

TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

KHOA CÔNG NGHỆ THÔNG TIN

DỰ ÁN CÔNG NGHỆ THÔNG TIN

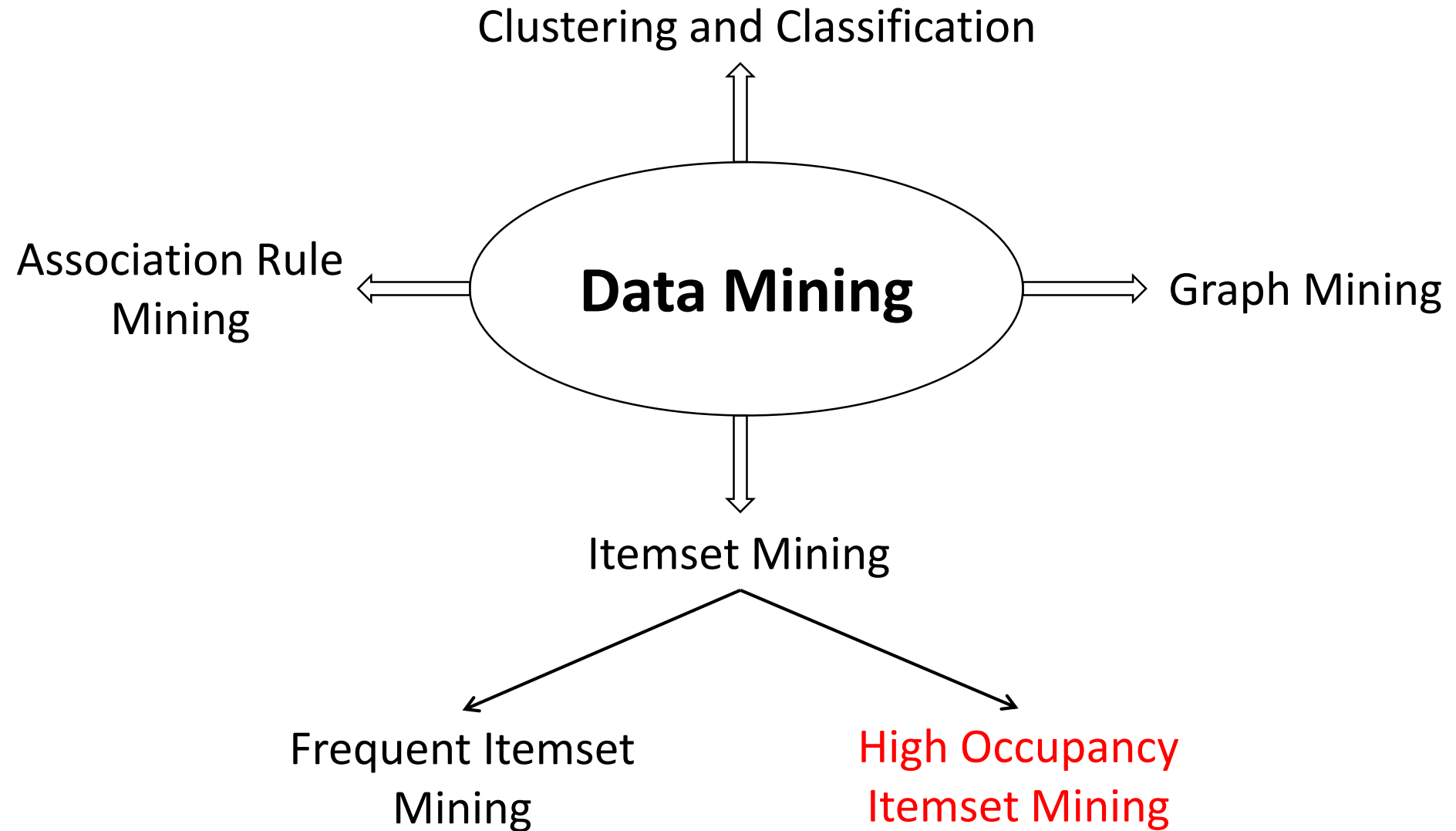
KHAI THÁC TẬP CHIẾM TỶ LỆ CAO TRONG CƠ SỞ DỮ LIỆU GIAO DỊCH

Hướng dẫn: ThS. Doãn Xuân Thanh

Thực hiện: Nghiêm Tiến Đạt – 52000025

1. GIỚI THIỆU
2. MỘT SỐ KHÁI NIỆM
3. THUẬT TOÁN
4. THỰC NGHIỆM
5. VẤN ĐỀ VÀ GIẢI PHÁP
6. ỨNG DỤNG (DEMO)
7. KẾT LUẬN

1. GIỚI THIỆU



1. GIỚI THIỆU

	Frequent Itemset Mining	High Occupancy Itemset Mining
Khởi đầu	Rakesh Agrawal (1993)	Zhi-Hong Deng (2017)
Quan tâm	Số lần xuất hiện của itemset trong transaction database (support)	Mức độ bao phủ của itemset trong transaction database (occupancy)
Mục đích	Tìm ra tất cả itemset có $\text{support} \geq \text{minSup}$	Tìm ra tất cả itemset có $\text{occupancy} \geq \text{minOcp}$
Thuật toán	APriori, FP-Growth, v.v.	HEP, DFHOI

2. MỘT SỐ KHÁI NIỆM

Định nghĩa 3. Occupancy của itemset P , kí hiệu $O(P)$, được tính như sau:

$$O(P) = \sum_{T \in STSet(P)} \frac{|P|}{|T|}$$

Định nghĩa 5. Occupancy-list của itemset P , kí hiệu $OL(P)$, là một danh sách mà mỗi phần tử chứa hai trường gồm:

1. tid là mã định danh của transaction T chứa P .
2. tsize là độ dài của T .

Định nghĩa 6. Giả sử các transaction chứa itemset P có u độ dài khác nhau.

Ngoài ra, gọi n_x ($1 \leq x \leq u$) là số lượng transaction có cùng độ dài l_x , và $\sum_{i=x}^u n_i \times \frac{l_x}{l_i}$ được kí hiệu là UBO_p^x . Giá trị cận trên occupancy (upper-bound occupancy) của P , kí hiệu $UBO(P)$, được tính như sau:

$$UBO(P) = \max_{1 \leq x \leq u} UBO_p^x$$

2. MỘT SỐ KHÁI NIỆM

Định lý 2. Gọi P, P_1, P_2 là ba itemset và occupancy-list của chúng là $OL(P)$, $OL(P_1)$, $OL(P_2)$ theo thứ tự. Nếu $P = P_1 \cup P_2$ thì $OL(P) = OL(P_1) \cap OL(P_2)$. Điều này cho phép ta có thể xây dựng occupancy-list của k -itemset từ occupancy-list của $(k - 1)$ -itemset mà không cần phải quét cơ sở dữ liệu nhiều lần.

