Trí tuệ ứng dụng https://doi.org/10.1007/s10489-024-05826-y

Các thuật toán ngẫu nhiên mới để khai thác tiện ích bảo đảm quyền riêng tư

Đức Nguyên 1.2 Bắc Lê1.2

Được chấp nhận: ngày 28 tháng 8 năm 2024

Tóm tắt Khai thác tập mục tiện ích cao (HUIM) là một kỹ thuật để trích xuất những hiểu biết sâu sắc có giá trị từ dữ liệu. Khi xử lý thông tin nhạy cảm, HUIM có thể nêu lên những lo ngại về quyền riêng tư. Do đó, khai thác tiện ích bảo vệ quyền riêng tư (PPUM) đã trở thành một hướng nghiên cứu quan trọng. PPUM liên quan đến việc chuyển đổi cơ sở dữ liệu giao dịch định lượng thành các phiên bản được làm sạch để bảo vệ dữ liệu nhạy cảm trong khi vẫn giữ lại các mẫu hữu ích. Các nhà nghiên cứu trước đây đã sử dụng các phương pháp tối ưu hóa ngẫu nhiên để che giấu các mẫu nhạy cảm trong cơ sở dữ liệu thông qua việc thêm hoặc xóa các giao dịch. Tuy nhiên, những cách tiếp cận này làm thay đổi cấu trúc cơ sở dữ liệu. Để giải quyết vấn đề này, bài viết này giới thiệu một cách tiếp cận mới để ẩn dữ liệu với tối ưu hóa ngẫu nhiên mà không thay đổi cấu trúc cơ sở dữ liệu. Chúng tôi thiết kế hàm mục tiêu linh hoạt để cho phép người dùng hạn chế các tác động tiêu cực của PPUM theo yêu cầu cụ thể của họ. Chúng tôi cũng phát triển một chiến lược chung để thiết lập ma trận ràng buộc. Ngoài ra, chúng tôi trình bày một thuật toán ngẫu nhiên áp dụng trình tối ưu hóa kiến sư tử cùng với thuật toán lai, kết hợp cả phương pháp tối ưu hóa chính xác và ngẫu nhiên, để giải quyết vấn đề ẩn. Kết quả của các thử nghiệm mở rộng được trình bày, chứng minh tính hiệu quả và tính linh hoạt của các thuật toán đề xuất.

Từ khóa Kiến sư tử · Trình tối ưu hóa · Quyền riêng tư · Khai thác tiện ích

1 Giới thiệu

Khai thác dữ liệu là việc trích xuất các mẫu thú vị từ cơ sở dữ liệu, chủ yếu nhắm mục tiêu những hiểu biết có giá trị từ các bộ dữ liệu phức tạp [1]. Khai thác tiện ích, một kỹ thuật phổ biến, tập trung vào việc xác định các tập mục có tiện ích cao (HUI) dựa trên lợi nhuận đơn vị và số lượng giao dịch. Cách tiếp cận này đã thu hút được sự chú ý đáng kể vì khả năng ứng dụng rộng rãi của nó. Tuy nhiên, việc khám phá các mô hình trong dữ liệu kinh doanh và sức khỏe làm tăng mối lo ngại về quyền riêng tư [2]. Các mô hình ẩn có thể làm lộ danh tính cá nhân, các mối quan hệ riêng tư hoặc bí mật thương mại bí mật. Một cách tiếp cận để giải quyết những mối lo ngại này là xử lý dữ liệu bằng các phương pháp phi tập trung như học tập liên kết [3]. Tuy nhiên, có một số thách thức; ví dụ: khối lượng dữ liệu trong các thiết bị phân tán cần là đủ và các giao thức truyền thông cần phải được bảo mật [4]. Là một cách tiếp cận truyền thống, các kỹ thuật khai thác tiện ích bảo vệ quyền riêng tư (PPUM) đã được phát triển để bảo vệ thông tin nhạy cảm trước khi chia sẻ và khai thác dữ liệu [5].

Trong hầu hết các nghiên cứu được công bố, các tập mục hữu ích cao (SHUI) nhạy cảm đều bị ẩn bằng cách sử dụng các phương pháp loại bỏ hoặc giảm bớt tiện ích của chúng trong cơ sở dữ liệu. Sớm nhất được phát triển bởi Yeh et al. [6], bao gồm hai thuật toán, đó là ẩn mục có ích cao trước (HHUIF) và xung đột tập mục nhạy cảm tối đa trước (MSICF). Các thuật toán này xác định các giao dịch và vật phẩm của nạn nhân, sau đó giảm dần số lượng vật phẩm của nạn nhân để làm giảm tiện ích của SHUI. Hai thuật toán heuristic bổ sung đã được đề xuất bởi Lin et al. [7], cụ thể là tiện ích mục tối đa-tiện ích nhạy cảm tối đa (MSU-MAU) và tiện ích mục tối thiểu-tiện ích nhạy cảm tối đa (MSU-MIU). Các thuật toán này đã giới thiệu khái niệm tiện ích có độ nhạy tối đa như một phương pháp phỏng đoán hiệu quả. Gần đây, các kỹ thuật sắp xếp và các khái niệm mới [8, 9] cũng đã được trình bày. Cách tiếp cận thứ hai được đề xuất bởi Lin et al. [10]. Trong phương pháp này, cơ sở dữ liệu gốc bị xáo trộn bằng cách chèn các giao dịch giả tạo hoặc loại bỏ các giao dịch hiện tại.

Những kỹ thuật này sử dụng các khái niệm heuristic để ẩn các tập mục nhạy cảm. Tuy nhiên, chúng có tính tổng quát thấp và

B Bắc Lê
lhbac@fit.hcmus.edu.vn
Đức Nguyễn
nnduc@fit.hcmus.edu.vn

1 Khoa Công nghệ thông tin, Đại học Khoa học Tự nhiên, Thành phố Hồ Chí Minh, Việt Nam

2 Vietnam National University, Ho Chi Minh City, Vietnam

Published online: 27 September 2024

tỷ lệ tác động tiêu cực. Đối với một chiến lược chính thức hơn, Li et al. [11] trình bày thuật toán PPUM-ILP. Họ đã phát triển bảng HI để lưu trữ các HUI sẽ bị ảnh hưởng trong quá trình ẩn giấu. Trạng thái của các tập mục này được thay đổi một cách hiệu quả bằng cách sử dụng các kỹ thuật lập trình số nguyên. Vì việc loại bỏ các ràng buộc lặp đi lặp lại để hình thành và giải quyết một mô hình khả thi dẫn đến việc thực thi chậm, Duc et al. [12] đã thiết kế một phương pháp thư giãn tối ưu hóa toán học. Hơn nữa, họ sử dụng phương pháp tiền xử lý song song để giải quyết vấn đề ẩn.

Tối ưu hóa là quá trình tìm kiếm một giải pháp tốt từ tập hợp khả thi của một vấn đề tối ưu hóa. Thách thức là hầu hết các vấn đề tối ưu hóa đều là NP-hard. Thay vì tìm giải pháp chính xác, các nhà nghiên cứu sử dụng phương pháp ngẫu nhiên để tạo ra một quá trình tìm kiếm lặp lại có thể thoát khỏi cực tiểu cục bộ [13]. Trong những năm gần đây, các thuật toán tối ưu hóa ngẫu nhiên mới đã xuất hiện và được áp dụng cho nhiều lĩnh vực khác nhau [14, 15], một số thuật toán đã đạt được hiệu suất vượt trội. Các hành vi ngẫu nhiên trong nhiều thuật toán này được lấy cảm hứng từ các hiện tượng tự nhiên, bao gồm các hành vi săn bắn (sư tử [16], tắc kè hoa [17], linh cẩu [18]), lỗ đen [19] và quá trình chính trị xã hội [20], trong số những người khác [21–23]. Kỹ thuật tối ưu hóa ngẫu nhiên cũng được áp dụng rộng rãi để bảo vệ quyền riêng tư trong khai thác dữ liệu [24].

Trong bài báo này, hai thuật toán PPUM, được gọi là PPUM-ALO và PPUM-IALO, được đề xuất để giải quyết vấn đề vệ sinh bằng cách tiếp cận linh hoạt hơn cho phép sử dụng cả phương pháp tối ưu hóa chính xác và ngẫu nhiên để giải quyết vấn đề ẩn. Những đóng góp chính của chúng tôi được tóm tắt như sau:

- Đề xuất chiến lược tổng thể để hình thành bài toán tối ưu hóa việc che giấu các mẫu nhạy cảm.
- Giới thiệu một bài toán tối ưu hóa mới cho việc làm sạch dữ liệu, bài toán này có thể được giải bằng các thuật toán ngẫu nhiên. Mô hình tối ưu hóa mới có thể được các chuyên gia miền sửa đổi bằng các siêu tham số cho các tác dụng phụ, cho phép họ tùy chỉnh giải pháp theo yêu cầu của mình.
- Phát triển hai phương pháp ngẫu nhiên, cụ thể là PPUM-ALO và PPUM-IALO, để giải quyết vấn đề ẩn giấu.
- Chứng minh tính linh hoạt và sức mạnh của các thuật toán đề xuất bằng thực nghiệm và so sánh với các phương pháp thay thế.

Các phần còn lại của bài viết này được cấu trúc như sau. Việc xem xét các công trình đã xuất bản về HUIM, PPUM và tối ưu hóa ngẫu nhiên được trình bày trong Phần 2. Phần 3 cung cấp thông tin cơ bản về vấn đề PPUM. Chi tiết cụ thể của phương pháp PPUM đề xuất được trình bày trong Phần 4. Phần 5 trình bày các kết quả thử nghiệm so sánh trên các bộ dữ liệu điển hình và chứng minh hiệu quả và hiệu quả vượt trội của các thuật toán được đề xuất. Cuối cùng, Phần 6 tóm tắt bài viết và thảo luận các hướng nghiên cứu trong tương lai.

2 công trình liên quan

2.1 Khai thác tập mục có tiện ích cao

Khai thác tiện ích là một chủ đề nghiên cứu rộng rãi trong khai thác dữ liệu, giải quyết vấn đề tìm kiếm thông tin hữu ích trong cơ sở dữ liệu giao dịch định lượng. Công trình đầu tiên trong lĩnh vực này được xuất bản vào năm 2004 [25]. Các tác giả đã phát triển thuật toán khai thác tập mục hữu ích (HUIM) bằng cách sử dụng hai loại thông tin tiện ích để xác định các mẫu ẩn hữu ích. Vì HUI không có thuộc tính đóng xuống, Liu [26] đã giới thiệu khái niệm sử dụng theo trọng số giao dịch (TWU) để cắt bớt các tập mục không cần thiết.

Để khai thác các mẫu một cách hiệu quả, các cấu trúc dữ liệu chuyên biệt cần được thiết kế. Lin và cộng sự. [27] trình bày HUP-Tree. Tseng [28] đã thiết kế UP-Tree theo sau là hai thuật toán tiếp theo, UP-growth [28] và UP-growth+ [29]. Ngoài ra, cấu trúc danh sách tiện ích [30] được phát triển để tìm HUI trực tiếp. Thuật toán EFIM [31] biểu diễn cơ sở dữ liệu và hợp nhất các giao dịch với hai giới hạn trên: tiện ích cây con được sửa đổi và tiện ích cục bộ. Kim và cộng sự. [32] áp dụng phương pháp cửa sổ trượt bằng cách chèn dữ liệu dòng chảy.

Đối với các vấn đề khai thác với các tiện ích bên ngoài không chắc chắn, Gan et al. [33] đã giới thiệu một phương pháp khai thác để thay đổi các mẫu. Cơ sở dữ liệu động đã được khai thác bằng thuật toán MCH-miner [34]. Hơn nữa, thuật toán MCUI-miner [35] kết hợp thuật toán di truyền và phương pháp MapReduce để tìm kiếm các mẫu đóng. Trong môi trường phân tán, Lin et al. [36] các khái niệm phân cụm đã điều chỉnh cho phù hợp với các giao dịch nhóm dựa trên mối tương quan của chúng.

