TẠP CHÍ KHOA HỌC ĐẠI HỌC ĐÀ LẠT Tập 8, Số 2, 2018 45–58

CẢI THIỆN THUẬT GIẢI CUCKOO TRONG VẤN ĐỀ ẨN LUẬT KẾT HỢP

Đoàn Minh Khuê[a*], Lê Hoài Bắc[b]

[a]Khoa Công nghệ Thông tin, Trường Đại học Đà Lạt, Lâm Đồng, Việt Nam
[b]Khoa Công nghệ Thông tin, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia TP. Hồ Chí Minh, TP. Hồ Chí Minh, Việt Nam
[*]Tác giả liên hệ: Email: khuedm@dlu.edu.vn

Lịch sử bài báo
Nhận ngày 23 tháng 01 năm2018
Chỉnh sửa ngày 24 tháng 04 năm 2018 | Chấp nhận đăng ngày 14 tháng 05 năm 2018

Tóm tắt

Hiện nay, vấn đề bảo mật dữ liệu ngày càng được quan tâm hơn trong quá trình khai thác dữ liệu. Làm sao để vừa có thể khai thác hợp pháp mà vừa tránh lộ ra các thông tin nhạy cảm. Có rất nhiều hướng tiếp cận nhưng nổi trội trong số đó là khai thác luật kết hợp đảm bảo sự riêng tư nhằm ẩn các luật nhạy cảm. Gần đây, có một thuật toán meta heuristic khá hiệu quả để đạt mục đích này, đó là thuật toán tối ưu hóa Cuckoo (COA4ARH). Trong bài báo này, một đề xuất cải tiến của COA4ARH được đưa ra để tính toán số lượng tối thiểu các item nhạy cảm cần được xóa để ẩn luật, từ đó hạn chế việc mất các luật không nhạy cảm. Các kết quả thực nghiệm tiến hành trên ba tập dữ liệu thực cho thấy trong một số trường hợp thì cải tiến đề xuất có kết quả khá tốt so với thuật toán ban đầu.

Từ khóa: Ẩn luật nhạy cảm; Khai thác dữ liệu đảm bảo sự riêng tư; Tác dụng phụ; Thuật toán tối ưu hóa Cuckoo.

Mã số định danh bài báo: http://tckh.dlu.edu.vn/index.php/tckhdhdl/article/view/410

Loại bài báo: Bài báo nghiên cứu gốc có bình duyệt

IMPROVEMENT OF CUCKOO ALGORITHM FOR ASSOCIATION RULE HIDING PROBLEM

Doan Minh Khue[a*], Le Hoai Bac[b]

[a]The Faculty of Information Technology, Dalat University, Lamdong, Vietnam
[b]The Faculty of Information Technology, University of Science, VNU Hochiminh City, Hochiminh City, Vietnam
[*]Corresponding author: Email: khuedm@dlu.edu.vn

Article history
Received: January 23rd, 2018
Received in revised form: April 24th, 2018 | Accepted: May 14th, 2018

Abstract

Nowadays, the problem of data security in the process of data mining receives more attention. The question is how to balance between exploiting legal data and avoiding revealing sensitive information. There have been many approaches, and one remarkable approach is privacy preservation in association rule mining to hide sensitive rules. Recently, a meta-heuristic algorithm is relatively effective for this purpose, which is cuckoo optimization algorithm (COA4ARH). In this paper, an improved version of COA4ARH is presented for calculating the minimum number of sensitive items which should be removed to hide sensitive rules, as well as limit the loss of non-sensitive rules. The experimental results gained from three real datasets showed that the proposed method has better results compared to the original algorithm in several cases.

Keywords: Cuckoo optimization algorithm; Privacy-preserving data mining; Sensitive association rule hiding; Side effect.

Article identifier: http://tckh.dlu.edu.vn/index.php/tckhdhdl/article/view/410

Article type: (peer-reviewed) Full-length research article

1. GIỚI THIỆU

Trong hợp tác kinh doanh, việc chia sẻ dữ liệu giữa các tổ chức là rất phổ biến. Việc này có thể cung cấp các thông tin có giá trị, chẳng hạn như mô hình mua sắm của khách hàng tại các siêu thị hay phát hiện gian lận xảy ra trong ngành ngân hàng. Tuy nhiên, điều này dẫn đến một lo ngại là để lộ ra các thông tin nhạy cảm không mong muốn cho các bên thứ ba. Trong tình huống này rất cần thiết có một lĩnh vực nghiên cứu để vừa có thể khai thác dữ liệu vừa đảm bảo không làm lộ những tri thức nhạy cảm trong cơ sở dữ liệu ấy. Chính vì lý do này, lĩnh vực *khai thác dữ liệu đảm bảo sự riêng tư* đã ra đời và đang được phát triển trong những năm gần đây.

