân hóa trong việc đề xuất sản phẩm.

Ngoài ra, Transformer còn có khả năng tập trung vào các node thực sự liên quan thông qua attention có điều hướng. Thay vì lan truyền thông tin một cách đồng đều từ toàn bộ lân cận như trong GCN truyền thống, Transformer có thể chọn lọc và gán trọng số phù hợp cho từng node, giúp giảm nhiễu từ các tương tác không cần thiết và làm tăng độ chính xác của mô hình.

Một điểm mạnh khác là tính linh hoạt trong việc tích hợp các loại đặc trưng đa dạng. Mô hình không chỉ xử lý embedding từ dữ liệu hành vi (như lịch sử tương tác giữa người dùng và sản phẩm), mà còn có thể kết hợp thêm đặc trưng hình ảnh được trích xuất bằng CNN và thông tin bổ trợ từ metadata như danh mục, thương hiệu hoặc mùa vụ. Khả năng xử lý đa dạng dữ liệu này góp phần nâng cao chất lượng khuyến nghị trong các hệ thống thực tế.

2.5 Dự đoán tương tác và tối ưu hóa mô hình với BPR Loss

Sau khi hoàn tất các bước lan truyền thông tin trong mạng đồ thị, mô hình sinh ra embedding cuối cùng cho các node người dùng và sản phẩm. Các embedding này sẽ được sử dụng để dự đoán mức độ tương tác giữa người dùng và sản phẩm, từ đó hỗ trợ quá trình xếp hạng.

2.5.1 Hàm dự đoán tương tác

Tương tự như mô hình Matrix Factorization trong BPRMF (Bayesian Personalized Ranking Matrix Factorization), điểm tương tác giữa người dùng u và

sản phẩm i được tính thông qua tích vô hướng giữa vector embedding của người dùng và sản phẩm:

$$\widehat{y_{ui}} = e_u^\top e_i \quad (2.3)$$

Trong đó:

- $e_u \in R^d$ là vector embedding cuối cùng của người dùng u .
- $e_i \in R^d$ là vector embedding cuối cùng của sản phẩm i .
- $\widehat{y_{ui}}$ biểu thị điểm dự đoán mức độ phù hợp giữa người dùng và sản phẩm.

2.5.2 Hàm mất mát BPR

Để tối ưu hóa mô hình theo hướng học thứ tự ưu tiên (pairwise ranking), chúng tôi áp dụng hàm mất mát Bayesian Personalized Ranking (BPR) – một phương pháp hiệu quả trong các hệ thống gợi ý sử dụng dữ liệu tương tác ngầm.

Giả sử tại thời điểm $t + 1$, người dùng u tương tác với một sản phẩm thực tế i_{t+1} , gọi là positive item). Một sản phẩm không được người dùng tương tác ký hiệu là i_{t+1}^- được lấy ngẫu nhiên từ tập các sản phẩm chưa tương tác, đóng vai trò là negative item.

Mô hình được huấn luyện để đảm bảo rằng điểm tương tác của sản phẩm thực tế luôn cao hơn sản phẩm không tương tác. Hàm mất mát BPR được định nghĩa như sau:

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \sum_{t=1}^n \log \sigma (P(i_{t+1}) - P(i_{t+1}^-)) \quad (2.4)$$

Trong đó:

- \mathcal{U} : tập hợp người dùng.

- i_{t+1} : sản phẩm thực tế mà người dùng tương tác tại thời điểm $t + 1$ (positive item).
- i_{t+1}^- : sản phẩm không tương tác được chọn ngẫu nhiên (negative item).
- $P(i) = \widehat{y_{ui}}$: điểm dự đoán giữa người dùng u và item i .
- $\sigma(\cdot)$: hàm sigmoid để tính xác suất.

Hàm mất mát này khuyến khích mô hình học được thứ tự ưu tiên giữa các sản phẩm, thay vì dự đoán giá trị tuyệt đối như trong các mô hình rating truyền thống. Điều này đặc biệt phù hợp với các bài toán gợi ý thời trang, nơi mà hành vi tương tác ngầm (như xem, nhấp, mua) đóng vai trò chính trong đánh giá mức độ yêu thích của người dùng đối với sản phẩm.

2.6 Các độ đo đánh giá hiệu suất mô hình

Recall đo lường tỷ lệ các mục tiêu liên quan được truy xuất thành công trong danh sách gợi ý, được tính bằng công thức:

$$Recall = \frac{\text{Số lượng mục tiêu liên quan được truy xuất}}{\text{Tổng số mục tiêu liên quan}} \quad (2.5)$$

NDCG (Normalized Discounted Cumulative Gain) đánh giá chất lượng thứ hạng của các mục tiêu liên quan, trong đó các mục tiêu xuất hiện ở đầu danh sách sẽ được gán trọng số cao hơn. Công thức tính NDCG:

$$NDCG = \frac{1}{Q} \sum_{q=1}^Q \frac{DCG_q}{IDCG_q} \quad (2.6)$$

Trong đó DCG_q là độ lợi tích lũy có chiết khấu cho truy vấn q , còn $IDCG_q$ là giá trị DCG lý tưởng.

MRR (Mean Reciprocal Rank) phản ánh vị trí của mục tiêu liên quan đầu tiên xuất hiện trong danh sách gợi ý, thông qua giá trị trung bình nghịch đảo của thứ hạng:

$$MRR = \frac{1}{Q} \sum_{q=1}^Q \frac{1}{rank_q} \quad (2.7)$$

Hit Rate đo lường tỷ lệ các truy vấn mà trong đó có ít nhất một mục tiêu liên quan xuất hiện trong top-k:

$$Hit Rate = \frac{\text{Số lượng truy vấn có kết quả đúng}}{\text{Tổng số truy vấn}} \quad (2.8)$$

AUC (Area Under the ROC Curve) đo lường xác suất mô hình xếp hạng một mục tiêu liên quan cao hơn một mục tiêu không liên quan, được tính bằng công thức:

$$AUC = \frac{\text{Số lần xếp đúng (positive > negative)}}{\text{Tổng số cặp (positive, negative)}} \quad (2.9)$$

MAP (Mean Average Precision) đo trung bình độ chính xác tại các vị trí mà mục tiêu liên quan xuất hiện trong danh sách gợi ý, được tính bằng công thức:

$$MAP = \frac{1}{Q} \sum_{q=1}^Q \frac{1}{|R_q|} \sum_{k=1}^K Precision@k \cdot I(r_k \in R_q) \quad (2.10)$$

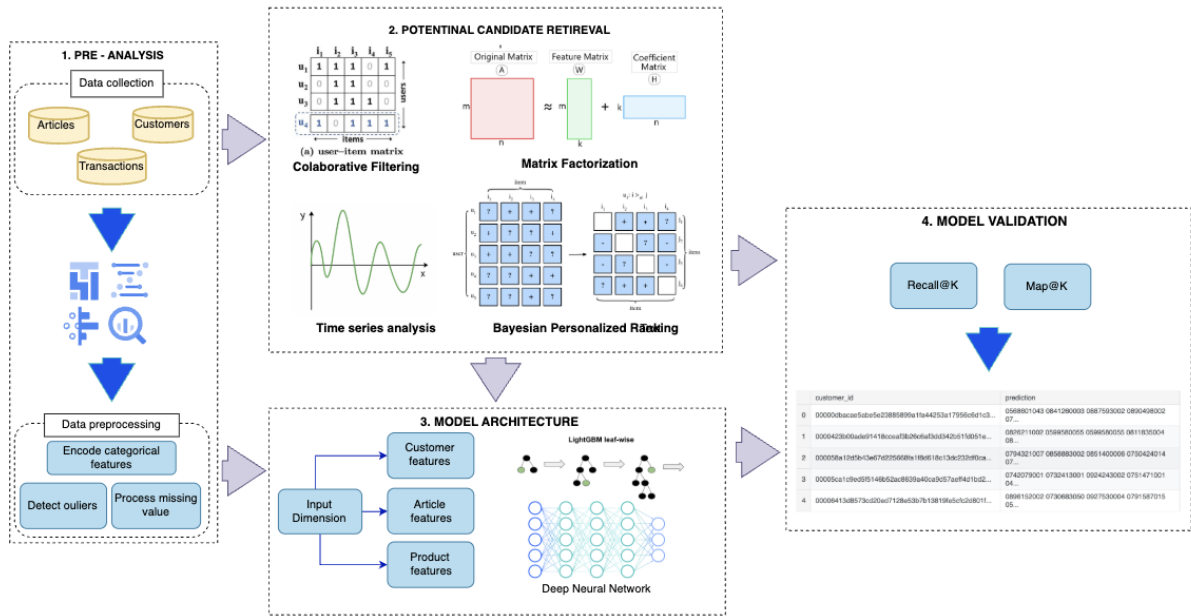
Trong đó:

- Q : số lượng truy vấn (người dùng).
- R_q : tập mục tiêu liên quan cho truy vấn q .
- r_k : mục tiêu tại vị trí thứ kkk trong danh sách gợi ý.
- $I(\cdot)$: hàm chỉ thị (1 nếu đúng, 0 nếu sai).
- $Precision@k$: Độ chính xác tại vị trí k.

CHƯƠNG 3 - KIẾN TRÚC MÔ HÌNH ĐỀ XUẤT VÀ PHƯƠNG PHÁP NGHIÊN CỨU

3.1 Light Gradient Boosting Machine (LightGBM)

LightGBM (Light Gradient Boosting Machine) là một thuật toán tăng cường gradient (gradient boosting) hiện đại, được thiết kế nhằm tối ưu hiệu năng xử lý và thời gian huấn luyện trên các tập dữ liệu lớn. Khác với các mô hình truyền thống, LightGBM ứng dụng một số kỹ thuật đặc thù để tăng tốc độ huấn luyện mà vẫn đảm bảo độ chính xác cao.



Hình 3.1 Quy trình làm việc của mô hình đề xuất

3.1.1 Thuật toán dựa trên histogram

LightGBM sử dụng phương pháp phân nhóm theo histogram để xử lý các đặc trưng có giá trị liên tục. Cụ thể, các giá trị liên tục được ánh xạ thành các khoảng rời rạc

(bins), từ đó giúp giảm số lượng phép tính cần thực hiện trong quá trình xây dựng cây quyết định. Nhờ đó:

- Thời gian huấn luyện được rút ngắn đáng kể.
- Bộ nhớ sử dụng được tiết kiệm hơn.
- Hiệu suất mô hình được cải thiện rõ rệt trên tập dữ liệu lớn.

3.1.2 Chiến lược tăng trưởng cây: *Leaf-wise* thay vì *Level-wise*

Trong quá trình xây dựng cây quyết định, LightGBM áp dụng chiến lược phát triển theo lá (leaf-wise growth) thay vì theo mức (level-wise) như các thuật toán boosting truyền thống.

- Level-wise: Cây được phát triển đồng đều từ gốc đến các cấp độ tiếp theo. Chiến lược này giúp cây cân bằng nhưng tốc độ học chậm hơn.
- Leaf-wise: Cây phát triển từ lá có độ giảm lỗi lớn nhất (largest loss reduction), giúp giảm lỗi nhanh và cải thiện độ chính xác. Tuy nhiên, chiến lược này có thể dẫn đến overfitting nếu áp dụng trên tập dữ liệu nhỏ.

Nhờ áp dụng leaf-wise, LightGBM có tốc độ học nhanh hơn và thường đạt độ chính xác cao hơn, nhưng đi kèm với yêu cầu kiểm soát chặt chẽ các siêu tham số như `max_depth`, `num_leaves` để tránh quá khớp.

3.1.3 Kỹ thuật *GOSS* và *EFB* đặc trưng

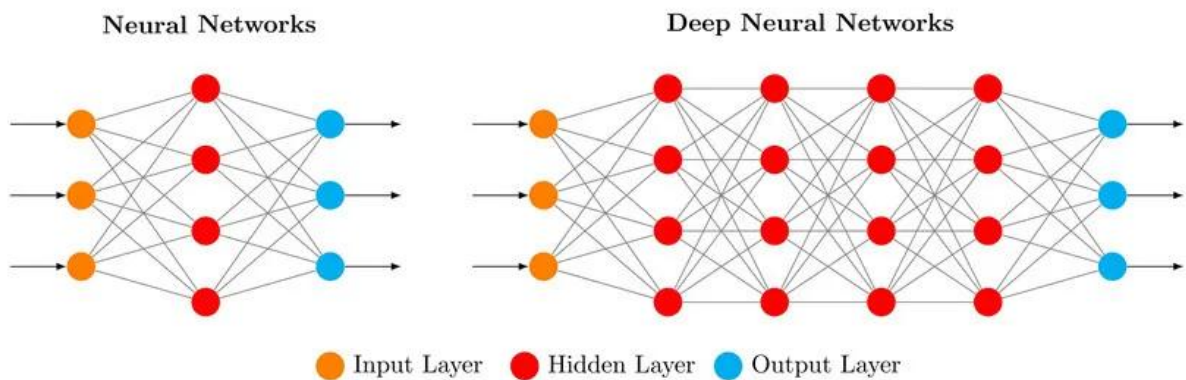
LightGBM giới thiệu hai kỹ thuật nổi bật nhằm tối ưu hóa cả về kích thước dữ liệu và số lượng đặc trưng, bao gồm Gradient-based One-Side Sampling (GOSS) và Exclusive Feature Bundling (EFB).