2.2 Khai thác tiện ích bảo vệ quyền riêng tư

Với sự ra đời của thuật toán HUIM, việc khai thác tiện ích bảo vệ quyền riêng tư (PPUM) đã trở thành một chủ đề nghiên cứu quan trọng. Vấn đề PPUM ban đầu được giải quyết bởi Yeh et al. [6], người đã đề xuất hai thuật toán heuristic, HHUIF và MSICF. Mục đích của các thuật toán này là che giấu SHUI bằng cách giảm số lượng vật phẩm trong giao dịch.

Để đẩy nhanh quá trình lẫn trốn, Yun et al. [37] đã phát triển một cấu trúc cây gọi là FPUTT. Với FPUTT, quá trình dọn dẹp chỉ cần ba lần quét cơ sở dữ liệu. Mặc dù nhanh hơn các thuật toán trước đó, FPUTT có thể tạo ra các cây không nén trên các tập dữ liệu lớn, dẫn đến mức sử dụng bộ nhớ cao và chi phí tính toán. Vì việc quét cơ sở dữ liệu nhiều lần rất tốn thời gian, Liu và cộng sự. [38] đã xây dựng bảng T và bảng HUI. Vì nó cập nhật cả bảng T và

Bảng HUI cho từng mục được sửa đổi, thời gian chạy của thuật toán vẫn ở mức cao. Trong một nỗ lực khác nhằm cải thiện tốc độ của quá trình khử trùng, Yin và Yi [39] đã trình bày cấu trúc từ điển danh sách tiện ích, có cách tiếp cận tương tự với bảng HI [11], bảng IT [40] và bảng GIT [12].

Lin và cộng sự. [7] đã giới thiệu khái niệm tiện ích có độ nhạy tối đa và hai thuật toán heuristic: MSU-MAU và MSU-MIU. Họ cũng đề xuất ba phép đo về tác dụng phụ của quá trình ẩn náu. Ngoài ra, Lin và cộng sự. [10, 41] đã đề xuất hai phương pháp liên quan đến thuật toán di truyền, PPUMGA+insert và PPUMGAT. Các mục nhạy cảm được ẩn bằng cách chèn các giao dịch giả tạo hoặc xóa các giao dịch hiện có. Tuy nhiên, những thay đổi của họ trong cơ sở dữ liệu có thể tạo ra những hiểu biết sai lệch.

Để giải quyết thách thức còn lại trong việc giảm thiểu tác dụng phụ, Liu và cộng sự. [42] đã đề xuất một phiên bản nâng cao của thuật toán MSICF, IMSICF. Quá trình dọn dẹp phải chịu chi phí tính toán cao hơn vì tính toán động của thuật toán này về xung đột xem xét từng mục nhạy cảm. Gần đây, ba phương pháp đã được giới thiệu bởi Jangra et al. [8] dựa trên kỹ thuật sắp xếp dữ liệu. Hiệu quả của các kỹ thuật sắp xếp đã được chứng minh bằng ba thuật toán nữa [9]. Các thử nghiệm cho thấy các thuật toán này có khả năng giảm tác dụng phụ tốt hơn so với các thuật toán heuristic trước đó. Một khái niệm mới có tên là tiện ích nhạy cảm với sản phẩm thực (RISU) cũng được mô tả trong tác phẩm này.

Một cách tiếp cận tổng quát hơn đã được giới thiệu bởi Li et al. [11], trong đó quá trình dọn dẹp được xây dựng dưới dạng một bài toán tối ưu hóa và được giải bằng các bộ giải lập trình số nguyên. Mặc dù hiệu suất tổng thể của PPUM-ILP [11] là tốt, nhưng việc xây dựng các bài toán thỏa mãn ràng buộc với nhiều biến dẫn đến thời gian chạy đáng kể. Đức và cộng sự. [40] đã giải quyết nhược điểm này bằng cách trình bày một quy trình xây dựng nhanh chóng các phương pháp tiếp cận ILP. Ngoài ra, một phương pháp song song để tiền xử lý và xây dựng bài toán vệ sinh đã được đề xuất [12] cùng với việc giải quyết bài toán ẩn.

2.3 Tối ưu hóa ngẫu nhiên

Trong các thuật toán tối ưu hóa ngẫu nhiên, các toán tử ngẫu nhiên thêm các điều chỉnh ngẫu nhiên vào các giải pháp ứng viên, hỗ trợ họ thoát khỏi sự tối ưu cục bộ. Những kỹ thuật tối ưu hóa này lấy cảm hứng từ thế giới tự nhiên. Chúng tuân theo một khuôn khổ chung bắt đầu bằng một tập hợp các giải pháp ngẫu nhiên, sau đó được cải tiến và nâng cao thông qua các phương pháp khác nhau dành riêng cho mỗi thuật toán.

Điều làm nên sự khác biệt của các thuật toán này là cách tiếp cận cụ thể mà chúng áp dụng để cải thiện bộ giải pháp của mình. Thuật toán di truyền (GA) [43] bắt chước chọn lọc tự nhiên. Tối ưu hóa bầy đàn (PSO) [44] bắt nguồn từ thói quen bầy đàn. Tối ưu hóa đàn kiến (ACO) [45] mô phỏng chuyển động

Bảng 1 Số lượng mặt hàng trong giao dịch

nhận dạng	1	2	3	4	5
1	10	5	0	7	2
2	6	0	2	0	0
3	0	0	0	5	1
4	0	0	3	0	8
5	1	6	0	10	0
6	0	0	0	1	3
7	8	7	0	0	0
8	10	0	6	3	0
9	2	0	8	0	0
10	10	0	1	0	0

của loài kiến. Các thuật toán này rất hiệu quả và có thể giải quyết được nhiều vấn đề trong thế giới thực.

Một số thuật toán tối ưu hóa gần đây hơn là trình tối ưu hóa sói xám (GWO) [46], trình tối ưu hóa linh cẩu đốm (SHO) [18], trình tối ưu hóa kiến sư tử (ALO) [16] và điều tra dựa trên pháp y (FBI) [47]. Các biến thể của các thuật toán này đã được phát triển để giải quyết các vấn đề cụ thể. Bài viết này đề xuất các phương pháp ngẫu nhiên cho PPUM. ALO được chọn để tìm giải pháp vì tính đơn giản của nó và vì nó có thể được hướng dẫn bằng bước khởi tạo thích hợp.

3 vòng sơ loại

Cho T là tập hợp n giao dịch: T = {T1, T2, . . . , Th }. Giao dịch Th ∈ T chứa các mục riêng biệt. Mỗi mặt hàng đều có số lượng hỗ trợ giao dịch. Cho I = {i1, i2, . . . , ik } là tập hợp các phần tử riêng biệt trong cơ sở dữ liệu. Giao dịch Th là tập con của I. Một mục ik trong cơ sở dữ liệu có lợi nhuận riêng. Cơ sở dữ liệu giao dịch định lượng D có thể được biểu diễn bằng hai bảng: Bảng 1 liệt kê chi tiết giao dịch và Bảng 2 liệt kê tiện ích của các mục.

Định nghĩa 1 (itemset) Thuật ngữ "itemset" biểu thị một tập hợp bao gồm các mục.

Định nghĩa 2 (tidset) Một tidset đại diện cho một tập hợp các mã định danh duy nhất cho các giao dịch.

Bảng 2 Lợi nhuận mặt hàng

Mục	Lợi nhuận
1	7
2	3
3	9
4	5
5	6

Định nghĩa 3 (Tiện ích nội bộ) Tiện ích nội bộ của một vật phẩm trong giao dịch được xác định bởi số lượng của nó. Chúng ta biểu thị tiện ích nội tại của mục ik trong giao dịch Th bằng q(ik , Th).

Ví dụ, trong Bảng 1, tiện ích nội tại của mục 1 trong giao dịch T1 là 10, do đó q(1, T1) = 10.

Định nghĩa 4 (tiện ích bên ngoài) Tiện ích bên ngoài thể hiện tầm quan trọng của một mục trong cơ sở dữ liệu. Chúng ta biểu thị tiện ích bên ngoài của mục ik bằng e(ik).

Ví dụ, trong Bảng 2, tiện ích bên ngoài của mục 2 là 3, vì vậy e(2) = 3.

Định nghĩa 5 (Công dụng của một hạng mục trong giao dịch) Công dụng của hạng mục ik trong giao dịch Th, ký hiệu là u(ik , Th), là tích của các tiện ích bên trong và bên ngoài của nó:

$$u(ik , Th) = q(ik , Th) \times e(ik). \quad (1)$$

Tiện ích của mục 2 trong giao dịch T1 của cơ sở dữ liệu D là u(2, T1) = q(2, T1) × e(2) = 5 × 3 = 15.

Định nghĩa 6 (tiện ích của một tập mục trong một giao dịch) Gọi u(A, Th) biểu thị tiện ích của tập mục A trong giao dịch Th . Nếu tập mục A không được Th hỗ trợ thì u(X, Th) sẽ bằng 0. Mặt khác, u(A, Th) đề cập đến tổng hữu dụng của tất cả các mục trong A trong giao dịch Th :

$$u(A, Th) = \sum_{ik \in A} u(ik , Th). \quad (2)$$

Trong D, tiện ích của tập mục {3, 5} trong giao dịch T4 được tính như sau: u ({3, 5}, T4) = u(3, T4) + u(5, T4) = 3 × 9 + 8 × 6 = 75. Vì {3, 5} T1 , u({3, 5}, T1) = 0.

Định nghĩa 7 (Tiện ích của một tập mục trong cơ sở dữ liệu) Công dụng của tập mục A trong cơ sở dữ liệu D được xác định bằng tổng các tiện ích của nó trong việc hỗ trợ các giao dịch:

$$bạn(A) = \sum_{T \in D, A \subseteq Th} u(A, Th). \quad (3)$$

Ví dụ, tiện ích của tập mục {4, 5} trong cơ sở dữ liệu D được tính như sau: u({{4, 5}) = u({4, 5}, T1)+u({4, 5}, T3)+ u({4, 5}, T6) = (7 × 5 + 2 × 6) + (5 × 5 + 1 × 6) + (1 × 5 + 3 × 6) = 101.

Định nghĩa 8 (ngưỡng tiện ích tối thiểu) Ngưỡng tiện ích tối thiểu, ký hiệu là δ, là thước đo được sử dụng để xác định xem các tập mục có mang lại tiện ích cao hay không. Giá trị của nó được xác định bởi người dùng.

Định nghĩa 9 (tập mục hữu ích cao) Nếu tập mục A có tiện ích trong cơ sở dữ liệu bằng hoặc lớn hơn ngưỡng tối thiểu δ, thì nó đủ tiêu chuẩn là tập mục hữu ích cao (HUI).

Những thay đổi trong cơ sở dữ liệu xảy ra do quá trình ẩn ảnh hưởng đến cả các mẫu nhạy cảm và không nhạy cảm. Lin và cộng sự. [7] đề xuất ba biện pháp khắc phục tác động tiêu cực. Để làm rõ ảnh hưởng của quá trình che giấu đối với NSHUI, Li et al. [11] đưa ra khái niệm về chi phí che giấu.

Cho cơ sở dữ liệu D, gọi D' là cơ sở dữ liệu đã được chọn lọc, H là tập hợp các HUI trong D và H' là tập hợp các HUI trong D'. Gọi N là tập hợp các HUI không nhạy cảm (NSHUI) trong D và S là tập hợp các SHUI trong D.

