Sử dụng hai ngưỡng khai thác tập có thể xóa trên dữ liệu tăng cường

NGUYỄN THỊ HOÀI LINH *
Trường Đại học Kinh tế - Tài chính Thành phố Hồ Chí Minh
Ngày nhận: 21/08/2023 - Duyệt đăng: 31/08/2023
(*) Liên hệ: linhnth@uef.edu.vn

Tóm tắt:

Khai thác dữ liệu truyền thống thường được áp dụng trên các cơ sở dữ liệu (CSDL) tĩnh và xử lý theo lô. Trên thực tế, cơ sở dữ liệu thường xuyên biến động, việc xử lý theo lô không hiệu quả gây lãng phí khi một lượng nhỏ dữ liệu được thêm vào nhưng phải khai thác lại từ đầu. Vì vậy, khai thác dữ liệu trên cơ sở dữ liệu động đã thu hút sự nghiên cứu của nhiều tác giả. Trong đó, khai thác tập có thể xoá (EIs) trên cơ sở dữ liệu tăng cường là một trong những lĩnh vực thú vị. Mặc dù gần đây cũng đã có một vài công trình được phát triển để xử lý việc cập nhật EIs trên cơ sở dữ động nhưng hạn chế chính là xác suất quét lại CSDL lớn dẫn đến tốn nhiều thời gian cập nhật. Trong bài báo này chúng tôi đề xuất một thuật toán cập nhật EIs sử dụng hai ngưỡng để tránh việc quét lại nhiều lần CSDL gốc cũng như sử dụng các cấu trúc dữ liệu mới để xử lý dữ liệu tăng cường hiệu quả.

Từ khóa: Khai thác dữ liệu, tập có thể xóa, dữ liệu tăng cường, dữ liệu lớn, dữ liệu gốc.

Abstract:

Traditional data mining is often applied in static databases and bath processing. In fact, databases are often changed, bath processing is not suitable because it consumes more time to mine in the whole database. Therefore, mining in dynamic databases attracted many researchers where mining erasable itemsets (EIs) in incremental databases is one of interesting areas. Recent years, there are some publications developed for updated EIs in dynamic databases but they consume more time to rescan the original database. In this paper we propose an algorithm for updating EIs using two minimum thresholds to avoid rescanning databases, we also use new data structure to efficient process incremental databases.

Keywords: Data mining, erasable itemsets, incremental databases, big data, original data.

1. Giới thiệu tổng quan

Năm 2019, Deng và các đồng tác giả đã đưa ra bài toán khai thác tập có thể xóa (EI – erasable itemset) đồng thời đề xuất thuật toán META (mining erasable itemsets with the anti-monotone property), một thuật toán dựa vào thuật toán Apriori để khai thác tập có thể xóa (Deng và cộng sự, 2009). Sau đó, nhiều thuật toán đã được đề xuất như VME (vertical-format-based algorithm for mining erasable itemsets) được đề nghị năm 2010 (Deng và cộng sự, 2010), MERIT (fast mining erasable itemsets) được đề nghị năm 2012 (Le và cộng sự, 2014), và MEI (mining erasable itemsets) được đề nghị năm 2014 (Le và cộng sự, 2014) để giảm bộ nhớ sử dụng và thời gian khai thác.

Hầu hết các thuật toán trên sử dụng cơ sở dữ liệu tĩnh và khai thác theo lô. Tuy nhiên, trong thực tế, cơ sở dữ liệu thường thay đổi, các bản ghi mới thường được thêm vào cơ sở dữ liệu một cách thường xuyên. Việc phát triển một thuật toán khai thác tập có thể xoá khi cơ sở dữ liệu tăng cường là rất quan trọng. Chính vì vậy Hong và các đồng tác giả đã đề xuất một thuật toán dựa vào VME để cập nhật EIs trên CSDL tăng cường với một ngưỡng tối thiểu (Hong và cộng sự, 2017). Tuy nhiên, thuật toán này tốn nhiều thời gian khai thác do dựa vào ý tưởng của VME cũng như phải quét lại CSDL gốc nhiều lần.