GOSS là một kỹ thuật chọn mẫu có trọng số, cho phép giảm số lượng dữ liệu huấn luyện mà vẫn giữ được phân phối gradient quan trọng. Thay vì sử dụng toàn bộ tập dữ

liệu, GOSS giữ lại những mẫu có gradient lớn – tức là những mẫu đóng vai trò quan trọng trong việc cập nhật mô hình – và chỉ lấy ngẫu nhiên một phần từ các mẫu còn lại. Nhờ đó, thời gian tính toán được tiết kiệm đáng kể mà không làm suy giảm hiệu quả huấn luyện.

Trong khi đó, EFB là một phương pháp giảm số lượng đặc trưng bằng cách nhóm các đặc trưng phân tán (sparse features) nhưng không tương quan vào cùng một nhóm (bundle). Việc gom nhóm này giúp giảm đáng kể chiều dữ liệu đầu vào, từ đó rút ngắn thời gian huấn luyện, tiết kiệm bộ nhớ và giảm nhiều không cần thiết. Quan trọng hơn, kỹ thuật này vẫn đảm bảo giữ được độ chính xác của mô hình, giúp LightGBM hoạt động hiệu quả hơn trên tập dữ liệu lớn và nhiều chiều.

3.2 Deep Neural Network (DNN)



(Nguồn: *Deep neural network architecture. Adapted from Zemin (2021), Deep Neural Network (DNN) explained, Medium. Retrieved July 20, 2025*)

Hình 3.2 So sánh kiến trúc giữa Neural Networks và Deep Neural Networks

DNN là mạng nơ-ron nhiều lớp (deep multilayer perceptron), có khả năng học các mối quan hệ phức tạp phi tuyến giữa các đặc trưng đầu vào và kết quả dự đoán. Deep Neural Network (DNN) là một trong những mô hình học sâu được ứng dụng phổ biến

trong lĩnh vực hệ thống khuyến nghị nhờ khả năng học các mối quan hệ phức tạp giữa khách hàng và sản phẩm.

Trong nghiên cứu này, DNN nhận nhiều nhóm đặc trưng đầu vào như:

- Customer features (đặc trưng khách hàng: tuổi, giới tính, lịch sử mua, ...).
- Article features (đặc trưng sản phẩm: loại sản phẩm, thương hiệu, màu sắc, ...).
- Product features (có thể là các đặc trưng kỹ thuật khác hoặc đặc trưng thêm về sản phẩm).

3.2.1 Kiến trúc của DNN

Mô hình DNN trong hệ thống khuyến nghị được xây dựng dựa trên kiến trúc nhiều lớp (multilayer), bao gồm ba thành phần chính: tầng đầu vào (input layer), các tầng ẩn (hidden layers), và tầng đầu ra (output layer).

Ở tầng đầu vào, các nhóm đặc trưng khác nhau như đặc trưng người dùng (customer features), đặc trưng sản phẩm (article features), và các đặc trưng bổ sung liên quan đến sản phẩm (product features) sẽ được mã hóa và chuẩn hóa thành các vector số. Sau đó, các vector này được kết hợp lại để hình thành một vector biểu diễn duy nhất đại diện cho cặp người dùng – sản phẩm, làm đầu vào cho mạng nơ-ron.

Tiếp theo là các tầng ẩn, bao gồm nhiều lớp liên tiếp với số lượng lớp và số lượng neuron trong mỗi lớp được điều chỉnh linh hoạt theo bài toán (ví dụ từ 3 đến 6 lớp, với mỗi lớp có thể gồm 64, 128 hoặc 256 neuron). Tại mỗi tầng, đầu vào từ tầng trước sẽ được nhân với trọng số, cộng với hệ số điều chỉnh (bias), sau đó đưa qua hàm kích hoạt phi tuyến như ReLU, sigmoid hoặc tanh. Mục tiêu của các tầng ẩn là trích xuất các đặc trưng trừu tượng, đồng thời học được các mối quan hệ phi tuyến phức tạp giữa người

dùng và sản phẩm. Để giảm thiểu hiện tượng quá khớp (overfitting), mô hình có thể tích hợp các kỹ thuật như Dropout hoặc Batch Normalization.

Cuối cùng, tầng đầu ra sẽ trả về kết quả dự đoán dưới dạng xác suất hoặc điểm số. Trong các bài toán phân loại nhị phân (ví dụ: dự đoán khả năng mua hàng), hàm kích hoạt sigmoid thường được sử dụng. Đối với các bài toán xếp hạng hoặc phân loại nhiều lớp, hàm softmax sẽ phù hợp hơn để ánh xạ kết quả về dạng phân phối xác suất.

3.2.2 Quy trình huấn luyện

Quá trình huấn luyện của mô hình DNN bắt đầu từ việc chuẩn bị dữ liệu đầu vào dưới dạng các vector đặc trưng đã được mã hóa và chuẩn hóa. Các vector này được ghép đôi với nhãn đầu ra tương ứng, chẳng hạn như nhãn nhị phân biểu thị hành vi có hoặc không mua hàng của người dùng đối với một sản phẩm cụ thể.

Trong quá trình huấn luyện, mô hình sử dụng hàm mất mát (loss function) để đo lường sai số giữa dự đoán của mô hình và giá trị thực tế. Đối với các bài toán phân loại nhị phân, hàm Binary Cross-Entropy thường được sử dụng. Trong khi đó, nếu mục tiêu là tối ưu hóa thứ hạng (ranking), các hàm mất mát như pairwise loss hoặc ranking loss sẽ phù hợp hơn.

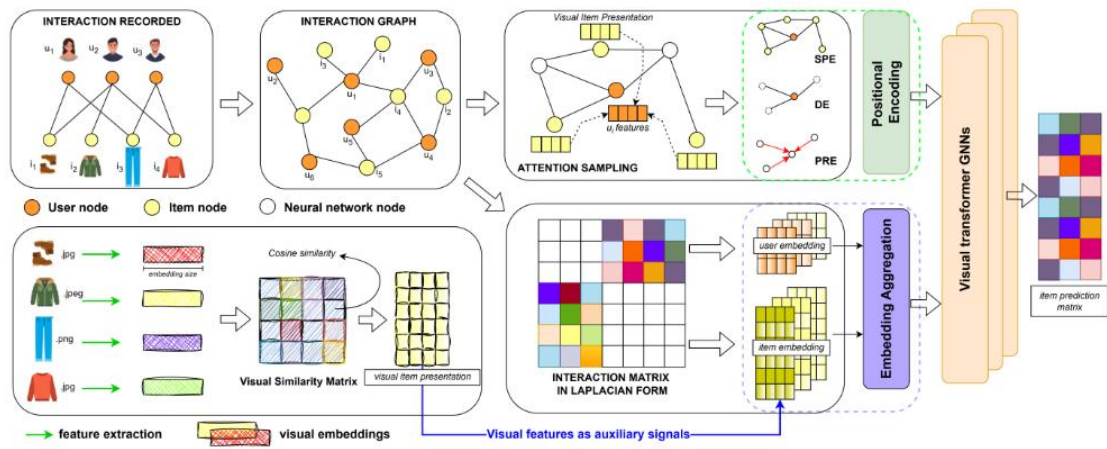
Tối ưu hóa mô hình được thực hiện thông qua thuật toán Gradient Descent hoặc các biến thể nâng cao như Adam, RMSProp,... Những thuật toán này giúp cập nhật trọng số của mạng nơ-ron theo hướng làm giảm sai số của mô hình sau mỗi vòng huấn luyện (epoch). Mỗi lần cập nhật, mô hình dần học được các mối quan hệ phức tạp giữa đặc trưng đầu vào và hành vi người dùng, từ đó cải thiện độ chính xác của dự đoán.

3.3 Visual-TransGNN (Visual Transformer Graph Neural Network)

Mô hình Visual-TransGNN là một kiến trúc kết hợp (hybrid framework) tích hợp tín hiệu tương đồng hình ảnh (visual similarity) và cấu trúc đồ thị (graph-based structural

signals) bằng cách sử dụng Mạng Nơ-ron Đồ thị có tích hợp Transformer (Transformer-enhanced Graph Neural Network). Mô hình được thiết kế nhằm học đồng thời vector đặc trưng cho người dùng và sản phẩm, sao cho thể hiện được cả hành vi tương tác lẫn mức độ phù hợp về mặt thị giác.

3.3.1 Tổng quan kiến trúc Visual-TransGNN



Hình 3.3 Tổng quan kiến trúc Visual-TransGNN

Tổng quan về kiến trúc được đề xuất được minh họa trong Hình 3.3. Mô hình hoạt động dựa trên bốn thành phần đầu vào chính như sau:

- Đồ thị tương tác người dùng – sản phẩm (A): Đây là một ma trận kề hai phía giữa người dùng và sản phẩm, mã hóa các phản hồi ngầm như lượt nhấp chuột, thuê hoặc mua hàng. Mỗi cạnh (u, i) biểu thị một tương tác giữa người dùng u và sản phẩm i . Về mặt hình thức, $A \in R^{n \times m}$, trong đó n và m lần lượt là số lượng người dùng và sản phẩm.
- Vector đặc trưng hình ảnh (E): Là tập hợp gồm m vector đặc trưng hình ảnh, mỗi vector được trích xuất từ ảnh sản phẩm bằng một mô hình CNN đã được

huấn luyện sẵn, chẳng hạn như EfficientNet hoặc ResNet. Các vector này tạo thành một ma trận $E \in R^{m \times d}$, với d là kích thước không gian đặc trưng.

- Các biểu diễn hình ảnh sản phẩm: Là biểu diễn thị giác của sản phẩm sau khi được tinh chỉnh thông qua các phép biến đổi có tính đến độ tương đồng giữa các sản phẩm. Biểu diễn này không chỉ phản ánh ngoại hình riêng lẻ của từng sản phẩm, mà còn thể hiện mối liên hệ cảm nhận với các sản phẩm khác. Đây là đầu vào quan trọng cho các module tiếp theo như Attention Sampling hoặc Message Passing.
- Ma trận tương đồng hình ảnh (S): Là một ma trận đối xứng được tính toán dựa trên độ tương đồng cosine giữa các vector đặc trưng hình ảnh. Mỗi phần tử S_{xy} biểu thị mức độ giống nhau về mặt thị giác giữa sản phẩm x và sản phẩm y .

Kiến trúc Visual-TransGNN bao gồm ba thành phần chính, được minh họa trong Hình 3.3, bao gồm: Attention Sampling, Positional Encoding và mô-đun cốt lõi Visual-TransGNN.

Ở đầu vào, các hình ảnh sản phẩm (định dạng như .jpg hoặc .png) được xử lý bởi một bộ trích xuất đặc trưng hình ảnh (chẳng hạn như CNN hoặc EfficientNet) để tạo ra các vector đặc trưng hình ảnh có số chiều lớn. Từ các vector này, một ma trận tương đồng hình ảnh được tính toán dựa trên độ tương đồng cosine, đóng vai trò như một tín hiệu phụ trợ cho các module xử lý phía sau.

Module Attention Sampling có nhiệm vụ lựa chọn các hàng xóm giàu thông tin nhất cho mỗi node, bằng cách kết hợp đồng thời giữa kết nối cấu trúc trong đồ thị và độ tương đồng thị giác giữa các node. Phương pháp sampling có điều hướng attention này

giúp cải thiện quá trình học biểu diễn, thông qua việc loại bỏ nhiễu không liên quan và tập trung vào các ngữ cảnh thực sự có ý nghĩa.

Module Positional Encoding giúp làm giàu biểu diễn của từng node thông qua các tín hiệu hình học của đồ thị, bao gồm: mã hóa đường đi ngắn nhất (Shortest Path Encoding – SPE), mã hóa theo bậc node (Degree Encoding – DE), và mã hóa dựa trên PageRank (PRE). Những kỹ thuật mã hóa này mang lại cho mô hình khả năng nhận thức về cấu trúc tổng thể và ngữ cảnh vị trí của mỗi node trong đồ thị.

Tiếp theo, mô-đun Visual-TransGNN cốt lõi được tổ chức thành ba thành phần chính: (i) Lớp Transformer, thực hiện attention đa đầu (multi-head attention) trên tập hàng xóm đã được sampling; (ii) Lớp GNN, truyền thông tin dọc theo đồ thị tương tác người dùng – sản phẩm, đồng thời tích hợp thông tin thị giác thông qua cơ chế lan truyền phụ trợ; (iii) Chiến lược cập nhật mẫu tự động, nhằm duy trì tính đa dạng trong quá trình huấn luyện qua các epoch.

