

MỤC LỤC

Hệ thống giám sát nhiệt độ tự động cho lợn bằng cảm biến thẻ tai tích hợp công nghệ bluetooth và học máy.	61
<i>Dương Thành Nam, Nguyễn Mạnh Sơn, Trần Thị Hoa, Trần Sơn Tùng, Nguyễn Hoàng Giang, Nguyễn Thị Kim Phương, Nguyễn Văn Huy</i>	
Phòng chống phương tiện bay không người lái dựa trên chiếm quyền điều khiển giao thức: Một số vấn đề kỹ thuật và giải pháp minh họa.	71
<i>Bùi Công Thành, Phùng Đức Dũng, Đồng Xuân Chinh</i>	
Phát hiện lỗi hỏng phần mềm nhờ học sâu.	79
<i>Bùi Văn Công, Vũ Thảo Nguyên, Vũ Đức Minh, Nguyễn Phương Lan</i>	
Áp dụng mạng CNN trong phân loại quang phổ Raman của dung dịch glucose tinh khiết với bộ dữ liệu tăng cường.	89
<i>Nguyễn Thanh Tùng, Ngô Quang Trí</i>	
Chiến lược vùng đệm thích nghi cho khai thác mẫu tuần tự lợi ích cao đều đặn trên dữ liệu tăng trưởng.	96
<i>Trần Minh Thái</i>	
Khai thác hiệu quả các luật kết hợp súc tích có lợi ích và tương quan cao.	113
<i>Trần Thống, Hoàng Minh Tiến, Dương Văn Hải, Trương Chí Tín</i>	

THÔNG TIN DÀNH CHO TÁC GIẢ

Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ Thông tin và Truyền thông - gọi tắt là Công nghệ TTTT, phiên bản tiếng Anh là ICT Research - là chuyên san khoa học của Tạp chí Thông tin và Truyền thông nay là Tạp chí Khoa học và Công nghệ Việt Nam, do Bộ Thông tin và Truyền thông và nay là Bộ Khoa học và Công nghệ xuất bản từ năm 1999.

Chuyên san Công nghệ TTTT có phản biện, mở trực tuyến (open access) và được phát hành hàng năm với Ấn phẩm tiếng Việt (ISSN: 1859-3526) có 2 số và Ấn phẩm tiếng Anh (ISSN: 1859-3534) - Research and Development on Information and Communication Technology (ICT Research) - có 2 số.

Chuyên san Công nghệ TTTT là một diễn đàn dành cho các nhà nghiên cứu và các chuyên gia phổ biến các ý tưởng mới và sáng tạo trong các lĩnh vực của Công nghệ Thông tin, Truyền thông và Điện tử ở Việt Nam và trên toàn thế giới. Hội đồng biên tập của Chuyên san CNTT-TT bao gồm các nhà khoa học uy tín đến từ các đơn vị nghiên cứu ở Việt Nam và các quốc gia khác.

HƯỚNG DẪN DÀNH CHO CÁC TÁC GIẢ

Các tác giả nên tìm hiểu cách viết bài báo kỹ thuật, chẳng hạn như các hướng dẫn trong tài liệu “How to write for technical periodicals & conferences” của Viện Kỹ sư Điện, Điện tử Hoa Kỳ (IEEE).

Chuyên san Công nghệ TTTT chỉ nhận các bản thảo được nộp trực tuyến trên trang web của Chuyên san Công nghệ TTTT tại <https://ictmag.vn/cntt-tt/>. Tác giả cần chế bản bản thảo theo mẫu mà Chuyên san Công nghệ TTTT hướng dẫn.

Các tác giả phải chịu trách nhiệm trong việc đảm bảo rằng bản thảo này chưa từng được công bố trước đó, cũng như hiện đang không trong quá trình phản biện hoặc chuẩn bị xuất bản ở nơi khác. Tác giả liên hệ cũng phải chịu trách nhiệm đảm bảo sự đồng ý của tất cả các đồng tác giả khi nộp bài cho Chuyên san Công nghệ TTTT. Bản thảo cần được nộp dưới dạng PDF. Để hỗ trợ phản biện kín hai chiều, tác giả cần nộp hai phiên bản của bản thảo, một phiên bản có đầy đủ thông tin của tất cả các tác giả, một không có thông tin gì.

Các bản thảo nộp lại sau khi chỉnh sửa cũng cần tuân theo các chỉ dẫn như của bản thảo nộp lần đầu, nhưng cần kèm theo một văn bản giải trình các ý kiến của phản biện, trong đó giải thích các nội dung được chỉnh sửa và/hoặc trả lời các ý kiến của các phản biện. Sau khi một bản thảo được chỉnh thức chấp nhận xuất bản, các tác giả cần phải chuẩn bị bản thảo cuối cùng bằng LATEX hay WORD theo mẫu trình bày cho IEEE Transactions.

ĐẠO ĐỨC TRONG XUẤT BẢN

Chuyên san Công nghệ TTTT cam kết tuân thủ các quy định, tiêu chuẩn và hướng dẫn được thiết lập bởi COPE (Committee on Publication Ethics) trong việc đảm bảo tính liêm chính của tạp chí cũng như xử lý các trường hợp vi phạm. Tất cả các bên tham gia (Biên tập viên, Tác giả, Phản biện và Nhà xuất bản) cần phải hiểu và tuân thủ các quy định, tiêu chuẩn và hướng dẫn được công bố trên trang web của COPE.

Ban biên tập của Chuyên san Công nghệ TTTT sẽ sàng lọc tất cả các bản thảo trước khi công bố. Tất cả các bản thảo sẽ được kiểm tra về mức độ tương tự với các công trình khác sử dụng các công cụ trực tuyến có được. Mọi hình thức đạo văn không được chấp nhận và bản thảo sẽ bị từ chối.