Trong bài báo này, chúng tôi đề xuất một thuật toán khai thác tập có thể xoá trên cơ sở dữ liệu tăng cường sử dụng hai ngưỡng để tránh việc quét lại nhiều lần CSDL gốc cũng như sử dụng các cấu trúc dữ liệu mới để xử lý dữ liệu tăng cường hiệu quả dựa trên thuật toán FUP-erasable (Hong và cộng sự, 2017).

Phần còn lại của bài báo như sau: Phần 2 trình bày một số công trình liên quan đến khai thác tập có thể xoá và khai thác tập có thể xoá trên cơ sở dữ liệu tăng cường. Phần 3 trình bày thuật toán đề xuất bao gồm một số định nghĩa liên quan đến bài toán khai thác tập có thể xóa và các ký hiệu được sử dụng trong bài báo, đưa ra khái niệm pre-erasable, đồng thời phát biểu định lý về việc sử dụng hai ngưỡng để khai thác tập có thể xoá trên cơ sở dữ liệu tăng cường, dựa vào đó đề xuất thuật toán khai thác tập có thể xoá sử dụng hai ngưỡng trên cơ sở dữ liệu tăng cường. Cuối cùng phần 4 trình bày kết luận của bài báo.

2. Tình hình nghiên cứu

2.1. Khai thác tập có thể xoá trên CSDL tĩnh

Bài toán khai thác tập có thể xóa được đề xuất bởi Deng và các đồng tác giả vào năm 2009 (Deng và cộng sự, 2009), đồng thời, họ cũng đề xuất thuật toán META (mining erasable itemsets with the anti-monotone property), một thuật toán dựa vào thuật toán Apriori. Sau đó, nhiều thuật toán được phát triển để cải thiện thời gian và không gian lưu trữ như VME (vertical-format-based algorithm for mining erasable itemsets) được đề nghị năm 2010 (Deng và cộng sự, 2010), MERIT (fast mining erasable itemsets) được đề nghị năm 2012 (Deng và cộng sự, 2012), và MEI (mining erasable itemsets) được đề nghị năm 2014 (Le và cộng sự, 2014).

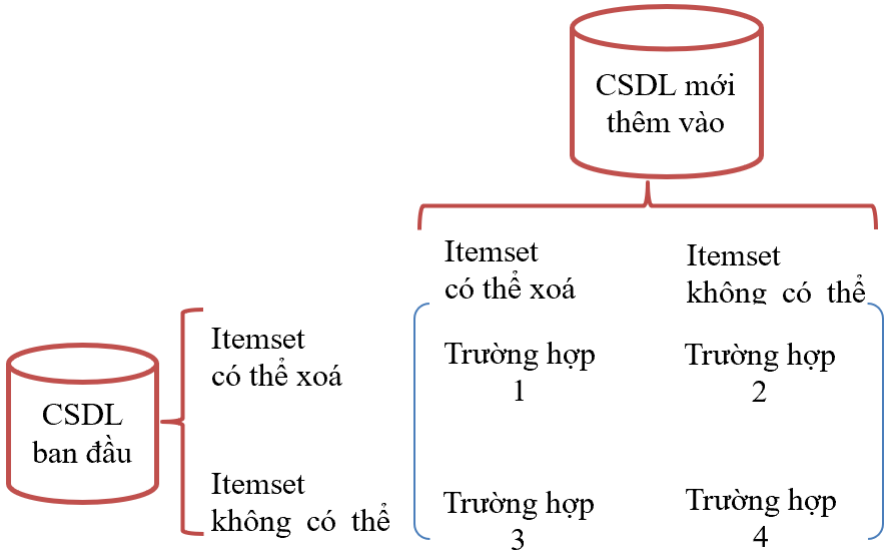
2.2. Khai thác tập có thể xoá trên CSDL tăng cường