Định nghĩa 10 (lỗi ẩn) Lỗi ẩn được ký hiệu là α và biểu thị tỷ lệ SHUI trong cơ sở dữ liệu trước và sau khi khử trùng:

$$\alpha = \frac{|S \cap H'|}{|S|}. \quad (4)$$

Ví dụ: nếu α = 0,5 thì 50% tập mục nhạy cảm không bị ẩn.

Định nghĩa 11 (chi phí bị thiếu) Tỷ lệ β, được gọi là chi phí còn thiếu, thể hiện tỷ lệ phần trăm NSHUI bị mất tiện ích và không thể xác định được trong cơ sở dữ liệu bị nhiễu loạn:

$$\beta = \frac{|N - N \cap H'|}{|N|}. \quad (5)$$

Ví dụ: nếu β = 0,6 thì 60% HUI không nhạy cảm sẽ trở thành tập mục có tiện ích thấp.

Định nghĩa 12 (chi phí nhân tạo) Chi phí nhân tạo γ là tỷ lệ các tập mục có tiện ích thấp chuyển đổi thành HUI trong cơ sở dữ liệu đã được làm sạch D'. Điều này có thể được tính toán như sau:

$$\gamma = \frac{|H' - H \cap H'|}{|N|}. \quad (6)$$

Ví dụ: nếu γ = 0,2 thì có 20% HUI dư thừa so với cơ sở dữ liệu ban đầu.

Định nghĩa 13 (chi phí ẩn) Chi phí ẩn thể hiện những thay đổi đối với các tập mục không nhạy cảm trong cơ sở dữ liệu:

$$= \frac{|N - H'|}{|N|}. \quad (7)$$

Trong nghiên cứu này, chúng tôi đánh giá kết quả bằng cách sử dụng lỗi ẩn, chi phí còn thiếu và chi phí nhân tạo. Do đó, việc đánh giá kết quả không tính đến chi phí ẩn giấu.

Báo cáo vấn đề 1 Cho một cơ sở dữ liệu giao dịch định lượng D, mục tiêu chính của PPUM là chuyển nó thành cơ sở dữ liệu sạch D', đồng thời giảm thiểu các tác động tiêu cực lên các mẫu hữu ích trong D.

Bảng 3 Cấu trúc bảng GIT [12]

Căn nhà	thời gian	Kích cỡ	Tính thiết thực
I1	T S1	s1	u1
I2	T S2	s2	u2
I3	T S3	s3	u3

4 Các thuật toán đề xuất

Trong phần này, chúng tôi thiết kế một cách tiếp cận mềm cho vấn đề PPUM. Mục tiêu của tất cả các thuật toán PPUM là che giấu các mẫu nhạy cảm đồng thời giảm thiểu tác dụng phụ. Để giải quyết vấn đề này, chúng tôi xây dựng quy trình vệ sinh như một bài toán tối ưu hóa.

4.1 Xây dựng ma trận ràng buộc

Bước đầu tiên liên quan đến việc duy trì mối tương quan giữa HUI và các giao dịch của chúng. Bảng GIT là một cấu trúc hiệu quả có thể phục vụ các nhiệm vụ tiền xử lý song song [12]. Sau khi tiền xử lý cơ sở dữ liệu gốc, chúng ta thu được hai bảng GIT, đó là S-GIT để lưu trữ SHUI và N-GIT cho NSHUI. Bảng 3 trực quan hóa cấu trúc bảng GIT. Ví dụ: các giao dịch có danh tính trong bộ T S1 hỗ trợ HUI I1, có kích thước s1 và tiện ích u1 trong cơ sở dữ liệu.

Bước thứ hai là xây dựng bài toán ẩn. Vì các tiện ích bên ngoài là các giá trị cố định nên để ẩn SHUI, chúng tôi làm xáo trộn cơ sở dữ liệu gốc bằng các tiện ích bên trong nhân tạo. Các tiện ích nội bộ được thay thế phải giảm tiện ích của SHUI xuống dưới ngưỡng tiện ích tối thiểu δ để ẩn SHUI khỏi các thuật toán khai thác. Đặt các tiện ích bên trong đó là các biến và để chúng được biểu thị bằng vectơ v:

$$v = [vkh],\ vkh \in N*,\ Th \in D,\ ik \in I. \tag{8}$$

Giá trị ban đầu của các biến được ký hiệu là o. Hệ số của một biến là tiện ích bên ngoài của hạng mục đại diện cho nó; ví dụ: hệ số của vhk là e(ik). Gọi p là hệ số của các biến trong v.

Ẩn lời giải bài toán là một nhiệm vụ dành cho v. Tất cả các SHUI mà chúng ta cần ẩn đều được lưu trữ trong bảng S-GIT. Bằng cách quét bảng S-GIT, chúng ta có thể định nghĩa v và sắp xếp vị trí của các biến trong cơ sở dữ liệu bằng cấu trúc băm có tên là danh sách biến.

Định nghĩa 14 (danh sách biến) Danh sách biến là một cấu trúc băm bao gồm các hệ số và giá trị ban đầu của các biến. Vị trí của một biến trong cơ sở dữ liệu (mục và giao dịch hỗ trợ nó) đóng vai trò là chỉ mục (Bảng 4).

Sau đó, chúng tôi quét lại S-GIT, xây dựng các ràng buộc để ẩn SHUI và biểu diễn chúng bằng ma trận ràng buộc S của

Bảng 4 Một danh sách biến ví dụ

	hệ số	Giá trị ban đầu
v21	e(i2)	q(i2,T1)
v32	e(i3)	q(i3,T2)
v51	e(i2)	q(i5,T1)

kích thước $|S| \times \|v\|0$. Đặt d là vectơ trong đó tất cả các giá trị đều là ngưỡng tiện ích tối thiểu δ. Để ẩn SHUI, chúng ta có

$$Sv - d \le 0. \tag{9}$$

Việc hạ thấp tiện ích của SHUI cũng ảnh hưởng đến NSHUI. Bảng N-GIT lưu trữ các HUI bị ảnh hưởng bởi quá trình ẩn. Đối với HUI X và tập hợp giao dịch TSX hỗ trợ nó, chúng tôi kết hợp từng mục ik ∈ X với giao dịch Th ∈ TSX . Một cặp mục-giao dịch u kh được tìm thấy trong danh sách biến và có hai trạng thái:

$$uk = \begin{cases} vkh, & \text{Nếu } vkh \in v, \\ q(ik, Th), & \text{nếu không thì.} \end{cases} \tag{10}$$

Sau khi quét N-GIT, chúng ta thu được ma trận ràng buộc N với kích thước $|N| \times \|v\|0$. Thuật toán 1 mô tả chi tiết toàn bộ quá trình thiết lập ma trận ràng buộc (Bảng 5).

4.2 Xây dựng bài toán tối ưu hóa

Giả sử các phần tử của r là tiện ích còn lại của các tập mục trong NSHUI sau khi thay thế biến. Để giữ lại NSHUI, chúng tôi có

$$Nv + r - d \ge 0. \tag{11}$$

Li và cộng sự. [11] đã đề xuất một mô hình quy hoạch tuyến tính số nguyên:

$$\begin{aligned} &\arg\min_{v} \|v\|1 && vkh \in N* \\ &\text{st} && Sv - d \le 0,\ Nv + r - d \ge 0. \end{aligned} \tag{12}$$

Tuy nhiên, mô hình này không phải lúc nào cũng có giải pháp khả thi. Gần đây, Đức và cộng sự. [12] đề xuất nới lỏng:

$$\begin{aligned} &\arg\min_{v} \|v\|1 + \|q\|1 && vkh \in N*, qi \in R \\ &\text{st} && Sv - d \le 0,\ Nv + r - d \ge q. \end{aligned} \tag{13}$$

Các vấn đề thỏa mãn ràng buộc (12) và (13) đã được chứng minh là những phương pháp hiệu quả để giảm thiểu việc ẩn

Bảng 5 Tác dụng phụ của thuật toán heuristic đối với bộ dữ liệu về nấm và siêu thị thực phẩm

Tập dữ liệu	Nhâm nhi(%)	SMRF b(%)	SDIF b(%)	máy ảnh DSLR b(%)	HHUIF b(%)	MSICF b(%)	MSU-MAU b(%)	MSU-MIU b(%)
nấm	0,5	71,0	46,4	62,1	77,2	56,2	76,8	62,9
	0,6	87,7	84,1	89,6	99,5	83,5	99,5	97,5
	0,7	92,1	97,5	94,1	99,9	92,4	99,9	98,8
	0,8	83,9	86,3	90,7	99,7	95,0	99,9	99,6
Siêu thị thực phẩm	0,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	0,6	22,9	24,9	25,7	26.1	26,9	26.1	21.0
	0,7	12.1	3.3	3.3	12.1	2.9	12.1	13,7
	0,8	12.8	14.0	14.0	12.1	7.4	12.1	12.8

Thuật toán 1 Thiết lập ma trận ràng buộc. Đầu vào: bảng N-GIT, S-GIT

Đầu ra: Vectơ biến v, vectơ hệ số p, ma trận ràng buộc S, N, vectơ tiện ích còn lại r 1: Khởi tạo vectơ biến v

2: Khởi tạo ma trận S, N 3: cho itemset, tidset, size, util trong S-GIT do 4: Kết hợp itemset, tidset thành t

5: cho biến vkh = ti trong t
6: do Thêm vkh vào
7: Thêm e(ik) vào

8: end for 9: end for 10: for itemset, tidset, size, util trong S-GIT do 11: Kết hợp itemset, tidset thành t

12: Khởi tạo vectơ t1 với giá trị ‖v‖0 bằng 0
13: cho tôi làm gì
14: nếu vi = ti trong v
15: thì t i1 = t i1 ∈ t1
16: kết thúc nếu
17: kết thúc cho

18: Thêm t1 vào S 19: end for 20: Khởi tạo vector util còn lại r 21: for itemset, tidset, size, util trong N-GIT do 22: Kết hợp itemset, tidset vào t

23: Khởi tạo vectơ t1 với giá trị ‖v‖0 bằng 0
24: cho tôi làm gì
25: nếu vi = ti trong v
26: thì t i1 = t i1 ∈ t1
27: r = util − p[ti] × o[ti]
28: kết thúc nếu
29: kết thúc cho
30: Thêm t1 vào NT

31: Cộng r vào r 32: kết thúc 33: S = Sp; N = N p
34: trả về v, p, r, S, N

chi phí của quá trình vệ sinh. Tuy nhiên, không có hạn chế cụ thể nào đối với thông tin dư thừa. Hơn nữa, chúng không tương thích với các thuật toán tối ưu hóa ngẫu nhiên. Trong [11, 12, 40], tác giả đã sử dụng bộ giải quy hoạch số nguyên để tìm lời giải.

