

# Hệ gợi ý mua sắm dựa theo phiên làm việc với mô hình mạng học sâu đồ thị

Nguyễn Tuấn Khang<sup>1</sup>, Nguyễn Tú Anh<sup>2</sup>, Mai Thúy Nga<sup>2</sup>, Nguyễn Hải An<sup>3</sup>, Nguyễn Việt Anh<sup>1</sup>

<sup>1</sup> Viện Công nghệ Thông Tin, Học Viện Khoa Học Công Nghệ, Hà Nội

<sup>2</sup> Trường Đại học Thăng Long, Hà Nội

<sup>3</sup> Phòng Khoa Học Công Nghệ, Tổng Công Ty Thăm Dò Khai Thác Dầu Khí, Hà Nội

Tác giả liên hệ: Nguyễn Tuấn Khang, khang\_nt@yahoo.com

Ngày nhận bài: 30/08/2022, ngày sửa chữa: 13/11/2022, ngày duyệt đăng: 25/11/2022

Định danh DOI: 10.32913/mic-ict-research-vn.v2022.n2.1135

**Tóm tắt:** Phân tích phiên làm việc của khách hàng để dự báo khả năng họ sẽ lựa chọn sản phẩm nào tiếp theo là một bài toán dự báo khá phổ biến trong ngành thương mại điện tử. Việc dự báo này giúp cho doanh nghiệp đưa ra các ý tưởng bán hàng phù hợp trong quá trình người dùng tương tác với hệ thống bán hàng của mình. Bài báo này đề xuất hướng sử dụng mạng học sâu đồ thị để xây dựng mô hình gợi ý dựa vào phiên làm việc của khách hàng. Kết quả thực nghiệm cho thấy việc sử dụng đồ thị rất phù hợp trong việc biểu diễn dữ liệu lựa chọn sản phẩm thông qua hành vi nhấp chuột trong phiên làm việc của khách hàng và mô hình gợi ý sử dụng GNN cho kết quả dự báo với 2 chỉ số đánh giá mô hình Recall@20 và MRR@20 tốt hơn so với các mô hình trước đây.

**Từ khóa:** dữ liệu nhấp chuột, hành vi mua sắm, hệ thống gợi ý, phiên làm việc, mạng học sâu đồ thị (Graph Neural Network).

---

**Title:** Session-based Recommendation using Graph Neural Network

**Abstract:** Customer behavior analysis based on the current active session to understand and predict what is the next product that customer might click is a potential usecase in ecommerce. This type of recommendation helps enterprise to promote the upselling opportunity to increase the purchase behavior. This paper proposes to use a GNN to develop a recommendation model using the current active session of customer during their purchasing clicks on the website. The experimental result shows that the GNN is suitable to model a product selection from sequential mouse clicks, and the session-based recommendation using GNN performs higher than other models with the two performance metrics Recall@20 and MRR@20.

**Keywords:** mouse click, purchase behaviour, recommendation system, session, graph neural network (GNN)

---

## I. TỔNG QUAN

### 1. Tổng quan bài toán

Khi một khách hàng vào một trang thương mại điện tử thì có hai xu hướng: hoặc họ đã định hướng được sản phẩm mà họ sẽ mua, hoặc là họ được định hướng được sản phẩm mà họ nên mua. Đối với kịch bản thứ hai, người dùng sẽ gặp khó khăn hơn nhiều vì họ sẽ phải chọn sản phẩm phù hợp nhất với nhu cầu của họ. Vấn đề đặt ra là làm sao họ có thể làm được điều đó trong vô số sản phẩm giống nhau mà họ đang tìm kiếm, đó chính là ý tưởng xây dựng hệ thống gợi ý [1].

Các hệ thống gợi ý ngày nay càng được chú trọng, nhất là đối với các nhà cung cấp dịch vụ trực tuyến như: Amazon, Netflix [2], Youtube... Một hệ thống gợi ý hiệu quả sẽ là

vấn đề sống còn đối với nhà cung cấp dịch vụ hoặc bán hàng, nó có thể làm tăng sự hài lòng của khách hàng và giữ chân người dùng lâu dài [3].

Hiện nay có hai hướng để xây dựng hệ thống gợi ý tùy theo ngữ cảnh tương tác người dùng như sau:

- Hệ gợi ý dựa vào thông tin lịch sử hoặc sở thích của người dùng đã để lại để tìm ra sản phẩm phù hợp nhất, hệ thống hoạt động kiểu này khá dễ hiểu nhưng lại gặp nhiều thách thức khi cần đưa ra gợi ý cho người dùng ngay cả khi họ không để lại thông tin lịch sử gì cho hệ thống.
- Hệ gợi ý dựa chỉ dựa vào quá trình tương tác hiện tại của người dùng với hệ thống, gọi là phiên làm việc, nhằm cho phép hệ thống có thể đưa ra gợi ý cho người

dùng chỉ sau vài ba chuỗi sự kiện tương tác của họ với hệ thống, mô hình này được gọi là hệ thống gợi ý dựa vào phiên làm việc (*Session-based Recommendation*), gọi tắt là bài toán SR.

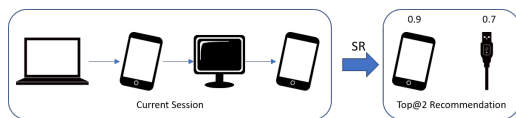
## 2. Đặt vấn đề

Đối tượng nghiên cứu của bài báo này là hành vi nhấp chuột (lựa chọn sản phẩm) của khách hàng trong một phiên mua hàng. Mục tiêu của bài báo này là nghiên cứu và đề xuất mô hình dự báo hành vi lựa chọn sản phẩm của khách hàng mô hình mạng học sâu, cụ thể hơn là trả lời câu hỏi "Với số lượng sản phẩm đã lựa chọn trong phiên tương tác hiện tại thì khả năng người dùng sẽ chọn sản phẩm nào tiếp theo". Ở mức độ tổng quan, mô hình gợi ý sẽ không chỉ đưa ra một sản phẩm tiếp theo mà sẽ đưa ra danh sách gợi ý  $K$  sản phẩm có xác suất cao nhất mà khách hàng có thể lựa chọn một, bài toán này còn gọi là bài toán gợi ý Top-K, ngắn gọn gọi là bài toán Top-K.

Bài toán SR có thể được mô tả như sau, giả sử  $V = \{v_1, v_2, \dots, v_m\}$  là một danh mục các đối tượng duy nhất (ví dụ như danh mục sản phẩm) được các người dùng tương tác trong các phiên làm việc của họ, như vậy ta có thể thể phiên làm việc được biểu diễn như sau  $s = [v_{s,1}, v_{s,2}, \dots, v_{s,n}]$  trong đó  $v_{s,i} \in V$  có tính thứ tự theo chuỗi thời gian để và thể hiện một hành động lựa chọn (click) nào đó của người dùng trong phiên làm việc  $s$  (ví dụ chọn một sản phẩm cụ thể).

Để giải quyết vấn đề này, ta cần xây dựng mô hình dự báo liệu người dùng sẽ lựa chọn đối tượng (sản phẩm)  $v_{s,n+1}$  tiếp theo trong phiên làm việc  $s$  đó. Với mô hình gợi ý này cho một phiên làm việc  $s$  cụ thể, hệ gợi ý sẽ trả về hàm ý là một véc-tơ chứa danh mục  $k$  sản phẩm gợi ý nào đó với xác suất được lựa chọn từ cao tới thấp, danh mục sản phẩm gợi ý này được gọi là *top-k* sản phẩm gợi ý cho người dùng.

Hình 1 minh họa mô hình SR đưa ra dự báo *top-2* sản phẩm cùng xác suất tương ứng mà khách hàng sẽ lựa chọn để click tiếp.