Mỗi vòng phản biện có thể cần khoảng hai tháng. Trong trường hợp các tác giả được Biên tập viên yêu cầu chỉnh sửa bản thảo, phiên bản được chỉnh sửa cần được nộp trong vòng hai tháng đối với yêu cầu chỉnh sửa lớn và một tháng với yêu cầu chỉnh sửa nhỏ. Tác giả nào cần nhiều thời gian hơn xin vui lòng liên hệ với Biên tập viên.

Mỗi bài báo được chấp nhận sẽ được xuất bản trực tuyến ngay sau đó, với số DOI, sau khi được biên tập và chế bản theo các quy tắc biên tập của Chuyên san Công nghệ TTTT. Đồng thời, khi có đủ một số lượng bài phù hợp, Chuyên san Công nghệ TTTT sẽ tập hợp lại và xuất bản một số in, kèm với mục lục các bài báo.

THÔNG TIN VỀ BẢN QUYỀN

Khi bài báo được chấp nhận, bản quyền của bài báo sẽ tự động được chuyển giao cho Tạp chí Khoa học và Công nghệ Việt Nam, là tổ chức chịu trách nhiệm xuất bản Chuyên san Công nghệ TTTT, với đảm bảo rằng mọi nội dung của bài báo được chấp nhận xuất bản sẽ được công bố miễn phí trên trang web của Chuyên san Công nghệ TTTT. Việc chuyển giao này cho phép bên xuất bản có toàn quyền giải quyết bất kỳ khiếu nại nào về việc sử dụng trái phép. Các tác giả có quyền đăng tải lại hay sử dụng lại bài báo được xuất bản dưới bất kỳ hình thức nào cho bất kỳ mục đích nào, miễn là có trích dẫn đầy đủ tới bài báo gốc xuất bản trên Chuyên san Công nghệ TTTT. Việc nộp bài báo được xem như là các tác giả đã đọc và đồng ý với chính sách bản quyền này.

Hệ thống giám sát nhiệt độ tự động cho lợn bằng cảm biến thẻ tai tích hợp công nghệ bluetooth và học máy

Dương Thành Nam^{1*}, Nguyễn Mạnh Sơn¹, Trần Thị Hoa², Trần Sơn Tùng¹,
Nguyễn Hoàng Giang², Nguyễn Thị Kim Phương², Nguyễn Văn Huy²

¹Trung tâm Nghiên cứu và Phát triển công nghệ cao, Viện Hàn lâm Khoa học và Công nghệ Việt Nam,
18 Hoàng Quốc Việt, phường Nghĩa Đô, Hà Nội, Việt Nam

²Viện Kiểm định Công nghệ và Môi trường, Liên hiệp Các hội Khoa học và Kỹ thuật Việt Nam, phường Long Biên, Hà Nội, Việt Nam

Ngày nhận bài 8/10/2024; ngày chuyển phân biện 10/10/2024; ngày nhận phân biện 22/11/2024; ngày chấp nhận đăng 28/11/2025

Tóm tắt:

Hệ thống giám sát nhiệt độ tự động cho lợn, tích hợp cảm biến thẻ tai kết hợp với công nghệ Bluetooth Low Energy (BLE) và các mô hình học máy, mang lại một giải pháp hiện đại và hiệu quả trong việc theo dõi sức khỏe đàn lợn. Các cảm biến được gắn trực tiếp lên thẻ tai có khả năng đo nhiệt độ cơ thể của từng cá thể liên tục, đồng thời thu thập thông tin về điều kiện môi trường như độ ẩm và nhiệt độ chuồng trại. Dữ liệu được truyền về trung tâm xử lý thông qua kết nối BLE và được phân tích trên nền tảng điện toán đám mây. Hệ thống sử dụng các mô hình học máy tiên tiến như PCA-SVC, PCA-Decision tree, PCA-Random forest và 1D-CNN để phát hiện sớm các dấu hiệu bất thường. Trong số đó, mô hình 1D-CNN cho kết quả nổi bật với độ chính xác lên tới 96,6%, chứng minh hiệu quả vượt trội trong việc xử lý dữ liệu chuỗi và phát hiện các vấn đề sức khỏe. Việc ứng dụng cảm biến thông minh không chỉ nâng cao độ chính xác, giảm chi phí nhân công, mà còn tối ưu hóa khả năng giám sát theo thời gian thực, góp phần quan trọng trong phòng ngừa dịch bệnh và nâng cao hiệu quả quản lý chăn nuôi.

Từ khóa: Bluetooth Low Energy, cảm biến thẻ tai, giám sát nhiệt độ lợn, học máy, 1D-CNN.

Automatic temperature monitoring system for pigs using ear tag sensors integrated with bluetooth and machine learning

Thanh Nam Duong^{1*}, Manh Son Nguyen¹, Thi Hoa Tran², Son Tung Tran¹,
Hoang Giang Nguyen², Thi Kim Phuong Nguyen², Van Huy Nguyen²

¹Center for Advanced Science and Technology Research, Vietnam Academy of Science and Technology,
18 Hoang Quoc Viet Street, Nghia Do Ward, Hanoi, Vietnam

²Institute of Environment, Technology and Verification, Vietnam Union of Science and Technology Associations, Long Bien Ward, Hanoi, Vietnam

Received 8 October 2024; revised 22 November 2024; accepted 28 November 2025

Abstract:

The automatic temperature monitoring system for pigs, integrated with ear tag sensors combined with Bluetooth Low Energy (BLE) technology and machine learning models, offers a modern and efficient solution for monitoring the health of pig herds. The sensors are directly attached to ear tags and are capable of continuously measuring each pig's body temperature, while also collecting environmental information such as humidity and ambient temperature inside the housing facility.

*Tác giả liên hệ: Email: namdt.vast@gmail.com

Data is transmitted to a central server via BLE and analysed on a cloud computing platform. The system employs advanced machine learning models such as PCA-SVC, PCA-Decision Tree, PCA-Random Forest, and 1D-CNN to enable early detection of abnormal signs. Among these models, 1D-CNN demonstrates outstanding performance, achieving an accuracy of 96.58%, proving its effectiveness in processing time-series data and identifying health issues. The application of smart sensor technology not only enhances monitoring accuracy and reduces labor costs but also optimises real-time surveillance capabilities. This contributes significantly to disease prevention and improves overall livestock management efficiency.