Để giải quyết vấn đề này, chúng tôi đã thiết kế một mô hình mới tương thích với nhiều phương pháp tối ưu hóa. Cho phép min(·) và max(·) là các hàm tối thiểu và tối đa theo phần tử, sao cho

$$m = \text{phút}(a, b) \quad a, b, m \in Rd$$
$$= \begin{bmatrix} \min(a1, b1) \\ \min(a2, b2) \\ \dots \\ \min(an, bn) \end{bmatrix} \quad (14)$$

Và

$$n = \text{tối đa}(a, b) \quad a, b, n \in Rd$$
$$= \begin{bmatrix} \max(a1, b1) \\ \max(a2, b2) \\ \dots \\ \max(an, bn) \end{bmatrix}. \quad (15)$$

Đặt w1 , w2 và w3 là các vectơ trọng số có thể điều chỉnh được xác định theo sở thích của người dùng. Những điều này lần lượt kiểm soát lỗi ẩn, chi phí bị thiếu và chi phí nhân tạo của quá trình ẩn. Cụ thể, mỗi giá trị wi1 trong w1 tương ứng với một SHUI. Về mặt hình thức, chúng tôi có

$$\text{minv tức giận} \quad \|w1 \circ \max(0, Sv - d)\|1 \quad vkh \in N*. \quad (16)$$

Tương tự, đối với chi phí còn thiếu,

$$\text{minv tức giận} \quad \|w2 \circ \max(0, d - (Nv + r)\|1 \quad vkh \in N*. \quad (17)$$

Tìm một giải pháp tối ưu cho PPUM là một bài toán khó. Số lượng tập mục là 2|I|, có nghĩa là có thể có cùng số lượng ràng buộc. Vì vậy, không thể tuyên bố một mô hình hạn chế hoàn toàn các tác dụng phụ. Chúng tôi xác định một cách tiếp cận mềm để hạn chế các mẫu dư thừa:

$$\text{minv tức giận} \quad \|w3 \circ p \circ v - p \circ o\|1 \quad vkh \in N*. \quad (18)$$

Về mặt hình thức, mô hình mềm của chúng tôi có thể được mô tả như sau:

$$\text{minv tức giận} \quad \|w1 \circ \text{tối đa}(0, Sv - d)\|1 + \|w2 \circ \max(0, d - (Nv + r))\|1 + \|w3 \circ (p \circ v - p \circ o)\|1. \quad \text{vkh} \in N* \tag{19}$$

4.3 Thuật toán PPUM-ALO và PPUM-IALO

Sau khi thiết lập mô hình tối ưu, bước tiếp theo là tìm giải pháp tối ưu. Trong bài báo này, chúng tôi đề xuất hai thuật toán: PPUM-ALO và PPUM-IALO. Thuật toán PPUM-ALO áp dụng ALO để tìm lời giải của bài toán tối ưu hóa (19). ALO [16] là một thuật toán tối ưu hóa lấy cảm hứng từ hành vi săn mồi của kiến sư tử. Những sinh vật này nổi tiếng với việc tạo ra các hố hình nón trên đất cát và chờ đợi con mồi rơi xuống đáy. Hình 1 minh họa một cái bẫy hình nón do kiến sư tử tạo ra.

Đầu tiên, chúng tôi mô hình hóa bước đi ngẫu nhiên của kiến khi tìm kiếm thức ăn. Gọi g là số lần lặp tối đa và t là bước đi ngẫu nhiên. Hàm ngẫu nhiên r(·) được định nghĩa như sau:

$$r(\cdot) = \begin{cases} 1, & \text{nếu } x \sim U(0, 1) > 0{,}5, \\ 0, & \text{nếu } x \sim U(0, 1) \le 0{,}5. \end{cases} \tag{20}$$

Cho c(·) là hàm tổng tích lũy, bước đi ngẫu nhiên ở bước t là

$$xt = [\,0, c(2r(t1) - 1), c(2r(t2) - 1), \ldots, c(2r(tg) - 1)\,]. \tag{21}$$

Giả sử bài toán tối ưu hóa có v biến. Vị trí của các con kiến trong ma trận ag × v:

$$A = \begin{bmatrix} a1 \\ a2. \\ = \end{bmatrix} \begin{bmatrix} a1,1 & a1,2 & \cdots \\ a1,v & a2,1 & a2,2 \cdots \\ \cdot & a2,v & \ldots \ldots \ldots \ldots \\ ag,1 & ag,2 & \cdots ag,v \end{bmatrix}. \tag{22}$$

Hình 1 ALO mô phỏng hành vi săn mồi của kiến sư tử bên trong bẫy hình nón

Thứ hai, giả sử rằng kiến sư tử ở đâu đó trong không gian tìm kiếm,

$$L = \begin{bmatrix} l1 \\ l2. \\ = \end{bmatrix} \begin{bmatrix} l1,1\ l1,2 \cdots \\ l1,v\ l2,1\ l2,2 \cdots \\ \cdot\ l2,v \ldots \ldots \ldots \ldots \\ lg,1\ lg,2 \cdots \\ lg,v \end{bmatrix} \quad (23)$$

Để hình thành các giới hạn của bài toán tối ưu hóa, chúng tôi giới hạn các bước đi ngẫu nhiên bên trong không gian tìm kiếm bằng cách chuẩn hóa x cho mỗi biến j:

$$\hat{x}\, j = \frac{xj}{\|xj\|}. \quad (24)$$

Để mô phỏng hành vi săn mồi của kiến sư tử, chúng ta định nghĩa một siêu cầu bằng hai vectơ:

$$c = \text{phút}(x) \quad (25)$$

$$d = \text{tối đa}(x). \quad (26)$$

Việc săn kiến sư tử được mô hình hóa bằng phương pháp chọn bánh xe roulette. Khi con mồi vào bẫy, kiến sư tử ném cát từ tâm hố. Điều này có nghĩa là bán kính siêu cầu được giảm thích ứng ở bước t theo tỷ lệ r = 10w tg . Giá trị của w thay đổi theo thời gian như đề xuất trong [16]:

$$c = \frac{1r}{c} \quad (27)$$

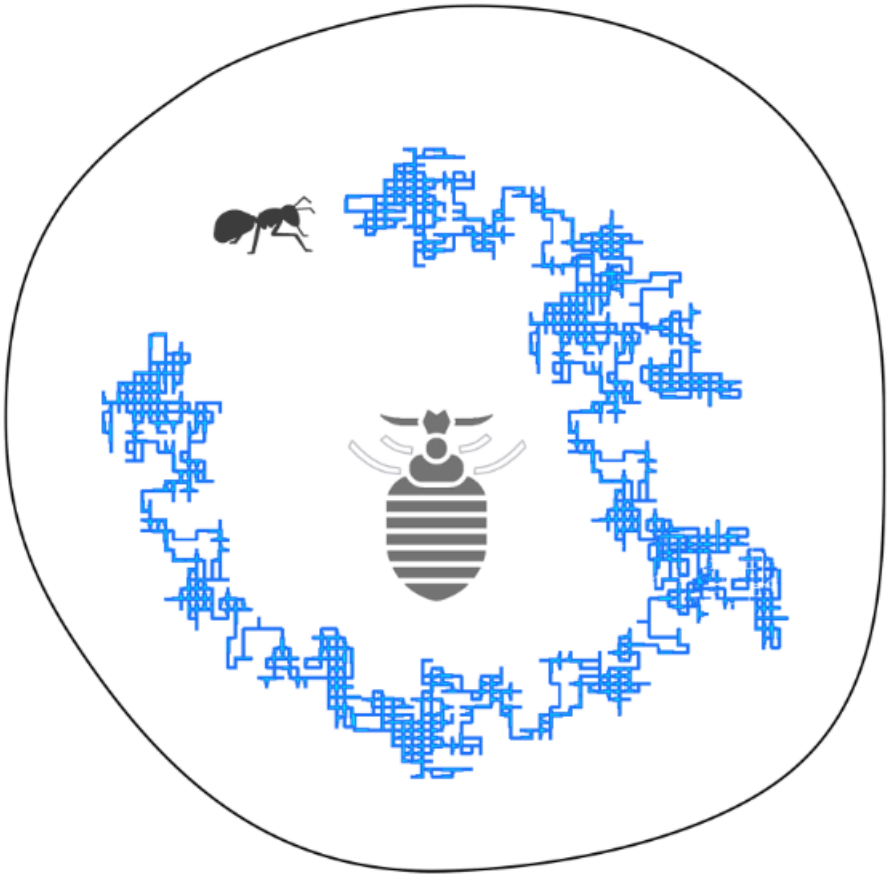
$$d = \frac{1r}{c.} \quad (28)$$

Hình 2 cho thấy bước đi ngẫu nhiên của một con kiến trong bẫy kiến sư tử. Khi sư tử bắt được một con kiến, nó sẽ di chuyển đến vị trí của nó để tiêu thụ. Con sư tử kiến đưa ra giải pháp tốt nhất là con ưu tú. Tinh hoa sẽ ảnh hưởng đến chuyển động của tất cả các loài kiến. Cho Rl là những bước đi ngẫu nhiên xung quanh con kiến, Re bao gồm những bước đi ngẫu nhiên xung quanh đàn kiến:

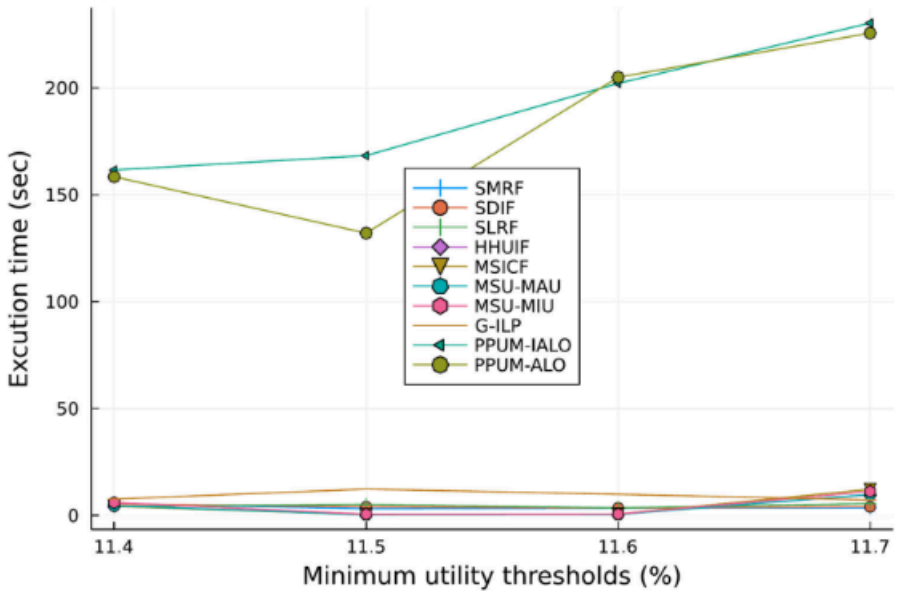
$$A = Rl + Re\ 2 \quad (29)$$

Lời giải của bài toán tối ưu hóa là vị trí của người ưu tú sau khi thuật toán ALO được thực thi. Các giá trị biến được sử dụng để thay thế các tiện ích nội bộ trong cơ sở dữ liệu gốc để có được tiện ích đã được làm sạch.

Hình 2 Bước đi ngẫu nhiên của một con kiến trong bẫy



Hình 3 Sự biến đổi của thời gian chạy thuật toán với ngưỡng tiện ích tối thiểu trên tập dữ liệu nấm



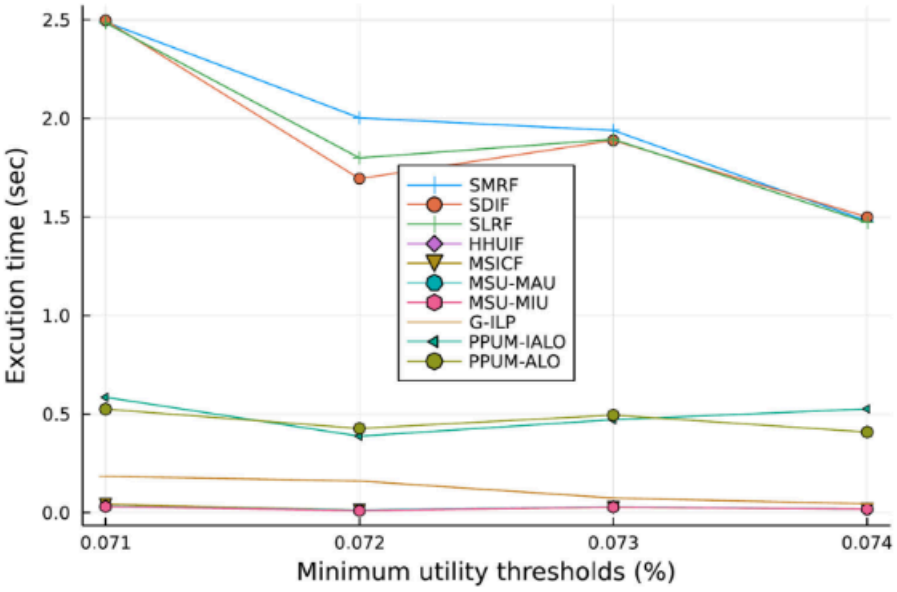
Chúng tôi kết hợp phương pháp chính xác của bộ giải toán với phương pháp ngẫu nhiên của ALO để tạo thành thuật toán lai. Đầu tiên, chúng tôi giải mô hình tối ưu hóa bằng phương pháp tối ưu hóa toán học, cụ thể là bộ tối ưu hóa Gurobi [48]. Giải pháp này được sử dụng để khởi tạo con sư tử kiến ưu tú vì nó sẽ ảnh hưởng đến chuyển động của tất cả các con kiến trong quá trình lặp lại. Chúng tôi thực hiện ALO chỉ trong 20 lần lặp. Thuật toán này được đặt tên là PPUM-IALO.