Hình 1. Bài toán gợi ý top-k sản phẩm

## II. NGHIÊN CỨU LIÊN QUAN

Bài toán gợi ý trong lĩnh vực thương mại điện tử không phải là vấn đề mới, ngay từ những năm 2000 JB Schafer và cộng sự [1] đã nêu ra vấn đề này để tìm cách nâng

cao khả năng bán kèm và bán chéo sản phẩm cho các website bán hàng thời đó, thuật toán được đề xuất là sử dụng các thông tin trong quá khứ cũng như sở thích cá nhân (*personalization*) của khách hàng để đề xuất sản phẩm cần bán trong tương lai. Sau đó, nhóm tác giả này tiếp tục cải thiện mô hình gợi ý với ý tưởng sử dụng dữ liệu tri thức và sự tương quan (*correlation*) giữa các sản phẩm hay giữa các người dùng để phân tích hành vi khách hàng (*customer behavior*) [4] với việc phân tích các hệ thống thương mại điện tử thời đó như amazon hay ebay.

Badrul M. Sarwar và cộng sự (2000) [5] nhận thấy các hệ thống gợi ý đang phải xử lý khối lượng thông tin khổng lồ về sản phẩm, khách hàng, đơn hàng gây ra thách thức trong việc đưa ra gợi ý, nhóm tác giả đề xuất một trong những thuật toán phổ biến của học máy là SVD (*Singular Value Decomposition*) với mục tiêu giảm số chiều thông tin để tăng tốc độ xử lý của hệ thống gợi ý, nghiên cứu này cũng đưa ra khái niệm "*top N list*" của mô hình gợi ý. Năm 2002, nhóm tác giả này tiếp tục đề xuất sử dụng thuật toán người láng giềng (*neighborhood*) [6] để phân nhóm khách hàng (*clustering*) từ đó xây dựng mô hình gợi ý theo mô hình bộ lọc cộng tác (*collaborative filtering*).

Năm 2004, Zan Huang và cộng sự [7] đưa khái niệm đồ thị vào bài toán gợi ý trong lĩnh vực thương mại điện tử, nhóm tác giả đề xuất đồ thị có hướng gồm hai lớp: lớp sản phẩm và lớp khách hàng mà mối quan hệ giữa hai lớp thể hiện thông tin mua sắm trong quá khứ thông qua trọng số của đồ thị. Mô hình đồ thị được thực nghiệm với cả ba kỹ thuật gợi ý gồm sử dụng bộ lọc cộng tác, dựa vào nội dung và hướng kết hợp, kết quả cho thấy mô hình này hoạt động tốt nhất với kỹ thuật kết hợp.

Năm 2006, Netflix tổ chức cuộc thi tìm kiếm giải thuật gợi ý tốt nhất nhằm dự đoán điểm đánh giá của người dùng cho các bộ phim của họ (*user ratings*) dựa vào các đánh giá trước đây mà không sử dụng thêm thông tin gì về người dùng hay bộ phim, đây là bài toán gợi ý lọc cộng tác. Yehuda Koren, Robert Bell và Chris Volinsky là thành viên đội thắng cuộc năm 2019 [2] trình bày mô hình phân tích ma trận thành nhân tử (*matrix factorization*) hoạt động tốt hơn các thuật toán của đối thủ khác như thuật toán SVD hoặc người láng giềng. Mô hình phân tích ma trận thành nhân tử tìm cách phân rã hai véc tơ đại diện cho người dùng ( $p_u$ ) và bộ phim ( $q_i$ ) (còn được gọi là véc tơ nhân tử, *factor vector*) vào một không gian nhân tử riêng (*joint-latent-factor space*), và vấn đề của mô hình là sao cho học được véc tơ nhân tử  $p_u$  và  $q_i$  với sai số *RMSE* (*root-mean-square error*) nhỏ nhất.

Balázs Hidasi và cộng sự (2015) [8] đưa ra mô hình mạng hồi quy RNN trong việc xây dựng hệ gợi ý, khác với bài toán của Netflix cần thông tin quá khứ của người dùng, hướng tiếp cận của nghiên cứu này tập

trung vào những phiên làm việc ngắn và hiện tại để đưa ra gợi ý cho người dùng, đó là khái niệm *Session-based Recommendation*. Mô hình này sử dụng thuật toán RNN phân cấp để tìm ra các đặc trưng ẩn trong phiên làm việc hiện tại của người dùng để đưa ra gợi ý cho sản phẩm tiếp theo, thuật toán RNN rất phù hợp với bài toán khi phải xử lý chuỗi sự kiện tuần tự (ví dụ như chuỗi nhấp chuột của người dùng trong phiên làm việc đó khi lựa chọn mua một số sản phẩm nào đó), nghiên cứu cho thấy mô hình RNN với biến thể HRNN (*Hierarchical RNN*) hoạt động tốt hơn so với các mô hình truyền thống cũng như mô hình RNN cơ sở.

Kể thừa nghiên cứu của Hidası, Yong Kiam Tan và cộng sự (2016) [9] đã đề xuất cải tiến mô hình RNN với thuật toán xử lý dữ liệu làm việc theo chuỗi (phiên) phù hợp hơn cho mô hình RNN. Y.K. Tan và cộng sự cũng sử dụng chung bộ dữ liệu của Hidası nhưng đã thực nghiệm đa dạng hơn để phân tích mức độ hiệu quả của việc xử lý dữ liệu phiên làm việc với mô hình RNN. Kết quả cho thấy nghiên cứu của Y.K. Tan và cộng sự cho kết quả tốt hơn và thuật toán xử lý dữ liệu được sử dụng tham chiếu trong một số nghiên cứu tiếp theo về bài toán này [10–12]

Với sự ra đời của mô hình mạng học sâu và rộng (*Wide & Deep Learning*) do Google phát triển năm 2016, Heng-Tze Cheng và cộng sự [13], Khang Nguyen và cộng sự [14] cũng áp dụng mô hình này trong việc cải thiện tính tương tác giữa các thuộc tính ở cả mức cao và mức thấp để giúp cho mô hình gợi ý có thể tìm ra được các đặc tính ẩn tốt hơn do nó vừa có tính tổng quát hóa của mô hình học rộng vừa có tính ghi nhớ của mô hình học sâu.

Shu Wu và cộng sự (2019) [12] sử dụng mô hình mạng học sâu và đồ thị cùng khá nhiều kỹ thuật xử lý đồ thị và các biến thể khác nhau của GNN để phân tích bài toán gợi ý dựa vào phiên làm việc. Nhóm tác giả đề xuất kỹ thuật biến đổi véc tơ phiên làm việc sang một không gian nhúng bằng cách sử dụng mạng GNN để huấn luyện và học được véc tơ nhúng của đồ thị biểu diễn phiên làm việc, các véc tơ nhúng này thể hiện được các đặc tính ẩn của từng phiên làm việc từ đó hỗ trợ đưa ra được gợi ý có tính chính xác hơn.

### III. PHƯƠNG PHÁP LUẬN

## 1. Ý tưởng xây dựng đồ thi

Ý tưởng của nghiên cứu là đề xuất một số phương án xây dựng đồ thị từ bộ dữ liệu lựa chọn sản phẩm của người mua hàng. Gọi  $d$  là số lượng sản phẩm trong bộ dữ liệu và các sản phẩm được đánh dấu theo số thứ tự từ 0 đến  $d - 1$ .

Cu thể nhóm tác giả đề xuất 3 dạng đề thi sau:

- **Đồ thị G:** đồ thị đơn thể hiện mối quan hệ trực tiếp khi lựa chọn sản phẩm.

- **Đồ thị H:** đồ thị đơn thể hiện cả mối quan hệ trực tiếp và gián tiếp khi lựa chọn các sản phẩm trong phiên làm việc.
- **Đồ thị K:** đồ thị đa quan hệ (còn gọi là đồ thị sâu) thể hiện đồng thời nhiều mối quan hệ khác nhau giữa các sản phẩm thuộc nhiều phiên làm việc khác nhau trong bộ dữ liệu.