Bên cạnh đó, module Embedding Aggregation thực hiện việc kết hợp các embedding học được từ đồ thị với các embedding được tăng cường bằng thông tin hình ảnh, từ đó tạo ra biểu diễn cuối cùng cho cả người dùng và sản phẩm. Những biểu diễn này sau đó được chuyển qua lớp dự đoán để tính toán điểm xếp hạng sản phẩm cuối cùng.

Tổng thể, Visual-TransGNN thực hiện việc tinh chỉnh embedding một cách lặp lại, bằng cách luân phiên giữa các bước lan truyền dựa trên attention và lan truyền theo cấu trúc đồ thị. Nhờ đó, mô hình có thể tận dụng đồng thời cả hành vi tương tác của người dùng và tính mạch lạc về mặt thị giác, giúp nâng cao hiệu quả gợi ý trong các hệ thống đề xuất sản phẩm thời trang.

3.3.2 Phương pháp đo độ tương đồng hình ảnh

Mỗi hình ảnh sản phẩm trước tiên sẽ được mã hóa thông qua một mạng CNN được huấn luyện sẵn (ví dụ: EfficientNet, ResNet hoặc MobileNet) trên tập dữ liệu ImageNet, nhằm sinh ra một vector đặc trưng có kích thước cố định $v_i \in R^d$ cho sản phẩm i .

Độ tương đồng giữa hai sản phẩm i và j được tính dựa trên cosine similarity giữa các vector embedding tương ứng như sau:

$$V_{i,j} = \frac{v_i \cdot v_j}{|v_i||v_j|} \in [-1,1] \quad (3.1)$$

Quá trình này tạo ra một ma trận tương đồng thị giác đối xứng $V \in R^{m \times m}$, trong đó mỗi phần tử $V_{i,j}$ thể hiện mức độ giống nhau về mặt thị giác giữa sản phẩm i và j .

Để tăng tính ổn định trong các phép tính dựa trên đồ thị, ma trận tương đồng thị giác V được chuẩn hóa bằng chuẩn hóa bậc đối xứng (symmetric degree normalization). Bậc (degree) của mỗi sản phẩm i được tính bằng tổng độ tương đồng của nó với tất cả các sản phẩm còn lại:

$$d_i = \sum_{j=1}^m V_{i,j} \quad (3.2)$$

Để đảm bảo ổn định số học, một hằng số nhỏ $\varepsilon > 0$ được thêm vào. Khi đó, độ tương đồng thị giác được chuẩn hóa giữa hai sản phẩm i và j được tính theo công thức:

$$\widehat{V}_{i,j} = \frac{V_{i,j}}{\sqrt{(d_i + \varepsilon)(d_j + \varepsilon)}} \quad (3.3)$$

Dưới dạng ma trận, phép chuẩn hóa trên có thể được biểu diễn như sau:

$$\widehat{V} = D^{-\frac{1}{2}} V D^{-\frac{1}{2}} \quad (3.4)$$

Trong đó D là ma trận đường chéo với các phần tử $D_{ii} = d_i + \varepsilon$. Ma trận tương đồng thị giác đã được chuẩn hóa \hat{V} giúp cân bằng ảnh hưởng của các node trong đồ thị, đồng thời phù hợp cho các bước lan truyền thông điệp (message passing). Việc sử dụng chuẩn hóa này giúp mô hình khai thác hiệu quả hơn các mối quan hệ cảm nhận giữa các sản phẩm có hình ảnh tương đồng.

3.3.3 Attention Sampling Module

Việc tính attention trên toàn bộ đồ thị tương tác người dùng - sản phẩm dẫn đến hai thách thức lớn:

- Chi phí tính toán cao: Việc tính attention toàn cục có độ phức tạp bậc hai $O(N^2)$, khiến nó trở nên không khả thi trên các hệ thống khuyến nghị quy mô lớn.
- Nhiều thông tin: Attention toàn cục có thể dẫn đến việc đưa vào các tương tác người dùng - sản phẩm không liên quan, ảnh hưởng tiêu cực đến hiệu suất mô hình.

Module này giới thiệu một chiến lược sampling có điều hướng attention, nhằm giảm chi phí tính toán so với attention toàn cục, đồng thời lọc bỏ các hàng xóm không liên quan. Quá trình bắt đầu bằng việc tính ma trận tương đồng ngữ nghĩa từ ma trận đặc trưng của các node $X \in R^{N \times d}$, theo công thức:

$$S = XX^T \quad (3.5)$$

Sau đó, độ lệch cấu trúc (structural bias) được đưa vào bằng cách tinh chỉnh ma trận tương đồng ngữ nghĩa thông qua cấu trúc đồ thị:

$$\hat{S} = S + \alpha \hat{A}S \text{ với } \hat{A} = A + I \quad (3.6)$$

Tại đây, A là ma trận kề của đồ thị tương tác, I là ma trận đơn vị, và $\alpha = 0.5$ là hệ số trọng số điều chỉnh mức độ ảnh hưởng của tín hiệu cấu trúc. Với mỗi node v_i , top-k hàng xóm có độ tương đồng cao nhất sẽ được lựa chọn từ hàng thứ i của ma trận S , dựa trên điểm tương đồng đã được tính chỉnh. Chiến lược sampling này giúp giảm tải tính toán cho attention toàn cục, đồng thời cho phép mô hình tập trung vào các hàng xóm thực sự giàu thông tin và có ý nghĩa về mặt cấu trúc.

3.3.4 Positional Encoding Module

Các tương tác giữa người dùng và sản phẩm trong hệ thống gợi ý chứa những thông tin cấu trúc phức tạp, đóng vai trò quan trọng trong việc tạo ra các đề xuất mang tính cá nhân hóa. Khác với dữ liệu dạng lưới (grid-like) – nơi mà các mô hình như Transformer có thể dễ dàng học được các mẫu theo trình tự – thì đồ thị tương tác lại có cấu trúc liên kết phức tạp hơn, gây khó khăn cho việc học thông tin.

Để giúp Transformer hiểu được đặc trưng cấu trúc của đồ thị, thông tin cấu trúc từ đồ thị được mã hóa vào biểu diễn của từng node thông qua ba loại Positional Encoding bổ trợ lẫn nhau, bao gồm:

Mã hóa theo đường đi ngắn nhất (Shortest-Path Encoding – SPE): Loại encoding này biểu diễn khoảng cách giữa các node bằng cách ánh xạ số bước nhảy ngắn nhất $P(i, j)$ giữa node i và node j vào không gian tiềm ẩn, thông qua một mạng perceptron nhiều lớp (MLP):

$$SPE(v_i, v_j) = MLP(P(i, j)) \quad (3.7)$$

Mã hóa dựa trên bậc node (Degree-based Encoding – DE): Encoding này phản ánh mức độ kết nối cục bộ của node, qua đó thể hiện được mức độ phổ biến hoặc tính đa dạng. Bậc của node i , ký hiệu là deg_i , được đưa vào một MLP để tạo ra một vector embedding có thể học được:

$$DE(v_i) = MLP(deg_i) \quad (3.8)$$

Mã hóa dựa trên PageRank (PageRank-based Encoding – PRE): Encoding này nắm bắt mức độ quan trọng toàn cục của node trong đồ thị, bằng cách sử dụng điểm PageRank của node và chuyển đổi nó thông qua MLP:

$$PRE(v_i) = MLP(Pr(v_i)) \quad (3.9)$$

Sau khi tính được ba biểu diễn cấu trúc nêu trên, vector đặc trưng ban đầu của node x_i sẽ được nối (concatenate) với SPE, DE và PRE, sau đó đưa qua một MLP khác để tạo ra biểu diễn cuối cùng đã được làm giàu:

$$h_i = MLP([x_i | SPE_i | DE_i | PRE_i]) \quad (3.10)$$

Tại đây, $MLP(\cdot)$ là mạng perceptron nhiều lớp, một mạng nơ-ron truyền thẳng (Feed Forward Neural Network) bao gồm một hoặc nhiều lớp fully-connected đi kèm các hàm kích hoạt phi tuyến.

3.3.5 Visual TransGNN Module

Module này tích hợp Transformer với Mạng Nơ-ron Đồ thị (GNN) nhằm kết hợp khả năng học các phụ thuộc xa (long-range dependencies) với cấu trúc đồ thị cục bộ, đồng thời khai thác độ tương đồng thị giác giữa các sản phẩm để làm giàu embedding bằng thông tin hình ảnh:

3.3.5.1 Transformer Layer

Với một node mục tiêu v_i và tập hàng xóm đã được sampling $Smp(v_i)$, các biểu diễn node tương ứng được xếp thành một ma trận $H \in R^{k \times d}$, sau đó đưa qua cơ chế self-attention đa đầu.

Biểu diễn đầu ra của node v_i được tính như sau:

$$h_i = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (3.11)$$

Trong đó:

- $Q = h_i W_q$ là ma trận query
- $K = H W_k$ là ma trận key
- $V = H W_v$ là ma trận value

Các ma trận W_q, W_k, W_v là các ma trận học được trong quá trình huấn luyện. Công thức này tuân theo cơ chế attention chuẩn đã được giới thiệu trong các mô hình Transformer.

3.3.5.2 GNN Message Passing Layer tích hợp tương đồng thị giác

Sau bước tổng hợp dựa trên attention, các embedding node được tiếp tục tinh chỉnh thông qua message passing trên đồ thị tương tác người dùng – sản phẩm.

Bước lan truyền chuẩn được biểu diễn như sau:

$$h^{(l+1)} = \hat{A} h^{(l)} \quad (3.12)$$

trong đó \hat{A} là ma trận kề đã chuẩn hóa.

Để tích hợp thông tin thị giác, một bước lan truyền bổ sung được thực hiện bằng ma trận tương đồng thị giác đã chuẩn hóa S_{vis} . Đối với các node sản phẩm, biểu diễn được cập nhật theo công thức:

$$h_i^{(l+1)} = h_i^{(l+1)} + \alpha \cdot S_{vis} \cdot h_i^{(l)} \quad (3.13)$$

Trong đó, α là một hệ số vô hướng (có thể học được hoặc cố định) điều chỉnh mức độ ảnh hưởng của tín hiệu hình ảnh.

Công thức này cho phép các node sản phẩm hấp thụ thông tin ngữ cảnh dựa trên ảnh từ các node có đặc điểm thị giác tương đồng, giúp embedding học được trở nên giàu tính liên kết cảm nhận (perceptual coherence) hơn.

3.3.5.3 *Sample Update Sub-Module*

Để đảm bảo hiệu quả tính toán, các mẫu attention sẽ được cập nhật định kỳ thông qua các heuristic sampling cục bộ hoặc random walk, thay vì phải tính toán lại toàn bộ ma trận tương đồng có độ phức tạp $O(N^2)$. Cơ chế này giúp mô hình duy trì tính hiệu quả và khả năng khái quát hóa trong suốt quá trình huấn luyện.

3.4 Các phương pháp sinh ứng viên nền tảng trong hệ thống khuyến nghị

Trong hệ thống khuyến nghị đa tầng, giai đoạn sinh ứng viên (candidate generation) đóng vai trò quan trọng nhằm thu hẹp không gian tìm kiếm, bằng cách chọn ra tập hợp các sản phẩm tiềm năng có khả năng phù hợp với từng người dùng.

3.4.1 *Dựa trên lịch sử giao dịch*

OrderHistory: Truy xuất các sản phẩm người dùng đã mua trong khoảng thời gian gần (ví dụ: 3 hoặc 7 ngày). Phù hợp với việc gợi ý lại các sản phẩm đã từng được ưa thích.

OrderHistoryDecay: Gán trọng số cao hơn cho các giao dịch gần đây và giảm dần theo thời gian. Phản ánh tốt hơn xu hướng mua sắm hiện tại.

ItemPair: Gợi ý các sản phẩm thường được mua cùng nhau bởi người dùng khác. Dựa trên phân tích tần suất xuất hiện đồng thời giữa các cặp sản phẩm.

TimeHistory: Phân tích các sản phẩm phổ biến nhất trong một khoảng thời gian nhất định (ví dụ: 7 ngày gần đây).

TimeHistoryDecay: Tương tự TimeHistory, nhưng bổ sung yếu tố thời gian làm trọng số – giúp nhấn mạnh các sản phẩm đang nổi.

SaleTrend: Phân tích sự thay đổi về doanh số sản phẩm trong thời gian ngắn, từ đó phát hiện xu hướng tiêu dùng tăng đột biến.

3.4.2 Dựa trên nhóm người dùng

UserGroupTimeHistory: Sử dụng thông tin phân nhóm người dùng (theo tuổi, giới tính, mức chi tiêu) để gợi ý các sản phẩm phổ biến trong từng nhóm.