Định lý 4. Với bất kì superset của itemset P , kí hiệu P' , ta có $O(P') \leq UBO(P)$. Dựa vào hệ quả của định lý, ta có thể giảm đáng kể phạm vi không gian tìm kiếm vì itemset có giá trị cận trên occupancy nhỏ hơn ngưỡng tối thiểu mà người dùng cung cấp thì các superset của nó không phải là tập chiếm tỷ lệ cao.

3. THUẬT TOÁN

Thuật toán HEP

- Do Zhi-Hong Deng đề xuất vào năm 2017.
- Khám phá k -itemset bằng cách kết hợp hai $(k - 1)$ -itemset với nhau.
- Sử dụng cấu trúc occupancy-list để không phải quét database nhiều lần.
- Sử dụng upper-bound occupancy để loại bỏ sớm các tập không đủ điều kiện.
- Thêm dòng số 16 để tránh việc các itemset trùng lặp có thể được duyệt qua nhiều lần.

$$P_1 \leftarrow \{a\}, P_2 \leftarrow \{b\}, \{c\}: \{a, b\}, \{a, c\}$$

$$P_1 \leftarrow \{b\}, P_2 \leftarrow \{a\}, \{c\}: \{b, a\}, \{b, c\}$$

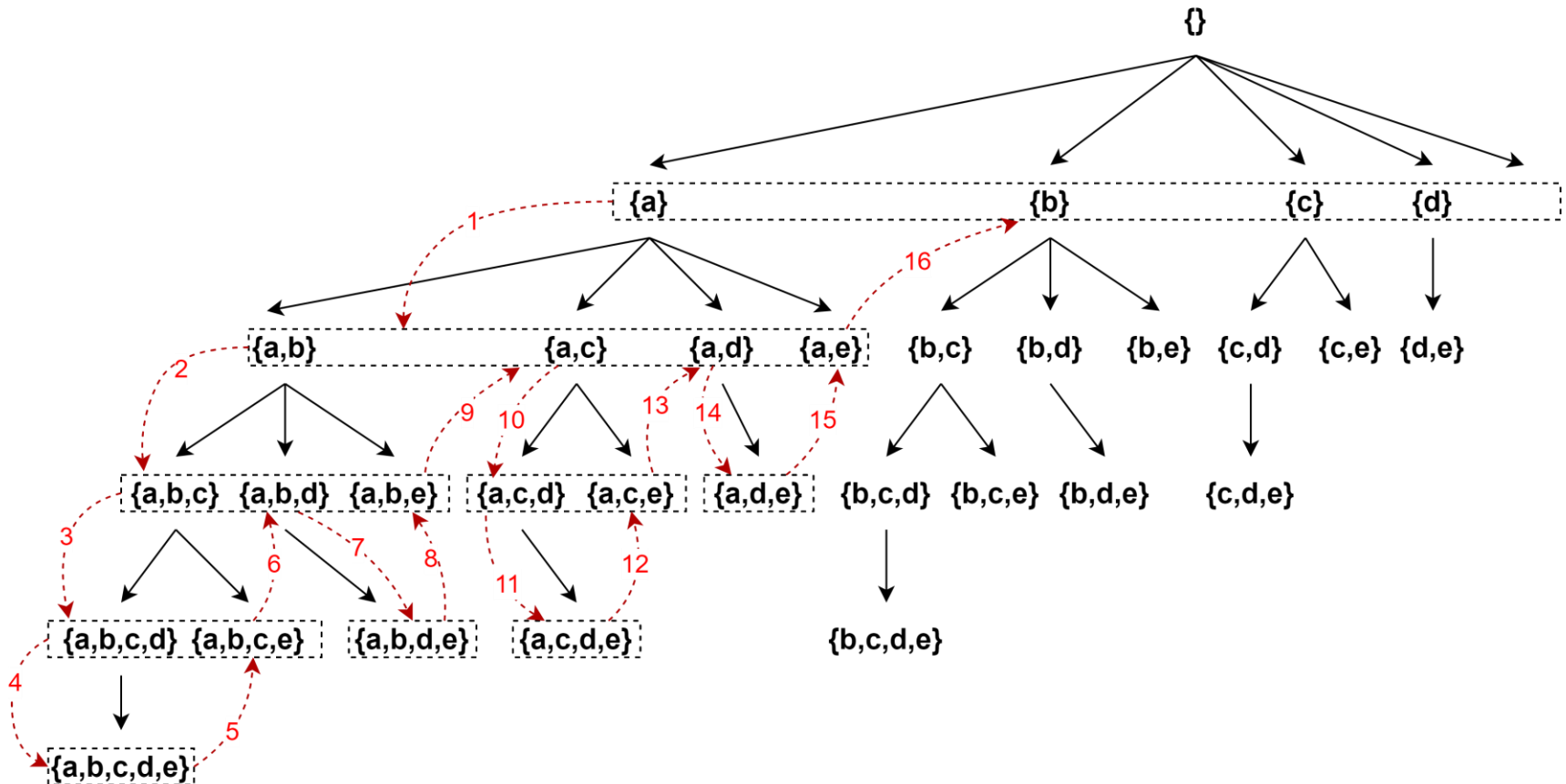
$$P_1 \leftarrow \{c\}, P_2 \leftarrow \{a\}, \{b\}: \{c, a\}, \{c, b\}$$

3. THUẬT TOÁN

Thuật toán DFHOI

- Do Loan Nguyen và các đồng tác giả khác đề xuất vào năm 2022.
- Sử dụng chiến thuật tìm kiếm theo chiều sâu (Depth First Search) để khám phá các itemset.
- Sử dụng mảng độ dài và support transaction set thay cho occupancy-list để tiết kiệm bộ nhớ.
- Không cần tính upper-bound occupancy trong trường hợp các transaction trong database có cùng độ dài.

