ĐẠI HỌC HOA SEN

Tạp chí Khoa học Trường Đại học Hoa Sen
https://www.hoasen.edu.vn/qlkh/ tapchidhhs.vn

MỘT THUẬT TOÁN KHAI THÁC TẬP LỢI ÍCH CAO LIÊN QUAN CÓ LỢI NHUẬN ÂM

Phạm Ngọc Diễm[1], Lý Quang Vinh*[2], Nguyễn Quí Trung[3], Cao Tùng Anh[4]

[2]Khoa Thiết Kế Nghệ Thuật
[3]Phòng Công Nghệ Thông Tin
[4]Trường ĐH Công nghệ TP. Hồ Chí Minh

Thông tin bài báo

Nhận bài: 12/2024
Chấp nhận: 02/2025
Xuất bản online: 03/2025

TÓM TẮT

Khai thác tập lợi ích cao (High-utility itemset - HUI) là một xu hướng nghiên cứu mạnh trong khai thác dữ liệu. Trong các nghiên cứu gần đây, nhiều thuật toán đã đề xuất việc khai thác các tập HUIs. Tuy nhiên, phần lớn các thuật toán chỉ quan tâm đến các bộ dữ liệu có giá trị lợi nhuận dương (LND). Như chúng ta biết, trong thực tế, tập các mặt hàng kinh doanh thường bao gồm cả giá trị LND (+) và âm (-). Để giải quyết vấn đề này, chúng tôi đề xuất một thuật toán để tìm HUIs có cả các mặt hàng mang lợi nhuận âm (LNA) trong cơ sở dữ liệu (CSDL).

Vấn đề đặt ra là làm thế nào để HUI liên quan có LNA một cách có hiệu quả? Từ CSDL giao dịch D và bảng lợi nhuận của các sản phẩm, làm sao để tìm được các tập lợi ích cao liên quan tốt nhất trên CSDL có mặt hàng có LNA? Từ những vấn đề được đặt ra, bài báo đã đề xuất một thuật toán COHUIs_CoHUN để "HUI liên quan có LNA" nhằm đáp ứng yêu cầu thực tế này.

High-Utility Itemset (HUI) mining is a prominent research trend in data mining. In recent studies, numerous algorithms have been proposed for mining HUIs. However, most of these algorithms focus exclusively on datasets with positive profit values. As we know, in real-world scenarios, business itemsets often include both positive (+) and negative (-) profit values. To address this issue, we propose an algorithm for mining HUIs that include items with negative profits in the database (DB).

The challenge lies in how to effectively mine high-utility itemsets associated with negative profits. From a transactional database (D) and a profit table of products, how can we identify the most relevant high-utility itemsets in a database containing items with negative profits? To address these challenges, this paper proposes the COHUIs_CoHUN algorithm to "mine high-utility itemsets associated with negative profits," aiming to meet these practical requirements.

Keywords: *Khai thác dữ liệu, lợi nhuận âm, tập lợi ích cao, tập liên quan*

* Tác giả liên hệ:
Email: *vinh.lyquang@hoasen.edu.vn*

1. TỔNG QUAN

Trong môi trường kinh doanh hiện nay, sự cạnh tranh gay gắt giữa các công ty đã thúc đẩy sự xuất hiện của nhiều chiến lược kinh doanh sáng tạo nhằm thu hút khách hàng. Một trong số đó là việc thực hiện các chiến dịch khuyến mãi và ưu đãi hấp dẫn, nhằm kích cầu tiêu dùng. Tuy nhiên, các sản phẩm khuyến mãi đi kèm với sản phẩm chính thường không mang lại lợi nhuận, hoặc thậm chí chịu lỗ, dẫn đến sự xuất hiện của các mặt hàng có giá trị LNA. Ngoài ra, các công ty đôi khi phải giảm giá bán thấp hơn giá vốn để thu hồi vốn nhanh, tạo ra các mặt hàng có LNA.

Nhiều thuật toán HUI đang phát triển trong những năm gần đây, vào năm 2015, thuật toán HUP-Miner [1] được đề xuất bởi tác giả S. Krishnamoorthy, đây là một phiên bản cải tiến của thuật toán HUI-Miner [2] với hai chiến lược cắt tỉa mới (PU-Prune and LA-Prune) để HUI. Lin và cộng sự (2016) đã đề xuất thuật toán FHN [3]. Năm 2018, Singh và cộng sự phát triển thuật toán EHIN để khai thác HUI trong CSDL có LNA [4].