UserGroupSaleTrend: Theo dõi xu hướng mua hàng trong nhóm người dùng. Nếu một sản phẩm đang phổ biến trong một nhóm, nó sẽ được đề xuất cho các thành viên khác cùng nhóm.

3.4.3 Dựa trên lọc cộng tác

ALS (Alternating Least Squares): Mô hình phân rã ma trận người dùng - sản phẩm, học các đặc trưng tiềm ẩn để dự đoán sản phẩm phù hợp cho người dùng.

UserGroupALS: Áp dụng mô hình ALS riêng cho từng nhóm người dùng, được phân chia theo hành vi hoặc nhân khẩu học.

BPR (Bayesian Personalized Ranking): Tối ưu hóa thứ tự xếp hạng sản phẩm dựa trên hành vi tương tác. Tập trung vào việc đặt sản phẩm phù hợp lên đầu danh sách.

UserGroupBPR: Huấn luyện BPR riêng cho từng nhóm người dùng, từ đó tạo ra đề xuất cá nhân hóa sâu hơn.

ItemCF (Item-based Collaborative Filtering): Dựa trên sự tương đồng giữa các sản phẩm mà người dùng đã tương tác để đề xuất sản phẩm mới.

UserGroupItemCF: Kết hợp ItemCF với phân nhóm người dùng. Mỗi nhóm có một mô hình ItemCF riêng, giúp gợi ý chính xác hơn.

CHƯƠNG 4 - THỰC NGHIỆM

4.1 Các tập dữ liệu được sử dụng

4.1.1 *Vibrent Clothes Rental*

4.1.1.1 *Tổng quan về tập dữ liệu*

Tập dữ liệu Vibrent Clothes Rental ghi lại đầy đủ các hoạt động cho thuê của một công ty thời trang vừa và nhỏ, chuyên cung cấp dịch vụ cho thuê quần áo. Tập dữ liệu này được thu thập, xử lý và công bố công khai trên nền tảng Kaggle – một hệ sinh thái trực tuyến nổi tiếng dành cho cộng đồng khoa học dữ liệu và học máy, nơi chia sẻ và phát triển các tập dữ liệu thực tế phục vụ nghiên cứu và ứng dụng.

Bộ dữ liệu gồm 4 thành phần chính:

- Outfits (Trang phục): Gồm 15.600 bộ trang phục duy nhất, được phân loại thành khoảng 9.700 nhóm trang phục. Mỗi bộ trang phục đi kèm với các thông tin mô tả như: tên, mô tả ngắn, tập các thẻ (tags) thể hiện thuộc tính (ví dụ: “Dress”, “Green”, “Wool”), hình ảnh minh họa (tổng cộng hơn 50.100 ảnh), thời gian bắt đầu cho thuê, thời hạn và giá thuê tương ứng.
- Users (Người dùng): Bao gồm 7.400 người dùng ẩn danh với lịch sử thuê trang phục được ghi nhận. Phần lớn người dùng là nữ giới, có nhu cầu thuê trang phục cho các hoạt động thường nhật hoặc các sự kiện như tiệc tùng.
- Transactions (Giao dịch): Ghi nhận 4.400 giao dịch cho thuê được thực hiện trong giai đoạn từ năm 2016 đến năm 2024, phản ánh hành vi sử dụng thực tế và lịch sử tương tác giữa người dùng và sản phẩm.
- Hình ảnh và Embeddings: Tập dữ liệu cung cấp 50.300 hình ảnh trang phục tương ứng với các bộ outfits. Đồng thời, đi kèm là thư mục chứa các vector

đặc trưng (embeddings) của ảnh, được trích xuất sẵn từ mô hình EfficientNet_V2, phục vụ cho các bài toán khai thác đặc trưng hình ảnh trong hệ thống khuyến nghị.

4.1.1.2 Tiền xử lý dữ liệu

Trong quá trình khai thác bộ dữ liệu Vibrent Clothes Rental, chúng tôi nhận thấy tồn tại một số vấn đề chất lượng dữ liệu cần được xử lý trước khi đưa vào huấn luyện mô hình. Cụ thể:

- Xử lý giá trị thiếu (Missing values): Ở tập dữ liệu outfits, nhiều cột chứa các giá trị khuyết (NaN). Để đảm bảo tính toàn vẹn của dữ liệu, chúng tôi đã loại bỏ các dòng dữ liệu không đầy đủ thay vì thay thế hoặc suy đoán, nhằm tránh ảnh hưởng tiêu cực đến chất lượng mô hình.
- Chuẩn hóa định dạng mảng: Một số cột chứa dữ liệu dạng mảng (ví dụ: danh sách các thẻ tags) được lưu trữ dưới dạng chuỗi. Chúng tôi đã xử lý và chuyển đổi các trường này về đúng định dạng danh sách (list of strings) để phục vụ việc trích xuất đặc trưng.
- Loại bỏ và đồng bộ ID ngoại lai: Trong quá trình kết hợp các bảng dữ liệu (outfits, users, transactions), chúng tôi phát hiện một số ID không còn tồn tại hoặc không đồng nhất giữa các bảng. Các dòng dữ liệu chứa khóa ngoại (foreign key) không hợp lệ đã được loại bỏ đồng thời ở các bảng liên quan, nhằm đảm bảo tính liên kết và tránh mất đồng bộ dữ liệu.
- Chuẩn hóa thời gian: Dữ liệu thời gian trong các bảng có định dạng không thống nhất. Chúng tôi đã chuyển toàn bộ các trường thời gian về định dạng chuẩn UNIX timestamp (Epoch Time) để phục vụ cho các phép tính thời gian và phân tích dòng thời gian sau này.

4.1.2 H&M Personalized Fashion Recommendations

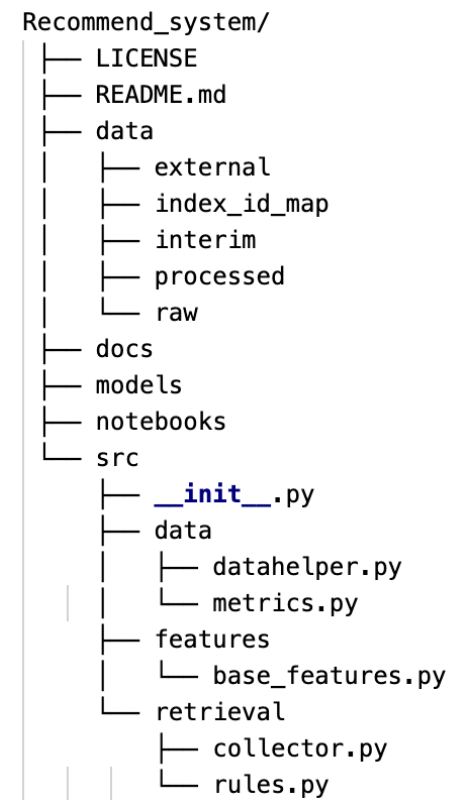
Tập dữ liệu H&M Personalized Fashion Recommendation, được công bố công khai trên nền tảng Kaggle. Tập dữ liệu phản ánh hoạt động bán hàng của công ty thời trang H&M, bao gồm thông tin chi tiết về sản phẩm, khách hàng, lịch sử giao dịch, và hình ảnh.

Cụ thể, tập dữ liệu bao gồm bốn thành phần chính:

- `articles.csv`: Chứa thông tin chi tiết về từng sản phẩm (*article_id*) có trong kho, với tổng cộng 105.542 sản phẩm. Tập tin này bao gồm 25 cột, cung cấp các siêu dữ liệu như tên sản phẩm, danh mục, màu sắc, chất liệu, và mã nhóm sản phẩm.
- `customers.csv`: Ghi nhận siêu dữ liệu về 1.371.980 khách hàng (*customer_id*) tham gia giao dịch. Trong đó có 7 trường thông tin bao gồm các đặc trưng nhân khẩu học và hành vi mua sắm, giúp hỗ trợ quá trình xây dựng hồ sơ người dùng.
- `transactions_train.csv`: Là tập dữ liệu huấn luyện, ghi nhận 31.788.324 dòng tương ứng với các giao dịch mua hàng theo từng ngày của người dùng. Mỗi dòng thể hiện một giao dịch giữa khách hàng và sản phẩm, bao gồm cả thông tin về số lượng, thời gian và mức giá. Lưu ý rằng các dòng dữ liệu trùng lặp có thể xuất hiện nếu khách hàng mua cùng một sản phẩm nhiều lần.
- Hình ảnh sản phẩm: Bộ dữ liệu còn bao gồm 105.104 hình ảnh, tương ứng với các *article_id*. Các hình ảnh được tổ chức thành 87 thư mục con, trong đó mỗi thư mục được đặt tên dựa trên ba chữ số đầu tiên của mã *article_id*. Hình ảnh giúp hỗ trợ các mô hình khai thác đặc trưng thị giác trong hệ thống khuyến nghị.

4.2 Triển khai hệ thống khuyến nghị đa mô hình với LightGBM và DNN

4.2.1 Cấu trúc thư mục mô hình LightGBM



Hình 4.1 Cấu trúc thư mục mô hình LightBGM

Về cấu trúc tổng thể, hệ thống được tổ chức thành các thư mục chức năng phục vụ cho toàn bộ quy trình xử lý dữ liệu và huấn luyện mô hình:

- `data/`: Chứa toàn bộ dữ liệu cần thiết cho pipeline bao gồm dữ liệu thô, dữ liệu trung gian, dữ liệu đã xử lý và embedding được tạo ra từ nhiều phương pháp khác nhau.
- `models/`: Lưu trữ các mô hình đã huấn luyện dưới nhiều định dạng, phục vụ cho bước suy luận hoặc đánh giá.

- notebooks/: Tập hợp các notebook Jupyter dùng để khám phá dữ liệu, thử nghiệm các chiến lược recall, xử lý embedding và trực quan hóa kết quả.
- src/: Thư mục mã nguồn chính của hệ thống, gồm các module tiền xử lý dữ liệu, tạo đặc trưng và triển khai các chiến lược recall (thu thập ứng viên).
- docs/: Chứa tài liệu kỹ thuật của hệ thống được xây dựng bằng Sphinx, mô tả chi tiết các thành phần và quy trình thực hiện.
- README.md: Tập tài liệu hướng dẫn sử dụng, bao gồm mô tả cấu trúc thư mục, cách cài đặt và thực thi hệ thống.

4.2.2 Thu thập các ứng viên

4.2.2.1 Mô hình LightGBM 1

Mô hình LightGBM 1 thực hiện thu thập ứng viên theo đơn vị thời gian tuần, nhằm khai thác hành vi mua sắm gần nhất của người dùng và huấn luyện mô hình một cách tuần tự, giả lập điều kiện thực tế trong thương mại điện tử.

Xử lý dữ liệu theo tuần:

- Vòng lặp chính duyệt qua từng tuần từ tuần 1 đến WEEK_NUM - 1.
- Ở mỗi tuần, hệ thống tính toán khoảng thời gian hợp lệ (start_date, end_date) thông qua hàm calc_valid_date().
- Dữ liệu tương tác được chia thành hai phần: Tập huấn luyện (train): chứa dữ liệu trước start_date và tập kiểm tra (valid): chứa dữ liệu từ start_date đến end_date.
- Thông tin nhân khẩu học như nhóm tuổi (age_bins) được bổ sung từ bảng người dùng vào tập huấn luyện để tăng độ chính xác cho mô hình.

Sinh ứng viên bằng lớp RuleCollector: Hệ thống sử dụng lớp RuleCollector để tích hợp nhiều chiến lược recall khác nhau, từ đó xây dựng danh sách các sản phẩm tiềm năng cho từng người dùng. Các chiến lược chính bao gồm:

- OrderHistory: Gợi ý dựa trên lịch sử mua hàng trong 3 ngày và 7 ngày gần nhất.
- OrderHistoryDecay: Giống như OrderHistory nhưng áp dụng trọng số giảm dần theo thời gian (càng gần hiện tại càng quan trọng).
- ItemPair: Khai thác mối quan hệ đồng xuất hiện giữa các sản phẩm (các sản phẩm thường được mua cùng nhau).
- UserGroupTimeHistory: Gợi ý dựa trên hành vi mua hàng của nhóm người dùng có đặc điểm tương đồng (ví dụ cùng độ tuổi).
- TimeHistoryDecay: Tận dụng xu hướng mua sắm cá nhân của người dùng theo thời gian, có áp dụng cơ chế suy giảm.
- SaleTrend: Phân tích xu hướng bán hàng chung trên toàn hệ thống để gợi ý những sản phẩm đang được quan tâm.