Keywords: Bluetooth Low Energy, ear tag sensor, machine learning, temperature monitoring, 1D-CNN.

1. Đặt vấn đề

Chăn nuôi lợn là một ngành truyền thống quan trọng, đóng góp lớn vào an ninh lương thực và kinh tế nông nghiệp, đặc biệt tại Việt Nam. Việc giám sát và cảnh báo sớm các vấn đề sức khỏe của lợn là yếu tố then chốt để ngăn chặn dịch bệnh và bảo vệ hệ thống chăn nuôi. Trên thế giới, nhiều nước phát triển đã áp dụng công nghệ tiên tiến trong giám sát và quản lý sức khỏe vật nuôi như: nhận dạng sinh trắc học, phân tích hành vi và robot tự động hóa trong quá trình chăn nuôi. Tuy nhiên, tại Việt Nam, công nghệ hiện đại vẫn chưa được áp dụng rộng rãi trong giám sát dịch bệnh.

Nhiệt độ cơ thể là chỉ số quan trọng để chẩn đoán sức khỏe lợn [1]. Nhiệt độ cơ thể là một chỉ số sinh lý quan trọng, nhiệt độ tăng lên khi xuất hiện các triệu chứng bệnh lý. Phương pháp đo truyền thống qua trực tràng gây căng thẳng và nguy cơ lây nhiễm chéo, trong khi hệ thống camera giám sát hiện tại yêu cầu nhân lực và không đáp ứng được việc giám sát liên tục [2, 3].

Xuất phát từ thực tế và các vấn đề nêu trên, nghiên cứu này đã tiến hành phát triển “Hệ thống giám sát nhiệt độ tự động cho lợn bằng cảm biến thẻ tai tích hợp công nghệ bluetooth và học máy” cho phép theo dõi liên tục, phát hiện sớm bất thường, từ đó cảnh báo kịp thời và nâng cao hiệu quả quản lý chăn nuôi. Ngoài ra, nghiên cứu này làm nổi bật tính mới trong việc áp dụng mạng nơ-ron tích chập

một chiều (1D-CNN) để phân tích và dự đoán các bất thường về nhiệt độ của lợn, một phương pháp chưa được sử dụng phổ biến trong lĩnh vực này. So với các mô hình học máy truyền thống như PCA-SVC, PCA-Decision tree hay PCA-Random forest, mô hình 1D-CNN cho thấy khả năng học các đặc trưng phức tạp từ dữ liệu chuỗi thời gian, mang lại độ chính xác vượt trội trong việc phát hiện sớm các bất thường nhiệt độ, từ đó giúp cải thiện hiệu quả giám sát sức khỏe lợn.

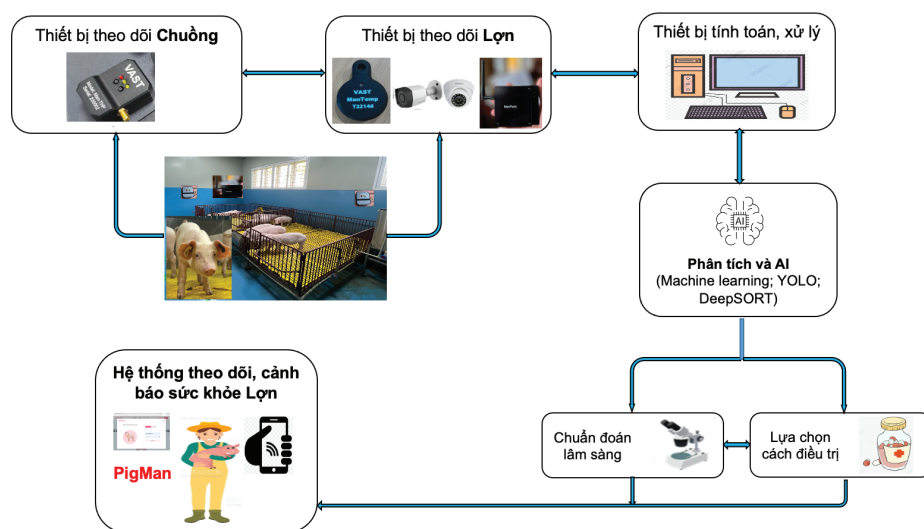
2. Thiết bị và phương pháp nghiên cứu

2.1. Hệ thống giám sát nhiệt độ bất thường ở lợn

Hệ thống giám sát nhiệt độ bất thường của lợn (hình 1) gồm: cảm biến thẻ tai (ManTemp), thiết bị đo điều kiện môi trường chuồng trại (Man-THP), camera giám sát hành vi, bộ nhận và truyền dữ liệu (ManPoint) và Cloud server (pigman.com).

Các cảm biến thẻ tai ManTemp được gắn vào tai lợn để đo và ghi nhận nhiệt độ cơ thể. Dữ liệu từ cảm biến ManTemp được truyền đến hệ thống thu nhận dữ liệu (ManPoint), sau đó chuyển đến máy chủ đám mây để xử lý và phân tích.

Thiết bị Man-THP chịu trách nhiệm giám sát các điều kiện môi trường chuồng trại thông qua các phép đo độ ẩm, nhiệt độ và áp suất. Hệ thống giúp phát hiện sự thay đổi môi trường có thể ảnh hưởng đến nhiệt độ cơ thể của lợn. Đồng thời, hệ thống camera giám sát được lắp đặt để ghi lại và theo dõi hoạt



Hình 1. Hệ thống giám sát hành vi và thân nhiệt.

động của lợn liên tục 24 giờ. Hình ảnh được lưu trữ và phân loại theo mã ID lợn và vị trí camera, sau đó gửi đến máy chủ đám mây pigman.com để phân tích.