Mặc dù các thuật toán trên khá tốt trong việc khai thác các tập HUI, tuy nhiên số lượng HUIs tìm được chứa nhiều mặt hàng có tương quan yếu và không mang lại ý nghĩa thực tế. Để giải quyết vấn đề này, vào năm 2018, W. Gan và cộng sự đề xuất thuật toán CoHUIM [5] xem xét cả mối liên quan và lợi nhuận giữa các sản phẩm trong giao dịch. Và đến năm 2020, thuật toán CoHUI-Miner [6] được đề xuất bởi Bay Vo và các đồng sự, thuật toán sử dụng phép chiếu CSDL để giảm kích thước và đưa ra một khái niệm mới, được gọi là lợi ích tiền tố của dự kiến giao dịch, để loại bỏ các tập hợp mẫu không đáp ứng ngưỡng tối thiểu trong quá trình khai thác; thuật toán đã đạt hiệu suất tốt hơn thuật toán CoHUIM [5] về cả thời gian chạy và sử dụng bộ nhớ. Thuật toán CoUPM, được giới thiệu vào năm 2019, cũng tiếp tục cải thiện hiệu suất khai thác bằng cách áp dụng cấu trúc Utility-List [7].

Dựa trên các nghiên cứu trên, bài báo này đề xuất thuật toán CoHUIs_CoHUN, với mục tiêu HUI liên quan có LNA một cách hiệu quả. Kết quả thực nghiệm (KQ-TN) cho thấy thuật toán đề xuất đã loại bỏ được nhiều tập HUI dư thừa và có hiệu suất cao hơn EHIN [4].

2. PHƯƠNG PHÁP

Trong các thuật toán tìm kiếm được gọi từ EHIN [4], các tác giả lần lượt tìm các tập lợi ích cao cho các mẫu có LND và sau đó là các mẫu có LNA. Điều này đã được các tác giả chứng minh là giảm không gian tìm kiếm do sử dụng 2 tập Primary và Secondary khác nhau cho các mẫu được coi là chính và phụ. Nó đảm bảo rằng một mẫu không phải đi hội với mọi mẫu khác để tìm ra các tập lợi ích cao mà chỉ cần tìm với các mẫu trong Primary và các mẫu trong Secondary đến khi có tập không thỏa ngưỡng min_util thì dừng (vì các tập Primary và secondary đã được sắp xếp theo thứ tự utility). Do vậy, việc cải tiến các thủ tục tìm kiếm này là không cần thiết, chúng tôi đề xuất vẫn sử dụng các thủ tục này khi thực hiện CoHUN để khai thác các mẫu có LNA không dư thừa.

Trong thuật toán đề xuất CoHUIs_CoHUN chúng tôi sẽ sử dụng thêm ngưỡng tương quan minCor (do người dùng đưa vào). Việc trả về tập CoHUIs (Bước 13) sẽ chứa tất cả các tập lợi ích cao liên quan có LNA.

Việc thêm ngưỡng liên quan minCor trong thuật toán EHIN [4] nhằm tăng tính hiệu quả trong việc HUI liên quan một cách hiệu quả trên tập CSDL giao dịch có LNA không dư thừa. Với tổng các giao dịch không đổi, kết quả tập lợi ích cao liên quan trên tập CSDL giao dịch có LNA đã loại bỏ được các tập dư thừa. Như vậy, thuật toán đề xuất CoHUIs_CoHUN sẽ có thời gian tính toán nhanh hơn so với thuật toán EHIN [4] và chúng ta có một thuật toán có thể khai thác được các tập lợi ích cao liên quan trên CSDL có LNA.

2.1. Thuật toán CoHUIs_CoHUN

Input:
D: tập các giao dịch
minUtil: ngưỡng lợi ích
minCor: ngưỡng giá trị liên quan.