*Đồ thi G*

Gọi  $G$  là một đồ thị thoả mãn ma trận kề  $M_G \in \mathbb{R}^{d \times d}$  với  $M_G^{i,j}$  là số lần sản phẩm  $j$  được nhấp (click) ngay sau sản phẩm  $i$  trong một phiên. Khái niệm *ngay sau* thể hiện sự tương tác trực tiếp từ khi nhấp từ sản phẩm  $i$  tới sản phẩm  $j$ .

Giả sử  $s = \{v_1, v_2, \dots, v_n\}$  đại diện cho một phiên làm việc với  $v_i$  ( $0 \leq v_i < d$ ) là đại diện cho một sản phẩm được nhập thứ  $i$ . Lúc đó ta thực hiện, với mỗi  $i$  ( $1 \leq i < n$ ):

$$M_G^{v_i, v_{i+1}} \leftarrow M_G^{v_i, v_{i+1}} + 1 \quad (1)$$

Nhân xét:

- $G$  là đồ thị có hướng, có trọng số.
- Trong  $G$ , trọng số cạnh nối từ đỉnh  $i$  tới  $j$  có giá trị là  $M_G^{i,j} \in \mathbb{R}$
- Theo thống kê, xác suất của để sản phẩm  $j$  được nhấp ngay sau sản phẩm  $i$  là:

$$P_G^{i,j} = \frac{M_G^{i,j}}{\sum_{x=0}^{d-1} M_G^{i,x}}$$

*Đồ thi H*

Gọi  $H$  là một đồ thị thoả mãn ma trận kề  $M_H \in \mathbb{R}^{d \times d}$  với  $M_H^{i,j}$  là số lần sản phẩm  $j$  được nhấp (click) sau khi nhấp sản phẩm  $i$  trong một phiên. Khái niệm sau khi thể hiện sự tương tác trực tiếp hoặc gián tiếp từ khi nhấp từ sản phẩm  $i$  tới sản phẩm  $j$  có thể đi qua một số sản phẩm nào khác giữa hai sản phẩm này.

Giả sử  $s = \{v_1, v_2, \dots, v_n\}$  đại diện cho một phiên làm việc với  $v_i$  ( $0 \leq v_i < d$ ) là đại diện cho một sản phẩm được nhập thứ  $i$ . Lúc đó ta thực hiện, với mỗi  $i, j$  ( $1 \leq i < j < n$ ):

$$M_H^{v_i, v_j} \leftarrow M_H^{v_i, v_j} + 1 \quad (2)$$

Nhân xét:

- $H$  là đồ thị có hướng, có trọng số.
- Trong  $H$ , trọng số cạnh nối từ đỉnh  $i$  tới  $j$  có giá trị là  $M_H^{i,j} \in \mathbb{R}$
- $M_G^{i,j} \leq M_H^{i,j} \quad \forall i, j : 0 \leq i < j < d$
- Theo thống kê, xác suất của để sản phẩm  $j$  được nhập sau sản phẩm  $i$  là:

$$P_H^{i,j} = \frac{M_H^{i,j}}{\sum_{x=0}^{d-1} M_H^{i,x}}$$

- Gọi  $E_X$  là số lượng cạnh của đồ thị  $X$ . Ta có,  $E_H > E_G$ .

### Đồ thị $K$

Gọi  $c$  là số lượng nhấp nhiều nhất của một phiên trong cơ sở dữ liệu. Gọi  $K$  là một đồ thị thỏa mãn ma trận kề  $M_K \in \mathbb{R}^{d \times d \times c}$  với  $M_K^{i,j}[t]$  là số lần sản phẩm  $j$  được nhấp (click) sau khi nhấp sản phẩm  $i$  đúng  $t$  lần nhấp trong một phiên.

Giả sử  $s = \{v_1, v_2, \dots, v_n\}$  đại diện cho một phiên làm việc với  $v_i$  ( $0 \leq v_i < d$ ) là đại diện cho một sản phẩm được nhấp thứ  $i$ . Lúc đó ta thực hiện, với mỗi  $i, j$  ( $1 \leq i < j < n$ ):

$$M_K^{v_i, v_j}[j - i - 1] \leftarrow M_K^{v_i, v_j}[j - i - 1] + 1 \quad (3)$$

Nhận xét:

- $K$  là đồ thị có hướng, có trọng số.
- Trong  $K$ , trọng số cạnh nối từ đỉnh  $i$  tới  $j$  có giá trị là  $M_K^{i,j} \in \mathbb{R}^c$ . Vì thế, nó chiếm bộ nhớ lưu trữ, mất thời gian truy cập lấy giá trị hơn nhiều đồ thị  $H$  và  $G$ .
- Mang thông tin của cả 2 đồ thị  $H$  và  $G$
- $M_G^{i,j} = M_K^{i,j}[0] \quad \forall i, j : 0 \leq i < j < d$
- $M_H^{i,j} = \sum_{t=0}^c M_K^{i,j}[t] \quad \forall i, j : 0 \leq i < j < d$
- Theo thống kê, xác suất của để sản phẩm  $j$  được nhấp sau khi nhấp sản phẩm  $i$  đúng  $t$  lần nhấp là:

$$P_K^{i,j}[t] = \frac{M_K^{i,j}[t]}{\sum_{x=0}^{d-1} M_K^{i,x}[t]}$$

- Gọi  $E_X$  là số lượng cạnh của đồ thị  $X$ . Ta có,  $E_K = E_H > E_G$ .

## 2. Mạng học sâu đồ thị (Graph Neural Network)

Mạng học sâu đồ thị (GNN) được giới thiệu đầu tiên vào năm 2005 [15], GNN là một loại mạng nơ-ron hoạt động trực tiếp trên cấu trúc đồ thị, với việc sử dụng nơ-ron như là các nút trong cấu trúc mạng, từng nút sẽ chứa thông tin của riêng nó và thu thập thêm các thông tin từ các nút lân cận thể hiện mối tương quan giữa các nút trong đồ thị.

Với hướng tiếp cận sử dụng đồ thị, GNN ngày càng trở nên phổ biến trong nhiều lĩnh vực khác nhau [16], tiềm năng của mô hình GNN cho thấy khả năng ứng dụng và xử lý được khá nhiều bài toán thực tế như xây dựng biểu đồ tri thức, đánh giá mối tương quan của mạng xã hội, hệ thống gợi ý bán hàng... Sức mạnh của GNN trong việc mô hình hóa được sự phụ thuộc giữa các nút trong đồ thị cho phép tạo ra bước đột phá trong lĩnh vực nghiên cứu liên quan đến phân tích đồ thị.

Trong nghiên cứu này, nhóm tác giả đề xuất sử dụng mô hình GNN để xây dựng mô hình gợi ý dựa và phiên làm việc của người dùng. Để có sự nhất quán trong phần mô tả các khái niệm, nhóm tác giả thiết lập một số thuật ngữ

chính như sau: khái niệm đồ thị (graph) sẽ dùng để biểu diễn phiên làm việc (session), trong đó đỉnh (node) của đồ thị mô tả sản phẩm (item) lựa chọn và cạnh (edge) của đồ thị mô tả việc người dùng nhấp chuột (click) từ sản phẩm trước sang sản phẩm tiếp theo trong phiên.

Với phương án xây dựng mạng nơ-ron đồ thị GNN cho các đồ thị đơn quan hệ  $G$ ,  $H$  và đồ thị sâu  $K$  mô tả ở phần trên, do  $K$  là đồ thị sâu nên cần có phương án phù hợp hơn để mô hình GNN có thể học được tính chất sâu của đồ thị  $K$ . Do đó, nhóm tác giả đề xuất hai mô hình khác nhau cho đồ thị đơn quan hệ và đồ thị sâu như sau:

### Mô hình mạng nơ-ron cho đồ thị $G$ và $H$

Mô hình mạng nơ-ron cho đồ thị đơn quan hệ  $G$  và  $H$  được biểu diễn như hình 2.