Gán nhãn (label): Sau khi thu thập được danh sách ứng viên, hệ thống sẽ đối chiếu với tập valid để gán nhãn nhị phân (1 nếu đã mua, 0 nếu chưa) cho từng cặp người dùng – sản phẩm. Tập dữ liệu có gán nhãn này là đầu vào cho mô hình xếp hạng LightGBM trong bước tiếp theo.

4.2.2.2 Mô hình LightGBM 2

Mô hình LightGBM 2 xây dựng quy trình huấn luyện theo tuần, nhằm khai thác các đặc trưng hành vi và nhân khẩu học của người dùng trong từng giai đoạn cụ thể. Mỗi tuần sẽ được xử lý như một phiên huấn luyện riêng biệt, mô phỏng việc huấn luyện và đánh giá mô hình trong điều kiện thực tế.

Phân chia dữ liệu theo tuần: Mỗi tuần được xác định thông qua hàm `calc_valid_date(week)`, trả về cặp ngày `start_date` và `end_date`. Dữ liệu giao dịch được chia thành: Train: Từ thời điểm bắt đầu đến trước `start_date` và Valid: Từ `start_date` đến `end_date`, dùng để kiểm tra độ chính xác của mô hình. Tập train được bổ sung thông tin từ bảng `user_info` như: độ tuổi (age), giới tính (gender), nhóm tuổi (age_bins), khả năng chi tiêu,...

Tạo các cửa sổ thời gian (time windows): Để phản ánh hành vi mua sắm theo thời gian, hệ thống lọc giao dịch trong các khoảng sau: `last_week`: 7 ngày gần nhất, `last_2week`: 14 ngày gần nhất, `last_60day`: 60 ngày gần nhất., `last_80day`: 80 ngày gần nhất.

Chiến lược thu thập ứng viên (Candidate Generation): Hệ thống kết hợp nhiều phương pháp để tạo danh sách sản phẩm tiềm năng cho từng người dùng:

- Phương pháp dựa trên Collaborative Filtering: ALS, BPR.
- Phương pháp dựa trên lịch sử mua hàng: OrderHistory: Dựa trên các đơn hàng trong 35 ngày gần nhất. OrderHistoryDecay: Tăng trọng số cho các sản phẩm mua gần đây (7 ngày), giảm dần theo thời gian. TimeHistory: Phân tích xu hướng mua trong 7 ngày gần nhất.
- Phương pháp nhóm người dùng: UserGroupTimeHistory: Phân tích hành vi mua theo nhóm độ tuổi, giới tính, khả năng chi tiêu. UserGroupItemCF: Kết hợp giữa nhóm người dùng và lọc cộng tác theo item.
- Phương pháp Item-based Collaborative Filtering (ItemCF): Gợi ý sản phẩm tương tự dựa trên độ tương đồng giữa các mặt hàng, áp dụng trên nhiều khung thời gian (80 ngày, 60 ngày, 14 ngày).

Lọc và xếp hạng ứng viên:

- Các sản phẩm ứng viên được giữ lại nếu đạt ngưỡng tối thiểu về tần suất xuất hiện ($\text{min_pos_rate} = 0.006$).
- Điểm số của từng ứng viên được chuẩn hóa bằng MinMaxScaler, sau đó thực hiện xếp hạng theo từng người dùng.

Gán nhãn và chuẩn bị cho mô hình ranking

- Sau khi thu thập được danh sách các ứng viên, hệ thống đối chiếu với tập valid để gán nhãn: $\text{Label} = 1$: nếu người dùng thực sự đã mua sản phẩm trong tuần đó, $\text{Label} = 0$: nếu không có hành vi mua.
- Tập dữ liệu có nhãn được dùng làm đầu vào cho mô hình LightGBM trong bước huấn luyện ranking.

Phương pháp LightGBM 2 này cho phép mở rộng chiến lược gợi ý bằng cách tích hợp linh hoạt nhiều nguồn đặc trưng, cả về hành vi và ngữ cảnh, từ đó cải thiện chất lượng và độ cá nhân hóa của kết quả đề xuất.

4.2.3 Huấn luyện mô hình

Quá trình huấn luyện mô hình được kiểm soát nghiêm ngặt nhằm đảm bảo hiệu quả và độ ổn định. Cụ thể, hai callback quan trọng được sử dụng là EarlyStopping và ModelCheckpoint. Trong đó, EarlyStopping theo dõi chỉ số AUC trên tập validation và tự động dừng quá trình huấn luyện nếu AUC không cải thiện sau 10 epoch liên tiếp. Cơ chế này giúp giảm thiểu hiện tượng quá khớp (overfitting) và tiết kiệm tài nguyên tính toán. Đồng thời, ModelCheckpoint đóng vai trò lưu lại các trọng số của mô hình tại thời điểm đạt AUC cao nhất, đảm bảo rằng mô hình cuối cùng là phiên bản tốt nhất trong toàn bộ quá trình huấn luyện.

Mô hình được huấn luyện trên tập dữ liệu lớn với batch size là 2048, phù hợp với quy mô dữ liệu giao dịch thực tế. Dữ liệu đầu vào gồm hai phần chính (X_{train1} và

X_train2), phản ánh cấu trúc đặc trưng phức tạp của bài toán. Việc xáo trộn dữ liệu sau mỗi epoch được thực hiện nhằm tăng tính ngẫu nhiên và độ khách quan trong huấn luyện. Số epoch tối đa được đặt là 30, tuy nhiên có thể dừng sớm nếu điều kiện của EarlyStopping được kích hoạt.

Tập validation riêng biệt (X_test1 , X_test2) được sử dụng để đánh giá độc lập hiệu năng của mô hình trong suốt quá trình training. Chỉ số AUC được lựa chọn làm tiêu chí giám sát chính do phù hợp với đặc thù của bài toán recommendation, nơi khả năng xếp hạng và phân biệt giữa các lựa chọn là yếu tố quan trọng hơn so với độ chính xác tuyệt đối. Cách tiếp cận này không chỉ đảm bảo chất lượng mô hình sau huấn luyện, mà còn tối ưu hóa toàn diện hiệu suất hệ thống.

4.2.4 Kết hợp hai mô hình *LightGBM 1* và *LightGBM 2*

Mô hình LightGBM 2 hiện đại hơn và sử dụng nhiều chiến lược học sâu hơn như ALS, BPR, ItemCF, trong khi LightGBM 1 chủ yếu dùng các rule-based heuristics và decay theo thời gian. LightGBM 2 chú trọng tối ưu hóa đầu ra (normalize score, lọc top-k), còn LightGBM1 giữ nguyên toàn bộ ứng viên. Dữ liệu lịch sử ở LightGBM 2 phong phú hơn về chiều sâu (80 ngày), cho phép các mô hình học nhiều khía cạnh hành vi hơn.

LightGBM 1 có thể dễ triển khai hơn với độ phức tạp thấp hơn, nhưng ít khả năng khái quát hóa và ít phù hợp với các dữ liệu lớn và đa chiều như trong LightGBM 2.

Thực hiện quá trình kết hợp (blending) kết quả dự đoán từ sáu mô hình khuyến nghị khác nhau, với mục tiêu khai thác ưu điểm của từng mô hình và giảm thiểu hiện tượng overfitting. Hệ thống được chia thành hai nhóm chính: nhóm mô hình phiên bản large (LightGBM rank, LightGBM binary và DNN) và nhóm mô hình phiên bản small tương ứng.

Trước hết, các kết quả dự đoán được đọc và xử lý từ từng mô hình. Đối với mỗi nhóm, hệ thống thực hiện việc ghép nối (merge) các cặp giá trị `customer_id` và `article_id` từ các kết quả đầu ra, sau đó sắp xếp danh sách sản phẩm theo xác suất dự đoán giảm dần. Danh sách sản phẩm này được nhóm lại theo từng người dùng, tạo thành tập hợp các sản phẩm được đề xuất riêng biệt cho mỗi mô hình.

Tiếp theo, quá trình kết hợp được thực hiện thông qua hàm `cust_blend`. Mỗi mô hình trong số bốn mô hình chính (bao gồm LightGBM rank và binary ở cả hai phiên bản) được gán một trọng số, trong đó các mô hình binary được ưu tiên hơn với hệ số 1.3 so với 1.0 cho các mô hình còn lại. Mỗi sản phẩm trong danh sách đề xuất sẽ được tính một điểm tổng hợp, phản ánh mức độ ưu tiên của sản phẩm đó, dựa trên công thức:

$$score = \sum_{i=1}^n \frac{w_i}{rank_i + 1} \quad (4.1)$$

Trong đó w_i là trọng số của mô hình thứ i , và $rank_i$ là vị trí của sản phẩm trong danh sách đề xuất tương ứng.

Từ điểm số tổng hợp này, hệ thống lựa chọn ra 12 sản phẩm có điểm cao nhất để đưa vào danh sách đề xuất cuối cùng cho mỗi người dùng. Các kết quả sau khi kết hợp được lưu lại trong một cấu trúc dữ liệu tổng hợp (DataFrame), với cột `prediction` chứa danh sách các sản phẩm đã được blend và sắp xếp theo mức độ ưu tiên.

Phương pháp này không chỉ giúp tận dụng được các tín hiệu mạnh từ từng mô hình riêng biệt mà còn nâng cao độ ổn định và khả năng khái quát hóa của hệ thống gợi ý, từ đó cải thiện đáng kể hiệu năng trong thực tế.

4.2.5 Kết quả thực nghiệm

4.2.5.1 Mô hình LightGBM 1

Bảng 4.1 Hiệu suất mô hình LightGBM 1

AUC	0.7317
Map@12	0.0289
Recall@12	0.0665
HitRate@12	0.1316

Mô hình LightGBM 1 đạt $AUC = 0.7317$, cho thấy khả năng phân biệt giữa các lớp khá tốt. Tuy nhiên, hiệu suất gợi ý còn hạn chế với $MAP@12 = 0.0289$, cho thấy sản phẩm đúng chưa nằm ở vị trí cao trong top-12; $Recall@12 = 0.0665$ phản ánh mức độ bao phủ còn thấp; và $HitRate@12 = 13.16\%$ cho thấy tỷ lệ người dùng nhận được ít nhất một gợi ý đúng vẫn còn khiêm tốn.

4.2.5.2 Mô hình LightGBM 2

Bảng 4.2 Hiệu suất mô hình LightGBM 2

AUC	0.6826
Map@12	0.02918
Recall@12	0.0653
HitRate@12	0.132

Mô hình LightGBM 2 đạt $\text{MAP@12} = 0.02918$ và $\text{HitRate@12} = 13.2\%$, nhìn hơn mô hình 1, cho thấy độ chính xác và khả năng gợi ý đúng có cải thiện nhẹ. Tuy nhiên, AUC giảm xuống 0.6826 phản ánh khả năng phân biệt giữa các lớp yếu hơn, trong khi $\text{Recall@12} = 0.0653$ gần như không đổi, cho thấy mức độ bao phủ vẫn còn hạn chế.

4.2.5.3 Mô hình LightGBM 1 kết hợp mô hình LightGBM 2

Bảng 4.3 Hiệu suất mô hình kết hợp giữa LightGBM 1 và LightGBM 2

Map@12	0.03091
Recall@12	0.0699
HitRate@12	0.14

Mô hình tổ hợp đạt hiệu suất tốt nhất trong ba mô hình, với $\text{MAP@12} = 0.03091$, $\text{Recall@12} = 0.0699$ và $\text{HitRate@12} = 14\%$. Các chỉ số này cho thấy mô hình không chỉ đưa ra gợi ý đúng ở vị trí cao hơn trong top-12, mà còn bao phủ nhiều sản phẩm đúng hơn và tăng tỷ lệ người dùng nhận được gợi ý phù hợp so với từng mô hình đơn lẻ.

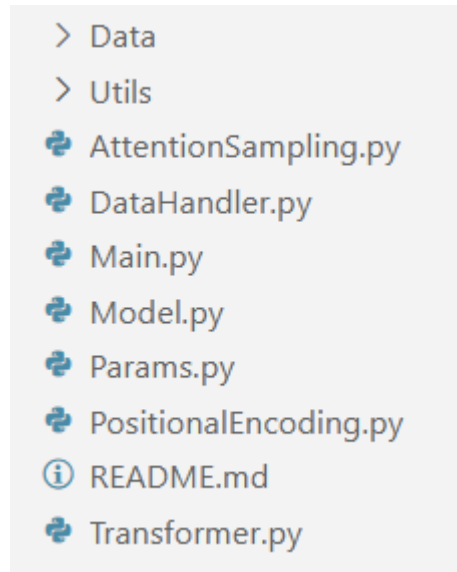
4.3 Triển khai mô hình Visual-TransGNN

4.3.1 Cấu trúc thư mục mô hình Visual-TransGNN

Để triển khai mô hình TransGNN, hệ thống được tổ chức thành một cấu trúc thư mục rõ ràng và phân tách chức năng, đảm bảo tính dễ quản lý và tái sử dụng. Trong đó:

- Data/: đóng vai trò lưu trữ toàn bộ tập dữ liệu đầu vào phục vụ cho quá trình huấn luyện và đánh giá mô hình. Các thư mục con bên trong Data/ tương ứng với từng tập dữ liệu cụ thể, đã được tiền xử lý để sẵn sàng đưa vào mô hình.