Hệ thống tích hợp các thành phần này có khả năng sử dụng thuật toán phân tích dữ liệu để phát hiện sớm các bất thường về nhiệt độ cơ thể lợn hoặc môi trường sống. Thông tin thu thập được giúp người quản lý áp dụng kịp thời các biện pháp cần thiết nhằm đảm bảo sức khỏe và an toàn cho đàn lợn.

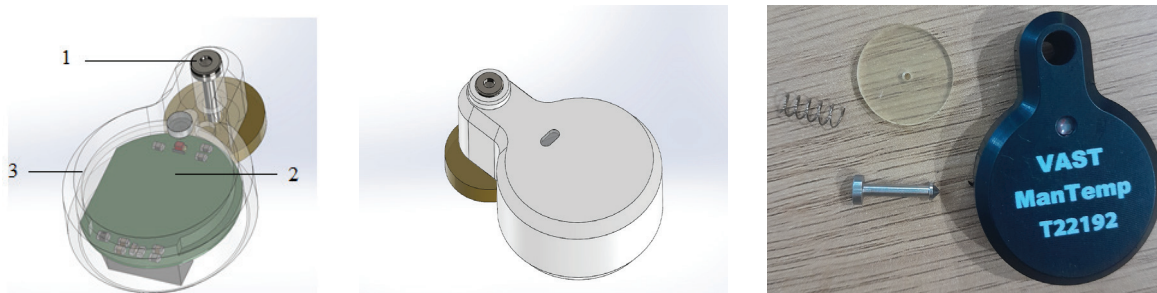
2.2. Cảm biến nhiệt độ thể tại ManTemp

Cảm biến nhiệt độ thể tại ManTemp được thiết kế với 3 thành phần chính: chốt khóa cố định có đệm lò xo; bo mạch điện tử với cảm biến nhiệt độ và phần thân vỏ (hình 2).

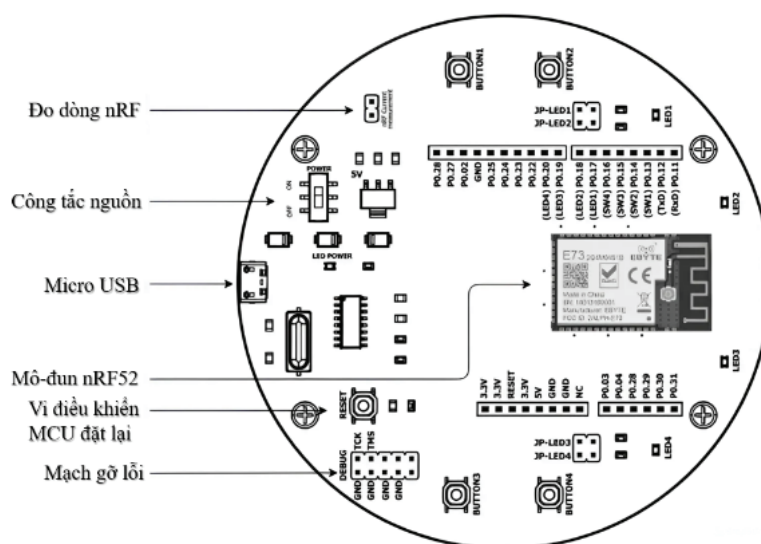
2.2.1. Bo mạch điện tử và cảm biến nhiệt độ

Bo mạch điện tử sử dụng công nghệ iBeacon và BLE nRF52832 (hình 3), cho phép kết nối và truyền dữ liệu nhiệt độ cùng với mã định danh của lợn đến hệ thống trung gian (ManPoint) và sau đó tới máy chủ đám mây. Bộ vi xử lý MCU (Microcontroller unit) được tích hợp trên bo mạch giúp tính toán nhiệt độ liên tục, phục vụ việc giám sát sức khỏe lợn theo thời gian thực.

Một cụm cảm biến (được hàn trên bảng mạch in tùy chỉnh PCB) được gắn vào thẻ đeo tai cho lợn để ghi lại thông số nhiệt độ và bên ngoài có định danh riêng cho từng con để phân biệt. Bộ xử lý trung tâm có thể thu thập và xử lý dữ liệu (ở đây là nhiệt độ của từng con lợn), từ đó xuất ra biểu đồ theo dõi nhiệt độ theo thời gian.



Hình 2. Sơ đồ cấu tạo ManTemp. 1: Chốt khóa cố định có đệm lò xo; 2: Bo mạch điện tử với cảm biến nhiệt độ; 3: Thân vỏ.



Hình 3. Thiết kế phần cứng nRF52832.

Mạch thu phát BLE Bluetooth 4.0 SoC nRF52832 với bộ vi xử lý ARM Cortex-M4 64 MHz hỗ trợ FPU, dung lượng bộ nhớ RAM và Flash tăng gấp đôi, hỗ trợ BLE 5.0, với ưu điểm vượt trội về khoảng cách và thông lượng.

Với thiết kế linh hoạt, bo mạch hỗ trợ nhiều tùy chọn cấp nguồn như cổng Micro USB 5 V, jack đo dòng nRF 3,3 V và pin CR2032 3 V. Cảm biến nhiệt độ NTC-MF52 tích hợp sẵn cho phép đo nhiệt độ trong khoảng $(-55 \div 125^{\circ}\text{C})$ với độ chính xác $\pm 0,5^{\circ}\text{C}$, đảm bảo dữ liệu đo được ổn định và chất lượng.

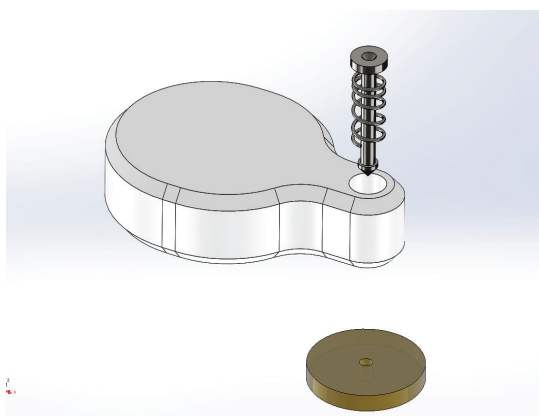
2.2.2. Thân vỏ và chốt khoá cố định

Phần thân vỏ của ManTemp được chế tạo từ nhựa tiêu chuẩn, bao gồm hai mặt trên và dưới tách rời, với các cạnh bo xung quanh để bảo vệ cảm biến.