Output:
CoHUIs: tập lợi ích cao không dư thừa

Begin
Bước 1: khởi tạo tập α rỗng
Bước 2: tạo tập ρ là tập các phần tử có giá trị dương trong D
Bước 3: tạo tập η là tập các phần tử có giá trị âm trong D
Bước 4: Duyệt D, tính RLU(α,z) $\forall z \in \rho$ và tính TU
Bước 5: Tìm tập Secondary(α) = {z|z ∈ ρ ∧ RLU(α,z) ≥ min_util x TU}
Bước 6: Sắp xếp các phần tử của tập Secondary(α) theo thứ tự tăng dần dựa vào giá trị RTWU(α)
Bước 7: Duyệt D, xóa các item z∉Secondary(α) và xóa các giao dịch T rỗng
Bước 8: Sắp xếp các phần tử trong giao dịch T ∈ D theo thứ tự tập Secondary(α) và sau đó là các phần tử giá trị âm.
Bước 9: Gắn giá trị offset cho từng giao dịch T∈D
Bước 10: Duyệt D, tính RSU(α,z)∀z∈Secondary(α) và Sup(X)∀z ∈ Secondary(α)
Bước 11: tìm tập Primary(α)={z|z∈Secondary(α)≥RSU(α,z)≥ min_util x TU}
Bước 12: Search_P(η,α,D,Primary(α),Secondary(α),min_util,min_cor)
Bước 13: return CoHUIs
End

2.2. Thủ tục Search_P

Input:
η: là tập các phần tử có giá trị âm trong D, *α*: là tập các phần tử, Tập *α – D*: tập dữ liệu giao dịch, *Primary(α)*: là tập Primary của tập *α*, Tập *Secondary(α)*: là tập Secondary của tập *α*, *min_util* là ngưỡng lợi ích, *min_cor*: giá trị liên quan giữa các phần tử

Output:

Tập các *CoHUIs* của tập *α* với phần tử dương

Duyệt mỗi phần tử *z* ∈*Primary(α)*

Bước 1: Tạo tập β = α∪{z}
Bước 2: duyệt *α – D*, tính *U(β)*, tính *Sup(β)*, tính *Kulc(β)* và tạo tập *β - D*
Bước 3: nếu *Kulc(β)* ≥ *min_cor* và *U(β)* ≥ *min_util x TU* thì output *β*
Bước 4: nếu *Kulc(β)* > *min_cor* và *U(β)* > *min_util x TU* thì *Search_N(η,β ,β - D,min_util,min_cor)*
Bước 5: Duyệt *β - D*, tính *RSU(β,z)* và *RLU(β,z)* ở phần tử *z*∈*Secondary(α)*
Bước 6: *Primary(α) = {z|z ∈ Secondary(α)|RSU(β,z) ≥ min_util x TU}*
Bước 7: *Secondary(α) = {z | z ∈ Secondary(α)||RLU(β,z) ≥ min_util x TU}*
Bước 8: *Search_P(η,β,β-D,Primary(β),Secondary (β),min_util,min_cor)*

2.3. Thủ tục Search_N

Input:
η: là tập các phần tử có giá trị âm trong D, *α:* là tập các phần tử, Tập *α – D*: tập dữ liệu giao dịch, *min_util* là ngưỡng lợi ích, *min_cor*: giá trị liên quan giữa các phần tử

Output:
Tập các *CoHUIs* của tập α với phần tử âm

Duyệt mỗi phần tử *z* ∈ *η*
Bước 1: *β = α∈{z}*
Bước 2: duyệt *α – D*, tính *U(β)*,tính *Sup(β)* tính *Kulc(β)* và tạo tập *β - D*
Bước 3: nếu *Kulc(β)* ≥ *min_cor* và *U(β)* ≥ *min_util x TU* thì output *β*
Bước 4: Duyệt *β - D*, tính *RSU (β,z)* phần tử *z* ∈*η*
Bước 5: *Primary(β) = {z|z ∈ η |RSU(β,z) ≥ min_util x TU}*
Bước 6: *Search_N(Primary(β),β,β - D,min_util,min_cor)*

Minh họa thuật toán: Ta có bảng CSDL giao dịch có LNA như sau:

Bảng 1: Dữ liệu giao dịch

T_{id}	a	b	c	d	e
T_1	2	2	-	1	3
T_2	-	1	5	-	1
T_3	-	2	1	3	2
T_4	-	-	2	1	3
T_5	2	-	-	-	-
T_6	2	1	4	2	1
T_7	-	3	2	-	2