3. THUẬT TOÁN



3. THUẬT TOÁN

Định nghĩa 7. Gọi $DB = \{T_1, T_2, \dots, T_n\}$ là cơ sở dữ liệu giao dịch, mảng L chứa index độ dài của các transaction trong database sao cho $L_i = |T_i|$. Mảng L được sử dụng để tiết kiệm bộ nhớ cho việc lưu trữ occupancy-list. Ví dụ, từ Bảng 1.1 ta có $L = \{3, 3, 4, 2, 3, 5\}$ và occupancy-list của mỗi itemset chỉ cần lưu thông tin support transaction set của itemset đó, nghĩa là $OL(P) = STSet(P)$. Dựa vào đó, ta được

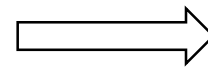
$$OL(\{a\}) = \{(T_1, 3), (T_2, 3), (T_4, 2), (T_6, 5)\}$$

$$OL(\{b\}) = \{(T_2, 3), (T_4, 2), (T_6, 5)\}$$

$$OL(\{c\}) = \{(T_1, 3), (T_3, 4), (T_5, 3), (T_6, 5)\}$$

$$OL(\{d\}) = \{(T_1, 3), (T_2, 3), (T_3, 4), (T_4, 2), (T_5, 3), (T_6, 5)\}$$

$$OL(\{e\}) = \{(T_3, 4), (T_5, 3), (T_6, 5)\}$$



$$OL(\{a\}) = \{T_1, T_2, T_4, T_6\}$$

$$OL(\{b\}) = \{T_2, T_3, T_6\}$$

$$OL(\{c\}) = \{T_1, T_3, T_5, T_6\}$$

$$OL(\{d\}) = \{T_1, T_2, T_3, T_4, T_5, T_6\}$$

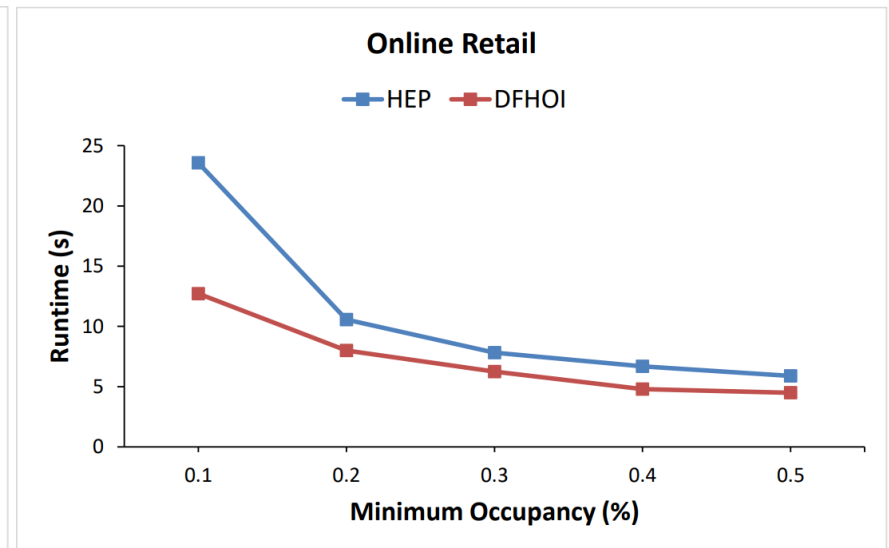
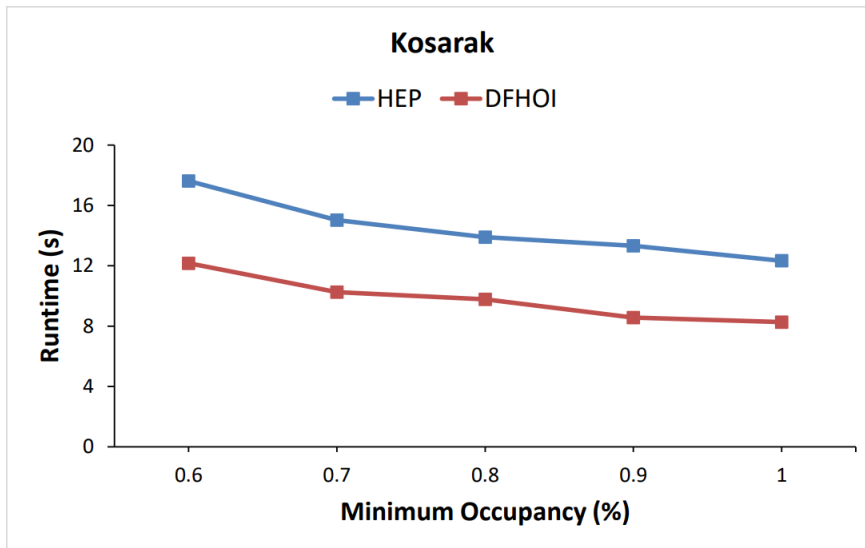
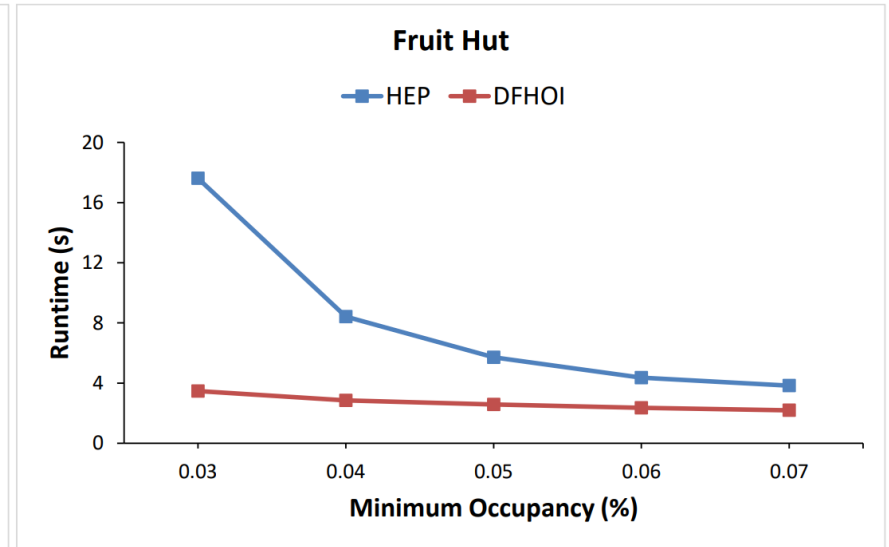
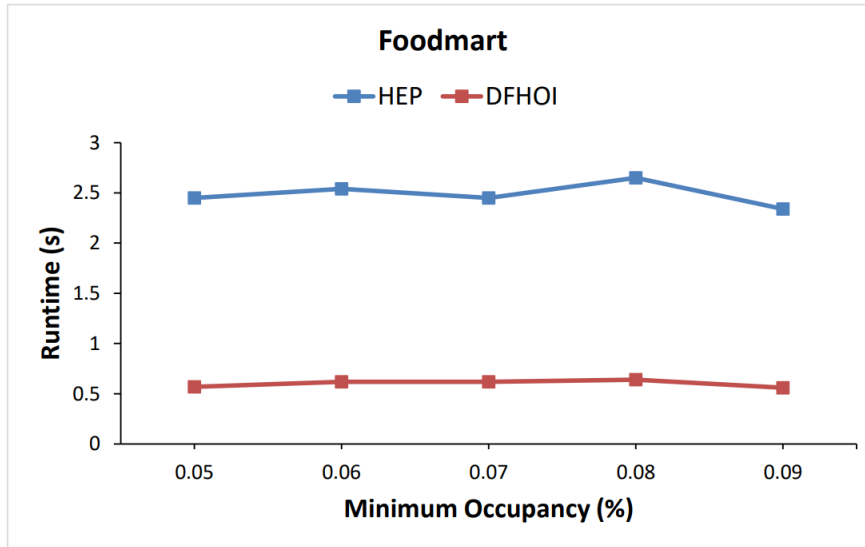
$$OL(\{e\}) = \{T_3, T_5, T_6\}$$