- Utils/ chứa các tập tin hỗ trợ kỹ thuật như các hàm tính toán hàm mất mát BPR, theo dõi tiến trình huấn luyện và đo thời gian thực thi theo từng epoch, giúp giám sát hiệu quả quá trình training.
- DataHandler.py: chịu trách nhiệm xử lý và chuẩn bị toàn bộ dữ liệu đầu vào cho TransGNN, bao gồm việc xây dựng ma trận kề user-item, tạo các mask cần thiết và khởi tạo các bộ nạp dữ liệu (dataloader) phục vụ cho cả huấn luyện và kiểm thử.
- Params.py: chứa tất cả các siêu tham số và cấu hình của mô hình giúp quá trình điều chỉnh mô hình trở nên thuận tiện và có tổ chức.
- Main.py: nơi triển khai pipeline huấn luyện chính, bao gồm các bước như khởi tạo mô hình, huấn luyện trên tập dữ liệu người dùng – sản phẩm, đánh giá mô hình qua các chỉ số đánh giá phổ biến như Recall, NDCG, MRR và HR, và cuối cùng là lưu lại trọng số của mô hình đạt kết quả tốt nhất.
- AttentionSampling.py, PositionalEncoding.py, Transformer.py và Model.py: Mỗi module đảm nhiệm một vai trò quan trọng trong kiến trúc tổng thể, từ việc chọn lọc các node quan trọng bằng attention sampling, bổ sung thông tin cấu trúc đồ thị thông qua positional encoding, đến lan truyền thông tin bằng các lớp Transformer và xây dựng forward logic tổng thể của mô hình.
- README.md: cung cấp tài liệu hướng dẫn chi tiết cho người dùng, bao gồm cách cài đặt môi trường, cách sử dụng hệ thống, mô tả chức năng của từng thành phần cũng như cách khởi chạy huấn luyện mô hình.



Hình 4.2 Cấu trúc thư mục mô hình Visual-TransGNN

Bên cạnh cấu trúc thư mục hệ thống, mỗi tập dữ liệu đầu vào nằm trong các thư mục con của Data/ cần đáp ứng các yêu cầu định dạng tối thiểu để đảm bảo khả năng tương thích với quy trình huấn luyện và suy luận của mô hình TransGNN:

- **user_list.txt** chứa danh sách tất cả người dùng xuất hiện trong tập hành vi tương tác, được biểu diễn dưới dạng các dòng văn bản, mỗi dòng là một mã định danh người dùng (mapping ID), được sắp xếp tăng dần từ 0 đến $n-1$, trong đó n là tổng số người dùng. Tương tự, **item_list.txt** ghi nhận toàn bộ sản phẩm từng xuất hiện trong các tương tác, với định dạng tương tự như danh sách người dùng.
- **train.txt** và **test.txt** đại diện cho tập huấn luyện và kiểm thử. Trong mỗi tệp, mỗi dòng tương ứng với một người dùng (theo mapping ID) và bao gồm danh sách các sản phẩm mà người dùng đó đã tương tác, cách nhau bằng khoảng trắng. Định dạng này giúp mô hình dễ dàng xây dựng ma trận kề (adjacency matrix) phục vụ cho lan truyền thông tin trong đồ thị.

- **interactions.txt** tổng hợp toàn bộ các tương tác người dùng – sản phẩm đã ghi nhận trong toàn bộ quá trình thu thập dữ liệu, được sử dụng trong quá trình xây dựng embedding và tạo các phép chiếu nền cho việc đánh giá.
- **similarity_matrix.npy** tệp định dạng nhị phân theo chuẩn NumPy, lưu trữ ma trận tương đồng giữa các sản phẩm dựa trên đặc trưng hình ảnh. Ma trận này được sử dụng trong bước truyền thông tin trực quan (visual propagation) của TransGNN.

4.3.2 Attention Sampling Module

Lớp AttentionSampling được thiết kế nhằm chọn ra các nút quan trọng nhất đối với từng nút trong đồ thị, dựa trên sự kết hợp giữa tương đồng ngữ nghĩa (semantic similarity) và cấu trúc đồ thị (structural similarity). Kết quả được sử dụng để tạo ra subgraph (đồ thị con) tối ưu cho quá trình lan truyền thông tin trong mô hình. Mục đích của việc sampling này là để giảm chi phí tính toán và làm nổi bật các node có ảnh hưởng cao, giúp nâng cao hiệu quả mô hình học.

Mã giả Attention Sampling Module

```

1. Class AttentionSampling
2.  Input:
3.  embeds: vector embedding của các node
4.  adj: ma trận kề gốc (sparse)
5.  Xử lý:
6.  Chuẩn hóa embedding: Chuẩn hóa L2 cho embeds  $\rightarrow X$ 
7.  Tính độ tương đồng:
8.      Semantic:  $S\_semantic = X \times X^t$ 
9.      Structural:  $\hat{A} = A + I$ , rồi  $S\_structural = \hat{A} \times S\_semantic$ 
10.  Trộn điểm attention:

```

11. $S = S_semantic + \alpha * S_structural$
12. Mask đường chéo ($S[i][i] = -\infty$)
13. Lấy top-k node có độ tương đồng cao nhất cho mỗi node
14. Xây lại ma trận kề:
15. Lấy chỉ số hàng - cột từ top-k
16. Xây adj_topk dạng sparse theo các cạnh được chọn (giá trị là score tương ứng)
17. Output:
18. sampled_nodes: danh sách node được chọn
19. adj_topk: ma trận kề mới sau attention sampling (sparse)

4.3.3 Positional Encoding Module

Lớp PositionalEncoding được thiết kế nhằm bổ sung thông tin cấu trúc toàn cục của đồ thị vào vector đặc trưng của mỗi node, qua ba loại đặc trưng vị trí:

- Khoảng cách ngắn nhất (Shortest-Path Distance) giữa các node để biểu diễn mức độ gần gũi theo topology.
- Bậc nút (Degree) để phản ánh mức độ kết nối trực tiếp.
- PageRank để đánh giá tầm quan trọng toàn cục của mỗi node trong đồ thị.

Mục tiêu chính là giúp mô hình có khả năng phân biệt vị trí tương đối và vai trò của từng node trong đồ thị, từ đó cải thiện chất lượng biểu diễn embedding trước khi truyền qua các lớp GNN hoặc Transformer.

Mã giả Positional Encoding Module:

1. Class PositionalEncoding
2. Input:
3. x: embedding ban đầu của user/item nodes
4. adj: ma trận kề (sparse)
5. node_indices (optional): các node cần tính PE

6. Xử lý:
7. Chuẩn bị dữ liệu
8. Chuyển adj sang dense nếu cần
9. Xác định node là user/item
10. Nếu chưa có node_indices, dùng toàn bộ node
11. Tính các positional encoding
12. Degree: Tính bậc của node → qua deg_mlp
13. PageRank: Tính PageRank → qua pre_mlp
14. Shortest-Path-Hop:
15. Nếu là user → hop = 0
16. Nếu là item → lấy từ dist_matrix[node_indices, node_indices]
17. → qua spe_mlp
18. Kết hợp đặc trưng
19. Ghép [x, spe, degree, pagerank] → qua comb_mlp
20. Output:
21. Embedding mới đã tích hợp positional encoding

4.3.4 Visual-TransGNN Module

Lớp TransformerEncoderLayer được xây dựng nhằm mục tiêu học biểu diễn (representation learning) của các node trong đồ thị hoặc chuỗi tương tác, bằng cách mô hình hóa mối quan hệ phụ thuộc dài hạn (long-range dependencies) thông qua cơ chế multi-head self-attention. Kiến trúc này cho phép mỗi node (hoặc phần tử trong chuỗi) cập nhật embedding của mình dựa trên toàn bộ các node còn lại, thay vì chỉ từ các node lân cận như trong GNN truyền thống.

Mã giả class Transformer Encoder:

1. Class TransformerEncoderLayer
2. Input:
3. x: Tensor đầu vào có shape (N, L, d_model) – batch gồm N node/item có độ dài L

4. `attn_mask`: mask attention dùng để bỏ qua các vị trí padding
5. Xử lý:
6. Attention Layer:
7. Áp dụng multi-head self-attention:
8. `attn_output = MultiheadAttention(x, x, x, mask=attn_mask)`
9. Residual + Norm:
10. $x \leftarrow x + \text{Dropout}(\text{attn_output})$
11. $x \leftarrow \text{LayerNorm}(x)$
12. Feedforward Layer:
13. $y \leftarrow \text{Linear1} \rightarrow \text{Activation} \rightarrow \text{Dropout} \rightarrow \text{Linear2} \rightarrow \text{Dropout}$
14. $x \leftarrow x + y \rightarrow \text{LayerNorm}$
15. Output:
16. Tensor có shape (N, L, d_{model}) , đã qua attention + feedforward + chuẩn hóa.

Hàm `gnn_message_passing` thực hiện bước lan truyền thông tin (message passing) trong mạng đồ thị (GNN), kết hợp cả thông tin cấu trúc từ đồ thị tương tác người dùng – sản phẩm và thông tin thị giác từ ma trận tương đồng hình ảnh giữa các sản phẩm.

Mã giả hàm truyền thông điệp `gnn_message_passing`:

1. Function `gnn_message_passing`
2. Input:
3. `adj`: ma trận kề user-item (sparse)
4. `embeds`: vector embedding ban đầu (user + item)
5. Xử lý:
6. Chuẩn hóa ma trận tương đồng ảnh (nếu có và chưa chuẩn hóa):
7. Tính $D^{-0.5} * A * D^{-0.5}$ với $A = \text{visual_sim}$
8. Lan truyền thông tin qua đồ thị gốc: `graph_embeds = adj @ embeds`
9. Nếu có `visual_sim_normalized`:
10. Tách phần `item_embeds`
11. Lan truyền trên đồ thị tương đồng ảnh
12. Kết hợp với `graph_embeds` bằng trọng số `alpha`
13. Output:

14. `graph_embeds`: embedding sau khi lan truyền và kết hợp

Hàm `forward()` là lõi lan truyền thông tin của mô hình TransGNN, kết hợp nhiều cơ chế học đặc trưng như attention sampling, positional encoding, transformer, và residual learning. Cụ thể, quy trình xử lý của hàm được chia thành ba bước chính:

1. Khởi tạo embedding đầu vào: Biểu diễn ban đầu của các node được tính bằng cách kết hợp tuyến tính giữa embedding học được và embedding thô từ đặc trưng ban đầu.
2. Lan truyền thông tin qua các khối (block): Với mỗi block được lặp lại:
 - Attention Sampling: Lựa chọn các node quan trọng từ toàn bộ đồ thị dựa trên độ tương đồng và cấu trúc (top-k), nhằm giảm độ phức tạp và tập trung vào các node liên quan.
 - GNN Message Passing: Lan truyền thông tin trên subgraph được lấy mẫu, cập nhật embedding của các node.
 - Positional Encoding (PE): Tính các đặc trưng định vị như Degree, PageRank và khoảng cách ngắn nhất (hop) cho các node được chọn sampling.
 - Gated Residual Fusion: Kết hợp biểu diễn hiện tại với thông tin positional thông qua cơ chế cổng (gating), cho phép mô hình điều chỉnh mức độ đóng góp của PE
 - Transformer Layer: Cập nhật embedding thông qua kiến trúc Transformer riêng biệt cho user và item, giúp khai thác mối liên hệ toàn cục và thứ tự giữa các node.
3. Tổng hợp các lớp: Các embedding qua từng block được gộp lại bằng tổng có trọng số, trong đó các lớp sâu hơn được gán trọng số thấp hơn.