- $c$  là số lượng nhấp được sử dụng làm đầu vào của mô hình,  $d$  là số lượng sản phẩm có trong toàn bộ phiên.
- Đầu vào của mô hình là một phiên gồm  $c$  nhấp có tính thứ tự lần là  $s = \{id_1, id_2, \dots, id_c\}$  với  $0 \leq id_i \leq d$ .
- Với mỗi nhấp  $id_i$  qua đồ thị, ta thu được một vector trọng số  $v_i$  với  $v_i \in \mathbb{R}^d$ .
- Sử dụng lớp *Norm* để chuẩn hóa  $v_i$  thành xác suất  $p_i \in \mathbb{R}^d$  với công thức sau:

$$p_i = \frac{v_i}{\text{sum}(v_i)} \quad (4)$$

- Cuối cùng, sử dụng một lớp *Fully connected layer* với hàm kích hoạt là *softmax* để tính toán đầu ra của mô hình.

### Mô hình mạng nơ-ron cho đồ thị $K$

Để cải tiến mô hình mạng nơ-ron đồ thị khi phải làm việc với đồ thị  $K$  (tức đồ thị có chiều sâu), nhóm tác giả đề xuất sử dụng thêm một lớp học sâu (Depth layer) vào mô hình như hình 3.

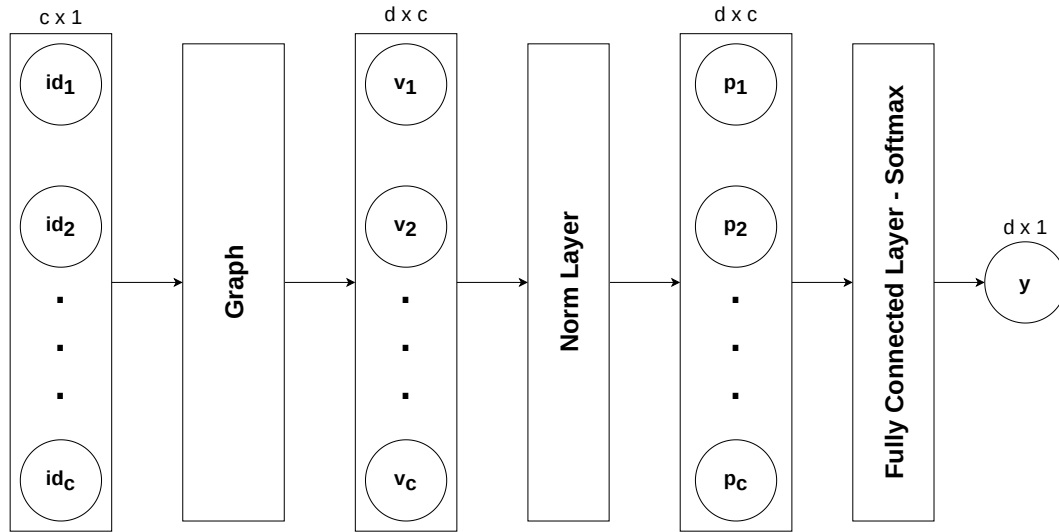
- Đầu vào của mô hình là một phiên gồm  $c$  nhấp có tính thứ tự lần lượt là  $s = \{id_1, id_2, \dots, id_c\}$  với  $0 \leq id_i \leq d$  ( $d$  là số lượng sản phẩm trong toàn bộ phiên).
- Với mỗi nhấp  $id_i$  qua đồ thị  $K$ , ta thu được một vector trọng số  $v_i$  với  $v_i \in \mathbb{R}^{d \times c}$ .
- Sử dụng lớp *Depth* để biến đổi chiều của  $v_i$  thành  $\mathbb{R}^d$  với công thức sau:

$$z_i = f(w_i v_i^T + b_i) \quad (5)$$

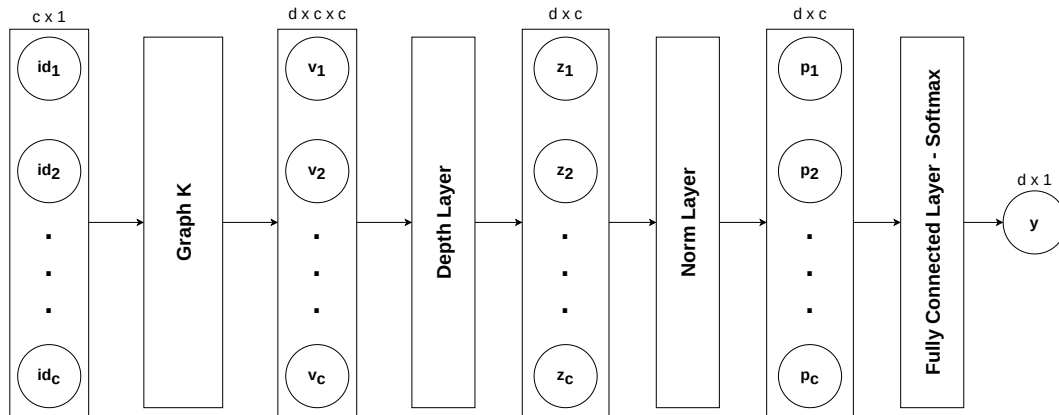
Với:

- $w_i \in \mathbb{R}^{1 \times c}$ : trọng số chiều sâu
- $b_i \in \mathbb{R}$ : trọng số tự do của chiều sâu
- $f(z)$ : là một hàm biến đổi  $z$ , nhóm tác giả sử dụng hàm tuyến tính  $f(z) = z$ .
- Sử dụng lớp *Norm* để chuẩn hóa  $z_i$  thành xác suất  $p_i \in \mathbb{R}^d$  với công thức sau:

$$p_i = \frac{z_i}{\text{sum}(z_i)} \quad (6)$$



Hình 2. Mô hình mạng nơ-ron cho đồ thị  $G$  và  $H$



Hình 3. Mô hình mạng nơ-ron cho đồ thị  $K$

- Cuối cùng, sử dụng một lớp *Fully connected layer* với hàm kích hoạt là *softmax* để tính toán đầu ra của mô hình.

## IV. NGUỒN DỮ LIỆU NGHIÊN CỨU

### 1. Nguồn Dữ Liệu (Dataset)

Nghiên cứu sử dụng bộ dữ liệu cung cấp bởi *YOO-CHOOSE GmbH*, đây là bộ dữ liệu được sử dụng trong cuộc thi *RecSys Challenge 2015* [17]. Bộ dữ liệu ghi lại tập hợp nhiều phiên làm việc của một trang web thương mại điện tử hoạt động trong lĩnh vực bán lẻ tại châu Âu, trong đó mỗi phiên làm việc chứa thông tin về chuỗi nhấp chuột và một danh sách sản phẩm mà khách hàng lựa chọn trong suốt phiên đó. Dữ liệu được ghi nhận kéo dài trong 6 tháng, từ tháng 04/2014 đến tháng 09/2014. Vì lý do quyền riêng tư, toàn bộ thông tin về người sử dụng đã được ẩn đi khỏi bộ dữ liệu.

Bộ dữ liệu bao gồm hai tập dữ liệu:

- **Dữ liệu nhấp chuột** (yoochoose-clicks.dat): chứa dữ liệu về chuỗi nhấp chuột của người dùng. Dữ liệu bao gồm các trường: (1) Session ID – ID của mỗi session. Trong mỗi session có thể có một hoặc nhiều sự kiện nhấp chuột. (2) Timestamp – thời gian xảy ra của sự kiện nhấp chuột. (3) Item ID – ID của sản phẩm được chọn. (4) Category – danh mục của sản phẩm được chọn.
- **Dữ liệu nhấp chuột đánh giá** (yoochoose-test.dat): giống với bộ dữ liệu nhấp chuột đã nêu ở trên nhưng những session trong đây không có trong tập trên. Nó được sử dụng để đánh giá mô hình.
- **Dữ liệu mua sắm** (yoochoose-buys.dat): chứa dữ liệu về chuỗi mua sắm của người dùng. Dữ liệu bao gồm các trường: (1) Session ID – ID của mỗi session. Trong mỗi session có thể có một hoặc nhiều sự kiện mua sắm.