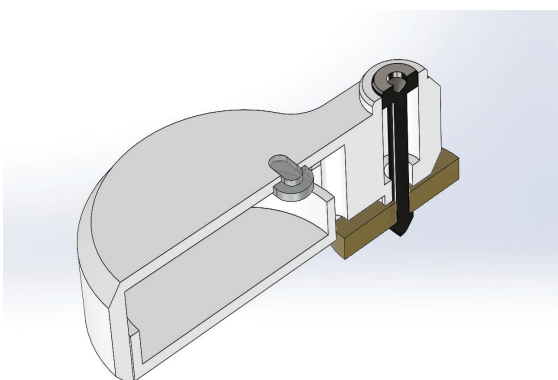
Thân vỏ được thiết kế một vị trí hạ bậc giúp cảm biến nhiệt tiếp xúc với da lợn, đảm bảo độ chính xác khi đo. Ngoài ra, có một vùng dành cho lò xo và chốt khóa (hình 4).

Chốt khóa bao gồm đỉnh chốt bằng vật liệu chống gỉ, với một đầu lõm dùng cho kim bấm và một đầu nhọn xuyên qua đệm cao su để giữ thiết bị cố định chắc chắn. Lò xo xoắn quanh đỉnh chốt tăng độ đàn hồi, cho phép thiết bị thích ứng với sự phát triển của lợn và đảm bảo phép đo liên tục chính xác.

ManTemp được thiết kế nhỏ gọn, thuận tiện trong sử dụng, giúp ngăn chặn phản ứng căng thẳng ở lợn. Phép đo nhiệt độ cố định tăng độ chính xác so với các phương pháp không tiếp xúc, giảm nguy cơ lây nhiễm. Thiết bị có khả năng truyền dữ liệu thời



Hình 4. Sơ đồ cấu tạo vỏ ManTemp.



gian thực tới các thiết bị di động hoặc hệ thống giám sát từ xa, nhờ kết hợp cảm biến nhiệt độ, MCU, các mô-đun cung cấp điện, lưu trữ dữ liệu và truyền dẫn.

2.3. Mô hình học máy

Nghiên cứu sử dụng bốn mô hình học máy để dự đoán sức khỏe lợn dựa trên dữ liệu nhiệt độ: PCA-SVC, PCA-Decision tree (PCA - cây quyết định), PCA-Random forest (PCA - rừng ngẫu nhiên) và 1D-CNN. Các kỹ thuật phân tích thành phần (Component analysis - CA) giúp giảm chiều dữ liệu, trong khi SVC, Decision tree và Random forest là các thuật toán phân loại mạnh mẽ. PCA-SVC và PCA-Decision tree sử dụng PCA để giảm số lượng đặc trưng, giúp các mô hình phân loại hoạt động hiệu quả hơn, nhưng có thể mất thông tin quan trọng trong quá trình giảm chiều. PCA-Random forest kết hợp với nhiều cây quyết định để cải thiện độ chính xác và giảm overfitting. Mô hình 1D-CNN, tuy không dùng PCA, nhưng lại mạnh mẽ trong việc xử lý dữ liệu chuỗi như tín hiệu thời gian và âm thanh nhờ khả năng trích xuất đặc trưng phức tạp từ dữ liệu chuỗi.

Trong đánh giá các mô hình học máy, đặc biệt là trong các bài toán phân loại, có một số chỉ số quan trọng được sử dụng để đo lường hiệu suất của mô hình. Những chỉ số này không chỉ cung cấp thông tin về khả năng dự đoán đúng hay sai của mô hình mà còn giúp người dùng hiểu rõ hơn về khả năng xử lý dữ liệu của nó, đặc biệt trong những tình huống có sự chênh lệch lớn giữa các lớp dữ liệu. Các chỉ số phổ biến bao gồm Độ chính xác (Accuracy), Độ chính xác dương tính (Precision), Độ phủ dương tính (Recall), và F1-Score. Mỗi chỉ số có một ý nghĩa riêng và được áp dụng tùy thuộc vào đặc thù của bài toán cũng như yêu cầu cụ thể của người sử dụng.

Accuracy (độ chính xác): Tỷ lệ dự đoán chính xác của mô hình trên tổng số dữ liệu.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision (độ chính xác dương tính): Tỷ lệ số lượng dương tính (tức là dự đoán bệnh) được phân loại chính xác trên tổng số các trường hợp được phân loại là dương tính.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall (độ phủ dương tính): Tỷ lệ số lượng dương tính được phân loại chính xác trên tổng số các trường hợp thực tế là dương tính.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

trong đó: TP: True positive (Dự đoán đúng dương tính); TN: True negative (Dự đoán đúng âm tính); FP: False positive (Dự đoán sai dương tính); FN: False negative (Dự đoán sai âm tính).

3. Kết quả và bàn luận

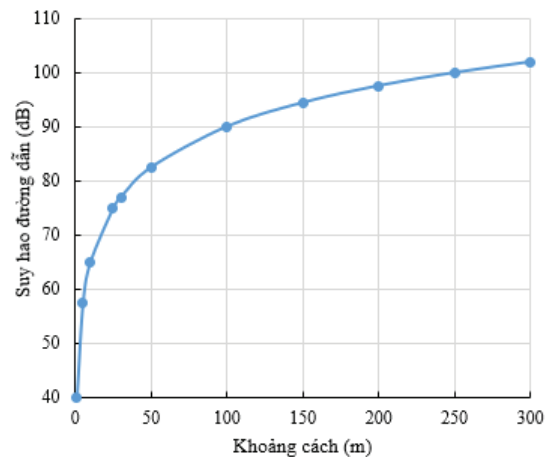
3.1. Thử nghiệm trong phòng thí nghiệm

3.1.1. Hiệu suất của Bluetooth Low Energy

Phạm vi hoạt động của mô-đun BLE phụ thuộc vào loại ăng-ten và suy hao đường dẫn (hình 5), được tính theo phương trình (4) dưới đây [4]:

$$\text{Suy hao đường dẫn} = 40 + 25 \times \log(d) \quad (4)$$

trong đó: d là khoảng cách giữa máy phát và máy thu. Độ nhạy của máy thu BLE nRF52832 là -70 dBm, với công suất phát từ -18 đến +8 dBm. Ví dụ, với công suất phát +8 dBm và độ nhạy -70 dBm, suy hao đường dẫn đạt +78 dB, tương ứng với khoảng cách giao tiếp lý thuyết là ~33 m.