4. THỰC NGHIỆM

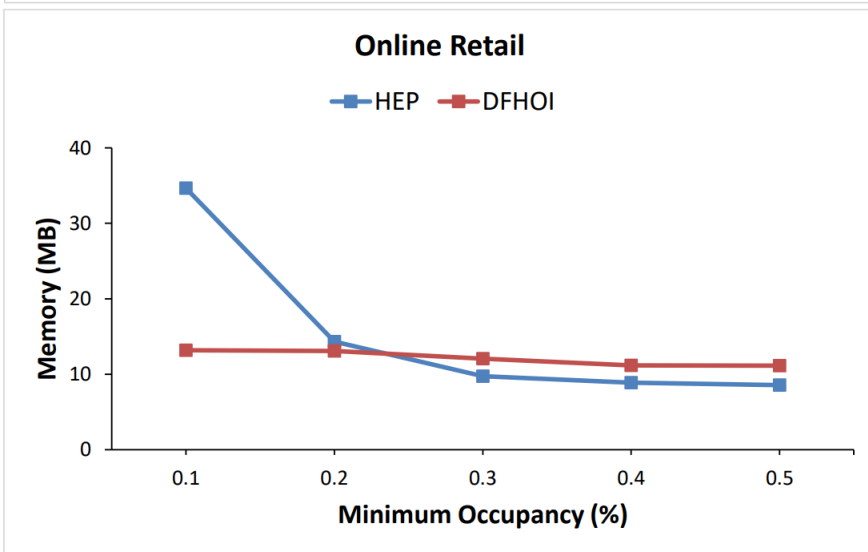
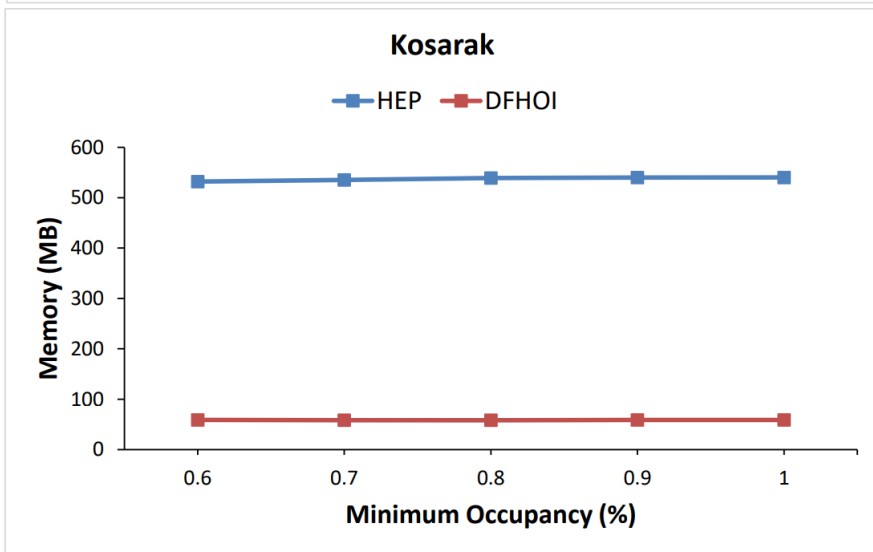
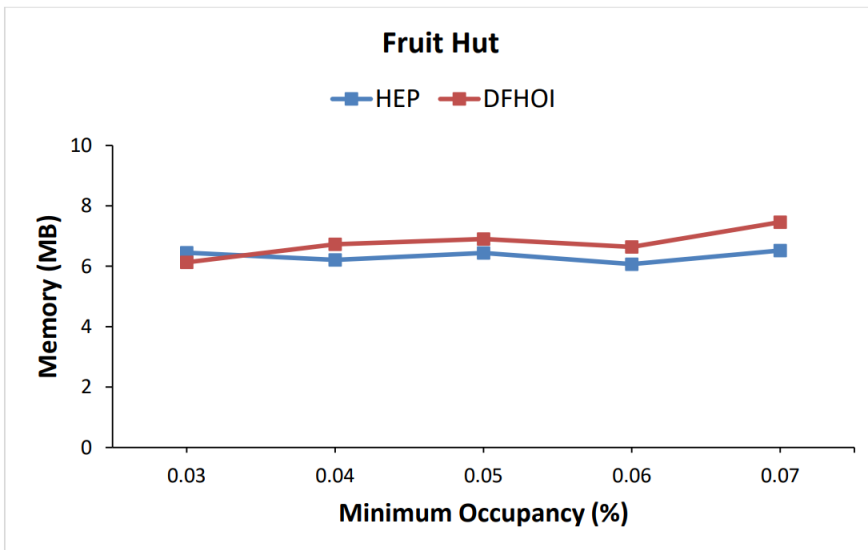
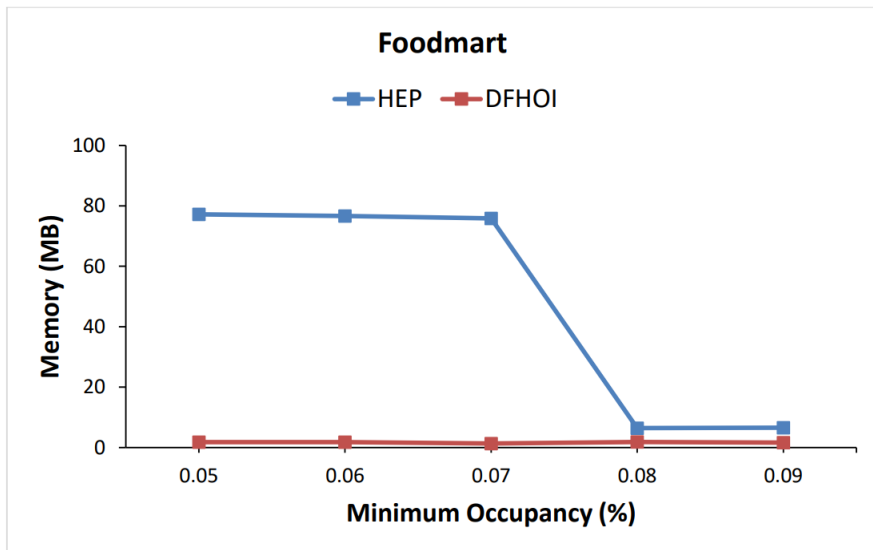
Triển khai thuật toán HEP và DFHOI bằng Python và Java, so sánh thời gian thực thi và bộ nhớ sử dụng trên 6 tập dữ liệu mẫu:

Dataset	#Trans	#Items	AvgLen	MaxLen
Foodmart	4,141	1,559	4.42	14
Fruit Hut	181,970	1,265	3.58	36
Kosarak	990,002	41,270	8.1	2,498
Online Retail	541,909	2,603	4.37	8
Retail	88,162	16,470	10.3	76
T10I4D100K	100,000	870	10.1	29

4. THỰC NGHIỆM



4. THỰC NGHIỆM

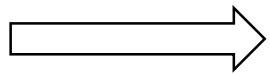


5. VẤN ĐỀ VÀ GIẢI PHÁP

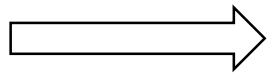
Làm thế nào để người dùng có thể chọn ngưỡng minOcp phù hợp?

TH1: minOcp quá thấp \rightarrow số lượng itemset tăng lên đáng kể
 \rightarrow tốn kém thời gian và tài nguyên tính toán.

TH2: minOcp quá cao \rightarrow không itemset nào thỏa mãn điều kiện.



Cần một thuật toán có khả năng lấy ngưỡng tự động



Thuật toán ATHOI

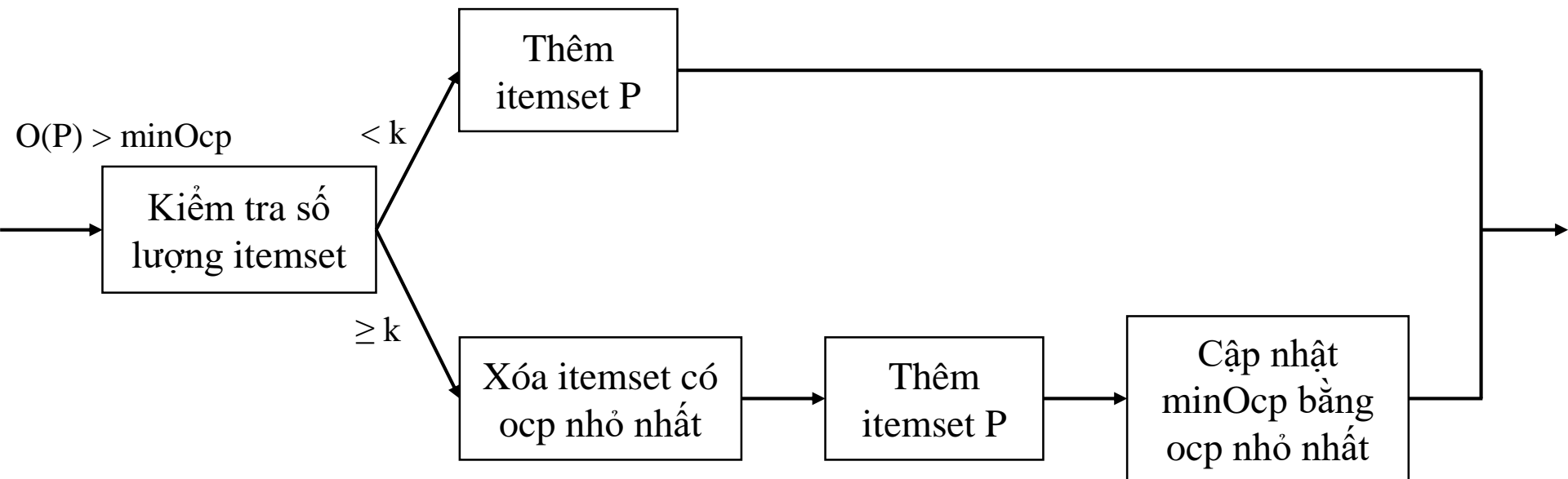
5. VẤN ĐỀ VÀ GIẢI PHÁP

	DFHOI	ATHOI
Chiến thuật tìm kiếm	Depth First Search (DFS)	Depth First Search (DFS)
Loại bỏ tập ứng viên	Upper-bound Occupancy (UBO)	Upper-bound Occupancy (UBO)
Input	Transaction database và minOcp	Transaction database và số lượng itemset cần khai thác (k)
Output	Tất cả itemset có occupancy \geq minOcp	Top k itemset có occupancy cao nhất