Kể từ khi công trình tiên phong của Agrawal và Srikant (2000, tr. 439-450) và Lindell và Pinkas (2000), một vài phương pháp đã được đề xuất nhằm mục đích đảm bảo sự riêng tư trong khai thác dữ liệu. Bài báo này sẽ tập trung vào hướng nghiên cứu tiếp cận được biết nhiều là *tập phổ biến và khai thác luật kết hợp đảm bảo sự riêng tư* nhằm ẩn các luật kết hợp nhạy cảm. Luật kết hợp là các phép kéo theo được rút ra từ một cơ sở dữ liệu giao dịch theo các tham số được chỉ định bởi người dùng. Luật kết hợp sẽ cung cấp tri thức cô đọng nhất cho người khai thác dữ liệu vì chúng là bản tóm tắt dữ liệu, trong đó có chứa các mối quan hệ giữa các *item* trong dữ liệu. Thuật ngữ "*ẩn luật kết hợp*" đã được đề cập lần đầu tiên bởi Atallah, Bertino, Elmagarmid, Ibrahim, và Verykios (1999, tr. 45-52) trong một hội thảo về kỹ thuật tri thức và dữ liệu. Các tác giả của công trình này đã tìm cách sửa đổi cơ sở dữ liệu ban đầu theo một cách sao cho các luật kết hợp nhạy cảm sẽ được ẩn, nhưng điều này có thể gây ảnh hưởng đến tập dữ liệu gốc và các luật kết hợp không nhạy cảm.

Có thể xem quá trình ẩn luật kết hợp nhạy cảm là quá trình biến đổi tập dữ liệu ban đầu, với các luật kết hợp nhạy cảm được chỉ định bởi người dùng, quá trình này thường được gọi là quá trình thanh trùng (*sanitization*) dữ liệu. Kết quả quá trình thanh trùng là sẽ tạo ra tập dữ liệu thanh trùng để cho dù bất kỳ thuật toán khai thác luật kết hợp nào được áp dụng trên tập dữ liệu này thì sẽ không có khả năng để phát hiện ra các luật nhạy cảm theo các thiết lập tham số chỉ định và sẽ có thể khai thác tất cả các luật không nhạy cảm đã xuất hiện trong các tập dữ liệu ban đầu (theo các thiết lập tham số tương tự hoặc cao hơn) và không có thêm các luật khác được tạo ra.

Do đó, thách thức đặt ra là làm thế nào để cân bằng giữa nhu cầu ẩn luật nhạy cảm với việc khai thác thông tin hợp pháp của dữ liệu người dùng. Trong trường hợp lý tưởng là tất cả các luật nhạy cảm được ẩn, không có luật không nhạy cảm từ cơ sở dữ liệu gốc bị mất và không có luật ma được tạo ra. Tuy nhiên, thực tế chứng minh rằng rất khó để đạt được một mục tiêu lý tưởng như vậy mà không phát sinh bất kỳ tác dụng phụ nào. Bởi lẽ nó còn phụ thuộc các yếu tố như tập dữ liệu ban đầu đa dạng ra sao hay các luật nhạy cảm được định nghĩa bởi người dùng là đơn giản hay phức tạp. Để giải quyết vấn đề này, đã có rất nhiều kỹ thuật được đề xuất, chẳng hạn như sửa đổi dữ liệu (*distortion*) (Oliveira & Zaïane, 2004) là thêm mới hay xóa bỏ các *item* hiện có trong dữ liệu gốc, còn kỹ thuật ngăn chặn (*blocking*) (Chang & Moskowitz, 1998) thì thay thế

các *item* bằng ẩn số (*unknowns*). Cho đến thời điểm hiện tại thì các hoạt động để bảo vệ sự riêng tư trong khai thác luật kết hợp có thể được chia thành ba loại chính là: i) Tiếp cận *heuristic*; ii) Tiếp cận dựa vào biên; và iii) Tiếp cận chính xác.

Trong những năm gần đây, các thuật toán *meta heuristic* đã được sử dụng để khai thác luật kết hợp nhằm đảm bảo sự riêng tư, chẳng hạn như *thuật toán tối ưu hóa Cuckoo* (Walton, Hassan, Morgan, & Brown, 2011) và *ẩn luật kết hợp sử dụng thuật toán tối ưu hóa Cuckoo* (Mahtab, Mohammad, & Mehdi, 2016). Tuy nhiên, các thuật toán này để bảo vệ thông tin nhạy cảm tránh bị tiết lộ thì chưa thật sự hoàn hảo và vẫn còn một số hiệu ứng phụ, đặc biệt là về việc mất luật còn rất cao. Chẳng hạn, trong công trình nghiên cứu và thực nghiệm "*thuật toán tối ưu hóa Cuckoo*" của Mahtab và ctg. (2016), chúng tôi nhận thấy rằng trong một vài trường hợp thì "*thuật toán tối ưu hóa Cuckoo*" không thể ẩn hoàn toàn các luật nhạy cảm. Đó là khi tập các luật kết hợp được định nghĩa bởi người dùng khá nhập nhằng. Do vậy, chúng tôi đề xuất một cải thiện để khắc phục nhược điểm này của thuật toán. Giải pháp của chúng tôi tập trung vào việc tính toán số lượng nhỏ nhất các *item* nhạy cảm để vừa đảm bảo ẩn hết các luật nhạy cảm, vừa hạn chế thao tác xóa lên dữ liệu ban đầu. Kết quả thực nghiệm với một số trường hợp đã cho thấy cải tiến của chúng tôi có thể ẩn hoàn toàn các luật nhạy cảm cùng một lúc. Để đạt được hiệu suất như vậy là do cải tiến dựa trên mối tương quan giữa các luật nhạy cảm mà tính toán ra *item* nhạy cảm.