5 thí nghiệm

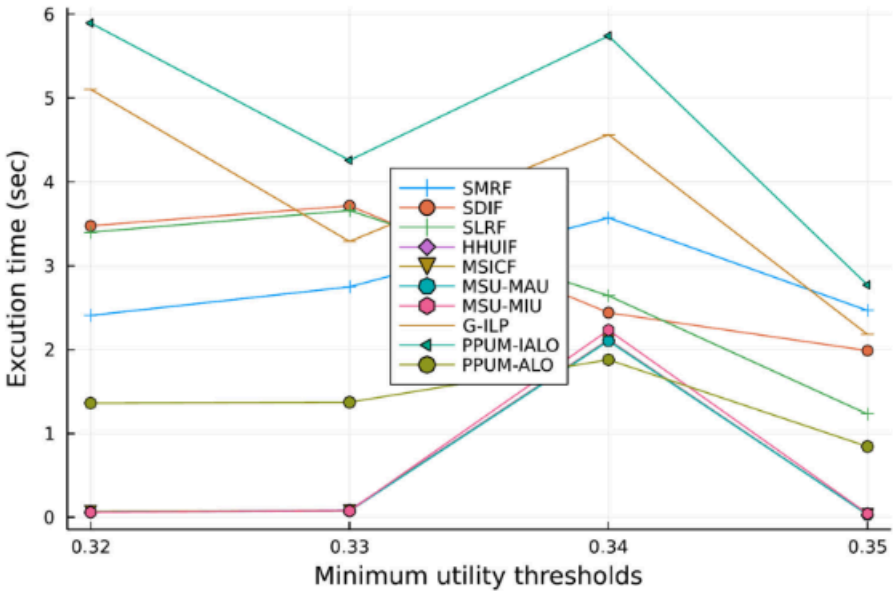
Chúng tôi đã tiến hành thử nghiệm trên CPU Intel Core i7-6700 với RAM 32 GB và GPU NVIDIA GeForce RTX2080. Các thuật toán được thực hiện trên bốn bộ dữ liệu điển hình. Mật độ của từng tập dữ liệu được tính bằng công thức sau:

Mật độ = AvgLen|I|

Hình 4 Sự biến đổi của thời gian chạy thuật toán với ngưỡng tiện ích tối thiểu trên tập dữ liệu foodmart



Hình 5 Sự biến đổi của thời gian chạy thuật toán với ngưỡng tiện ích tối thiểu trên tập dữ liệu t25i10d10k



Các thuộc tính của các tập dữ liệu thử nghiệm được mô tả trong Bảng 9. Tiện ích bên ngoài của từng mục được rút ra từ phân phối log-chuẩn bị cắt cụt thành khoảng R ∩ [1, 10]. Tiện ích bên trong của các mục trong giao dịch được lấy là số nguyên được lấy mẫu từ U (1, 10). Theo khảo sát của Zhang et al. [49], thuật toán d2HUP [50] được ưu tiên làm kỹ thuật khai thác. Hiệu suất của thuật toán đề xuất đã được đánh giá và so sánh với các thuật toán khác trên các bộ dữ liệu khác nhau về thời gian chạy, lỗi ẩn (α), chi phí bị thiếu (β) và chi phí nhân tạo (γ). Một số thuật toán heuristic, HHUIF, MSICF [6], MSU-MAU, MSU-MIU [7], SMRF, SLRF và SDIF [9] và phương pháp lập trình số nguyên mới nhất, GILP [12], đã được chọn để so sánh với các thuật toán được đề xuất. Bài toán lập trình số nguyên đã được giải bằng bộ giải Gurobi [48]. Tất cả các thuật toán được triển khai trong Julia phiên bản 1.6.6, một ngôn ngữ hiệu năng cao dành cho máy tính khoa học. Chúng tôi đã tiến hành thử nghiệm theo các ngưỡng tiện ích tối thiểu khác nhau và số lượng mẫu nhạy cảm (NSP) khác nhau và tỷ lệ phần trăm thông tin nhạy cảm (SIP). Mỗi thước đo đánh giá thuật toán được tính trung bình trong ba lần chạy.

Cả hai thuật toán PPUM-ALO và PPUM-IALO đều được sử dụng với 10 con kiến và bị dừng sớm sau 20 lần lặp. Để kiểm soát các tác dụng phụ, vectơ w1 được lấp đầy vô cực dương để loại bỏ hoàn toàn lỗi ẩn. Vì việc đưa ra thông tin sai lệch có thể gây ra vấn đề nghiêm trọng nên chúng tôi đặt w2 là một vectơ toàn một và đặt tất cả các giá trị của vectơ w3 thành 20 để giảm thiểu chi phí nhân tạo. Số lần lặp tối đa của PPUM-ALO và PPUM-IALO lần lượt được đặt thành 100 và 20.

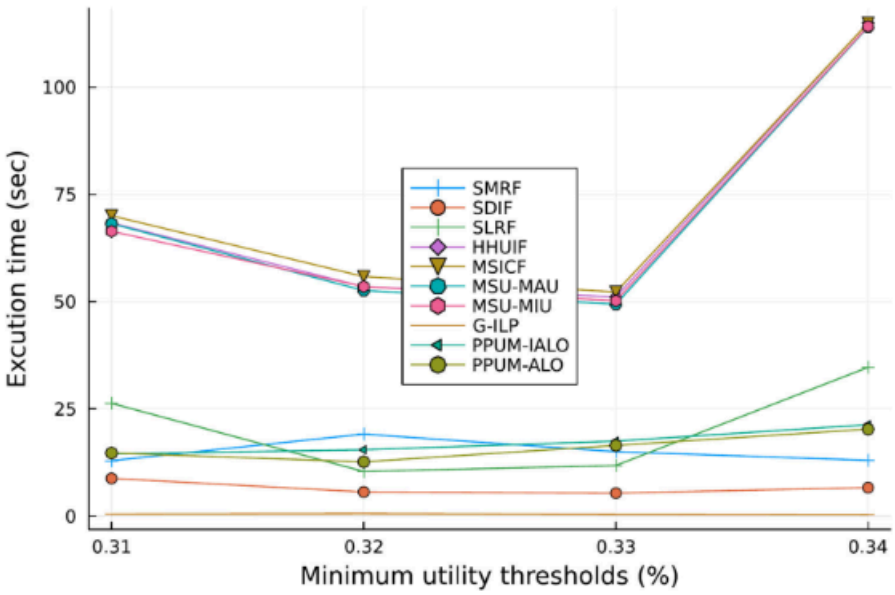
5.1 Thời gian chạy

Như được hiển thị trong Hình 3, các phương pháp ngẫu nhiên yêu cầu thời gian thực hiện cao trên tập dữ liệu nắm, nhỏ và dày đặc. Một tập dữ liệu mật độ cao có xu hướng có nhiều mục được chia sẻ giữa NSHUI và SHUI, tạo ra nhiều xung đột hơn giữa các mục tiêu giảm chi phí còn thiếu và ẩn

Bảng 6 Tác dụng phụ của thuật toán heuristic trên bộ dữ liệu t25i10d10k và t20i6d100k

Tập dữ liệu	NSP	SMRF b(%)	SDIF b(%)	máy ảnh DSLR b(%)	HHUIF b(%)	MSICF b(%)	MSU-MAU b(%)	MSU-MIU b(%)
t25i10d10k	1	50,6	50,6	68,3	68,9	67,3	74,3	56,7
	2	48,8	51,3	44,2	38,2	39,0	43,1	37,4
	3	41.1	41,0	71,4	59,0	58,9	51,6	40,2
	4	66,4	53,0	56,2	47,3	47,3	56,1	40,7
t20i6d100k	2	9,3	10,4	10,4	9,3	9,3	8,8	7,7
	3	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	4	8.3	7,8	7,8	7,8	5,6	8,9	6,7
	5	12.8	11.2	10.6	12.8	11.2	12.3	10.1

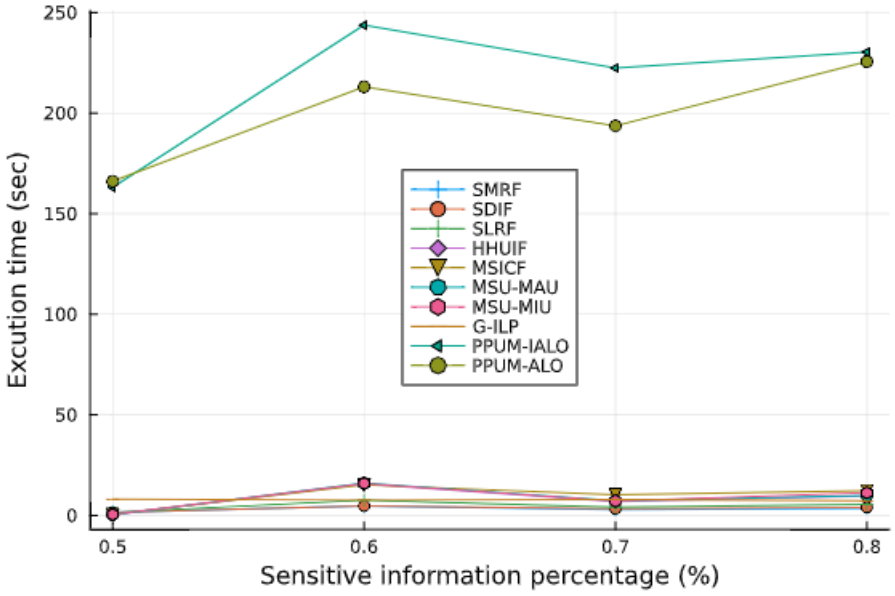
Hình 6 Sự biến đổi của thời gian chạy thuật toán với ngưỡng tiện ích tối thiểu trên tập dữ liệu t20i6d100k



nhập dữ liệu nhạy cảm. Điều này làm cho vấn đề ẩn được xây dựng trở nên khó giải quyết hơn. Các thuật toán heuristic ẩn các mẫu nhạy cảm bằng cách lặp đi lặp lại việc loại bỏ hoặc giảm bớt các tiện ích mục tiêu bên trong, mang lại lợi thế cho chúng trên các tập dữ liệu nhỏ. Tuy nhiên, thuật toán G-ILP hoạt động tốt trên nấm. Vì ngưỡng tiện ích tối thiểu được đặt thành 11,7% trong tổng số tiện ích của tập dữ liệu nên chúng tôi đã tăng tỷ lệ phần trăm thông tin nhạy cảm (Hình 7). Chúng ta có thể quan sát thấy các thuật toán được đề xuất gặp khó khăn trong các tập dữ liệu dày đặc. Bởi vì số lượng biến trong một tập dữ liệu dày đặc có thể lớn nên việc tính toán mức độ phù hợp chức năng cần thiết cho toàn bộ dân số là một công việc tốn nhiều thời gian.