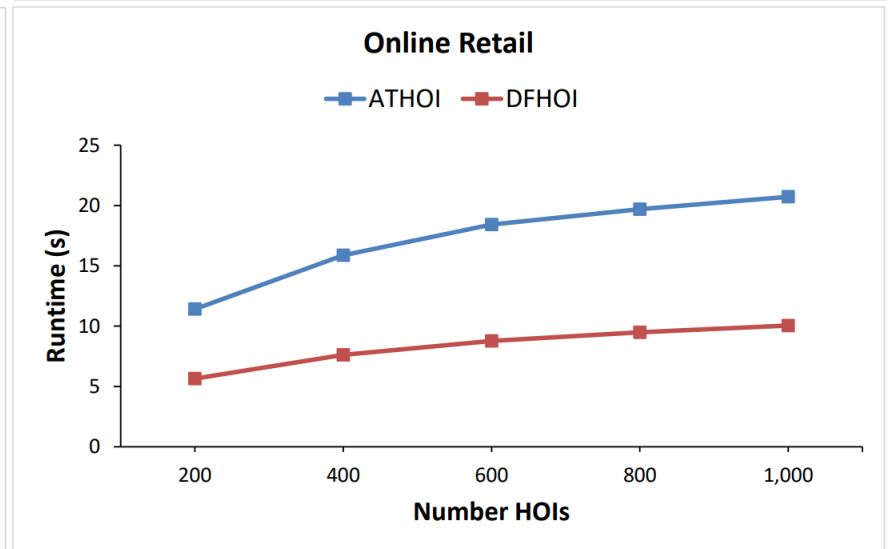
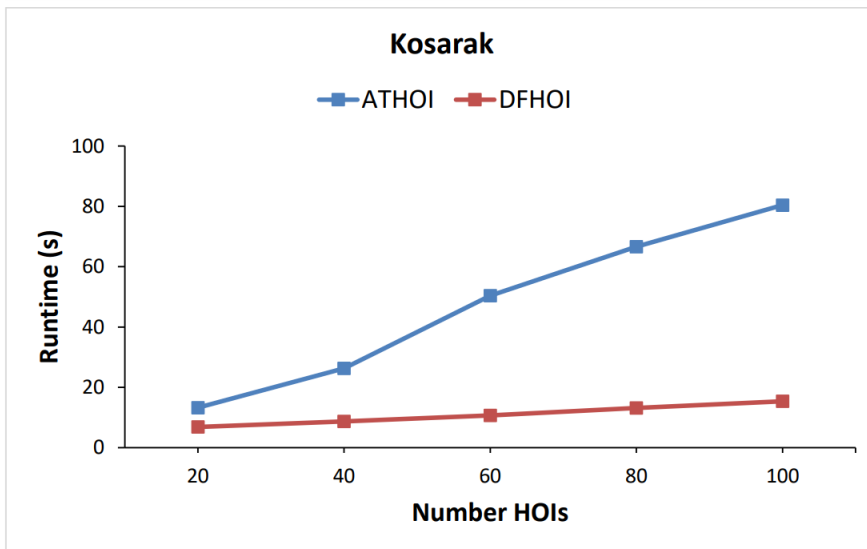
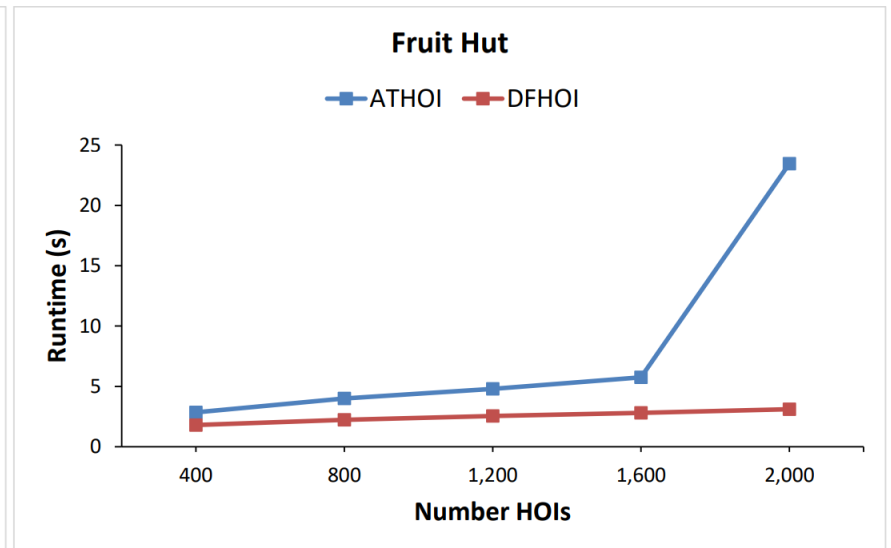
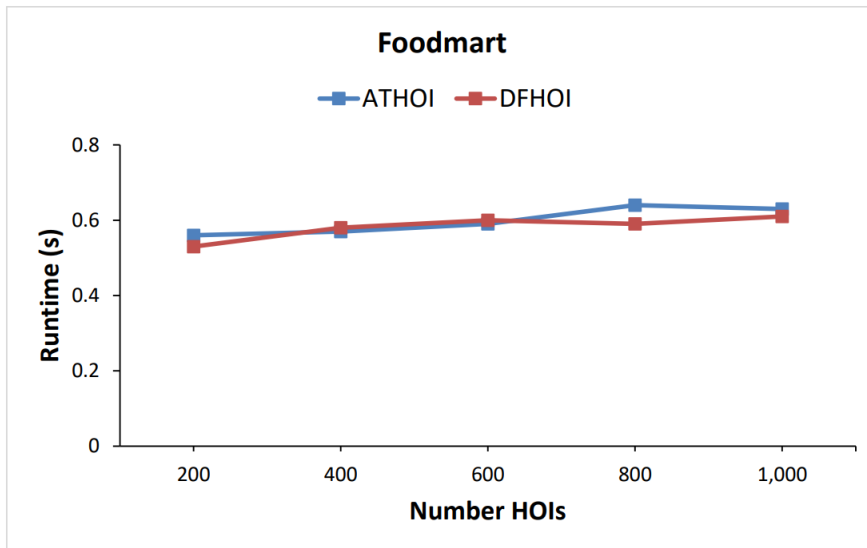
5. VẤN ĐỀ VÀ GIẢI PHÁP