2. ĐỊNH NGHĨA BÀI TOÁN

Giả sử rằng $I = \{i_1, i_2, i_3, ..., i_m\}$ là một tập của các *item*. D là một cơ sở dữ liệu bao gồm nhiều giao dịch. Mỗi giao dịch là một tập con của I. Tập các luật kết hợp được rút ra từ D là R và tập các luật kết hợp nhạy cảm là R_s, với $R_s \subset R$. Mỗi luật kết hợp được biểu diễn như $A \rightarrow B$ trong đó A là tiền đề hoặc phía bên trái và B là kết quả hoặc phía bên phải, để $A, B \subset I$ và $A \cap B = \emptyset$. Mục đích là để thay đổi D thành D', để các luật hiện hành trong R_s không thể được khai thác từ D' trong khi các luật hiện hành trong $R - R_s$ có thể được có thể khai thác. Hai tiêu chí được xem xét trong việc khai thác luật kết hợp bao gồm:

- Tiêu chí đầu tiên được gọi là độ hỗ trợ *item* cho biết tần suất của một luật trong cơ sở dữ liệu và có thể được tính bằng công thức (1):

$$Sup(A \rightarrow B) = \frac{|A \cup B|}{|D|} \quad (1)$$

Trong đó $Sup(A \rightarrow B)$ là độ hỗ trợ của luật kết hợp $A \rightarrow B$, $|A \cup B|$ là số giao dịch chứa tất cả các *items* trong cả hai tập A và B. D là tổng số giao dịch.

- Tiêu chí thứ hai là độ tin cậy luật cho biết độ mạnh luật trong cơ sở dữ liệu và có thể được tính bằng công thức (2):

$$Conf(A \rightarrow B) = \frac{|A \cup B|}{|A|} \quad (2)$$

Đối với mỗi luật kết hợp, một ngưỡng hỗ trợ nhỏ nhất (*Minimum Support Threshold - MST*) và một ngưỡng tin cậy nhỏ nhất (*Minimum Confidence Threshold - MCT*) được xác định. Một luật thỏa mãn khi độ hỗ trợ của nó là cao hơn *MST* và độ tin cậy của nó cao hơn *MCT*.

Khai thác luật kết hợp thường bao gồm hai giai đoạn: Giai đoạn 1 là tập các *item*set phổ biến được khai thác với một ngưỡng *MST* đã cho và Giai đoạn 2 là luật kết hợp mạnh được sinh ra từ các tập phổ biến thu được trong Giai đoạn 1 và phụ thuộc một ngưỡng *MCT* đã cho. Các luật được đề cập đến sau này đều là các luật mạnh. Có ba tác dụng phụ có thể có sau khi chuyển từ *D* thành *D'*: Tập con của các luật nhạy cảm không được ẩn trong *D'* gọi là *HF* (*Hiding Failure*), $HF = \{r \in R_S | r \in R'\}$. Bất kỳ các luật không nhạy cảm nào bị ẩn sai và bị mất trong *D'*, gọi là *LR* (*Lost Rules*), $LR = \{r \in (R - R_S) | r \notin R'\}$. Các luật giả tạo sinh ra trong *D'* được chứa vào *GR* (*Ghost Rules*), $GR = \{r \in R' | r \notin R\}$.

Giả định: Một luật được coi là được ẩn nếu độ hỗ trợ của nó là nhỏ hơn so với *MST* hoặc độ tin cậy của nó là nhỏ hơn *MCT*. Nói cách khác, nếu một luật mạnh trong *D* không còn là mạnh trong *D'*, chúng ta xem nó là được ẩn. Nhiệm vụ của phương pháp ẩn luật là bằng cách nào đó để chuyển từ *D* sang *D'* mà tất cả luật nhạy cảm R_S trở nên ẩn trong *D'* trong khi tránh được các dụng phụ (nghĩa là giảm thiểu các thành viên có trong *HF*, *LR*, và *GR*).

Do đó, để che giấu một luật thì đòi hỏi phải giảm độ hỗ trợ hoặc sự tự tin của nó đến một mức dưới ngưỡng. Quá trình ẩn thường có thể ảnh hưởng đến các luật không nhạy cảm trong *D* hoặc các luật tiền mạnh trong *D*. Các luật tiền mạnh là những luật với độ hỗ trợ không nhỏ hơn *MST* và độ tin cậy nhỏ hơn *MCT*. Một luật tiền mạnh có thể trở nên mạnh khi độ tin cậy của nó trên *MCT*. Một luật không nhạy cảm trong *D* có thể chấm dứt mạnh khi độ hỗ trợ của nó giảm xuống dưới *MST* hay độ tin cậy của nó giảm xuống dưới *MCT* trong *D'* do việc loại bỏ *item*.