Bảng 2: Bảng giá trị lợi nhuận

I	a	b	c	d	e
EU(X_i)	2	-3	1	4	1

Sau khi tính lợi nhuận của các mặt hàng trong giao dịch và tính lần lượt *U(X)* theo công thức [7][1], *TU(**T**$_j$)* theo công thức [7][2], *TWU(**T**$_j$)* theo công thức [7][3] ta được các bảng sau:

Bảng 3: Lợi nhuận theo giao dịch.

	a	b	c	d	e
	4	-6	-	4	3
	-	-3	5	-	1
	-	-6	1	12	2
	-	-	2	4	3
	4	-	-	-	-
	4	-3	4	8	1
	-	-9	2	-	2

Bảng 4: Lợi ích các mặt hàng trong giao dịch

	a	b	c	d	e	
T_1	4	-6	-	4	3	**5**
T_2	-	-3	5	-	1	**3**
T_3	-	-6	1	12	2	**9**
T_4	-	-	2	4	3	**9**
T_5	4	-	-	-	-	**4**
T_6	4	-3	4	8	1	**14**
T_7	-	-9	2	-	2	**-5**
U(X)	12	-27	14	28	12	
TWU(X)	**23**	**26**	**30**	**37**	**35**	**151**

2.4. Thực hiện thuật toán CoHUIs_CoHUN

Chọn: min_u*til* = 0,2; min _*cor* = 0,2

Bước 1: Tập $\alpha = \varnothing$
Bước 2: Tập ρ = tập phần tử có *U > 0*
Bước 3: Tập η = tập phần tử có *U < 0*
Bước 4: Tính *RLU(X),TU*
Bước 5: Tập *Secondary = {a,c,d,e} (Secondary = RLU(X) >min_util x TU)*
Bước 6: Sắp xếp tập *Secondary = a<c<d<e*
Bước 7: Xóa phần tử trong T mà không thuộc *Secondary*, xóa T rỗng
Bước 8: Gộp các *T,U(X)=U(T_{X_1})+U(T_{X_2}), TU(T_{new})=TU(T_1) + TU(T_2)*
Bước 9: Gắn Offset cho T
Bước 10: Duyệt D, tính *RSU(X), Sup(X)*

Bảng 5: Bảng sau khi thực hiện bước 4-10.

						Tập η
	a	**c**	**d**	**e**		**b**
T_1	4	-	4	3	11	(6)
T_2	-	7	-	3	10	(12)
T_3	-	1	12	2	15	(6)
T_4	-	2	4	3	9	-
T_5	4	-	-	-	4	-
T_6	4	4	8	1	17	(3)
Sup	**3**	**4**	**4**	**5**		**4**
RSU	**32**	**47**	**37**	**12**		

Bước 11: Tập *Primary={a,c,d,e}*
Bước 12: Search_P(η, α., *Primary, Secondary, min_uti, min_cor)*
Bước 13: Return *CoHUIs*

Sau Bước 12 thực hiện thủ tục Search_P ta có kết quả tập CoHUIs trả về tai bước 13 như sau:

Tập "***CoHUIs = ({a,c,d}, {a,c,d,e}, {a,c,d,e,b}, {a,d}, {a,d,e}, {a,d,e,b}, {c}, {c,d}, {c,d,b}, {c,d,e}, {c,d,e,b}, {c,e}, {d}, {d,e}, {d,e,b})***"

1 Công thức tính *U(X) "Lợi ích của một tập hợp mục trong tập dữ liệu": U(X)=* $\sum_{X \subseteq Tj \in D} U(X,T_j)$

2 Công thức tính *TU(T_j) (Lợi ích giao dịch): TU(T_j) =* $\sum_i^m U(X_i,T_j)$

3 Công thức tính *TWU(X) (Trọng số lợi ích giao dịch): TWU(X) =* $\sum_{X \subseteq Tj \in D} TU(T_j)$

Thuật toán đề xuất đã sử dụng thêm ngưỡng liên quan minCor (do người dùng đặt ra) có hiệu quả hơn trong việc HUI trên CSDL có trọng số âm không dư thừa. Ví dụ: Trong bài báo năm 2018 thuật toán EHIN [4] tìm được 20 tập CoHUIs, nhưng khi áp dụng dụng thuật toán đề xuất thì số lượng tập lợi ích cao là 15 tập (loại bỏ được 05 tập có lợi ích cao dư thừa).