Hình 5. Biểu đồ biểu diễn suy hao đường dẫn thu được từ phương trình (4).

Nghiên cứu đã thử nghiệm 10 cảm biến ManTemp ở nhiều khoảng cách và tốc độ lấy mẫu khác nhau để tối ưu hóa thời lượng pin. Với tốc độ lấy mẫu 5 lần/giây, mô-đun bluetooth tiêu thụ dòng điện 3 mA, trong khi cảm biến nhiệt độ tiêu thụ 1,5 mA.

Sử dụng pin CR2030 với dung lượng 230 mAh, thời lượng pin ước tính là 51 giờ.

Để kéo dài tuổi thọ pin, ManTemp được cấu hình để chỉ đo nhiệt độ khi phát hiện bất thường kèm theo tín hiệu đèn nháy sáng. Điều này giúp giảm tần suất đo không cần thiết, tập trung vào các thời điểm có biến đổi nhiệt độ, qua đó tiết kiệm năng lượng và tăng thời gian sử dụng thiết bị.

Việc kết hợp đèn nháy sáng và phát hiện nhiệt độ bất thường giúp tối ưu hóa hiệu suất và tiết kiệm pin, phù hợp cho các ứng dụng yêu cầu theo dõi nhiệt độ một cách thông minh và tiết kiệm năng lượng.

3.1.2. Đặc tính nhiệt độ và đánh giá hiệu chuẩn của thiết bị ManTemp

ManTemp là thiết bị đo nhiệt độ cơ thể lợn, hoạt động trong phạm vi nhiệt độ $-5\div 125^{\circ}\text{C}$. Để đảm bảo tính chính xác và ổn định, việc hiệu chuẩn thiết bị trước khi sử dụng là bắt buộc. Quy trình hiệu chuẩn được thực hiện tại ba điểm nhiệt độ đặc trưng (35 , 40 và 45°C), trong điều kiện môi trường ổn định với nhiệt độ $25\pm 2^{\circ}\text{C}$ và độ ẩm tương đối $50\div 60\%$. Quá trình hiệu chuẩn dựa trên so sánh trực tiếp giữa giá trị đo của ManTemp và tổ hợp thiết bị chuẩn (Rotronic).



Hình 6. Hệ thống giám sát nhiệt độ trại lợn (ManTemp, Man-THP, ManPoint).



Hình 7. Cảm biến thẻ tai (ManTemp) được bấm vào tai trái của lợn.

Bảng 1. Độ không đảm bảo đo nhiệt độ ($k=2$, $U=95\%$).

Điểm chuẩn (°C)	Giá trị trung bình của chuẩn (°C)	Giá trị trung bình của đối tượng (°C)	Số hiệu chỉnh (°C)	Độ không đảm bảo đo mở rộng (°C)
35	34,900	34,70	0,20	1,0
40	39,910	39,80	0,11	1,0
45	44,910	45,10	-0,19	1,0

Kết quả hiệu chuẩn cho thấy, sai số nhỏ ở tất cả các điểm: tại 35°C, sai số là 0,20°C; tại 40°C, sai số 0,11°C; tại 45°C, sai số -0,19°C. Độ không đảm bảo đo mở rộng ở các điểm là 1,0°C, nằm trong giới hạn cho phép.

Kết quả từ các điểm kiểm tra cho thấy, ManTemp hoạt động ổn định trong dải nhiệt độ được yêu cầu. Độ không đảm bảo đo mở rộng tại tất cả các điểm là 1,0°C, phù hợp với tiêu chuẩn của thiết bị và đảm bảo tính chính xác khi đo nhiệt độ cơ thể lợn (bảng 1). So với các thiết bị tương tự, ManTemp thể hiện độ chính xác cao và ổn định, đặc biệt là ở các điểm nhiệt độ quan trọng trong quy trình theo dõi sức khỏe động vật.

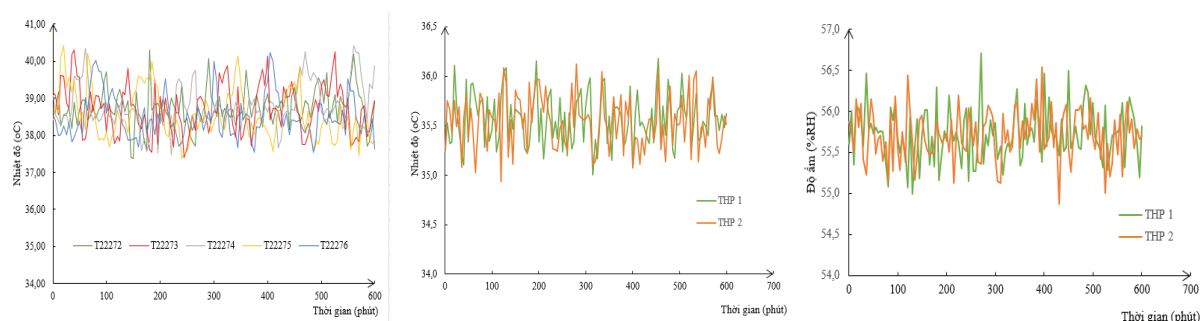
Qua quá trình hiệu chuẩn, thiết bị ManTemp đã chứng minh được khả năng hoạt động đáng tin cậy trong việc đo lường nhiệt độ. Với sai số nhỏ và độ không đảm bảo đo mở rộng nằm trong giới hạn tiêu chuẩn, ManTemp đáp ứng tốt các yêu cầu kỹ thuật của một thiết bị giám sát nhiệt độ trong chăn nuôi. Mặc dù có những sai số nhỏ ở một số điểm nhiệt độ, nhưng những sai số này đều nằm trong phạm vi chấp nhận được, không ảnh hưởng lớn đến kết quả đo lường và đánh giá sức khỏe động vật.