3. ĐỀ XUẤT CẢI THIỆN THUẬT TOÁN

3.1. Giới thiệu thuật toán

Phương pháp đề xuất của chúng tôi là một phiên bản cải tiến của COA4ARH (Mahtab & ctg., 2016). Vì vậy, hầu hết các bước của đề xuất là tương đồng với thuật toán ban đầu, chỉ có khác ở bước xác định các *item* nhạy cảm trong giai đoạn tiền xử lý. Các bước chính của thuật toán COA4ARH được trình bày trong Hình 1.

Input:
Tập dữ liệu gốc D, R_s, MST và MCT;
α, N_{pop}, N_{max}, MinNNS, MaxNNS, MaxIteration;
Output:
Một tập dữ liệu đã làm sạch D';
begin
Tiền xử lý mới cho tập dữ liệu gốc:
Khởi tạo 1 quần thể ban đầu;
call *FitnessFunction*;
call *BestSolutionFunction*;
repeat
// Tạo ra giải pháp mới
for each solution in population
K = (Max NNS - Min NNS) × Rand [0, 1] + Min NNS; // Xác định K
MR = $[\propto \times \frac{Current\ solution's\ K}{Total\ of\ all\ solution's\ K}] \times (Var_{hi} - Varlo_w)$; //Xác định MR
for K times do
for MR times do
Chọn ngẫu nhiên một *item* trong số các *item* nhạy cảm;
Đặt *Rand [0,1]* cho *item* đã chọn;
end for
end for
end for
call *FitnessFunction*;
call *BestSolutionFunction;*
Limit number of solutions to Nmax;
for each solution in population
call *ImmigrationFunction*; // Di chuyển tất cả các giải pháp hiện tại đến giải pháp tốt nhất
end for
call *FitnessFunction*;
call *BestSolutionFunction;*
until Điều kiện dừng được thỏa mãn;
end.

Hình 1. Thuật toán COA4ARH

3.2. Sửa đổi hoạt động tiền xử lý tập dữ liệu ban đầu

Như trình bày ở trên thì một luật kết hợp nhạy cảm được ẩn khi độ hỗ trợ hay độ tin cậy của nó nằm dưới ngưỡng. Để giảm độ tin cậy của một luật $X \rightarrow Y$, ta có thể tăng độ hỗ trợ của X, là các *item* bên trái luật, hay giảm độ hỗ trợ của *itemset* $X \cup Y$. Trường hợp thứ hai, nếu ta chỉ giảm độ hỗ trợ của Y, là các *item* bên phải luật, thì nó sẽ giảm độ tin cậy nhanh hơn khi chỉ giảm độ hỗ trợ của $X \cup Y$. Để giảm độ hỗ trợ của một *itemset*, ta sẽ sửa đổi một *item* bằng cách xóa *item* này trong các giao dịch hỗ trợ. Dựa vào những nhận định này, bài báo gốc của Mahtab và ctg. (2016) đã đưa ra cách xác định các *item* nhạy cảm của riêng nó trong hoạt động tiền xử lý như sau:

- Nếu một luật nhạy cảm chỉ có duy nhất một *item* ở bên phải, *item* độc nhất này sẽ được chọn làm *item* nhạy cảm.

- Nếu một luật nhạy cảm có nhiều hơn một *item* ở bên phải, một *item* trong số chúng sẽ được chọn làm *item* nhạy cảm mà *item* này có tần suất xuất hiện nhiều nhất ở phía bên phải của các luật nhạy cảm và có tần suất ít nhất của các luật không nhạy cảm.

Dễ nhận thấy rằng, trong trường hợp một luật nhạy cảm có nhiều *item* ở bên phải, một *item* có thể có tần suất xuất hiện nhiều nhất ở phía bên phải của các luật nhạy cảm và có thể có tần suất nhiều nhất ở phía bên phải của các luật không nhạy cảm so với các *item* còn lại trong luật nhạy cảm này. Điều này dẫn đến vấn đề là không có *item* nhạy cảm cho luật nhạy cảm, hay luật này không được ẩn ($HF \neq 0$). Như vậy, trong tình huống này, thuật toán ban đầu đã bị thất bại trong mục tiêu hàng đầu của mình là ẩn hết hoàn toàn và cùng một lúc các luật nhạy cảm. Do vậy, cần có một phương pháp để khắc phục nhược điểm này của "*thuật toán tối ưu hóa Cuckoo*". Chúng tôi đề xuất phương pháp xác định các *item* nhạy cảm bằng "*lược đồ nhạy cảm*", được mô tả chi tiết trong phần dưới đây.