Xử lý luồng dữ liệu động là một vấn đề thiết yếu trong khai thác tập có thể xoá. Vấn đề này đã thu hút sự nghiên cứu trong khai thác tập đại diện, khai thác tập tiện ích cao, khai thác tập Top-K. Ngoài ra, đã có những nỗ lực để xử lý với dữ liệu động trong lĩnh vực khai thác tập có thể xóa và hai phương pháp liên quan đã được đề xuất. WEPS (Yun và cộng sự, 2016) là một cách tiếp cận dựa trên cây tập trung vào dữ liệu mới nhất bằng cách áp dụng mô hình cửa sổ trượt vào khai thác mẫu có thể xóa. Nếu kích thước cửa sổ được cung cấp bởi người dùng, thuật toán thường xuyên chèn dữ liệu mới vào cửa sổ và xóa dữ liệu cũ. Cấu trúc dữ liệu để xử lý dữ liệu luồng được xây dựng với một lần quét cơ sở dữ liệu và được duy trì thông qua tái cấu trúc cây. IWEI (Lee và cộng sự, 2016) là một thuật toán dựa trên cây hiện có để xử lý dữ liệu gia tăng, thuật toán xây dựng cấu trúc cây IWEI, quét cơ sở dữ liệu một lần để xử lý dữ liệu gia tăng động. Tuy nhiên IWEI vẫn có những hạn chế vì chi phí xây dựng và tái cấu trúc cây phức tạp, đồng thời hiệu suất sử dụng cấu trúc cây thấp trong khi chi phí bảo trì cây lại cao.

2.3. Thuật toán FUP – erasable khai thác tập có thể xoá trên CSDL tăng cường

Hong và các đồng sự đề xuất thuật toán FUP-erasable (Hong và cộng sự, 2017), tương tự thuật toán FUP (Cheung và cộng sự, 1996) cho luật kết hợp, thuật toán cũng chia các trường hợp khai thác tập có thể xoá thành 4 trường hợp.

Trong 4 trường hợp trên, theo FUP, trường hợp 1 và trường hợp 4 sẽ không ảnh hưởng đến các itemset có thể xoá cuối cùng. Trường hợp 2 có thể



Hình 1. Bốn trường hợp trong thuật toán FUP – erasable

loại bỏ các itemset có thể xoá hiện có, và ngược lại, trường hợp 3 có thể thêm các tập có thể xoá mới. Dựa trên phương pháp FUP, các trường hợp 1, 2 và 4 được xử lý hiệu quả hơn các thuật toán khai thác hàng loạt thông thường. Tuy nhiên, thuật toán vẫn mất nhiều thời gian quét lại CSDL ban đầu trong trường hợp 3.

3. Thuật toán đề xuất

3.1 Các định nghĩa liên quan

Cho một CSDL sản phẩm P = {P_1, P_2,..., P_n} trong đó mỗi sản phẩm P_i chứa một tập các thành phần (itemset) và giá thành (*val*) để sản xuất sản phẩm tương ứng. Ví dụ về CSDL sản phẩm được minh họa trong Bảng 1.

Hai định nghĩa cơ sở của bài toán khai thác EI như sau (Deng và cộng sự, 2009).

Bảng 1. Ví dụ về một CSDL sản phẩm P

PID	Danh mục thành phần	val ($)
P1	a, b, e	2500
P2	a, b	1100
P3	a, e	1000
P4	b, c, e	150
P5	b, e	250
P6	c, e, g	150
P7	a, d, e, f	200
P8	c, d, e, f, h	100
P9	d,g	250
P10	b, f, h	100
P11	c, f, h	200

Định nghĩa 1. Cho X (⊆I) là một itemset, gain của X trong CSDL *P* được định nghĩa là:

$$g(X) = \sum_{\{P_k | X \cap P_k.itemset \neq \varnothing\}} P_k.val$$

Nghĩa là, gain của X được tính bằng tổng các val của các sản phẩm có chứa ít nhất một thành phần (item) có trong X.