Phương pháp lấy ngưỡng tự động của thuật toán ATHOI

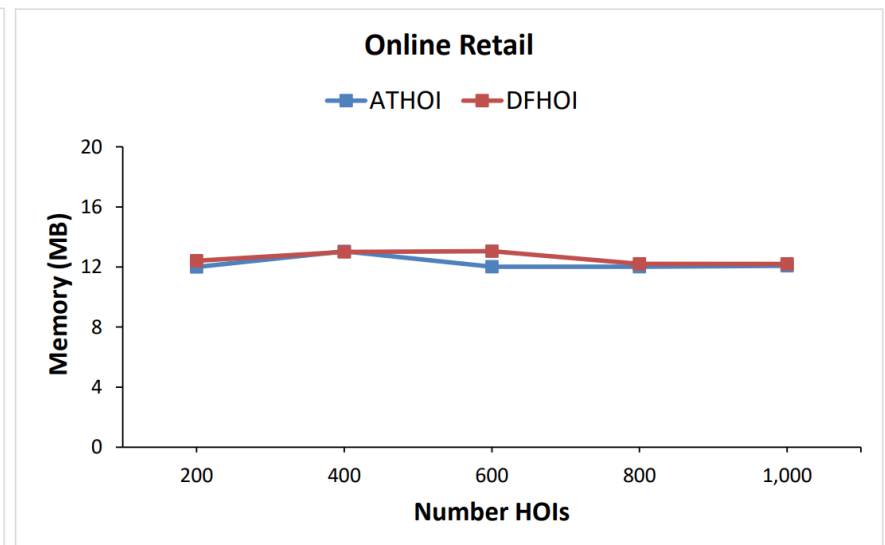
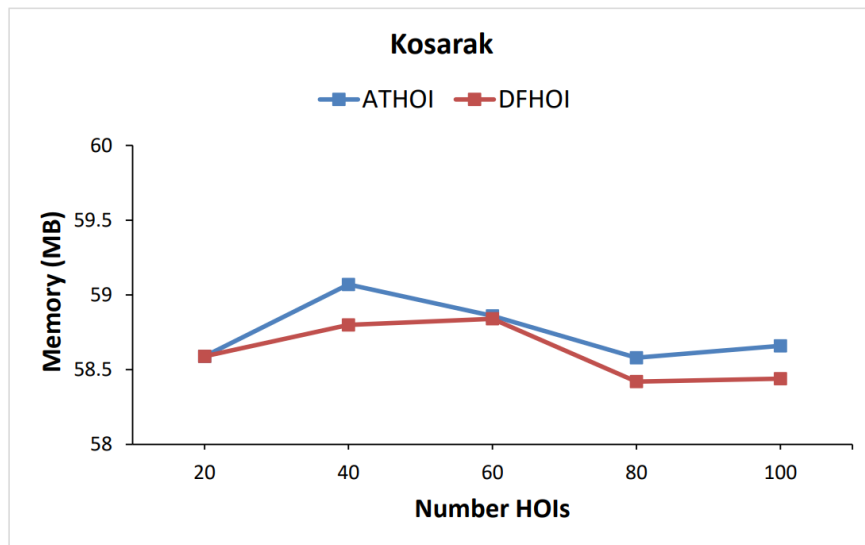
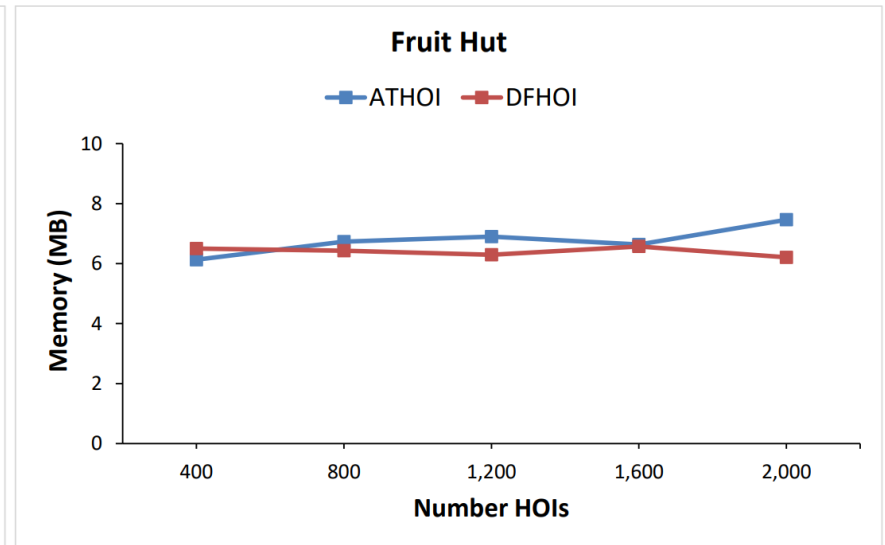
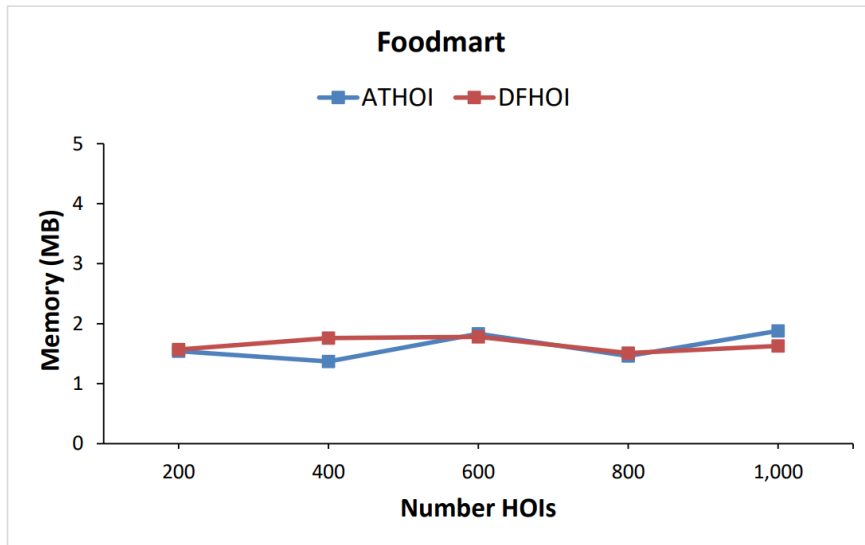
- Khởi đầu với $\text{minOcp} = 0$
- Duyệt qua từng itemset P, thực hiện:



5. VẤN ĐỀ VÀ GIẢI PHÁP



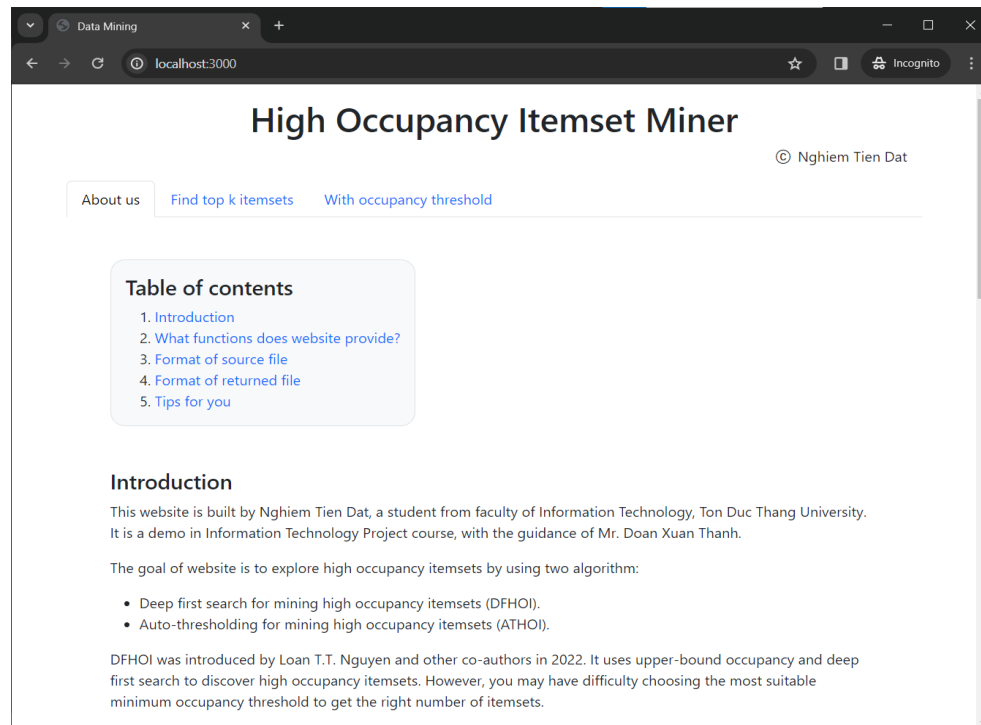
5. VẤN ĐỀ VÀ GIẢI PHÁP



6. ỨNG DỤNG DEMO

High Occupancy Itemset Miner là một ứng dụng trên nền tảng web để khai thác các tập chiếm tỷ lệ cao trong cơ sở dữ liệu giao dịch.