3.2.1. Các định nghĩa cho đề xuất

Dựa vào mối tương quan giữa các luật nhạy cảm mà chọn ra *item* nhạy cảm nhưng cũng phải đảm bảo gây ảnh hưởng đến cơ sở dữ liệu ở mức thấp nhất có thể được. Chúng tôi đưa ra định nghĩa một "*lược đồ nhạy cảm*" để đại diện cho một *item* nhạy cảm. Một lược đồ cho biết độ phủ các luật nhạy cảm và các ảnh hưởng lên cơ sở dữ liệu ban đầu của *item* mà nó đại diện. Một thời điểm chỉ chọn ra được một lược đồ có lợi nhất. "*Lược đồ nhạy cảm*" tốt nhất (có lợi) là lược đồ có độ phủ thông tin về số lượng các luật nhạy cảm nhiều nhất, đồng thời có tác động thấp nhất đến cơ sở dữ liệu các giao dịch.

- *Định nghĩa 1*: "*Lược đồ nhạy cảm"* của một *item* nhạy cảm được định nghĩa như sau:

$$L = <SI, CV, AM> \tag{3}$$

Trong đó: *SI* là *item* nhạy cảm mà lược đồ đại diện; *CV* là độ phủ luật của lược đồ; và *AM* (*Affect Measure*) là độ đo mức độ ảnh hưởng của một lược đồ đến các giao dịch.

- *Định nghĩa 2*: (*Giao dịch của lược đồ*): Một giao dịch *t* được xem là chứa trong một lược đồ *L* nếu thỏa mãn hai điều kiện sau: i) $NI_L \subseteq t$ và ii) $SI_L \in t$. Với tập tất cả các giao dịch được chứa trong *L* là T_L. $\forall\, t \in T_L$.

- *Định nghĩa 3*: (*Độ bao phủ luật nhạy cảm r của một lược đồ L*): Giả sử tập tất cả các luật nhạy cảm được phủ bởi *L* là SR_L. Một luật nhạy cảm $r: X \rightarrow Y$ được xem là phủ bởi một lược đồ *L* nếu $SI_L \in Y$. $\forall\, r \in SR_L$. Độ phủ (*CV*) của một *L* được định nghĩa là số lượng các luật trong SR_L. Độ phủ *CV* được

tính theo công thức (4).

$$CV = |SR_L| \quad (4)$$

- *Định nghĩa 4*: (*Minimal Count - MC của một luật nhạy cảm*): Cho một lược đồ L và một luật $r: X \rightarrow Y$. Để đặc trưng cho tác động ẩn của một luật nhạy cảm r chứa trong một lược đồ L gây ra ảnh hưởng lên cơ sở dữ liệu các giao dịch T_L. Để ảnh hưởng này là nhỏ nhất ta đưa ra định nghĩa MC_r. MC_r chính là số lượng nhỏ nhất các giao dịch bị ảnh hưởng khi ta ẩn các luật r trong SR_L và MC của luật r được tính như công thức (5).

$$MC_{r,L} = min \{MSC_r, MCCC_r\} \quad (5)$$

Trong đó: MSC là số lượng tối thiểu các giao dịch mà cần phải được sửa đổi để giảm độ hỗ trợ của luật r; $MCCC$ là số lượng tối thiểu các giao dịch mà cần phải được sửa đổi để giảm độ tin cậy của luật r. Công thức tính MSC và $MCCC$ đã được Wu, Chiang, và Arbee (2007) trình bày chi tiết tại hội nghị IEEE về tri thức và dữ liệu.

- *Định nghĩa 5*: (*Độ đo ảnh hưởng - AM*): Cho một lược đồ L, tập các luật nhạy cảm SR_L và các giao dịch T_L. Xét luật $r: X \rightarrow Y, \forall r \in SR_L$. Luật r có một giá trị $MC_{r,L}$. Mức độ ảnh hưởng của một lược đồ L đến các giao dịch T_L được định nghĩa là số lượng ảnh hưởng MC lớn nhất của luật r được phủ trong L lên các giao dịch T_L để đảm bảo ẩn tất cả các luật trong SR_L.

$$AM_L = max \{MC_{ri,L}\} \quad (6)$$

Trong đó: $r_i \in SR_L$ và được phủ trong L.

- *Định nghĩa 6*: (*Rút gọn lược đồ*): Một lược đồ L_1 có thể rút gọn được nếu tồn tại lược đồ L_2 sao cho $SI_{L1} = SI_{L2}$.

3.2.2. Chiến lược tìm kiếm các item nhạy cảm dựa vào "lược đồ nhạy cảm"

Chiến lược này có thể theo các bước như sau:

- *Bước 1*: Duyệt các luật nhạy cảm ban đầu, tính toán giá trị MSC và $MCCC$ cho từng luật nhạy cảm;

- *Bước 2*: Xây dựng bảng "*lược đồ nhạy cảm*": Duyệt tất cả các luật nhạy cảm; Tạo các lược đồ tương ứng với mỗi *item* bên phải trong luật; Rút gọn các lược đồ dư thừa;

- *Bước 3*: Chọn ra một lược đồ tốt nhất trong số các lược đồ còn lại ở Bước 2. Mục tiêu là để xác định ra *item* nhạy cảm: Nếu lược đồ có CV lớn nhất được chọn làm *item* nhạy cảm; Nếu nhiều lược đồ có cùng giá trị CV thì lược đồ

có *AM* nhỏ nhất được chọn làm *item* nhạy cảm; Nếu nhiều lược đồ có cùng giá trị *AM* nhỏ nhất thì chọn ngẫu nhiên một lược đồ làm *item* nhạy cảm; Bộ các luật nhạy cảm hiện hành = bộ các luật nhạy cảm ban đầu - *CV* của lược đồ tốt nhất;

- *Bước 4*: Xuất ra danh sách các *item* nhạy cảm.