Ví dụ 1: Xét CSDL như Bảng 1, {P_1, P_2, P_3, P_4, P_5, P_7, P_{10}} là các sản phẩm có chứa *a*, *b*, hoặc *ab* nên g(ab) = P_1.val + P_2.val + P_3.val + P_4.val + P_5.val + P_7.val + P_{10}.val = 5300 ($).

Định nghĩa 2. Cho một ngưỡng ξ và một CSDL sản phẩm *P*, gọi T là tổng các *val* trong *P*. Khi đó, một tập X được gọi là tập có thể xóa nếu:

$$g(X) \leq T \times \xi$$

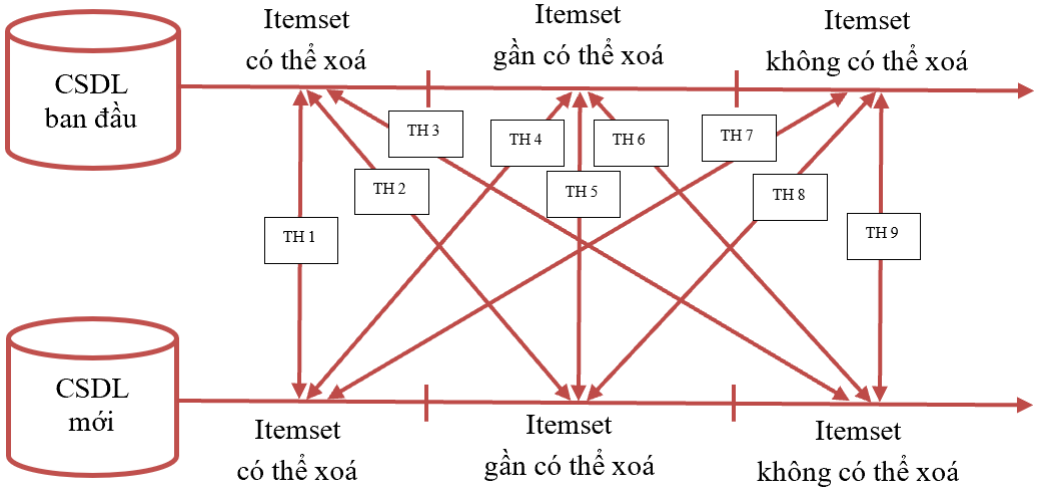
Ví dụ 2: Từ CSDL trong Bảng 1, ta có T = 6000 ($), giả sử ξ = 15%, từ ví dụ 1, ta có:

g(*ab*) = 5300 > 6000 × 15% = 900 nên *ab* không là tập có thể xóa.

3.2 Pre-erasable

Mặc dù thuật toán FUP-erasable tập trung vào các sản phẩm mới được chèn do đó tiết kiệm nhiều thời gian xử lý trong việc duy trì các quy tắc, nhưng nó vẫn phải quét lại CSDL ban đầu để xử lý trường hợp 3 khi 1 tập hợp các ứng viên là có thể xoá trong các sản phẩm mới nhưng không là tập có thể xoá trong CSDL ban đầu. Do đó, nếu xử lý trường hợp 3 hiệu quả, thời gian xử lý có thể giảm hơn nữa.