3. KẾT QUẢ THỰC NGHIỆM

Thuật toán CoHUIs_CoHUN được cài đặt bằng ngôn ngữ lập trình Java và thử nghiệm trên máy tính "Dell Vostro 3500, Intel Core i7-1165G7 @2.80GHz, bộ nhớ RAM 16GB", hệ điều hành Windows 10. Các CSDL thử nghiệm được tải từ thư viện SPMF là các CSDL giao dịch có LNA gồm Chess, Mushroom, Accidents. Thực nghiệm của thuật toán CoHUN được so sánh với thuật toán mới nhất cùng khai thác tập CoHUIs là EHIN [4].

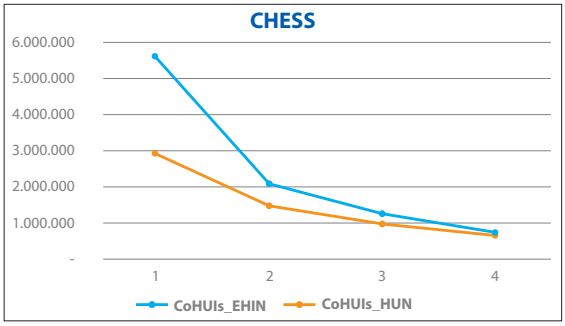
KQ-TN được đánh giá dựa trên kết quả các tập lợi ích cao liên quan có LNA thu được.

Bảng 6: Dữ liệu thực nghiệm

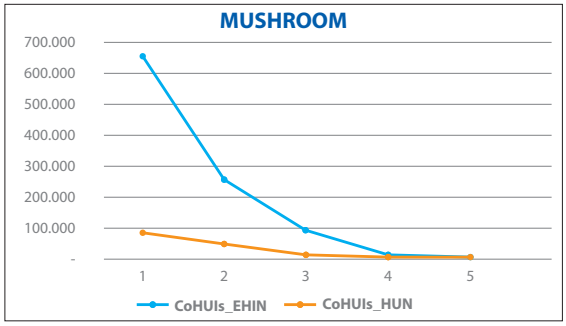
CSDL	Số lượng giao dịch	Số lượng item (I)	Độ dài trung bình
Chess	3.196	75	37
Mushroom	8.124	119	23
Retail	88.162	16.47	10.3
Accidents	340.183	468	33.8

Trong bài báo này, giá trị minCor = 0.2 được sử dụng cho tất cả các tập dữ liệu Chess, Mushroom, Retail, Accidents vì đây là giá trị tham chiếu từ bài báo gốc, giúp đảm bảo tính nhất quán khi so sánh với các thuật toán trước đây EHIN [4].

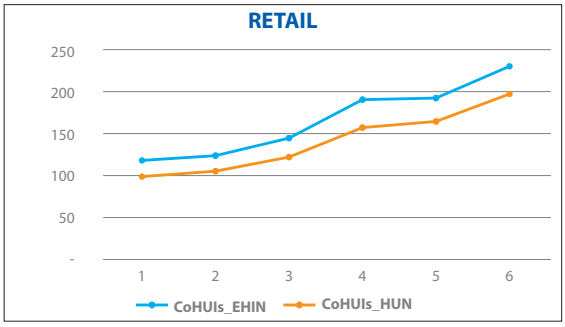
KQ-TN cho thấy thuật toán CoHUN có kết quả tập lợi ích cao HUIs thu được hiệu quả hơn thuật toán EHIN [4] trên tất cả các CSDL: Chess, Mushroom, Retail, Accidents.



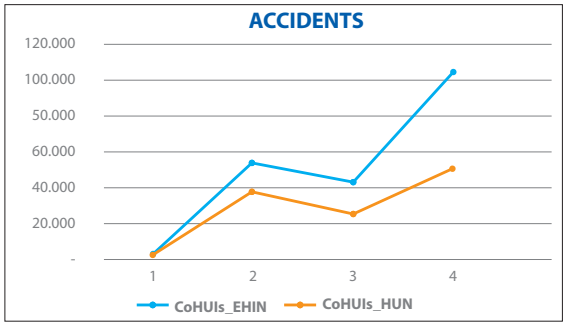
Hình 1: Kết quả trên bộ dữ liệu Chess



Hình 2: Kết quả trên bộ dữ liệu Mushroom



Hình 3: Kết quả trên bộ dữ liệu Retail



Hình 4: Kết quả trên bộ dữ liệu Accidents

KQ-TN cho thấy, các chiến lược cắt tỉa, mối tương quan và cấu trúc dữ liệu sử dụng trong thuật toán đề xuất CoHUIs_CoHUN là phù hợp khi HUI CoHUIs liên quan trên CSDL có LNA.