Mã giả hàm forward mô hình Visual-TransGNN:

```

1.Function forward
2.  Input:
3.    adj: ma trận kề user-item
4.  Xử lý:
5.    Khởi tạo embedding ban đầu:
6.      Tính raw_embed từ đặc trưng gốc bằng raw_embedding_proj
7.      Ghép user_embedding và item_embedding → learned_embed
8.      Kết hợp 2 loại embedding:
9.      combined_embed = alpha * raw_embed + (1 - alpha) * learned_embed
10.     Truyền qua các khối GNN:
11.       Với mỗi block trong block_num:
12.         Attention Sampling:
13.           Lấy các sampled_nodes và adj_topk tương ứng
14.           Nếu lỗi → dùng toàn bộ adj
15.         Message Passing:
16.           Lan truyền thông tin: current_embeds = gnn_message_passing(adj_topk,
embeds_prev)
17.         Positional Encoding:
18.           Tính PE cho sampled_nodes bằng pos_encoder(...)
19.         Gated Residual PE:
20.           Áp dụng cổng sigmoid(pe_gate(...))
21.           Cộng PE vào current_embeds tại các node đã sampling
22.         Transformer Layers:
23.           Áp dụng Transformer riêng cho user và item
24.           Ghép lại thành current_embeds
25.           → Lưu current_embeds vào danh sách embeds_list
26.     Tổng hợp embedding:
27.       Trọng số giảm dần theo chiều sâu (layer càng sâu trọng số càng thấp)
28.       final_embeds = tổng có trọng số từ embeds_list

```


29. Output:

30. final_embeds: embedding toàn bộ node

31. user_embeds, item_embeds: tách từ final_embeds

4.3.5 Đánh giá mức độ ảnh hưởng của từng thành phần trong mô hình Visual-TransGNN

Bảng 4.4 So sánh ảnh hưởng các thành phần của Visual-TransGNN trên tập dữ liệu Vibrent

Thành phần	Variants	Recall	NDCG
Visual Similarity Matrix	-VSM	0.1206	0.1009
Attention Sampling	-AS	0.0127	0.0120
Positional Encoding	-PE	0.1331	0.1045
TransGNN	-Trans	0.0033	0.0042
Visual-TransGNN		0.1221	0.0991

Để đánh giá mức độ ảnh hưởng của từng thành phần chính trong kiến trúc Visual-TransGNN, tiến hành một nghiên cứu cắt thành phần (ablation study) bằng cách lần lượt loại bỏ từng mô-đun và quan sát sự thay đổi hiệu suất của mô hình. Kết quả được thể hiện trong Bảng X (Vibrent dataset) với hai chỉ số chính là Recall và NDCG:

- Visual Similarity Matrix (-VSM), độ chính xác Recall giảm nhẹ từ 0.1221 xuống còn 0.1206, trong khi NDCG tăng nhẹ lên 0.1009. Điều này cho thấy ma trận tương đồng hình ảnh đóng vai trò hỗ trợ bổ sung, giúp cải thiện khả năng đánh giá sự phù hợp về mặt hình ảnh giữa các sản phẩm, nhưng mức ảnh hưởng không quá lớn.

- Attention Sampling (-AS) dẫn đến sự sụt giảm nghiêm trọng về hiệu suất, với Recall giảm xuống chỉ còn 0.0127 và NDCG là 0.0120. Điều này nhấn mạnh vai trò quan trọng của việc chọn lọc các nút lân cận có liên quan. Khi không thực hiện sampling, mô hình bị ảnh hưởng bởi nhiễu và mất khả năng tập trung vào bối cảnh quan trọng.
- Positional Encoding (-PE), Recall lại tăng nhẹ lên 0.1331 nhưng NDCG lại giảm còn 0.1045. Điều này cho thấy mô hình có thể truy xuất được nhiều mục hơn, nhưng mức độ liên quan giữa các mục lại kém chính xác hơn, từ đó khẳng định vai trò của các tín hiệu cấu trúc đồ thị trong việc nâng cao chất lượng xếp hạng.
- Transformer (-Trans), hiệu suất mô hình gần như sụp đổ hoàn toàn với Recall chỉ đạt 0.0033 và NDCG là 0.0042. Kết quả này khẳng định rằng cơ chế attention trong Transformer là trung tâm của toàn bộ mô hình, giúp mô hình tổng hợp thông tin một cách hiệu quả.
- Mô hình Visual-TransGNN đạt được hiệu suất cân bằng nhất với Recall là 0.1221 và NDCG là 0.0991. Mỗi thành phần trong kiến trúc đều có đóng góp rõ rệt, và việc loại bỏ bất kỳ phần nào cũng gây ra sự suy giảm đáng kể hoặc bất ổn định trong hiệu suất. Điều này khẳng định tính cần thiết và hiệu quả của kiến trúc tích hợp toàn diện mà mô hình đề xuất.

CHƯƠNG 5 – KẾT QUẢ THỰC NGHIỆM

5.1 Các mô hình nền tảng để so sánh

Để đánh giá hiệu quả của mô hình Visual-TransGNN và LightGBM được đề xuất, chúng tôi tiến hành so sánh với nhiều mô hình đối chứng nổi bật trong lĩnh vực hệ thống gợi ý. Những mô hình này đại diện cho các hướng tiếp cận phổ biến, bao gồm các mô hình dựa trên phân rã ma trận, học sâu với mạng nơ-ron đồ thị, mô hình sử dụng đặc trưng hình ảnh, và các thuật toán học máy truyền thống. Cụ thể:

- BPR-MF (Rendle et al., 2009): Đây là một trong những mô hình khuyến nghị nền tảng, kết hợp kỹ thuật phân rã ma trận với mục tiêu tối ưu hóa xếp hạng cặp thông qua hàm mất mát BPR. Mô hình học được sở thích tương đối giữa các cặp sản phẩm dựa trên phản hồi ẩn (implicit feedback) của người dùng. Phương pháp này tuy đơn giản nhưng vẫn được sử dụng rộng rãi do hiệu quả và dễ triển khai.
- NGCF (Wang et al., 2019): Mô hình này sử dụng mạng nơ-ron đồ thị (GNN) để khai thác cấu trúc đồ thị tương tác người dùng – sản phẩm. Bằng cách lan truyền thông tin qua nhiều bước (multi-hop) và áp dụng các lớp biến đổi phi tuyến, NGCF có khả năng học được biểu diễn ngữ nghĩa phong phú từ các kết nối trong đồ thị.
- LightGCN (He et al., 2020): LightGCN là phiên bản tối giản của GCN, loại bỏ các thành phần phi tuyến như hàm kích hoạt và tầng biến đổi để giảm độ phức tạp mô hình. Thay vào đó, mô hình chỉ giữ lại quá trình lan truyền (propagation) giữa các nút trong đồ thị. Cách tiếp cận này mang lại hiệu suất cao nhờ giảm thiểu nhiễu và tăng cường khả năng tổng quát hóa trong các hệ thống gợi ý dựa trên cộng tác.

- Image-embed (Borgersen et al., 2024): Đây là mô hình sử dụng các vector đặc trưng hình ảnh được trích xuất từ mô hình CNN được huấn luyện trước (pre-trained). Mỗi sản phẩm được biểu diễn bằng embedding hình ảnh, sau đó tính toán độ tương đồng giữa các sản phẩm để đưa ra gợi ý. Mô hình này không sử dụng thông tin tương tác người dùng, do đó chủ yếu dựa trên độ giống về mặt thị giác để đề xuất sản phẩm.
- Visual-GCN (Van & Tran, 2025): chỉ giữ lại ma trận tương đồng hình ảnh (visual similarity matrix) mà không áp dụng lan truyền thông tin qua mạng đồ thị. Việc so sánh với Visual-GNN giúp làm rõ ảnh hưởng riêng biệt của tín hiệu thị giác trong toàn bộ kiến trúc mô hình gợi ý.

5.2 Kết quả và đánh giá

Bảng 5.1 So sánh hiệu suất các mô hình với topK=5

Dataset	Vibrent Clothes Rental				H&M Fashion Dataset			
	recall	ndcg	mrr	hit_rate	recall	ndcg	mrr	hit_rate
BPR-MF	0,00137	0,01175	0,00891	0,02174	0,00302	0,02675	0,02444	0,04592
NGCF	0,00082	0,01373	0,0133	0,02174	0,0001	0,00012	0,0001	0,00015
LightGCN	0,00125	0,00371	0,00011	0,00023	0,00162	0,00473	0,0001	0,00018
Image-embed	0,00731	0,0102	0,0143	0,04541	0,04193	0,03712	0,0273	0,06777
VisualGCN	0,01269	0,01738	0,02369	0,06605	0,02837	0,04146	0,01616	0,08327

LightGBM	<u>0,05101</u>	<u>0,02984</u>	0,06981	<u>0,09448</u>	0,05668	<u>0,03872</u>	<u>0,0548</u>	<u>0,09044</u>
Visual-TransGNN	0,07213	0,09331	<u>0,03856</u>	0,20875	<u>0,04951</u>	0,05103	0,02225	0,09392

Bảng 5.2 Bảng So sánh hiệu suất các mô hình với topK=20

Dataset	Vibrent Clothes Rental				H&M Fashion Dataset			
	recall	ndcg	mrr	hit_rate	recall	ndcg	mrr	hit_rate
BPR-MF	0,00417	0,01902	0,01145	0,06957	0,00879	0,04802	0,03079	0,11291
NGCF	0,00516	0,01333	0,00686	0,05021	0,00001	0,00018	0,00012	0,00038
LightGCN	0,00593	0,00537	0,00016	0,00085	0,00449	0,0046	0,00013	0,00048
Image-embed	0,04591	0,01719	0,02382	0,23982	0,05298	0,05864	0,03155	0,08447
VisualGCN	0,07151	0,02556	0,03492	0,30823	0,0395	0,03883	0,01864	0,0957
LightGBM	<u>0,06927</u>	<u>0,03064</u>	0,07686	<u>0,14173</u>	<u>0,0699</u>	<u>0,04465</u>	0,06017	0,1441
Visual-TransGNN	0,12773	0,11068	<u>0,04235</u>	0,51301	0,07221	0,07064	<u>0,03168</u>	<u>0,14353</u>

Để phân tích mức độ hiệu quả của các mô hình đề xuất, hai bảng 5.1 và 5.2 trình bày kết quả đánh giá hiệu suất của các phương pháp gợi ý trên hai tập dữ liệu Vibrent Clothes Rental và H&M Fashion Dataset, tương ứng với hai ngưỡng đánh giá phổ biến là Top-5 và Top-20. Bốn chỉ số đánh giá chính được sử dụng bao gồm: Recall, NDCG,

MRR và Hit Rate. Trong số các mô hình, LightGBM và Visual-TransGNN nổi bật nhất với hiệu suất cao hơn đáng kể so với các mô hình còn lại.

5.2.1 Đánh giá hiệu suất mô hình với $topK=5$

Trên tập Vibrent Clothes Rental, mô hình Visual-TransGNN cho kết quả tốt nhất ở cả bốn chỉ số, với Recall = 0.07213, NDCG = 0.09331, và Hit Rate = 0.20875. Trong khi đó, LightGBM tuy đạt MRR = 0.06981 cao hơn Visual-TransGNN (0.03856), nhưng các chỉ số còn lại như Recall và NDCG đều thấp hơn đáng kể (Recall = 0.05101, NDCG = 0.02984). Điều này cho thấy LightGBM có thể sắp xếp thứ hạng một số item tốt hơn nhưng không đủ khả năng thu hồi các item phù hợp toàn diện như Visual-TransGNN.

Trên tập H&M Fashion Dataset, kết quả tương tự được ghi nhận. LightGBM vẫn giữ MRR cao nhất (0.05480), nhưng Visual-TransGNN lại vượt trội về NDCG (0.05103) và Hit Rate (0.09392), cho thấy khả năng gợi ý các sản phẩm phù hợp hơn ở những vị trí đầu danh sách.

5.2.2 Đánh giá hiệu suất mô hình với $topK=20$

Với $topK=20$, khoảng cách hiệu suất giữa Visual-TransGNN và LightGBM còn rõ ràng hơn. Trên Vibrent Clothes Rental, Visual-TransGNN đạt Recall = 0.12773, NDCG = 0.11068, và Hit Rate = 0.51301 – cao vượt trội so với LightGBM (Recall = 0.06927, NDCG = 0.03064, Hit Rate = 0.14173). Đây là minh chứng rõ ràng cho khả năng mở rộng phạm vi gợi ý hiệu quả của mô hình đề xuất.

Tương tự, trên H&M, Visual-TransGNN đạt Recall = 0.07221 và NDCG = 0.07064, cao hơn đáng kể so với LightGBM (Recall = 0.06990, NDCG = 0.04465). Mặc dù LightGBM vẫn duy trì MRR nhỉnh hơn (0.06017 so với 0.03168), nhưng nhìn chung, hiệu suất toàn diện của Visual-TransGNN cao hơn ở các chỉ số then chốt như Recall và

NDCG – vốn thể hiện năng lực phát hiện và xếp hạng các sản phẩm phù hợp với người dùng.

5.2.3 Đánh giá tổng quan

Qua việc dùng các chỉ số Recall, NDCG, Hit Rate, MRR để đánh giá các mô hình ở topK=5 và topK=20, có thể thấy rằng:

- Visual-TransGNN có hiệu suất toàn diện và ổn định hơn, đặc biệt ở Recall, NDCG và Hit Rate, nhờ khả năng kết hợp thông tin hình ảnh và cấu trúc đồ thị người dùng – sản phẩm.
- LightGBM, dù có lợi thế nhất định ở chỉ số MRR nhờ khả năng học theo cây tăng cường, nhưng thiếu khả năng xử lý các đặc trưng đa chiều như quan hệ trực quan giữa sản phẩm và cấu trúc lân cận của đồ thị tương tác.