Việc so sánh với các thiết bị đo nhiệt độ khác cho thấy, ManTemp có lợi thế về tính ổn định và khả năng duy trì độ chính xác trong suốt quá trình vận hành. Điều này khiến ManTemp trở thành một giải pháp hiệu quả và tin cậy trong lĩnh vực theo dõi nhiệt độ trong chăn nuôi lợn, đặc biệt khi so sánh với các phương pháp đo nhiệt độ không tiếp xúc khác.

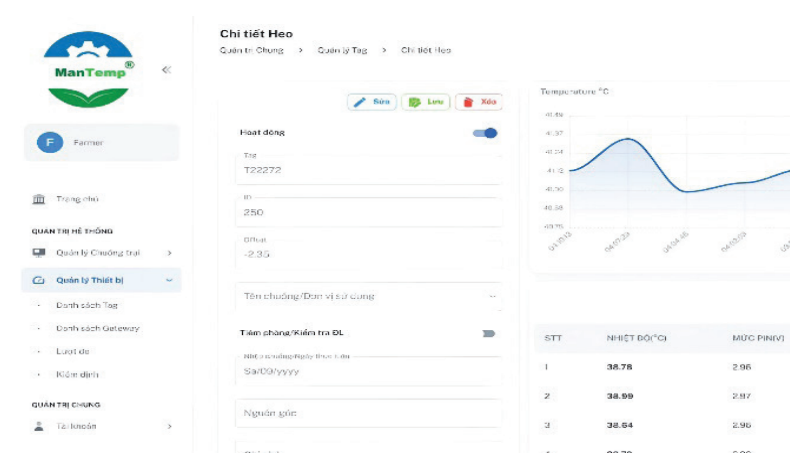
3.2. Thử nghiệm trên cá thể lợn

Trong thử nghiệm thực địa, hệ thống cảm biến thể tại ManTemp được kiểm tra trên 5 cá thể lợn có trọng lượng trung bình 24 kg, theo dõi liên tục trong 600 phút để đánh giá hiệu quả và tính ổn định trong điều kiện chuồng trại thực tế (hình 6). Cảm biến được gắn vào tai trái của lợn (hình 7), không gây ra phản ứng căng thẳng, đảm bảo an toàn và không ảnh hưởng đến hành vi hay sức khỏe của lợn. Thân nhiệt bình thường của lợn khỏe mạnh dao động trong khoảng 38,5-39,5°C. Tuy nhiên, trong suốt quá trình theo dõi, nhiệt độ cơ thể của lợn có thể dao động 37,5-40,5°C do các hoạt động sinh lý như ăn uống, di chuyển và tắm rửa. Dữ liệu thu thập từ ManTemp cho thấy, khả năng giám sát nhiệt độ cơ thể gần với thời gian thực, đạt độ chính xác cao, từ đó giúp phát hiện sớm các dấu hiệu bất thường về sức khỏe của lợn.

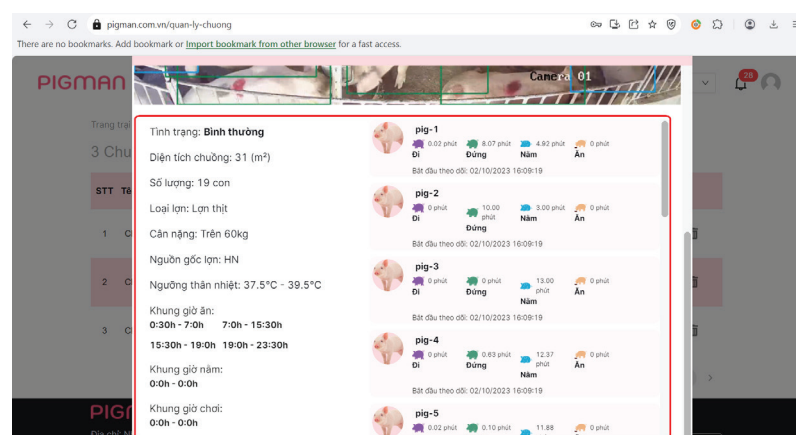
Hệ thống cảm biến ManTemp đã được thử nghiệm trong điều kiện thực tế tại chuồng trại, theo dõi nhiệt độ cơ thể lợn trong khoảng cách 25-30 m, với độ chính xác cao và khả năng giám sát gần thời gian thực. Hệ thống phát hiện và cảnh báo sớm các dấu hiệu bất thường về sức khỏe thông qua sự thay đổi nhiệt độ. Đồng thời, hai cảm biến môi trường (Man-THP) đặt tại các vị trí khác nhau trong chuồng trại ghi nhận nhiệt độ trung bình 35,5°C và độ ẩm trung bình 55,7%RH, với độ lệch chuẩn thấp, cho thấy tính ổn định cao. Hệ thống ManTemp kết hợp với Man-THP cung cấp dữ liệu toàn diện, giúp giám sát hiệu quả môi trường sống của lợn và phát hiện sớm các vấn đề sức khỏe tiềm ẩn (hình 8).



Hình 8. Dữ liệu thu thập từ 5 thiết bị ManTemp và 2 thiết bị Man-THP.



Hình 9. Dữ liệu ManTemp ID T22272 được xuất ra từ phần mềm ManTemp.



Hình 10. Hệ thống theo dõi nhiệt độ ManTemp.