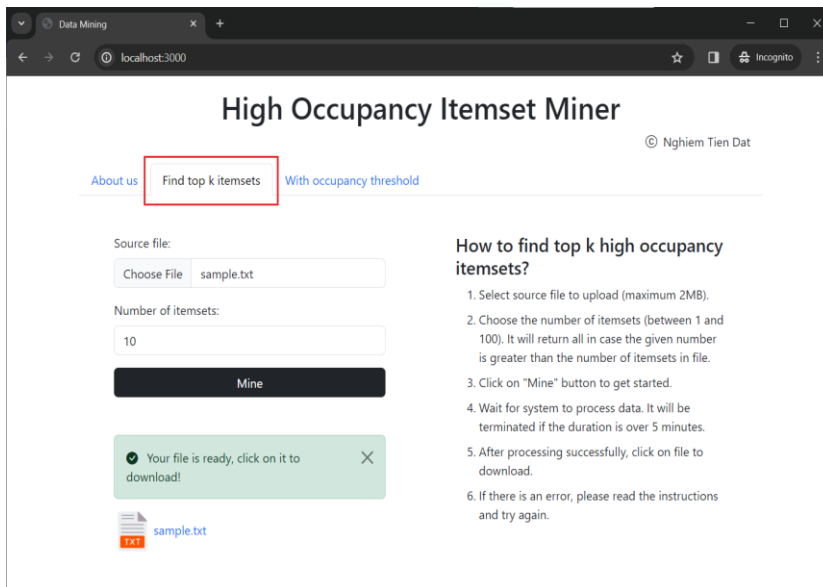
- Front-end: HTML, Bootstrap, Javascript
- Back-end: Nodejs, Python



6. ỨNG DỤNG DEMO

High Occupancy Itemset Miner cung cấp cho người dùng hai tính năng:

1. Tìm top k high occupancy itemset (ATHOI)
2. Tìm tất cả high occupancy itemset dựa vào minOcp (DFHOI)



High Occupancy Itemset Miner © Nghiem Tien Dat

About us **Find top k itemsets** With occupancy threshold

Source file:
Choose File sample.txt

Number of itemsets:
10

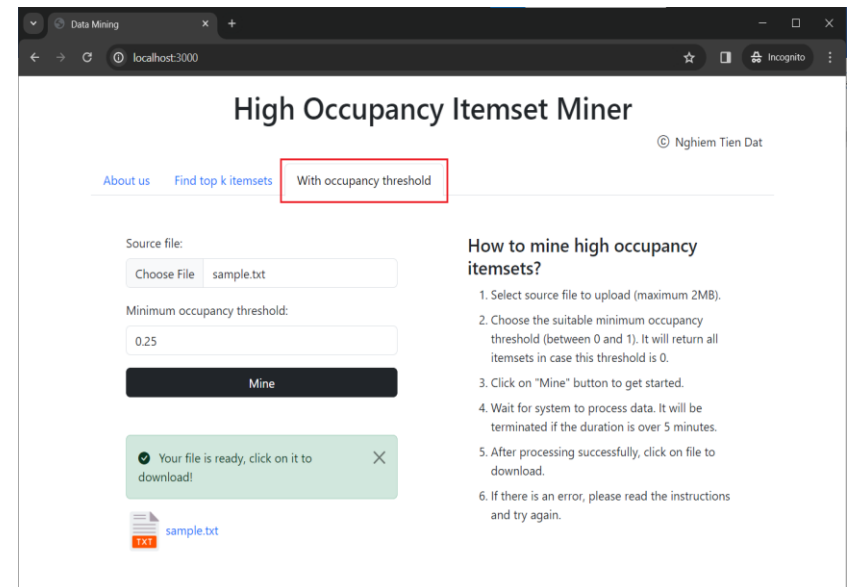
Mine

How to find top k high occupancy itemsets?

1. Select source file to upload (maximum 2MB).
2. Choose the number of itemsets (between 1 and 100). It will return all in case the given number is greater than the number of itemsets in file.
3. Click on "Mine" button to get started.
4. Wait for system to process data. It will be terminated if the duration is over 5 minutes.
5. After processing successfully, click on file to download.
6. If there is an error, please read the instructions and try again.

✓ Your file is ready, click on it to download!

sample.txt



High Occupancy Itemset Miner © Nghiem Tien Dat

About us Find top k itemsets **With occupancy threshold**

Source file:
Choose File sample.txt

Minimum occupancy threshold:
0.25

Mine

How to mine high occupancy itemsets?

1. Select source file to upload (maximum 2MB).
2. Choose the suitable minimum occupancy threshold (between 0 and 1). It will return all itemsets in case this threshold is 0.
3. Click on "Mine" button to get started.
4. Wait for system to process data. It will be terminated if the duration is over 5 minutes.
5. After processing successfully, click on file to download.
6. If there is an error, please read the instructions and try again.

✓ Your file is ready, click on it to download!

sample.txt

7. KẾT LUẬN

Nội dung đã thực hiện:

- Tìm hiểu về High Occupancy Itemset Mining.
- Triển khai thuật toán HEP và DFHOI bằng Python và Java.
- So sánh hiệu suất của hai thuật toán trên các tập dữ liệu mẫu.
- Đề xuất thuật toán lấy ngưỡng tự động ATHOI.
- Triển khai và thực nghiệm thuật toán ATHOI.
- Xây dựng ứng dụng High Occupancy Itemset Miner.

Phương hướng phát triển đề tài:

- Tiếp tục cải tiến thuật toán ATHOI.
- Thêm trọng số cho các item để khai thác tập sản phẩm chiếm lợi nhuận cao nhất (High Utility Occupancy Itemset Mining).

Deng, Z. H. (2017, February 18). Mining high occupancy itemsets. *Elsevier*, 222-229. Retrieved February 25, 2024, from <https://www.sciencedirect.com>

Nguyen, T. T., Mai, T., Pham, G. H., Yun, U., & Vo, B. (2022, February 20). An efficient method for mining high occupancy itemsets based on equivalence class and early pruning. *Elsevier*, 1-21. Retrieved February 25, 2024, from <https://www.sciencedirect.com>