Nếu mô hình tốt nhất tìm được ở Bước 3 có *CV* bằng số lượng các luật nhạy cảm ban đầu thì dừng, ngược lại, lặp lại Bước 2, Bước 3, Bước 4 để tìm các *item* nhạy cảm khác.

4. VÍ DỤ MINH HỌA

Cho một cơ sở dữ liệu gốc bao gồm 10 giao dịch (được thể hiện trong Bảng 1). Với *MST* = 10% và *MCT* = 70%. Giả sử tập các luật nhạy cảm cần được ẩn được trình bày trong Bảng 2.

Bảng 1. Tập dữ liệu ban đầu

TID	Items	TID	Items
T1	a, b, c, e, f	T6	b, c, e
T2	e	T7	a, b, c, d, e, f
T3	b, c, e, f	T8	a, b
T4	d, f	T9	c, e, f
T5	a, b, d, f	T10	a, b, c, e

Bảng 2. Các luật nhạy cảm

ID	SAR
1	c, d→ f, a
2	a, e, d → c, f
3	e → b

Đề xuất của chúng tôi được tiến hành như sau:

- *Bước 1*: Tính toán *MSC* và *MCCC* cho từng luật nhạy cảm (Bảng 3);

Bảng 3. *MSC* và *MCCC* cho từng luật nhạy cảm

ID	SAR	MSC	MCCC
1	c, d→ f, a	1	1.3
2	a, e, d → c, f	1	1.3
3	e → b	5	1.1

- *Bước 2*: Xây dựng bảng các "*lược đồ nhạy cảm*" (Bảng 4): Bộ các luật nhạy cảm bao gồm *{c, d → f, a; a, e, d → c, f; e → b}*; Sau đó ta rút gọn các lược đồ dư thừa. Kết quả thể hiện trong Bảng 5;

Bảng 4. Các lược đồ nhạy cảm

ID	SI	CV	AM
1	f	2	Max (1, 1) = 1.0
2	a	1	Max (1) = 1.0
3	c	1	Max (1) = 1.0
4	f	2	Max (1, 1) = 1.0
5	b	1	Max (1.1) = 1.1

Bảng 5. Lược đồ nhạy cảm sau khi tinh chỉnh

ID	SI	CV	AM
1	f	2	Max (1, 1) = 1.0
2	a	1	Max (1) = 1.0
3	c	1	Max (1) = 1.0
4	b	1	Max (1.1) = 1.1

- *Bước 3*: Xác định lược đồ có *CV* lớn nhất là lược đồ tốt nhất: Bảng 5 cho thấy *Lược đồ 1* có *CV* = 2 là lớn nhất, như vậy *Lược đồ 1* là tốt nhất;

- *Bước 4*: Tập hợp các *item* nhạy cảm *S = {1} = {f}*: Do *Lược đồ 1* có *CV = 2* bé hơn số lượng các luật nhạy cảm ban đầu nên bộ các luật nhạy cảm hiện hành là *{e → b}* có số lượng = số lượng các luật nhạy cảm ban đầu trừ đi *CV* của *Lược đồ 1* (*3 - 2 = 1*). Lặp lại Bước 2 với bộ các luật nhạy cảm hiện hành là *{e → b}* có số lượng là 1 (Bảng 6);

Bảng 6. Các lược đồ nhạy cảm

ID	SI	CV	AM
1	b	1	Max (1.1) = 1.1

- *Bước 5*: Bảng 6 cho thấy chỉ còn một lược đồ và lược đồ này là lược đồ tốt nhất;

- *Bước 6*: Tập hợp các *item* nhạy cảm *S = {f, b}*.

Thuật toán dừng khi bộ các luật nhạy cảm hiện hành là rỗng ({∅}). Từ đó ta có kết luận là tập hợp các *item* nhạy cảm *S = {f, b}*. Trong khi đó với thuật toán ban đầu

chỉ xác định được *b* là *item* nhạy cảm. Điều này không đảm bảo ẩn hết ba luật nhạy cảm đã chỉ định trước đó.

5. ĐÁNH GIÁ HIỆU SUẤT

5.1. Tập dữ liệu

Các tập dữ liệu được sử dụng để làm thực nghiệm bao gồm các tập dữ liệu thực lấy từ UCI (2018): *Mushroom; Chess*; và *Cylinder Bands*. Đặc điểm của các bộ dữ liệu cụ thể được trình bày trong Bảng 7.