- (2) Timestamp – thời gian xảy ra của sự kiện mua sắm. (3) Item ID – ID của sản phẩm được mua. (4) Price – giá sản phẩm. (5) Quantity – số lượng sản phẩm được mua.

Mỗi Session ID trong yoochoose-buys.dat luôn xuất hiện trong yoochoose-clicks.dat – các dữ liệu cùng Session ID kết hợp lại tạo thành chuỗi nhấp chuột của một khách hàng cụ thể trong suốt một phiên làm việc. Thời gian của một phiên có thể rất ngắn (vài phút) hoặc rất dài (vài giờ), có thể bao gồm một hoặc nhiều sự kiện nhấp chuột và mua hàng, phụ thuộc vào hành vi tương tác của người sử dụng. Thông tin chi tiết về hai tập dữ liệu này được thể hiện ở Bảng I

Bảng I  
THỐNG KÊ VỀ KÍCH THƯỚC BỘ DỮ LIỆU [17]

	yoochoose-clicks.dat	yoochoose-buys.dat
Số lượng sự kiện	33.003.944	1.150.753
Số lượng sản phẩm	52.739	19.949
Số lượng session	9.249.729	509.696

## 2. Một số phân tích bộ dữ liệu

Bảng thống kê dữ liệu của hai tập dữ liệu huấn luyện và kiểm thử được mô tả ở Bảng II.

Bảng II  
THỐNG KÊ VỀ BỘ DỮ LIỆU NHẬP

	Bộ huấn luyện	Bộ kiểm tra	Tất cả
Số lượng phiên	9.249.729	2.312.432	11.562.161
Số lượng sản phẩm	52.739	42.155	54.287
Số lượng nhấp	33.003.944	8.251.791	41.255.735
Số nhấp lớn nhất	200	200	200
Số nhấp trung bình	3,5681	3,56845	3,56817
Số nhấp nhỏ nhất	1	1	1
Số phiên 1 nhấp	13,619%	13,602%	13,616%
Số phiên 2 nhấp	38,467%	38,463%	38,466%
Số phiên 3 nhấp	17,442%	17,467%	17,447%
Số phiên 4 nhấp	10,118%	10,099%	10,114%
Số phiên 5 nhấp	5,866%	5,903%	5,874%
Số phiên 6 nhấp	3,926%	3,914%	3,924%
Số phiên 7 nhấp	2,565%	2,566%	2,565%
Số phiên 8 nhấp	1,828%	1,833%	1,829%
Số phiên 9 nhấp	1,297%	1,298%	1,297%
Số phiên 10 nhấp	0,979%	0,962%	0,976%
Số phiên hơn 10 nhấp	3,892%	3,894%	3,892%

Với thống kê dữ liệu ở Bảng II, ta có một số nhận xét sau:

- Bộ dữ liệu chứa hơn 11 triệu phiên, trong đó bộ huấn luyện chứa hơn 9 triệu phiên và bộ kiểm tra chứa hơn 2 triệu phiên.
- Có tất cả 54.287 sản phẩm, bộ huấn luyện có 52.739 sản phẩm. Vì vậy có 1.548 sản phẩm có trong bộ kiểm tra mà không có trong bộ huấn luyện, dẫn đến việc không thể xác định (học) số sản phẩm này. Vì vậy, ta cần loại bỏ các phiên có sản phẩm này ra khỏi tập kiểm tra (các nghiên cứu liên quan về bộ dữ liệu này cũng loại bỏ dữ liệu này).
- Phiên có ít nhất 1 nhấp và có thể lên tới 200 nhấp.
- Số lượng phiên chỉ có 1 nhấp chiếm 13,6%, dữ liệu này gần như không có giá trị vì không đủ thông tin nên cần loại bỏ phiên này. Phiên làm việc có số lượng nhấp nhiều nhất là 2 nhấp và 3 nhấp với tỷ lệ lần lượt là 38,5% và 17,4%. Như vậy phiên có số lượng nhấp càng lớn thì càng ít xuất hiện.
- Trung bình mỗi phiên làm việc thì khoảng 4 nhấp (3,5)

## 3. Tiền xử lý dữ liệu

Các bước tiền xử lý được thực hiện như sau:

- Chuẩn hóa phiên:
  - Tổng hợp phiên theo danh sách nhấp (danh sách sản phẩm)
  - Loại bỏ một số thuộc tính dữ liệu không cần thiết như thời gian nhấp, category, ...
- Bỏ các phiên chỉ có 1 nhấp.
- Loại bỏ phiên làm việc trong bộ kiểm tra có chứa sản phẩm mà không xuất hiện trong bộ huấn luyện.

Bộ dữ liệu sau bước tiền xử lý được mô tả ở Bảng III.

## 4. Phương Án Chia Dữ Liệu

Bộ kiểm tra chứa tới gần 2 triệu phiên, đây là số lượng khá lớn nên phần thực nghiệm của nghiên cứu này chỉ lấy 25% của bộ kiểm tra làm tập kiểm tra cuối cùng (test) và 25% khác cũng thuộc bộ này làm tập đánh giá (validate). Lưu ý, tập kiểm tra và tập đánh giá có những phiên khác nhau và được chọn ngẫu nhiên. Ta có Bảng IV.

## V. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

### 1. Tham Số Đánh Giá Mô Hình

Trong các bài toán dự báo bán hàng thực tế, việc gợi ý cho khách hàng một sản phẩm tiếp theo thường không có lợi ích gì, thay vào đó hệ thống gợi ý cần đưa ra một danh sách  $k$  sản phẩm đề xuất tiếp theo. Vì lý do đó một số tham số cơ bản trong quá trình đánh giá mô hình như *Precision*, *Recall* hay *Accuracy* đơn lẻ không còn phù hợp nữa. Thay vào đó các nghiên cứu gần đây xuất sử

Bảng III  
THỐNG KÊ VỀ BỘ DỮ LIỆU NHẬP SAU TIỀN XỬ LÝ

	Bộ huấn luyện	Bộ kiểm tra	Tất cả
Số lượng phiên	7.990.018	1.996.408	9.986.426
Số lượng sản phẩm	52.069	38.733	52.069
Số lượng nhấp	31.744.233	7.926.322	39.670.555
Số nhấp lớn nhất	200	200	200
Số nhấp trung bình	3,97299	3,97029	3,97245
Số nhấp nhỏ nhất	2	2	2
Số phiên 2 nhấp	44,532%	44,518%	44,529%
Số phiên 3 nhấp	20,192%	20,223%	20,198%
Số phiên 4 nhấp	11,713%	11,691%	11,709%
Số phiên 5 nhấp	6,791%	6,833%	6,800%
Số phiên 6 nhấp	4,545%	4,531%	4,543%
Số phiên 7 nhấp	2,969%	2,970%	2,969%
Số phiên 8 nhấp	2,117%	2,121%	2,117%
Số phiên 9 nhấp	1,502%	1,502%	1,502%
Số phiên 10 nhấp	1,133%	1,113%	1,129%
Số phiên hơn 10 nhấp	4,505%	4,498%	4,504%

Bảng IV  
THỐNG KÊ VỀ BỘ KIỂM TRA VÀ ĐÁNH GIÁ

	Bộ kiểm tra	Bộ đánh giá
Số lượng phiên	499.102	499.102
Số lượng sản phẩm	30.179	30.278
Số lượng nhấp	1.982.109	1.978.213
Số nhấp lớn nhất	200	200
Số nhấp trung bình	3,971	3,963
Số nhấp nhỏ nhất	2	2
Số phiên 2 nhấp	44,580%	44,585%
Số phiên 3 nhấp	20,132%	20,146%
Số phiên 4 nhấp	11,643%	11,786%
Số phiên 5 nhấp	6,881%	6,769%
Số phiên 6 nhấp	4,532%	4,528%
Số phiên 7 nhấp	2,988%	2,979%
Số phiên 8 nhấp	2,129%	2,101%
Số phiên 9 nhấp	1,478%	1,519%
Số phiên 10 nhấp	1,133%	1,131%
Số phiên hơn 10 nhấp	4,504%	4,457%

dụng tham số  $Recall@K$ ,  $MRR@K$  và  $ACCs@K$  để đánh giá việc gợi ý cùng lúc  $k$  sản phẩm. Trong nghiên cứu này, nhóm tác giả cũng đề xuất sử dụng các tham số này, trong đó:

### $Recall@K$

Để đánh giá hiệu suất của mô hình với một hệ gợi ý, nghiên cứu sử dụng chỉ số  $Recall@K$  với công thức sau:

$$Recall@K = \frac{1}{n} \sum_{i=0}^n \frac{|S_{rec}^i \cap S_{rel}^i|}{|S_{rel}^i|} \quad (7)$$

Trong đó, với  $n$  là số lượng phiên của bộ dữ liệu,  $S_{rec}^i$  là tập các sản phẩm gợi ý (gợi ý bởi  $topK$  -  $K$  trọng số lớn nhất) và  $S_{rel}^i$  là tập các sản phẩm được nhấp thực tế của phiên thứ  $i$  với  $0 \leq i < n$ . Thông thường trong bài toán gợi ý, tập các nhấp thực tế ( $S_{rel}^i$ ) chỉ có duy nhất một thành phần được gọi là next-click. Vì vậy,  $S_{rel}^i = \{id_*\}$  với  $id_*$  là sản phẩm được nhấp gợi ý tiếp theo sau các sản phẩm đã nhấp trong phiên làm việc thứ  $i$ .

### $MRR@K$

$MRR@K$  (Mean Reciprocal Rank) là mức trung bình của các cấp bậc tương hỗ của các sản phẩm mong muốn. Xếp hạng đối ứng được đặt thành 0 nếu thứ hạng lớn hơn  $K$ .  $MRR$  tính đến thứ hạng của mặt hàng, điều này rất quan trọng trong các cài đặt có thứ tự đề xuất quan trọng.  $MRR@K$  có công thức tính như sau:

$$MRR@K = \frac{1}{n} \sum_{i=0}^n RR(id_*^i, S_{rel}^i) \quad (8)$$

Trong đó, với  $n$  là số lượng phiên của bộ dữ liệu,  $S_{rel}^i$  là tập  $K$  sản phẩm gợi ý được sắp xếp theo trọng số từ lớn đến bé (gợi ý bởi  $top - K$  - trong đó  $K$  trọng số lớn nhất) và  $id_*$  là sản phẩm được nhấp gợi ý tiếp theo sau các sản phẩm đã nhấp trong phiên làm việc thứ  $i$  với  $0 \leq i < n$ .  $RR(id, S)$  là 0 nếu sản phẩm  $id$  không có trong  $S$ , là  $\frac{1}{r+1}$  với  $r$  là vị trí của  $id$  trong tập  $S$  tính từ 0.

Như vậy, nếu hệ gợi ý càng trả về đúng sản phẩm tiếp theo với điểm càng cao thì  $MRR$  càng cao. Lưu ý, chỉ số này chỉ áp dụng với một nhấp sản phẩm tiếp theo thực tế, không phù hợp cho việc gợi ý một chuỗi các nhấp (số nhấp lớn hơn 1).

### $ACCs@K$

Bài báo này đề xuất một chỉ số  $ACCs$  để tính độ chính xác trong một hệ gợi ý  $K$  nhân với trọng số (xác suất) lớn nhất với nhiều nhãn thực tế. Đây là độ đo để tính cho một bài toán nhiều nhãn đầu ra (một quan sát nhưng có nhiều nhãn).

$$ACCs@K = \frac{1}{n} \sum_{i=0}^n \min(1, |S_{rec}^i \cap S_{rel}^i|) \quad (9)$$

Trong đó, với  $n$  là số lượng phiên của bộ dữ liệu,  $S_{rec}^i$  là tập các sản phẩm gợi ý (gợi ý bởi  $topK$  -  $K$  trọng số lớn nhất) và  $S_{rel}^i$  là tập các sản phẩm được nhấp thực tế của

phiên thứ  $i$  với  $0 \leq i < n$ . Công thức  $\min(1, |S_{rec}^i \cap S_{rel}^i|)$  chỉ ra rằng ở quan sát thứ  $i$ , có tồn tại 1 sản phẩm nào đó trong danh sách nhân nằm trong  $K$  nhân dự đoán có trọng số lớn nhất hay không, giá trị này bằng 1 nếu có tồn tại và ngược lại bằng 0.

## 2. Kết quả và Nhận xét

Trong thí nghiệm này nhóm tác giả có chạy thử nghiệm với mô hình thống kê để làm kết quả cơ sở của thí nghiệm. Kết quả của quá trình huấn luyện của hai mô hình với ba đồ thị G, H, K được mô tả ở Bảng V:

Bảng V  
BẢNG KẾT QUẢ TỔNG QUAN

Mô hình	Stats.H	Stats.G	GNN.G	GNN.H	GNN.K
<b>Recall@1</b>	0,227	0,223	0,218	0,214	0,221
<b>Recall@5</b>	0,500	0,505	0,518	0,512	0,519
<b>Recall@10</b>	0,610	0,615	0,625	0,620	0,626
<b>Recall@20</b>	0,696	0,701	0,708	0,703	0,709
<b>ACCs@1</b>	0,253	0,248	0,241	0,239	0,245
<b>ACCs@5</b>	0,533	0,538	0,549	0,545	0,550
<b>ACCs@10</b>	0,639	0,643	0,651	0,647	0,652
<b>ACCs@20</b>	0,719	0,723	0,728	0,724	0,729
<b>MRR@1</b>	0,227	0,223	0,218	0,214	0,221
<b>MRR@5</b>	0,327	0,327	0,330	0,325	0,332
<b>MRR@10</b>	0,342	0,342	0,344	0,339	0,346
<b>MRR@20</b>	0,348	0,348	0,350	0,345	0,352

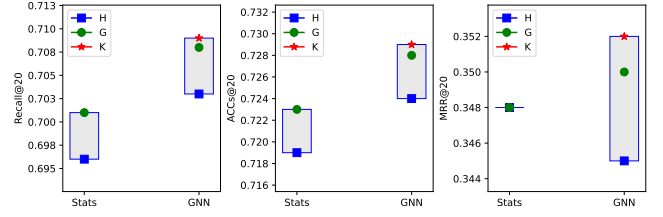
Qua Bảng V ta thấy:

- Thử nghiệm với 5 mô hình khác nhau, gồm 2 mô hình thống kê truyền thống và 3 mô hình học sâu sử dụng đồ thị. Lưu ý mô hình thống kê không hỗ trợ làm việc với đồ thị sâu K.
- Đánh giá mô hình dựa trên 3 tham số chính là  $Recall@K$ ,  $ACCs@K$ ,  $MRR@K$  với  $K \in [1, 5, 10, 20]$
- Mô hình thống kê với đồ thị H ( $Stats.H$ ) có kết quả tệ nhất, còn tốt nhất là mô hình sử dụng mạng học sâu GNN với đồ thị K ( $GNN.K$ ).
- Tuy nhiên kết quả cho thấy khoảng cách giá trị giữa các mô hình với từng tham số không lớn.

Hình 4 biểu diễn trực quan kết quả so sánh của từng nhóm mô hình khi thực hiện với các loại đồ thị khác nhau, trong khuôn khổ bài báo này tác giả chỉ biểu diễn với top-K=20, các kết quả của K còn lại có thể tham khảo ở Bảng V:

## 3. So sánh với các nghiên cứu liên quan

Để so sánh hướng tiếp cận và kết quả đạt được của nghiên cứu này, tác giả lựa chọn một số bài báo tương tự cùng giải



Hình 4. Biểu đồ kết quả cho các mô hình với top-K=20

quyết bài toán hệ gợi ý dựa vào phiên làm việc và cùng sử dụng bộ dữ liệu *Yoochoose* của cuộc thi RecSys Challenge 2015 [17] như trong nghiên cứu này.