Như thể hiện trong hình. Như trong hình 4 và 8, tất cả các thuật toán đều có thời gian chạy thấp trên tập dữ liệu foodmart, nhưng các thuật toán SMRF, SDIF và SLRF dường như chạy chậm hơn một chút. Kết quả này được mong đợi vì các thuật toán quét tập dữ liệu và SHUI để xây dựng cấu trúc dữ liệu bảo toàn trọng số giao dịch và giá trị tiện ích tập mục thực (RISU) trước khi che giấu SHUI. Với ngưỡng tiện ích tối thiểu cố định, việc ẩn các mẫu nhạy cảm hơn đòi hỏi thời gian thực hiện thuật toán cao hơn.

Hình 7 Sự thay đổi thời gian chạy thuật toán với số lượng mẫu nhạy cảm trên tập dữ liệu nấm



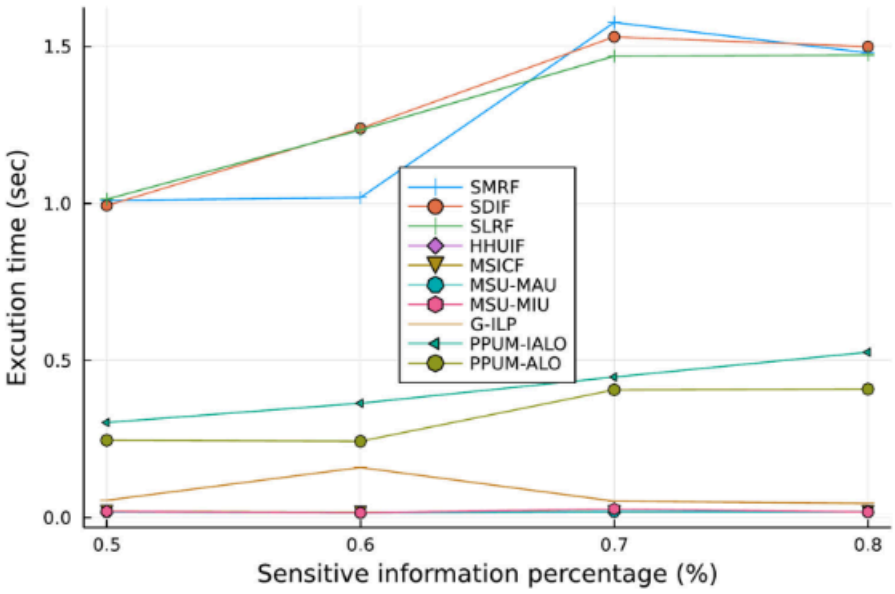
Bảng 7 Tác dụng phụ của các thuật toán dựa trên tối ưu hóa đối với bộ dữ liệu về nấm và siêu thị thực phẩm

Tập dữ liệu	Nhâm nhí(%)	G-ILP		PPUM-IALO		PPUM-ALO	
		b(%)	c (%)	b(%)	c (%)	b(%)	c (%)
nấm	0,5	4.7	472,3	4.7	472,3	không có	không có
	0,6	9,6	466,5	9,6	466,5	không có	không có
	0,7	24,4	636,4	24,4	636,4	không có	không có
	0,8	5,9	112,5	5,9	112,5	không có	không có
Siêu thị thực phẩm	0,5	0,0	0,0	0,0	0,0	0,0	0,0
	0,6	5,5	7,8	27,9	0,0	27,9	0,0
	0,7	0,0	399,3	25,2	0,0	25,3	0,0
	0,8	0,8	321,9	24,7	0,0	24,7	0,0

Bảng 8 Tác dụng phụ của các thuật toán dựa trên tối ưu hóa trên bộ dữ liệu t25i10d10k và t20i6d100k

Tập dữ liệu	NSP	G-ILP		PPUM-IALO		PPUM-ALO	
		b(%)	c (%)	b(%)	c (%)	b(%)	c (%)
t25i10d10k	1	31,6	8.1	31,6	8.1	95,2	0,0
	2	1.3	81,7	1.3	81,7	95,2	0,0
	3	10.9	21.2	10.9	21.2	95,2	0,0
	4	không có	không có	95,2	0,0	95,2	0,0
t20i6d100k	2	0,0	9,9	0,0	9,9	23,6	0,0
	3	0,0	0,0	0,0	0,0	0,0	0,0
	4	0,6	23,9	11.7	0,0	11.7	0,0
	5	0,0	12.3	0,0	12.3	24.0	0,0

Hình 8 Sự biến đổi của thời gian chạy thuật toán với số lượng mẫu nhạy cảm trên tập dữ liệu foodmart



Bảng 9 Bộ dữ liệu và thuộc tính của chúng

Tập dữ liệu	|D|	|Tôi|	Tỉ trọng(%)
nấm	8124	119	19.3
Siêu thị thực phẩm	4141	1559	0,25
t25i10d10k	9976	893	2,66
t20i6d100k	99922	929	2,22

Các thuật toán có xu hướng chạy chậm hơn trên các tập dữ liệu lớn. Trong Hình 5, các phương pháp tiếp cận dựa trên ILP, tức là G-ILP và PPUM-IALO, có thời gian thực hiện tổng thể cao. Phương pháp ngẫu nhiên PPUM-ALO có kết quả hợp lý. Trong Hình 9, có một trường hợp trong đó bài toán ILP khó giải hơn về mặt toán học. Điều này gây ra thời gian chạy cao bất ngờ với các phương pháp dựa trên ILP (Bảng 6).

Nhược điểm của thuật toán heuristic là rõ ràng trong các tập dữ liệu lớn và thưa thớt (Hình 6). Thời gian chạy của chúng rất cao trên tập dữ liệu t20i6d100k. Chúng ta cũng có thể thấy rằng thời gian thực hiện của một thuật toán bị ảnh hưởng bởi các mẫu nhạy cảm đã chọn. Bởi vì các thuật toán heuristic chỉ sửa đổi một tiện ích nội bộ trong một lần lặp nên SHUI xuất hiện trong nhiều giao dịch cần quét cơ sở dữ liệu nhiều hơn để ẩn. Như có thể được quan sát trong hình. Trong Hình 10 và 6, các thuật toán dựa trên các bài toán tối ưu hóa hoạt động hiệu quả hơn trên các tập dữ liệu lớn và thưa thớt.

5.2 Tác dụng phụ

Chi phí còn thiếu và nhân tạo của các thuật toán trên bộ dữ liệu về nấm và siêu thị thực phẩm được trình bày trong Bảng 5 và 7. Trên tập dữ liệu về nấm, thuật toán PPUM-ALO được đề xuất đã không thể ẩn SHUI và thuật toán PPUM-IALO đạt được kết quả tương tự tới G-ILP. Điều này có nghĩa là nếu không có một người ưu tú làm người hướng dẫn, ALO không thể thoát khỏi mức tối thiểu cục bộ của vấn đề lẩn trốn. PPUM-IALO được khởi tạo bằng giải pháp G-ILP và thực hiện ít bước hơn để tìm ra giải pháp phù hợp hơn cho các yêu cầu cụ thể tương ứng với siêu tham số do người dùng xác định. Hành vi này có thể

Bảng 11 TMR của các thuật toán dựa trên tối ưu hóa trên bộ dữ liệu về nấm và siêu thị thực phẩm

Tập dữ liệu	Nhầm nhi(%)	G-ILP	PPUM-IALO	PPUM-ALO
nấm	0,5	20,5	20,5	0,0
	0,6	67,3	67,3	0,0
	0,7	51,7	51,7	0,0
	0,8	49,1	49,1	0,0
Siêu thị thực phẩm	0,5	1.1	1.1	1.2
	0,6	0,5	0,5	0,5
	0,7	1.0	1.0	1.1
	0,8	1,5	1,5	81,1

được cho là do tác dụng phụ của thuật toán trên tập dữ liệu foodmart (Hình 7).

Đối với các tập dữ liệu lớn (xem Bảng 6 và 8), các thuật toán dựa trên tối ưu hóa thường hoạt động tốt hơn với sự cân bằng về chi phí nhân tạo. Tuy nhiên, do những hạn chế chặt chẽ của G-ILP, đã có trường hợp nó không thể giải quyết đúng vấn đề ẩn giấu. Ngoài ra, PPUM-IALO được đề xuất thu được kết quả phù hợp hơn G-ILP trên tập dữ liệu t20i6d100k (Hình 8).

5.3 Tỷ lệ sửa đổi giao dịch

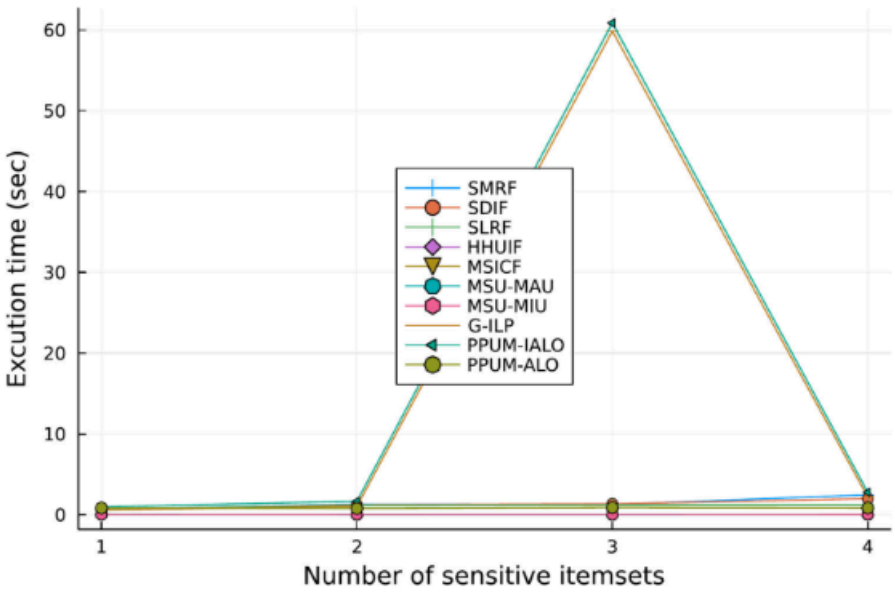
Để đo lường tác động của quá trình ẩn, chúng tôi đã sử dụng tỷ lệ sửa đổi giao dịch (TMR) để xác định tỷ lệ phần trăm giao dịch được sửa đổi cho từng tập dữ liệu thử nghiệm. TMR thấp hơn đối với các phương pháp heuristic, như được chỉ ra trong Bảng 9, 10 và 11. Có một trường hợp không lường trước được trong bộ dữ liệu siêu thị thực phẩm khiến các phương pháp này có TMR cao bất ngờ. Trong tập dữ liệu foodmart, thuật toán PPUM-ALO cũng có một giải pháp nhiễu loạn ảnh hưởng đến 80% giao dịch của nó. Thuật toán PPUM-ALO không tìm được giải pháp ẩn cho tập dữ liệu nấm, dẫn đến TMR bằng 0 (Hình 9).

Bảng 12 và 13 thể hiện TMR của thuật toán trên các tập dữ liệu lớn. Trong Bảng 12, có thể thấy rằng ngay cả với TMR nhỏ hơn, các thuật toán heuristic vẫn có hiệu suất thực thi cao hơn.