Bảng 7. Đặc tính của các tập dữ liệu

Cơ sở dữ liệu	Số lượng giao dịch	Chiều dài trung bình giao dịch	Số lượng *item*
Mushroom	8124	23	119
Chess	3196	37	75
Cylinder Bands	512	39	1756

5.2. Các độ đo hiệu suất

Các tiêu chí quan trọng sẽ được sử dụng để so sánh *đề xuất cải tiến* so với thuật toán ban đầu COA4ARH bao gồm:

- *Hiding Failure (HF)*: Cho biết số lượng các luật nhạy cảm mà thuật toán làm sạch không thể ẩn và vẫn đang được khai thác từ cơ sở dữ liệu đã làm sạch *D'*. Công thức tính toán như sau:

$$HF = \frac{|R_S(D')|}{|R_S(D)|} \tag{7}$$

Trong đó: $R_s(D')$ là số lượng luật nhạy cảm tìm thấy trong cơ sở dữ liệu làm sạch; *D'* và $R_S(D)$ là số lượng luật nhạy cảm trong cơ sở dữ liệu gốc ban đầu *D*.

- *Lost Rules (LR)*: Cho biết số lượng *các luật không nhạy cảm* bị mất do hoạt động thanh trùng *sanitization* và sẽ không còn được khai thác từ tập dữ liệu đã thanh trùng *D'*. Công thức tính toán là:

$$LR = \frac{|\sim R_S(D)| - |\sim R_S(D')|}{|\sim R_S(D)|} \tag{8}$$

Trong đó: $|\sim R_s(D)|$ là số lượng các luật không nhạy cảm trong tập dữ liệu ban đầu *D*; $|\sim R_s(D')|$ là số lượng các luật không nhạy cảm trong tập dữ liệu đã làm sạch *D'*.

- *Ghost Rules (GR)*: Cho biết số lượng *các luật giả* không có trong cơ sở dữ

liệu gốc ban đầu D và được tạo ra do hoạt động làm sạch *sanitization* và được khai thác từ cơ sở dữ liệu D'. Công thức tính toán:

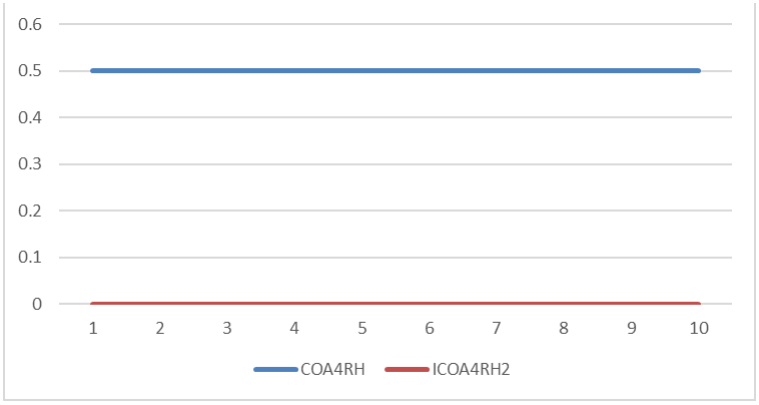
$$GR = \frac{|R'| - |R \cap R'|}{|R'|} \quad (9)$$

Trong đó: $|R'|$ là số lượng các luật khai thác từ D'; $|R|$ là số lượng các luật khai thác từ D.

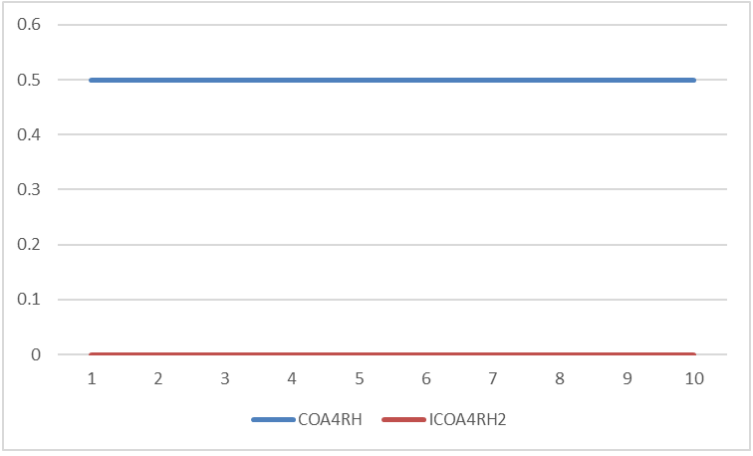
- *Số vòng lặp*: Số lần lặp yêu cầu để đạt được giải pháp tối ưu.