Hai nghiên cứu của Balázs Hidasi (2015) [8] và Yong Kiam Tan (2016) [9] đều sử dụng mạng nơ ron hồi quy (RNN) cho bài toán này, trong đó Yong Kiam Tan đã đề xuất cải tiến mô hình RNN với một thuật toán làm giàu dữ liệu và cho kết quả tốt hơn hẳn so với mô hình RNN của Balázs Hidasi. Thuật toán làm giàu dữ liệu trong quá trình tiền xử lý dữ liệu của Y.K. Tan và cộng sự để sinh dữ liệu là: với một phiên  $s = [v_{s,1}, v_{s,2}, \dots, v_{s,n}]$  thì sẽ tạo ra một chuỗi phiên con và nhân là  $([v_{s,1}], v_{s,2}), ([v_{s,1}, v_{s,2}], v_{s,3}), \dots, ([v_{s,1}, v_{s,2}, \dots, v_{s,n-1}], v_{s,n})$  với  $[v_{s,1}, v_{s,2}, \dots, v_{s,n-1}]$  là chuỗi nhập đầu vào và  $v_{s,n}$  là nhân *next-click*. Điểm lưu ý là bộ dữ liệu này có số sản phẩm là 37.483 sau quá trình tiền xử lý, khác với thống kê gốc của bộ dữ liệu này là 52.739 sản phẩm [17], điểm khác biệt này sẽ là đáng kể với các mô hình gợi ý phân lớp đa nhãn.

Do bộ dữ liệu này khá lớn, Y.K. Tan và cộng sự gặp một số khó khăn trong quá trình huấn luyện khi bị giới hạn phần cứng, có lẽ vì lý do đó Y.K. Tan và cộng sự đưa ra ý tưởng chia nhỏ bộ dữ liệu để thực nghiệm thành các bộ *Yoochoose* nhỏ hơn (1/4, 1/16, 1/64, 1/256). Trong quá trình thực nghiệm, Y.K. Tan và cộng sự nhận thấy việc sử dụng bộ dữ liệu đầy đủ mang đến kết quả kém hơn so với việc dùng một phần của dữ liệu. Lý do chính mà Y.K. Tan và cộng sự đưa ra nhận xét này là do số lượng nhân quá lớn trên bộ dữ liệu đầy đủ, con số này sẽ bị giảm đáng kể khi tách thành từng phần nhỏ hơn nên mô hình sẽ học nhẹ nhàng hơn nhiều. Cho dù Y.K. Tan và cộng sự kết luận mô hình *M2* cho kết quả tốt nhất với bộ dữ liệu con *Yoochoose1/64* với  $Recall@20$  là 0,7129 và  $MRR@20$  là 0,3091, tuy nhiên mô hình này không thể thực hiện được trên bộ dữ liệu đầy đủ do giới hạn phần cứng. Còn với bộ kết dữ liệu đầy đủ, mô hình *M3* của Y.K. Tan và cộng sự cho kết quả tốt nhất là  $Recall@20$  xấp xỉ 0,680 và  $MRR@20$  xấp xỉ 0,290.

Jing Li và các cộng sự (2017) [11] đề xuất sử dụng mô hình NARM (*Neural Attentive Recommendation Machine*) để xây dựng hệ gợi ý phiên làm việc và cũng sử dụng bộ dữ



liệu con *Yoochoose* 1/4 và 1/64. Thực nghiệm của Jing Li sử dụng bộ dữ liệu kiểm tra với 55.898 phiên, số lượng sản phẩm là 16.766 với thực nghiệm 1/64 và 29.618 với thực nghiệm 1/4. Jing Li đạt được kết quả *Recall@20* là 0,6973 và *MRR@20* là 0,2923, với bộ dữ liệu 1/4. Trong bảng so sánh kết quả với nghiên cứu sử dụng RNN của Hidasi [8] và Y.K. Tan và cộng sự [9], Jing Li cho rằng kết quả của mình tốt hơn, dù rằng nếu so sánh kỹ thì kết quả của Jing Li không thực sự đầy đủ và nổi trội như nghiên cứu của Tan, ví dụ như không thực nghiệm với bộ dữ liệu đầy đủ.

Cũng với hướng tiếp cận sử dụng bộ dữ liệu con của Tan, Shu Wu và cộng sự (2019) [12] sử dụng mô hình mạng học sâu đồ thị (GNN) cùng khá nhiều kỹ thuật xử lý đồ thị và các biến thể khác nhau của GNN để phân tích bài toán gợi ý dựa vào phiên làm việc. Nhóm nghiên cứu này đã đề xuất một mô hình sử dụng kỹ thuật biến đổi véc tơ phiên làm việc sang một không gian nhúng bằng cách sử dụng mạng đồ thị để huấn luyện và học. Khác với các nghiên cứu trước sử dụng tham số *Recall@K*, Shu Wu đề xuất sử dụng tham số *Precision@K* và một điểm lưu ý là Shu Wu lại so sánh kết quả *Precision@K* của mình với *Recall@K* của các nghiên cứu trước đây, nên khả năng có sự nhầm lẫn trong nghiên cứu này và việc so sánh kết quả là không hợp lý.

Trong nghiên cứu của Kiewan và cộng sự (2018) [10] cũng sử dụng mô hình RNN như hai bài báo của Balázs Hidasi [8] và Yong Kiam Tan [9], tuy nhiên Kiewan không chia nhỏ bộ dữ liệu như Y.K. Tan và cộng sự mà có hướng tiếp cận khác biệt là đưa ra khái niệm phiên dài và phiên ngắn (*long and short sequential session*), cụ thể Kiewan đánh giá mô hình theo 3 nhóm phiên có độ dài là [2-5], [6-25] và [26-200]. Kiewan đề xuất một số kỹ thuật khác nhau (bao gồm chuẩn hóa lớp LN, ma trận nhúng đầu vào RE hoặc xếp chồng lớp GRU) để xây dựng mô hình phù hợp với các loại dữ liệu phiên có độ dài ngắn khác nhau. Mô hình của Kiewan cho kết quả tốt nhất *Recall@20* là 0,691. Như vậy, kết quả này là tốt hơn với kết quả của Y.K. Tan và cộng sự khi kiểm tra với bộ dữ liệu đầy đủ nhân có *Recall@20* xấp xỉ 0,680 và tốt hơn hẳn so với kết quả của Balázs Hidas với *Recall@20* là 0,632. Nếu so sánh kết quả này với kết quả tốt nhất của Y.K. Tan và cộng sự với tập dữ liệu con 1/64 với số nhân ít hơn thì không tốt bằng (*Recall@20* trên tập 1/64 là 0,7129), tuy nhiên việc so sánh như vậy cũng không hoàn toàn hợp lý do số lượng nhân của tập dữ liệu đầy đủ mà Kiewan kiểm tra (khoảng 37 nghìn nhân) nhiều hơn so với tập con 1/64 của Y.K. Tan và cộng sự (gần 17 nghìn nhân).

Sau khi phân tích các nghiên cứu liên quan ở trên, tác giả có một số nhận xét như sau về kết quả của mình so với các nghiên cứu trước đây:

- Nghiên cứu này sử dụng cả tập dữ liệu huấn luyện và kiểm thử từ bộ dữ liệu gốc với số lượng sản phẩm, tức số lượng nhân, lên tới hơn 52 nghìn. Các nghiên cứu trước đây không sử dụng bộ dữ liệu kiểm thử riêng biệt của bộ dữ liệu gốc, mà trích ra từ tập dữ liệu huấn luyện. Điều này làm giảm số lượng sản phẩm, tức số lượng nhân của mô hình xuống còn từ 10 tới 37 nghìn nhân.
- Nghiên cứu này đề xuất và xây dựng mô hình có tính mở rộng cao khi hoạt động với đồ thị với hơn 52 nghìn đỉnh, đặc biệt với số lượng cạnh rất lớn với các loại đồ thị gián tiếp H hay đồ thị sâu K. Một số nghiên cứu liên quan trình bày không thể chạy được mô hình với bộ dữ liệu đầy đủ, do đó họ phải thực nghiệm với bộ dữ liệu nhỏ hơn với số lượng nhân thậm chí còn ít hơn.
- Mô hình đề xuất của nghiên cứu này cho kết quả *Recall@20* là 0,712 và *MRR@20* là 0,363, tốt hơn kết quả của Kiewan có *Recall@20* là 0,691 và của Y.K. Tan và cộng sự có *Recall@20* là 0,680 (Y.K. Tan và cộng sự có nhiều kết quả khác nhau, đây là kết quả chạy với bộ dữ liệu đầy đủ), và tốt hơn hẳn nghiên cứu đầu tiên của Balázs Hidas với *Recall@20* là 0,632.