Như vậy, mô hình Visual-TransGNN cho thấy tiềm năng ứng dụng thực tế vượt trội trong các hệ thống khuyến nghị thời trang, đặc biệt trong bối cảnh đòi hỏi sự kết hợp giữa yếu tố thị giác và hành vi người dùng.

CHƯƠNG 6 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

6.1 Kết luận

Luận văn đã hoàn thành việc xây dựng và đánh giá hai hướng tiếp cận hiện đại trong hệ thống khuyến nghị: LightGBM – một mô hình học máy tăng cường dựa trên cây quyết định, và Transformer ứng dụng trong đồ thị – thông qua mô hình Visual-TransGNN. Hai mô hình được triển khai độc lập, nhằm khảo sát khả năng cải tiến chất lượng gợi ý trong bối cảnh dữ liệu thời trang phức tạp và mang tính cá nhân hóa cao.

Trong hướng tiếp cận thứ nhất, LightGBM cho thấy hiệu quả rõ rệt trong việc xử lý các đặc trưng phi tuyến như hành vi người dùng, lịch sử mua sắm và embedding ảnh sản phẩm. Mô hình có khả năng huấn luyện nhanh, tiêu tốn ít tài nguyên và dễ tích hợp vào pipeline có nhiều đặc trưng đa dạng. Các kết quả thực nghiệm khẳng định rằng LightGBM là lựa chọn hiệu quả cho các hệ thống cân cân bằng giữa tốc độ và độ chính xác.

Ở hướng tiếp cận thứ hai, luận văn áp dụng Transformer trong học biểu diễn đồ thị để xây dựng mô hình Visual-TransGNN – một kiến trúc có khả năng mở rộng trường tiếp nhận thông tin, học được mối quan hệ sâu giữa người dùng và sản phẩm thông qua nhiều bước tương tác. Transformer giúp lựa chọn các node liên quan một cách linh hoạt hơn so với các mô hình đồ thị truyền thống, đồng thời kết hợp tốt với các đặc trưng hình ảnh từ CNN.

Kết quả thực nghiệm trên hai tập dữ liệu thực tế (Vibrent và H&M) cho thấy cả hai mô hình đều đạt hiệu suất cao, trong đó Visual-TransGNN vượt trội ở các chỉ số Recall và HitRate, còn LightGBM đạt kết quả tốt trên các chỉ số chính xác như MRR và NDCG. Điều này cho thấy mỗi mô hình đều phát huy thế mạnh riêng, và việc áp dụng

Transformer và LightGBM đã góp phần cải thiện rõ rệt hiệu quả gợi ý so với các phương pháp baseline truyền thống.

Quá trình thực hiện luận văn giúp chúng tôi nắm vững kiến thức về hai công cụ quan trọng trong lĩnh vực học máy hiện đại – LightGBM và Transformer, đồng thời rèn luyện kỹ năng xây dựng pipeline, tiền xử lý dữ liệu, thiết kế mô hình và đánh giá hiệu năng. Bên cạnh đó, chúng tôi cũng phát triển tư duy nghiên cứu và khả năng triển khai mô hình vào ngữ cảnh thực tế.

Tuy vẫn còn một số giới hạn như chưa tích hợp thông tin văn bản hoặc metadata, chưa tối ưu hóa cho thời gian thực và giao diện tương tác, nhưng luận văn đã chứng minh rõ ràng tiềm năng của việc cải tiến hệ thống khuyến nghị thông qua hai kỹ thuật cốt lõi là LightGBM và Transformer.

Tóm lại, luận văn đã hiện thực hóa thành công một hệ thống gợi ý sản phẩm thời trang dựa trên hai kỹ thuật tiên tiến, đồng thời mở ra hướng nghiên cứu sâu hơn về các phương pháp kết hợp đa mô hình, đa đặc trưng và đa nguồn dữ liệu trong lĩnh vực thương mại điện tử cá nhân hóa.

6.2 Hướng phát triển trong tương lai

Từ những kết quả đã đạt được, đề tài mở ra nhiều hướng phát triển tiềm năng nhằm nâng cao hơn nữa hiệu quả và khả năng ứng dụng của hệ thống khuyến nghị. Một trong những hướng đi đáng chú ý là việc tích hợp hai mô hình LightGBM và Transformer-based GNN vào trong một kiến trúc thống nhất. Trong nghiên cứu hiện tại, hai phương pháp được triển khai độc lập để đánh giá hiệu quả riêng biệt. Tuy nhiên, việc kết hợp điểm mạnh của mỗi mô hình có thể mang lại lợi ích đáng kể. Cụ thể, các embedding thu được từ mô hình dựa trên Transformer có thể được sử dụng làm đặc trưng đầu vào cho LightGBM – một mô hình có khả năng xử lý tốt dữ liệu phi tuyến và kết hợp nhiều loại đặc trưng khác nhau. Điều này không chỉ tận dụng được khả năng học

biểu diễn mạnh mẽ từ mạng nơ-ron mà còn khai thác hiệu quả tính linh hoạt và khả năng tổng hợp đặc trưng của mô hình cây tăng cường.

Bên cạnh đó, việc mở rộng đầu vào mô hình theo hướng đa phương thức là một lựa chọn cần được ưu tiên trong các bước phát triển tiếp theo. Hiện tại, hệ thống chủ yếu dựa vào embedding hình ảnh và quan hệ người dùng – sản phẩm được biểu diễn dưới dạng đồ thị. Trong thực tế, dữ liệu về sản phẩm và người dùng rất phong phú, bao gồm cả văn bản mô tả, phản hồi, đánh giá, nhãn phân loại, thông tin hành vi, hoặc thậm chí là thời gian tương tác. Việc tích hợp thêm những nguồn dữ liệu này sẽ giúp mô hình có cái nhìn toàn diện và giàu ngữ cảnh hơn, từ đó nâng cao chất lượng gợi ý và đáp ứng tốt hơn nhu cầu cá nhân hóa.

Một vấn đề khác cần được quan tâm là hiệu năng triển khai thực tế của mô hình, đặc biệt là các kiến trúc sử dụng Transformer. Mặc dù Transformer mang lại khả năng học mối quan hệ toàn cục và hiệu quả trong việc chọn lọc thông tin, nhưng độ phức tạp tính toán của attention toàn cục lại là một rào cản lớn khi triển khai trên quy mô lớn. Do đó, việc áp dụng các kỹ thuật tối ưu như sparse attention, local attention hoặc sử dụng các biến thể nhẹ hơn của Transformer như Linformer, Performer,... sẽ là hướng phát triển thiết thực giúp cải thiện tốc độ suy diễn mà vẫn giữ được hiệu suất gợi ý.

Song song với đó, việc xây dựng hệ thống có khả năng mở rộng (scalability) cũng là yếu tố quan trọng. Trong môi trường thương mại điện tử thực tế, số lượng người dùng và sản phẩm có thể lên đến hàng triệu, điều này đòi hỏi mô hình phải được thiết kế theo hướng phân tán, hỗ trợ huấn luyện và cập nhật liên tục với tốc độ cao. Các chiến lược như mini-batch sampling, incremental learning hoặc cơ chế cache để lưu trữ các embedding thường dùng sẽ đóng vai trò then chốt trong việc hiện thực hóa khả năng triển khai mô hình vào môi trường thật.

Cuối cùng, để đánh giá chính xác hiệu quả của hệ thống khuyến nghị, cần mở rộng phương pháp đánh giá theo hướng định tính, chẳng hạn như khảo sát mức độ hài lòng của người dùng hoặc thử nghiệm A/B trực tiếp trên hệ thống. Các chỉ số định lượng như Recall, NDCG hay MRR tuy phản ánh hiệu năng mô hình ở mức độ thống kê, nhưng đôi khi không phản ánh đầy đủ trải nghiệm thực tế của người dùng. Do đó, kết hợp giữa đánh giá định lượng và định tính sẽ mang lại cái nhìn toàn diện hơn, từ đó định hướng tốt hơn cho việc tinh chỉnh mô hình trong tương lai.

Tổng thể, đề tài đã đặt nền móng cho một hệ thống khuyến nghị hiện đại dựa trên hai hướng tiếp cận mạnh mẽ là LightGBM và Transformer. Tuy nhiên, những gợi ý trên cho thấy vẫn còn nhiều không gian để khai thác nhằm tiếp tục cải thiện độ chính xác, tốc độ xử lý và khả năng thích ứng với dữ liệu thực tế phức tạp – đặc biệt trong bối cảnh các nền tảng thương mại điện tử đang không ngừng phát triển và đòi hỏi ngày càng cao về khả năng cá nhân hóa khuyến nghị.

TÀI LIỆU THAM KHẢO

1. Deldjoo, Y., Elahi, M., & Cremonesi, P. (2023). *Multimodal recommender systems: A comprehensive survey and future research directions*. ACM Computing Surveys (CSUR), 55(7), 1–39. <https://doi.org/10.1145/3695461>
2. Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to recommender systems handbook*. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 1–35). Springer. https://doi.org/10.1007/978-0-387-85820-3_1
3. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). *A comprehensive survey on graph neural networks*. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24. <https://doi.org/10.1109/TNNLS.2020.2978386>
4. Zhang, X., Liu, Y., Wang, Y., Liu, H., & Zhang, M. (2024). *TransGNN: Harnessing the Collaborative Power of Transformers and Graph Neural Networks for Recommender Systems*. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. ACM. <https://doi.org/10.1145/3626772.3657721>
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). *Neural collaborative filtering*. In *Proceedings of the 26th International Conference on World Wide Web (WWW)* (pp. 173–182). <https://doi.org/10.1145/3038912.3052569>
6. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). *LightGBM: A highly efficient gradient boosting decision tree*. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 3146–3154).
7. Kipf, T. N., & Welling, M. (2017). *Semi-supervised classification with graph convolutional networks*. In *International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1609.02907>

8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). *Attention is all you need*. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 6000–6010). <https://arxiv.org/abs/1706.03762>
9. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. Q. (2020). *Simplifying graph convolutional networks*. In *International Conference on Machine Learning (ICML)* (pp. 6861–6871). <https://arxiv.org/abs/1902.07153>
10. Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2018). *A visual-aware recommender system for clothing e-commerce*. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(4), 1–23. https://doi.org/10.1007/978-3-030-92185-9_24
11. Van, M.M., Tran, T.T.: *Belightrec: A lightweight recommender system enhanced with bert*. In: *Thai-Nghe, N., Do, T.N., Benferhat, S. (eds.) Intelligent Systems and Data Science*. pp. 30-43. Springer Nature Singapore, Singapore (2025). <https://arxiv.org/abs/2503.20206>
12. Borgersen, K.A.K., Goodwin, M., Grundetjern, M., Sharma, J.: *A dataset for adapting recommender systems to the fashion rental economy*. In: *18th ACM Conference on Recommender Systems*. pp. 945{950. ACM, Bari, Italy (2024). <https://doi.org/10.1145/3640457.3688174>
13. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: *Lightgcn: Simplifying and powering graph convolution network for recommendation*. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. p. 639-648. SIGIR '20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3397271.3401063>
14. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: *Neural graph collaborative filtering*. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. p. 165-174. SIGIR'19, Association for

- Computing Machinery, New York, NY, USA (2019).
<https://doi.org/10.1145/3331184.3331267>
15. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: *Bayesian personalized ranking from implicit feedback*. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*. pp. 452-461. UAI '09, AUAI Press, Arlington, Virginia, USA (2009). <https://arxiv.org/abs/1205.2618>
 16. kaborg15. (2023). *Vibrent Clothes Rental Dataset*. Kaggle.
<https://www.kaggle.com/datasets/kaborg15/vibrent-clothes-rental-dataset>
 17. H&M Group. (2022). *H&M Personalized Fashion Recommendations*. Kaggle.
<https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations>
 18. Velićković, P., Cucurull, G., Casanova, A., Romero, A., Li'o, P., Bengio, Y.: *Graph attention networks* (2018). <https://arxiv.org/abs/1710.10903>
 19. Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., Liu, T.Y.: *Do transformers really perform bad for graph representation?* In: *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS 2021)*. NIPS '21, Curran Associates Inc., Red Hook, NY, USA (2021).
<https://doi.org/10.48550/arXiv.2106.05234>
 20. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: *Graph convolutional neural networks for web-scale recommender systems*. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. p. 974-983. KDD '18, Association for Computing Machinery, New York, NY, USA (2018).
<https://doi.org/10.1145/3219819.3219890>

21. Zzh, Zhang, W., Wentao: *Industrial solution in fashion-domain recommendation by an efficient pipeline using gnn and lightgbm*. In: *Proceedings of the Recommender Systems Challenge 2022*. p. 45-49. RecSysChallenge'22, Association for Computing Machinery, New York, NY, USA (2022).
<https://doi.org/10.1145/3556702.3556850>