Trong bài báo này, chúng tôi đề xuất khái niệm



Hình 2. Chín trường hợp khi sử dụng pre-erasable

"pre-erasable" để giải quyết các vấn đề trên dựa trên ngưỡng trên và ngưỡng dưới nhằm giảm thiểu số lần quét lại cơ sở dữ liệu ban đầu. Ngưỡng hỗ trợ dưới (ξ_{era}) nhằm xác định các itemset có thể xoá giống các thuật toán khai thác thông thường. Ngược lại, ngưỡng hỗ trợ trên (ξ_{pre}) xác định tỉ lệ lớn nhất cho một item là gần có thể xoá (pre-erasable itemset). Một pre-erasable itemset không phải là thật sự có thể xoá nhưng có thể dễ dàng trở thành có thể xoá trong tương lai thông qua các dữ liệu trong quá trình thêm vào CSDL. Xem xét cơ sở dữ liệu ban đầu và các sản phẩm mới được chèn với hai ngưỡng, do đó, một itemset có thể có chín trường hợp (TH) được minh họa trong Hình 2.

Trong 9 trường hợp, các trường hợp 1, 5, 6, 8 và 9 sẽ không ảnh hưởng tới kết quả cuối cùng. Các trường hợp 2 và 3 có thể loại bỏ bớt các itemset có thể xoá hiện có. Và ngược lại, các trường hợp 4 và 7 có thể thêm các itemset có thể xoá mới vào kết quả. Trường hợp 2, 3 và 4 có thể được xử lý dễ dàng nếu chúng tôi giữ lại tất cả các itemset có thể xoá và itemset gần có thể xoá. Trong thực tế, tổng gain của các sản phẩm thêm vào luôn nhỏ hơn rất nhiều so với tổng gain CSDL ban đầu. Một itemset là không có thể xoá trong CSDL ban đầu nhưng có thể xoá trong CSDL mới thêm vào (trường hợp 7) có thể là tập không có thể xoá trong CSDL cập nhật miễn là tổng gain của các sản phẩm mới thêm vào nhỏ hơn so với tổng gain của CSDL ban đầu. Điều này sẽ được chứng minh bên dưới.

Các ký hiệu được sử dụng trong bài báo này được định nghĩa như sau:

D: Cơ sở dữ liệu ban đầu

T: Tập các sản phẩm mới được thêm vào

U: D ∪ T

G(D): Tổng gain của các sản phẩm trong D

G(T): Tổng gain của các sản phẩm trong T

ξ_{era}: Ngưỡng cho tập có thể xoá

ξ_{pre}: Ngưỡng cho tập pre-erasable,

$$\xi_{pre} > \xi_{era}$$

X: một itemset

Tr^D: Cây lưu các tập có thể xoá và pre-erasable từ D

Tr^U: Cây lưu các tập có thể xoá và pre-erasable từ U

$G^D(X)$: gain của itemset X trong CSDL D

$G^T(X)$: gain của itemset X trong CSDL T

$G^U(X)$: gain của itemset X trong CSDL U

E_k^D tập k-itemset có thể xoá và gần có thể xoá trong CSDL D

E_k^T tập k-itemset có thể xoá và gần có thể xoá trong CSDL T

E_k^U tập k-itemset có thể xoá và gần có thể xoá trong CSDL U

Định lí về việc sử dụng hai ngưỡng để khai thác tập có thể xoá trên cơ sở dữ liệu tăng cường

Định lí 1. Cho ξ_{era} và ξ_{pre} lần lượt là các ngưỡng dưới (ngưỡng có thể xoá) và ngưỡng trên (ngưỡng xác định các tập pre-erasable) và G(D), G(T) lần

lượt là tổng val của cơ sở dữ liệu ban đầu và tập các sản phẩm mới thêm vào. Nếu

$$G(T) < \frac{(1-\xi_{pre}) * G(D)}{\xi_{pre} - \xi_{era}}$$

Thì một tập X là tập không thể xoá (không là tập có thể xoá và cũng không là tập pre – erasable) trong cơ sở dữ liệu ban đầu nhưng là tập có thể xoá trong tập các sản phẩm mới được chèn vào là tập không có thể xoá trong cơ sở dữ liệu cập nhật mới.