## VI. KẾT LUẬN

Nghiên cứu này có hai đóng góp chính gồm: (1) sử dụng đồ thị để mô tả chuỗi nhấp chuột trong quá trình lựa chọn sản phẩm trong phiên làm việc hiện tại, bao gồm ba loại đồ thị trực tiếp G, gián tiếp H và sâu K và (2) sử dụng mạng học sâu đồ thị GNN để học dữ liệu đồ thị để đưa ra các mô hình gợi ý dựa theo bài toán *top-K*. Thực nghiệm cho thấy hướng đề xuất của nghiên cứu này cho kết quả *Recall@20* và *MRR@20* tốt hơn hẳn các nghiên cứu trước đây, đặc biệt là khả năng xử lý đồ thị với số lượng đỉnh rất lớn và có khả năng mở rộng hơn nữa, đây chính là điểm hạn chế của các nghiên cứu trước đây gặp nhiều khó khăn khi phải xử lý bộ dữ liệu có số lượng sản phẩm (nhân) khá lớn với hơn 50 nghìn nhân, một số nghiên cứu trước đây còn phải chia nhỏ bộ dữ liệu và như vậy đã phải giảm nhân của mô hình gợi ý.

Hướng mở rộng của nghiên cứu này là tiếp tục sử dụng một số kỹ thuật biến đổi đồ thị (ví dụ phép nhúng đồ thị) sao cho mô hình GNN hoạt động hiệu quả hơn, hoặc phương án thiết kế tối ưu lớp sâu của mô hình GNN để nó có thể hoạt động hiệu quả hơn với đồ thị sâu K.

## TÀI LIỆU THAM KHẢO

- [1] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM conference on Electronic commerce*, 1999, pp. 158–166.
- [2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.
- [4] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data mining and knowledge discovery*, vol. 5, no. 1, pp. 115–153, 2001.
- [5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," *Minnesota Univ Minneapolis Dept of Computer Science, Tech. Rep.*, 2000.
- [6] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," in *Proceedings of the fifth international conference on computer and information technology*, vol. 1. Citeseer, 2002, pp. 291–324.
- [7] Z. Huang, W. Chung, and H. Chen, "A graph model for e-commerce recommender systems," *Journal of the American Society for information science and technology*, vol. 55, no. 3, pp. 259–274, 2004.
- [8] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015. [Online]. Available: <https://arxiv.org/abs/1511.06939>
- [9] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," *CoRR*, vol. abs/1606.08117, 2016. [Online]. Available: <http://arxiv.org/abs/1606.08117>
- [10] K. Villatell, E. Smirnova, J. Mary, and P. Preux, "Recurrent neural networks for long and short-term sequential recommendation," *CoRR*, vol. abs/1807.09142, 2018. [Online]. Available: <http://arxiv.org/abs/1807.09142>
- [11] J. Li, P. Ren, Z. Chen, Z. Ren, and J. Ma, "Neural attentive session-based recommendation," *CoRR*, vol. abs/1711.04725, 2017. [Online]. Available: <http://arxiv.org/abs/1711.04725>
- [12] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 346–353.
- [13] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Isir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," 2016.
- [14] K. Nguyen, A. Nguyen, L. Vu, N. Mai, and B. Nguyen, "An efficient deep learning method for customer behaviour prediction using mouse click events," 11 2018.
- [15] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, 2005., vol. 2, pp. 729–734 vol. 2, 2005.
- [16] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [17] Yoochoose Dataset, "Recsys challenge," 2015, [https://github.com/RUCAIBox/RecSysDatasets/tree/master/dataset\\_info/YOOCHOOSE](https://github.com/RUCAIBox/RecSysDatasets/tree/master/dataset_info/YOOCHOOSE).

## SƠ LƯỢC VỀ TÁC GIẢ

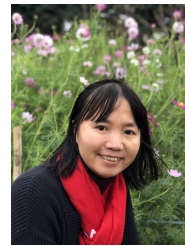


vi.

Email: [khang\\_nt@yahoo.com](mailto:khang_nt@yahoo.com)

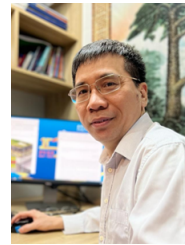


Email: [anh.nt@thanglong.edu.vn](mailto:anh.nt@thanglong.edu.vn)



thống thông tin quản lý, trí tuệ nhân tạo và khoa học dữ liệu.

Email: [ngamt@thanglong.edu.vn](mailto:ngamt@thanglong.edu.vn)



trong tối ưu phát triển khai thác các mỏ trên thềm lục địa Việt Nam và nâng cao hệ số thu hồi dầu và khí.

Email: [annh1@pvep.com.vn](mailto:annh1@pvep.com.vn)



Việt Nam. Lĩnh vực nghiên cứu: học máy, dữ liệu lớn và phân tích mạng xã hội.

Email: [anhnv@ioit.ac.vn](mailto:anhnv@ioit.ac.vn)

**Nguyễn Tuấn Khang** là nghiên cứu sinh tiến sĩ tại Viện Công nghệ thông tin - Viện Hàn lâm khoa học và công nghệ Việt Nam. Hiện đang là tư vấn giải pháp công nghệ tại IBM Việt Nam. Lĩnh vực nghiên cứu: tư vấn các giải pháp công nghệ thông tin trong lĩnh vực tài chính ngân hàng, chuyên sâu vào mảng học sâu và phân tích hành

**Nguyễn Tú Anh** tốt nghiệp trường Đại học Thăng Long năm 2020, hiện đang theo học chương trình thạc sĩ tại Viện Công nghệ thông tin - Viện Hàn lâm khoa học và công nghệ Việt Nam. Hiện đang là giảng viên khoa Công nghệ thông tin của trường Đại học Thăng Long. Lĩnh vực nghiên cứu: thị giác máy tính và phân tích dữ liệu.

**Mai Thúy Nga** đạt học vị Tiến sĩ ngành Công nghệ thông tin tại Viện Công nghệ thông tin - Viện Hàn lâm khoa học và công nghệ Việt Nam năm 2017. Hiện đang là giảng viên khoa Công nghệ thông tin của trường Đại học Thăng Long. Lĩnh vực nghiên cứu: xây dựng chương trình đào tạo đại học, phân tích thiết kế phần mềm, hệ

**Nguyễn Hải An** đạt học vị Tiến sĩ ngành Kỹ thuật khai thác dầu khí tại trường ĐH Mỏ Địa chất Hà Nội năm 2012; thạc sĩ Công nghệ dầu khí và phát triển mỏ tại Viện dầu lửa Pháp. Hiện là trưởng phòng Khoa học công nghệ, TCT Thăm dò Khai thác Dầu khí. Lĩnh vực nghiên cứu: ứng dụng kỹ thuật mới, công nghệ thông minh

**Nguyễn Việt Anh** đạt học vị Tiến sĩ ngành Khoa học máy tính tại đại học Kyoto, Nhật bản và được phong hàm Phó giáo sư năm 2022 tại Viện Công nghệ thông tin - Viện Hàn lâm khoa học và công nghệ Việt Nam. Hiện là nghiên cứu viên cao cấp tại Viện Công nghệ thông tin - Viện Hàn lâm khoa học và công nghệ