5.3. Các kết quả thực nghiệm

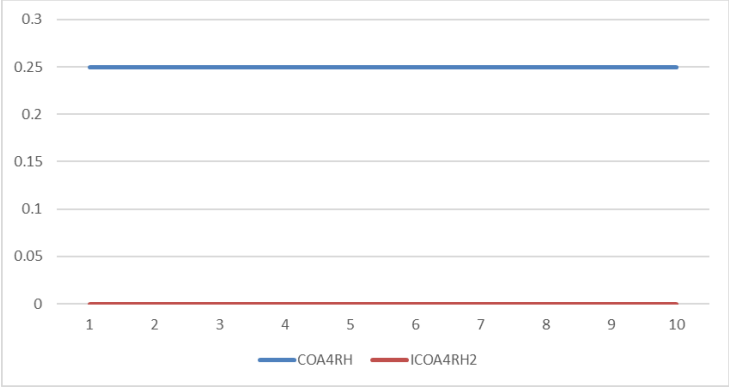
Các thí nghiệm ở cả hai thuật toán đều được tiến hành trên cùng một nền tảng là ngôn ngữ Java, thực hiện trên một máy tính PC với cấu hình @Intel CPU core i7 2.50 GHz và RAM 16GB, Windows 10 (64-bit). Qua một số kiểm thử cụ thể, chúng tôi thu được các kết quả như trong Hình 2, Hình 3, và Hình 4.



Hình 2. So sánh *HF* trên tập dữ liệu *Mushroom*



Hình 3. So sánh *HF* trên tập dữ liệu *Chess*



Hình 4. So sánh *HF* trên tập dữ liệu *Cylinder Bands*

Các hình trên cho thấy trong cả ba tập cơ sở dữ liệu, giá trị *HF* trong phương pháp cải tiến đều bằng 0, trong khi thuật toán ban đầu thì *HF* khác 0. Điều này nói lên rằng trong một số trường hợp thì thuật toán ban đầu không thể ẩn hết các luật nhạy cảm, còn phương pháp của chúng tôi thì khắc phục được nhược điểm này một cách triệt để. Có được hiệu quả này là do đề xuất cải tiến của chúng tôi tìm thấy được mối tương quan giữa luật nhạy cảm và các giao dịch hỗ trợ của nó, từ đó đưa ra phương pháp tính toán số lượng nhỏ nhất các *item* nhạy cảm để vừa đảm bảo ẩn hết các luật nhạy cảm và vừa gây ảnh hưởng thấp nhất lên các giao dịch hỗ trợ.

6. KẾT LUẬN

Bảo đảm sự riêng tư trong khai thác luật kết hợp là một chủ đề nghiên cứu quan trọng trong lĩnh vực bảo mật cơ sở dữ liệu. Bài báo này đã đề xuất một phương pháp cải tiến khá hiệu quả để giải quyết vấn đề ẩn hoàn toàn các luật kết hợp nhạy cảm bị nhập nhằng mà trước đó thuật toán gốc không xử lý được. Đề xuất cải tiến đã đề ra phương pháp tính toán chính xác số lượng các *item* nhạy cảm đảm bảo phủ hết các luật nhạy cảm nhưng cũng ít gây ảnh hưởng nhất đến các giao dịch dữ liệu. Các kết quả thực nghiệm trên các tập dữ liệu thực đã chứng tỏ hiệu quả của phương pháp cải tiến để ẩn hết các luật nhạy cảm. Trong tương lai, chúng tôi sẽ cố gắng tìm ra một cơ chế thích nghi cho các tham số đầu vào của thuật toán. Ngoài ra, tìm hiểu thêm về các cách tiếp cận khác (biên, chính xác) để từ đó cải tiến thuật toán tốt hơn trong trường hợp mất các luật không nhạy cảm cũng như đẩy nhanh tốc độ xử lý của thuật toán nhằm đạt được giải pháp tối ưu nhanh hơn.

TÀI LIỆU THAM KHẢO

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. *SIGMOD Record, 29*(2), 439-450.

Atallah, M., Bertino, E., Elmagarmid, A., Ibrahim, M., & Verykios, V. (1999). *Disclosure limitation of sensitive rules*. Paper presented at The IEEE Knowledge and Data Engineering Exchange Workshop (KDEX), USA.

Chang, L., & Moskowitz, I. (1998). *Parsimonious downgrading and decision trees applied to the inference problem*. Paper presented at The Workshop on New Security Paradigms (NSPW), USA.

Lindell, Y., & Pinkas, B. (2000). Privacy-preserving data mining. *Journal of Cryptology, 15*(3), 36-54.

Mahtab, H. A., Mohammad, N. D., & Mehdi, A. (2016). Association rule hiding using Cuckoo optimization algorithm. *Expert Systems with Applications, 64*, 340-351.

Oliveira, S., & Zaïane, O. (2004). *Achieving privacy preservation when sharing data for clustering*. Paper presented at The International Conference on Data Mining (SDM), Canada.

UCI. (2018). *Machine learning repository*. Retrieved from https://archive.ics.uci.edu/ml/index.php

Walton, S., Hassan, O., Morgan, K., & Brown, M. (2011). Modified Cuckoo search: A new gradient-free optimisation algorithm. *Chaos, Solitons & Fractals, 44*, 710-718.

Wu, Y. H., Chiang, C. M., & Arbee, L. P. C. (2007). Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data Engineering, 19*(1), 29-42.