4. KẾT LUẬN

Trong bài báo này, chúng tôi đã đề xuất một thuật toán mới mang tên CoHUIs_CoHUN nhằm khai thác tập lợi ích cao liên quan có LNA. Thuật toán sử dụng cấu trúc dữ liệu PNU-List, cho phép tách biệt và xử lý hiệu quả các giá trị LND và LNA. Đồng thời, các chiến lược cắt tỉa thông minh như U-Prune, Kulc-Prune, và LA-Prune đã được áp dụng để giảm thiểu không gian tìm kiếm và loại bỏ các tập lợi ích cao dư thừa.

KQ-TN trên các bộ dữ liệu chuẩn từ thư viện SPMF đã cho thấy thuật toán CoHUIs_CoHUN vượt trội hơn so với EHIN [4] về cả hiệu suất và tính hiệu quả. Cụ thể, thuật toán không chỉ giảm thời gian xử lý mà còn tiết kiệm bộ nhớ và loại bỏ được các tập dư thừa không mang lại ý nghĩa thực tiễn.

Hướng nghiên cứu tiếp theo

Dựa trên những kết quả đạt được, chúng tôi đề xuất các hướng nghiên cứu trong tương lai như sau:

1. Cải tiến cấu trúc dữ liệu: Phát triển các cấu trúc dữ liệu mới nhằm tối ưu hóa việc lưu trữ và truy xuất trên các CSDL lớn.
2. Nâng cao ngưỡng tương quan: Khám phá các chiến lược sử dụng ngưỡng tương quan động để thích nghi với nhiều loại CSDL khác nhau.
3. Ứng dụng trên CSDL động: Mở rộng thuật toán để HUI trên CSDL động hoặc CSDL tăng trưởng theo thời gian.

Bài báo đã cung cấp một cách tiếp cận mới mẻ và hiệu quả trong việc xử lý dữ liệu giao dịch có LNA, góp phần mở rộng tiềm năng ứng dụng của HUI cao trong thực tế kinh doanh.

TÀI LIỆU THAM KHẢO

[1] S. Krishnamoorthy, "Pruning strategies for mining high utility itemsets," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2371-2381, 2015.

[2] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 55-64.

[3] J. C.-W. Lin, P. Fournier-Viger, and W. Gan, "FHN: An efficient algorithm for mining high-utility itemsets with negative unit profits," *Knowledge-Based Systems*, vol. 111, pp. 283-298, 2016.

[4] K. Singh, H. K. Shakya, A. Singh, and B. Biswas, "Mining of high-utility itemsets with negative utility," *Expert Systems*, vol. 35, no. 6, p. e12296, 2018.

[5] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and H. Fujita, "Extracting non-redundant correlated purchase behaviors by utility measure," *Knowledge-Based Systems*, vol. 143, pp. 30-41, 2018.

[6] B. Vo et al., "Mining correlated high utility itemsets in one phase," *IEEE Access*, vol. 8, pp. 90465-90477, 2020.

[7] W. Gan, J. C.-W. Lin, H.-C. Chao, H. Fujita, and S. Y. Philip, "Correlated utility-based pattern mining," *Information Sciences*, vol. 504, pp. 470-486, 2019.

[8] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, "FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning," in *Foundations of Intelligent Systems: 21st International Symposium, ISMIS 2014, Roskilde, Denmark, June 25-27, 2014. Proceedings 21*, 2014: Springer, pp. 83-92.

[9] C. T. Anh, N. Q. Huy, and V. H. Khang, "Khai thác tập mục lợi ích cao có lợi nhuận âm trong cơ sở dữ liệu phân tán dọc," *Dalat University Journal of Science*, pp. 25-38, 2020.