Chứng minh định lí 1:

Một itemset X là tập không thể xoá trong cơ sở dữ liệu cập nhật U, tức là:

$G^U(X) > \xi_{pre} * G(U)$

Tương đương

$G^D(X) + G^T(X) > \xi_{pre} * (G(D) + G(T))$

$$\frac{G^D(X) + G^T(X)}{G(D) + G(T)} > \xi_{pre} \quad (1)$$

Một itemset X là tập không thể xoá trong cơ sở dữ liệu ban đầu D, tức là:

$$G(D) \geq G^D(X) > \xi_{pre} * G(D) \quad (2)$$

Một itemset X là tập có thể xoá trong T, tức là:

$$\xi_{era} * G(X) \geq G^T(X) \quad (3)$$

Từ (1), (2) và (3) ta có:

$$\frac{G(D) + G(T) * \xi_{era}}{G(D) + G(T)} > \xi_{pre}$$

Tương đương

$G(D) + G(T) * \xi_{era} > (G(D) + G(T)) * \xi_{pre}$

$G(D) + G(T) * \xi_{era} > G(D) * \xi_{pre} + G(T) * \xi_{pre}$

$G(D) - G(T) * \xi_{pre} > G(T) * \xi_{pre} - G(T) * \xi_{era}$

$(1 - \xi_{pre}) * G(D) > (\xi_{pre} - \xi_{era}) * G(T)$

$$\frac{(1-\xi_{pre}) * G(D)}{(\xi_{pre} - \xi_{era})} > G(D)$$ (ĐPCM)

3.5 Thuật toán đề xuất

Thuật toán đề xuất

Vào: Ngưỡng dưới ξ_{era} xác định các tập có thể xoá và ngưỡng trên ξ_{pre} xác định các tập gần có thể xoá, tập các itemset có thể xoá và gần có thể xoá E^D trong CSDL ban đầu D, tổng gain G^D CSDL ban đầu D và tập CSDL các sản phẩm mới được thêm vào T.

Ra: Các itemset có thể xoá kết quả trong CSDL cập nhật U.

Bước 1: tính giá trị an toàn f theo định lí 1 như

$$f = \frac{(1-\xi_{pre}) * G(D)}{\xi_{pre} - \xi_{era}}$$

Bước 2: tính tổng lợi nhuận G^T của CSDL sản phẩm mới thêm vào

Bước 2: liệt kê tập 1 ứng viên C_1^T và gain của chúng

Bước 3: đẩy tất cả các itemset trong C_1^T mà thoả ngưỡng vào E_1^T

Bước 4: với mỗi itemset X tồn tại trong E_1^T nhưng không tồn tại trong E_1^D, ta tiến hành các bước sau:

Bước 4-1: nếu G(T)+c <f, nếu X không tồn tại trong D thì bổ sung X vào E_1^U

Bước 4-2: nếu G(T)+c ≥ f

Bước 4-2-1: nếu X có tồn tại trong D thì tiến hành quét lại D, tính $G^U(X) = G^D(X) + G^T(X)$

Bước 4-2-2: nếu X không tồn tại trong D,

$G^U(X) = G^T(X)$

Nếu $G^U(X) \leq \xi_{pre} * (G^D + G^T)$ thì bổ sung X vào E_1^U

Bước 5: với mỗi itemset X tồn tại trong Tr^D (có thể xoá hoặc gần có thể xoá trong CSDL ban đầu) và đồng thời tồn tại trong E_1^U, ta tiến hành các bước sau:

Bước 5-1: $G^U(X) = G^D(X) + G^T(X)$

Bước 5-2: nếu $G^U(X) \leq (G(D) + G(T) + c) \times \xi_{era}$ thì cập nhật lại $G^U(X)$

Bước 5-3: ngược lại, xoá X khỏi E_1^U

Bước 6: với mỗi itemset X tồn tại trong E_1^D (có thể xoá hoặc gần có thể xoá trong CSDL ban đầu) và không tồn tại trong E_1^U (không xuất hiện trong CSDL mới), đẩy X vào E_1^U