Bảng 10 TMR của các thuật toán heuristic trên bộ dữ liệu về nấm và siêu thị thực phẩm

Tập dữ liệu	Nhầm nhi(%)	SMRF	SDIF	máy ảnh DSLR	HHUIF	MSICF	MSU-MAU	MSU-MIU
nấm	0,5	3.2	1.7	1.7	1.6	1.9	1.3	1,5
	0,6	67,3	67,3	67,3	67,3	67,3	67,3	67,3
	0,7	29,2	29,2	29,3	29,2	29,2	29,2	29,6
	0,8	40,8	40,8	40,8	40,8	41.1	40,8	40,8
Siêu thị thực phẩm	0,5	0,5	0,5	0,5	0,4	0,4	0,4	0,4
	0,6	46,4	46,4	46,4	46,3	46,3	46,3	46,3
	0,7	0,5	0,5	0,5	0,4	0,4	0,4	0,4
	0,8	0,6	0,5	0,6	0,4	0,4	0,4	0,4

Hình 9 Sự biến đổi của thời gian chạy thuật toán với số lượng mẫu nhạy cảm trên tập dữ liệu t25i10d10k



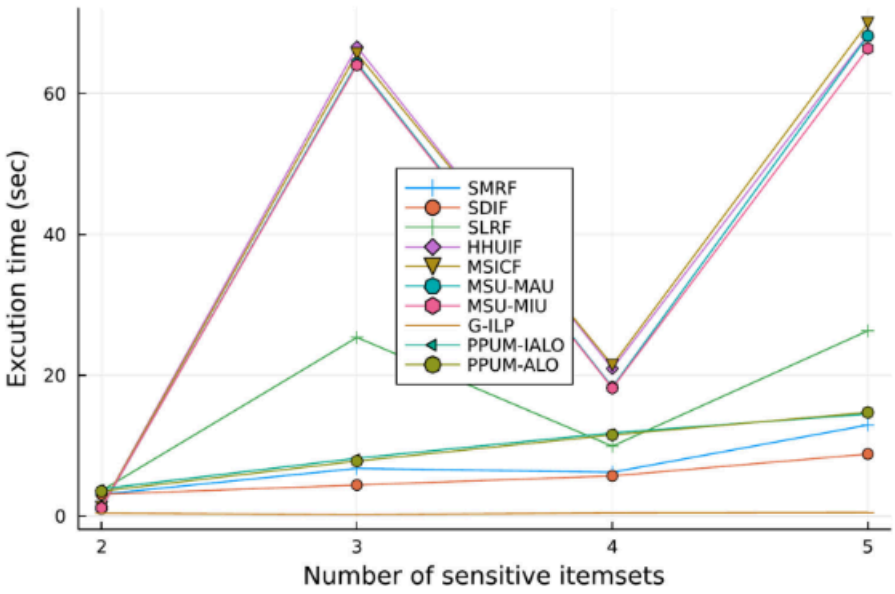
Bảng 12 TMR của thuật toán heuristic trên bộ dữ liệu t25i10d10k và t20i6d100k

Tập dữ liệu	NSP	SMRF	SDIF	máy ảnh DSLR	HHUIF	MSICF	MSU-MAU	MSU-MIU
t25i10d10k	1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
	2	0,1	0,1	0,1	0,1	0,1	0,1	0,1
	3	0,1	0,1	0,2	0,1	0,1	0,1	0,1
	4	0,2	0,1	0,1	0,1	0,1	0,1	0,1
t20i6d100k	2	0,0	0,1	0,1	0,0	0,0	0,0	0,0
	3	3.1	0,5	3.1	1.9	1.9	1.9	1.9
	4	0,8	0,8	0,8	0,6	0,6	0,6	0,6
	5	3.3	3.3	3.3	2.0	2.0	2.0	2.0

Bảng 13 TMR của các thuật toán dựa trên tối ưu hóa trên bộ dữ liệu t25i10d10k và t20i6d100k

Tập dữ liệu	NSP	G-ILP	PPUM-IALO	PPUM-ALO
t25i10d10k	1	0,9	0,9	0,9
	2	0,9	0,9	0,9
	3	1.0	1.0	1.0
	4	0,0	1.0	1.0
t20i6d100k	2	0,6	0,6	0,6
	3	12.3	12.3	12.3
	4	8,4	8,4	8,4
	5	17,4	17,4	17,4

Hình 10 Sự biến đổi của thời gian chạy thuật toán với số lượng mẫu nhạy cảm trên tập dữ liệu t20i6d100k



thời gian cắt hơn các thuật toán dựa trên tối ưu hóa. Tác dụng phụ của các thuật toán dựa trên tối ưu hóa có thể khác nhau, nhưng TMR của chúng vẫn giữ nguyên (Hình 10).

6 Kết luận

Với mối quan tâm ngày càng tăng về quyền riêng tư, các nhà nghiên cứu đang tập trung vào các vi phạm quyền riêng tư tiềm ẩn do khai thác tập mục tiện ích cao (HUIM). Trong nghiên cứu này, chúng tôi đề xuất hai thuật toán PPUM-ALO và PPUM-IALO cho PPUM. Để xây dựng các bài toán ẩn, chúng tôi thiết kế một chiến lược thiết lập các ma trận ràng buộc và giới thiệu một mô hình ẩn mới. Tối ưu hóa ngẫu nhiên là phương pháp chúng tôi sử dụng để tìm giải pháp ẩn. Các thuật toán của chúng tôi cho thấy hiệu suất ổn định trong thời gian chạy và các tác dụng phụ thông qua phân tích và so sánh thử nghiệm sâu rộng. Tuy nhiên, việc giấu đi những thông tin nhạy cảm là một bài toán NP-khó với nhiều trường hợp không mong muốn. Vì vậy, điều quan trọng là phải thử nghiệm các thuật toán tối ưu hóa khác nhau. Hơn nữa, phương pháp tối ưu hóa không cho phép xóa các mục khỏi giao dịch. Có những trường hợp việc giảm tiện ích bên trong không đủ để ẩn các tập mục. Việc phát triển một thuật toán nhiễu loạn cho phép loại bỏ các mục khỏi giao dịch là cần thiết cho nghiên cứu trong tương lai.

Lời cảm ơn Nghiên cứu này được tài trợ bởi Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM theo số tài trợ CNTT 2024-01.

Đóng góp của tác giả Đức Nguyễn: Khái niệm hóa, Phương pháp luận, Xác nhận, Phần mềm, Viết—bản thảo gốc. Bắc Lê: Khái niệm hóa, Phương pháp luận, Xác nhận, Giám sát, Đánh giá, Tuyên bố.

Tính sẵn có của dữ liệu Các bộ dữ liệu được tạo và/hoặc phân tích trong nghiên cứu hiện tại có sẵn trong kho GitHub, https://github.com/4ree/ILP.

Mã sẵn có Mã để xử lý trước và phân tích được cung cấp trong kho GitHub, https://github.com/4ree/ILP.

Tuyên bố

Các lợi ích cạnh tranh Các tác giả tuyên bố rằng họ không có lợi ích tài chính hoặc mối quan hệ cá nhân cạnh tranh nào có thể ảnh hưởng đến công việc được báo cáo trong bài viết này.

Sự đồng ý có hiểu biết Sự đồng ý có hiểu biết được lấy từ tất cả các đối tượng tham gia vào nghiên cứu này.

Tài liệu tham khảo

1. Gheisari M, Hamidpour H, Liu Y, Saedi P, Raza A, Jalili A, Rokhsati H, Amin R (2022) Kỹ thuật khai thác dữ liệu để khai thác web: một cuộc khảo sát. Ứng dụng Artif Intell 1(1):3–10. https://doi.org/10.47852/bonviewAIA2202290

2. Wu JM-T, Gautam S, Jolfaei A, Fournier-Viger P, Lin JC-W (2021) Ẩn thông tin nhạy cảm trong bộ dữ liệu sức khỏe điện tử. Hệ thống máy tính thế hệ tương lai 117:169–180. https://doi.org/10.1016/j.future.2020.11. 026

3. Zhang C, Xie Y, Bai H, Yu B, Li W, Gao Y (2021) Khảo sát về học tập liên kết. Hệ thống dựa trên kiến thức 216:106775. https://doi.org/10.1016/j.knosys.2021.106775

4. Liu Y, Kang Y, Xing C, Chen T, Yang Q (2020) Khung học tập chuyển tiếp liên kết an toàn. IEEE Intell Syst 35(4):70–82. https://doi.org/10.1109/MIS.2020.2988525

5. Yun U, Kim D (2017) Phân tích các phương pháp bảo vệ quyền riêng tư trong khai thác mô hình tiện ích cao. Trong: Park JJJH, Pan Y, Yi G, Loia V (eds.) Những tiến bộ trong khoa học máy tính và máy tính phổ biến-

ing, Singapore, trang 883–887. https://doi.org/10.1007/978-981-10- 3023-9_137

6. Yeh JS, Hsu PC (2010) HHUIF và MSICF: các thuật toán mới để khai thác tiện ích bảo đảm quyền riêng tư. Ứng dụng Hệ thống Chuyên gia 37(7):4779– 4786. https://doi.org/10.1016/j.eswa.2009.12.038

7. Lin JC-W, Wu TY, Fournier-Viger P, Lin G, Zhan J, Voznak M (2016) Thuật toán nhanh để ẩn các tập mục hữu ích cao nhạy cảm trong khai thác tiện ích bảo vệ quyền riêng tư. Eng Appl Artif Intell 55:269– 284. https://doi.org/10.1016/j.engappai.2016.07.003

8. Jangra S, Toshniwal D (2022) Các thuật toán hiệu quả để lựa chọn mục nạn nhân trong khai thác tiện ích bảo vệ quyền riêng tư. Hệ thống máy tính thế hệ tương lai 128:219–234. https://doi.org/10.1016/j.future.2021.10.008 9. Ashraf M, Rady S, Abdelkader T, Gharib TF (2023) Hiệu quả

các thuật toán bảo vệ quyền riêng tư để ẩn các bộ vật phẩm có tính tiện ích cao nhạy cảm. Máy tính & Bảo mật 132:103360. https://doi.org/10.1016/ j.cose.2023.103360

10. Lin JC-W, Hong TP, Fournier-Viger P, Liu Q, Wong JW, Zhan J (2017) Che giấu hiệu quả các tập vật phẩm bí mật có tiện ích cao với tác dụng phụ tối thiểu. Tạp chí Trí tuệ nhân tạo thực nghiệm và lý thuyết. 29(6):1225–1245. https://doi.org/10.1080/ 0952813X.2017.1328462

11. Li S, Mu N, Le J, Liao X (2019) Một thuật toán mới để bảo vệ quyền riêng tư khai thác tiện ích dựa trên lập trình tuyến tính số nguyên. Eng Appl Artif Intell 81:300–312. https://doi.org/10.1016/j.engappai. 2018.12.006

12. Nguyễn D, Trần MT, Lê B (2023) Một thuật toán mới sử dụng sự nới lỏng lập trình số nguyên để đảm bảo tính riêng tư trong khai thác tiện ích. Appl Intell. https://doi.org/10.1007/s10489-023-04913-w

13. Karimi-Mamaghan M, Mohammadi M, Meyer P, Karimi- Mamaghan AM, Talbi EG (2022) Học máy phục vụ siêu chẩn đoán để giải quyết các vấn đề tối ưu hóa tổ hợp: một công nghệ tiên tiến. Eur J Oper Res 296(2):393–422. https:// doi.org/10.1016/j.ejor.2021.04.032

14. Bao X, Kang H, Li H (2024) Trình tối ưu hóa rắn nhị phân cải tiến với chức năng truyền đột biến gaussian và khoảng cách hamming để lựa chọn tính năng. Ứng dụng điện toán thần kinh 36(16):9567–9589. https:// doi.org/10.1007/s00521-024-09581-6