Bước 7: tiến hành mining E_1^U để tạo ra E^U

Bước 8: nếu $G^T + c > f$

$G^D = G^D + G^T + c$

ngược lại: $c = c + G^T$

Bước 9: $E^D = E^U$

Kết luận

Trong bài báo này, chúng tôi đã đề xuất khái niệm về các tập gần có thể xoá. Dựa trên đó chúng tôi đưa ra một thuật toán khai thác tập có thể xoá trên cơ sở dữ liệu tăng cường hiệu quả. Thuật toán

sử dụng hai ngưỡng do người dùng chỉ định, ngưỡng dưới xác định các tập có thể xoá và ngưỡng trên xác định các tập gần có thể xoá. Các tập gần có thể xoá đóng vai trò như một bộ đệm để tránh trường hợp các tập không thể xoá trở thành có thể xoá trong CSDL được cập nhật khi các sản phẩm mới được thêm vào. Thuật toán đề xuất có thể xử lý hiệu quả trường hợp các tập không thể xoá trong CSDL gốc nhưng là tập có thể xoá trong CSDL các sản phẩm mới thêm vào, mặc dù thuật toán cần thêm không gian lưu trữ để ghi lại các tập gần có thể xoá.

Tiếp theo, chúng tôi sẽ tiến hành chạy thực nghiệm để so sánh đánh giá kết quả của thuật toán đề xuất so với FUP – erasable. Ngoài ra, chúng tôi cũng sẽ tiến hành nghiên cứu phương pháp hiệu quả để cập nhật lại EIs trên CSDL luồng.

TÀI LIỆU THAM KHẢO

Deng, Z., Fang, G., and Wang, Z. (2009). *Mining erasable itemsets*. Proceedings of the 8th International Conference on Machine Learning and Cybernetics, vol. 1, pp. 67-73, Jul. 2009.

Deng, Z., and Xu, X. (2010). An Efficient Algorithm for Mining Erasable Itemsets. *Advanced Data Mining and Applications*, pp. 214-225, Nov. 2010.

Deng, Z., and Xu, X. (2012). Fast mining erasable itemsets using NC_sets. *Expert Systems with Applications*, vol. 39, no. 4, pp. 4453-4463, Mar. 2012.

Hong, T. P., Huang, W.M., Lan, G. C., Chiang, M. C., Lin, J. C. W. (2021). A Bitmap Approach for Mining Erasable Itemsets. *IEEE Access* 9, 106029-106038.

Hong, T. P., Lin, K. Y., Lin, C. W., and Vo, B. (2017). *An incremental mining algorithm for erasable itemsets.* IEEE International Conference on innovations in intelligent systems and applications, pp. 286–289, 2017.

Le, T., and Vo, B. (2014). MEI: An efficient algorithm for mining erasable itemsets. *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 155-166, Jan. 2014.

Lee, C., Baek, Y., Ruy, T., Hyeonmo Kim, Heonho Kim, Lin, J. C. W., Vo, B., Yun, U. (2022). An efficient approach for mining maximized erasable utility patterns. *Inf. Sci.* 609:1288-1308.

Lee, G., Yun, U. (2017). Single-Pass based Efficient Erasable Pattern Mining using List Data Structure on Dynamic Incremental Databases. *Future Generation Computer Systems,* S0167-739X(16)30712-9.

Lee, G., Yun, U., Ryang, H., and Kim, D. (2016). Erasable itemset mining over incremental databases with weight conditions. *Engineering Applications of Artificial Intelligence,* vol. 52, pp. 213-234, 2016.

Ryang, H., and Yun, U. (2016). High utility pattern mining over data streams with sliding window technique. *Expert Systems with Applications*, vol. 57, pp. 214-231, 2016.

Yun, U., and Lee, G. (2016). Sliding window based weighted erasable stream pattern mining for stream data applications. *Future Generation Computer Systems*, vol. 59, pp. 1-20, 2016.