15. Kovaˇcevi´c A, Luburi´c N, Slivka J, Proki´c S, Grují´c KG, Vidakvi´c D, Sladi´c G (2024) Tự động phát hiện mùi mã bằng cách sử dụng số liệu và phần nhúng codet5: một nghiên cứu điển hình trong c#. Ứng dụng máy tính thần kinh 36(16):9203–9220. https://doi.org/10.1007/s00521-024- 09551-y

16. Mirjalili S (2015) Trình tối ưu hóa kiến. Adv Eng Softw 83:80–98.

https://doi.org/10.1016/j.advengsoft.2015.01.010 17. Braik MS (2021) Thuật toán đàn tắc kè hoa: một phương pháp tối ưu lấy cảm hứng từ sinh học

Mizer để giải quyết các vấn đề thiết kế kỹ thuật. Ứng dụng hệ thống chuyên gia 174:114685. https://doi.org/10.1016/j.eswa.2021.114685

18. Dhiman G, Kumar V (2017) Công cụ tối ưu hóa linh cầu đốm: một kỹ thuật siêu dữ liệu dựa trên cảm hứng sinh học mới cho các ứng dụng kỹ thuật. Adv Eng Softw 114:48–70. https://doi.org/10.1016/j. advengsoft.2017.05.014

19. Hatamlou A (2013) Lỗ đen: một phương pháp tối ưu hóa heuristic mới cho phân cụm dữ liệu. Inf Khoa học 222: 175–184. https://doi.org/10.1016/j.ins.2012.08.023 . Bao gồm Phần đặc biệt về các xu hướng mới trong hệ thống trí tuệ xung quanh và lấy cảm hứng từ sinh học

20. Ramezani F, Lotfi S (2013) Thuật toán dựa trên xã hội (sba). Máy tính mềm Appl 13(5):2837–2856. https://doi.org/10.1016/j.asoc. 05/05/2012

21. Mohammadi-Balani A, Dehghan Nayeri M, Azar A, Taghizadeh- Yazdi M (2021) Trình tối ưu hóa đại bàng vàng: một thuật toán siêu hình lấy cảm hứng từ thiên nhiên. Máy tính & Kỹ thuật công nghiệp. 152:107050. https://doi.org/10.1016/j.cie.2020.107050

22. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mir-jalili SM (2017) Thuật toán bầy Salp: một công cụ tối ưu hóa lấy cảm hứng từ sinh học cho các vấn đề thiết kế kỹ thuật. Adv Eng Softw 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

23. Kaur A, Jain S, Goel S (2019) Thuật toán tối ưu hóa Sandpiper: một cách tiếp cận mới để giải quyết các vấn đề kỹ thuật trong đời thực. Ứng dụng Intell 50(2):582–619. https://doi.org/10.1007/s10489-019-01507- 3

24. Afshari MH, Dehkordi MN, Akbari M (2016) Ẩn luật kết hợp bằng thuật toán tối ưu hóa chim cu. Ứng dụng Hệ thống Chuyên gia 64:340–351. https://doi.org/10.1016/j.eswa.2016.08.005

25. Yao H, Hamilton HJ, Butz CJ (2004) Một cách tiếp cận nền tảng để khai thác các tiện ích tập mục từ cơ sở dữ liệu. Trong: Kỷ yếu hội nghị quốc tế SIAM 2004 về khai thác dữ liệu, trang 482–486. https://doi.org/10.1137/1.9781611972740.51

26. Liu Y, Liao Wk, Choudhary A (2005) Thuật toán hai giai đoạn để phát hiện nhanh các tập mục hữu ích cao. Trong: Ho TB, Cheung D, Liu H (eds) Những tiến bộ trong khám phá kiến thức và khai thác dữ liệu, trang 689–695. Springer, Berlin, Heidelberg. https://doi.org/10.1007/ 11430919_79

27. Lin CW, Hong TP, Lu WH (2011) Cấu trúc cây hiệu quả để khai thác các tập mục hữu ích cao. Ứng dụng Hệ thống Chuyên gia 38(6):7419–7424. https://doi.org/10.1016/j.eswa.2010.12.082

28. Tseng VS, Wu CW, Shie BE, Yu PS (2010) Up-growth: một thuật toán hiệu quả để khai thác tập mục có tính tiện ích cao. Trong: Kỷ yếu hội nghị quốc tế ACM SIGKDD lần thứ 16 về khám phá tri thức và khai thác dữ liệu. KDD '10, trang 253–262. Hiệp hội Máy tính, New York, Hoa Kỳ. https://doi.org/10.1145/ 1835804.1835839

29. Tseng VS, Shie B, Wu C, Yu PS (2013) Thuật toán hiệu quả để khai thác các tập mục hữu ích cao từ cơ sở dữ liệu giao dịch. IEEE Trans Knowl Data Eng 25(8):1772–1786. https://doi.org/10.1109/ TKDE.2012.59

30. Liu M, Qu J (2012) Khai thác các tập mục hữu ích cao mà không cần tạo ứng cử viên. Trong: Kỷ yếu hội nghị quốc tế ACM lần thứ 21 về quản lý thông tin và tri thức. CIKM '12, trang 55–64, New York, Hoa Kỳ. https://doi.org/10.1145/2396761. 2396773

31. Zida S, Fournier Viger P, Lin CW, Wu CW, Tseng V (2016) EFIM: một thuật toán nhanh và hiệu quả về bộ nhớ để khai thác tập mục có tiện ích cao. Hệ thống thông tin kiến thức 51:595–625. https://doi.org/10.1007/s10115-016-0986-0

32. Kim H, Yun U, Baek Y, Kim H, Nam H, Lin JC-W, Fournier-Viger P (2021) Khai thác mô hình định hướng tiện ích dựa trên trượt giảm chắn trên dữ liệu luồng. Hệ thống dựa trên kiến thức 213:106653. https://doi.org/ 10.1016/j.knosys.2020.106653

33. Gan W, Lin JC-W, Chao HC, Fournier-Viger P, Wang X, Yu PS (2020) Khai thác thông tin xu hướng theo định hướng tiện ích cho hệ thống thông minh. ACM Trans Manag Inf Syst 11(3):1–28. https://doi.org/ 10.1145/3391251

34. Võ B, Nguyễn LTT, Nguyễn TDD, Fournier-Viger P, Yun U (2020) Cách tiếp cận đa lõi để khai thác hiệu quả các tập mục có tính tiện ích cao trong cơ sở dữ liệu lợi nhuận động. Truy cập IEEE. 8:85890–85899. https:// doi.org/10.1109/ACCESS.2020.2992729

35. Lin JC-W, Djenouri Y, Gautam S, Fournier-Viger P (2022) Mô hình tính toán tiến hóa hiệu quả của việc khai thác tập mục tiện ích cao khép kín. Ứng dụng Intell 52(9):10604–10616. https://doi.org/10.1007/ s10489-021-03134-3

36. Lin JC-W, Djenouri Y, Gautam S, Yun U, Fournier-Viger P (2021) Một mô hình dựa trên ga dự đoán để khai thác tập mục tiện ích cao khép kín. Máy tính mềm Appl 108:107422. https://doi.org/10.1016/j.asoc. 2021.107422

37. Yun U, Kim J (2015) Thuật toán nhiễu loạn nhanh sử dụng cấu trúc cây để khai thác tiện ích bảo đảm quyền riêng tư. Ứng dụng Hệ thống Chuyên gia 42(3):1149–1165. https://doi.org/10.1016/j.eswa.2014.08.037 38. Liu X, Wen S, Zuo W (2020) Các phương pháp vệ sinh hiệu quả

để bảo vệ kiến thức nhạy cảm trong khai thác tập mục có tiện ích cao. Ứng dụng Intell 50(1):169–191. https://doi.org/10.1007/s10489-019-01524- 2

39. Yin C, Li Y (2023) Thuật toán khai thác tiện ích bảo toàn quyền riêng tư nhanh dựa trên từ điển danh sách tiện ích. Ứng dụng Intell 53(23):29363–29377. https://doi.org/10.1007/s10489-023-04791-2

40. Nguyễn D, Lê B (2022) Một thuật toán nhanh để khai thác tiện ích bảo đảm quyền riêng tư. Tạp chí Công nghệ thông tin và Truyền thông. 2022(1):12–22. https://doi.org/10.32913/mic-ict-research. v2022.n1.1026

41. Lin CW, Hong TP, Wong JW, Lan GC, Lin WY (2014) Cách tiếp cận dựa trên GA để ẩn các tập mục có tính tiện ích cao nhạy cảm. Tạp chí Khoa học Thế giới 2014:2356–6140. https://doi.org/10.1155/ 2014/804629

42. Liu X, Chen G, Wen S, Song G (2020) Thuật toán dọn dẹp được cải tiến trong khai thác tiện ích bảo đảm quyền riêng tư. Bài toán Eng 2020:1–14. https://doi.org/10.1155/2020/7489045

43. Hatjimihail AT (1993) Thiết kế dựa trên thuật toán di truyền và tối ưu hóa các quy trình kiểm soát chất lượng thống kê. Lâm sàng Hóa học 39(9):1972–1978

44. Eberhart R, Kennedy J (1995) Một công cụ tối ưu hóa mới sử dụng lý thuyết bầy đàn hạt. Trong: MHS'95. Kỷ yếu của hội nghị chuyên đề quốc tế lần thứ sáu về máy vi tính và khoa học con người, trang 39–43. https://doi.org/10.1109/MHS.1995.494215 . IEEE

45. Colorni A, Dorigo M, Maniezzo V và cộng sự (1991) Tối ưu hóa phân tán bởi đàn kiến. Trong: Kỷ yếu hội nghị châu Âu đầu tiên về sự sống nhân tạo, tập 142, trang 134–142. Paris, Pháp 46. Mirjalili S, Mirjalili SM, Lewis A (2014) Trình tối ưu hóa sói xám. Adv

Tiếng Anh Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

47. Chou JS, Nguyễn NM (2020) Fbi lấy cảm hứng từ việc tối ưu hóa meta. ứng dụng. Máy tính mềm 93:106339. https://doi.org/10.1016/j.asoc.2020.106339

48. Tối ưu hóa Gurobi L (2020) Tài liệu tham khảo Trình tối ưu hóa Gurobi-

ual. http://www.gurobi.com 49. Zhang C, Almpanidis G, Wang W, Liu C (2018) Một nghiên cứu thực nghiệm

Đánh giá các thuật toán khai thác tập mục có tính tiện ích cao. Ứng dụng Hệ thống Chuyên gia 101:91–115. https://doi.org/10.1016/j.eswa.2018.02.008

50. Liu J, Wang K, Fung BC (2012) Khám phá trực tiếp các tập mục hữu ích cao mà không cần tạo ứng cử viên. Trong: Hội nghị quốc tế lần thứ 12 của IEEE về khai thác dữ liệu năm 2012, trang 984–989. https://doi.org/10.1109/ICDM.2012.20. IEEE

Đức Nguyễn nhận bằng Cử nhân năm 2018 và Thạc sĩ năm 2021 tại Trường Đại học Khoa học Tự nhiên - ĐHQG-HCM. Hiện anh đang công tác tại Khoa Khoa học Máy tính, Trường Đại học Khoa học Tự nhiên - ĐHQG-HCM. Mối quan tâm nghiên cứu của ông là học máy, nhận dạng mẫu, khai thác dữ liệu và bảo vệ quyền riêng tư trong khai thác dữ liệu.

Bắc Lê hiện là Giáo sư và Trưởng bộ môn Khoa học Máy tính, Khoa Công nghệ Thông tin, Đại học Khoa học Tự nhiên, Đại học Quốc gia Việt Nam, Thành phố Hồ Chí Minh, Việt Nam. Nghiên cứu chính của ông bao gồm trí tuệ nhân tạo, điện toán mềm, học máy và khai thác dữ liệu.