So với các hệ thống giám sát môi trường khác trong chăn nuôi, hệ thống ManTemp kết hợp với các cảm biến môi trường Man-THP mang lại sự ổn định và độ chính xác cao hơn. Việc đặt các cảm biến ở nhiều vị trí khác nhau giúp theo dõi và phát hiện những thay đổi trong môi trường một cách toàn diện hơn, đảm bảo rằng dữ liệu thu thập không chỉ phản ánh một phần của chuồng trại mà còn bao quát toàn

bộ môi trường sống của lợn. Điều này giúp tăng cường khả năng dự báo và phát hiện sớm các vấn đề tiềm ẩn, liên quan đến sức khỏe của lợn, đặc biệt là các thay đổi đột ngột về nhiệt độ hoặc độ ẩm.

Hệ thống ManTemp đã chứng minh khả năng hoạt động ổn định và cung cấp dữ liệu có giá trị trong việc giám sát sức khỏe lợn. Cảm biến ManTemp theo dõi nhiệt độ cơ thể lợn một cách chính xác mà không ảnh hưởng đến hành vi tự nhiên của chúng, điều này rất phù hợp cho ứng dụng trong thực tế chăn nuôi. So với phương pháp thủ công hay dùng camera, ManTemp tự động hóa quá trình thu thập và xử lý dữ liệu, giảm thiểu chi phí và công sức, đồng thời cung cấp khả năng giám sát từ xa và gần như trong thời gian thực, giúp phát hiện sớm các bất thường về sức khỏe.

Hệ thống vượt trội hơn so với các giải pháp hiện tại về độ chính xác, độ ổn định và khả năng phát hiện sớm. Bên cạnh theo dõi nhiệt độ, ManTemp còn tích hợp các cảm biến môi trường để giám sát các yếu tố như độ ẩm, giúp cải thiện hiệu quả chăm sóc và nuôi dưỡng. Thử nghiệm cho thấy ManTemp có độ chính xác cao và hỗ trợ ngăn ngừa dịch bệnh qua việc phát hiện sớm các triệu chứng bất thường.

Thiết bị ManTemp được gắn trực tiếp lên lợn, ghi nhận liên tục nhiệt độ và truyền dữ liệu lên hệ thống để phân tích (hình 9). Thuật toán phát hiện sự thay

đổi nhiệt độ bất thường, cảnh báo kịp thời về nguy cơ bệnh dịch, cho phép người quản lý và chuyên gia thú y can thiệp nhanh chóng. Dữ liệu này cũng được lưu trữ để tạo lịch sử sức khỏe, giúp người quản lý hiểu rõ hơn về yếu tố ảnh hưởng đến sức khỏe của đàn lợn, từ môi trường nuôi đến dinh dưỡng và chăm sóc (hình 10).

3.3. Dự đoán sự bất thường nhiệt độ bằng mô hình học máy

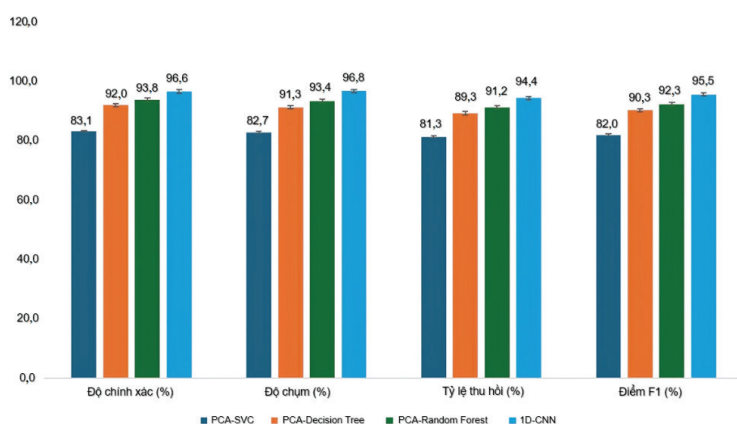
Để tăng cường khả năng phát hiện sớm các bất thường về nhiệt độ ở lợn, nghiên cứu đã sử dụng bốn mô hình học máy khác nhau: PCA-SVC, PCA-Decision tree, PCA-Random forest và 1D-CNN. Mục tiêu của việc áp dụng các mô hình này là phân tích và dự đoán các trạng thái nhiệt độ thông thường (BT) và bất thường (BBT) từ việc thu thập nhiệt độ dữ liệu được thông qua hệ thống cảm biến thể tại ManTemp.

Quá trình chuẩn bị dữ liệu cho hệ thống giám sát thân nhiệt lợn PigMan bao gồm việc lựa chọn và xử lý dữ liệu từ 1500 thẻ tại ManTemp của cả lợn bình thường và bất thường. Dữ liệu được phân loại dựa trên ngưỡng nhiệt độ: lợn có nhiệt độ $\leq 40^{\circ}\text{C}$ (TN) được coi là bình thường, trong khi lợn có nhiệt độ $> 40^{\circ}\text{C}$ (TP) được coi là bất thường. Chúng tôi đã sử dụng các mô hình học máy, bao gồm; Phân tích thành phần chính kết hợp với PCA-SVC, PCA-Decision tree, PCA-Random forest và 1D-CNN để đánh giá hiệu suất dự đoán trên cùng một bộ dữ liệu. Kết quả dự đoán lợn bất thường từ các mô hình thu được tại bảng 2.

Dữ liệu trên phản ánh kết quả phân loại của bốn mô hình học máy, bao gồm PCA-SVC, PCA-Decision tree, PCA-Random forest và 1D-CNN, dựa trên các chỉ số True positive (TP), True negative (TN), False positive (FP) và False negative (FN). Mô hình PCA-SVC có kết quả kém nhất với 694 trường hợp dự đoán đúng lợn bất thường và 806 trường hợp

Bảng 2. Kết quả dự đoán lợn bất thường từ các mô hình.

Mô hình	TP	TN	FP	FN
PCA-SVC	694	806	145	160
PCA-Decision tree	607	893	58	73
PCA-Random forest	591	909	42	57
1D-CNN	568	932	19	34



Hình 11. Biểu đồ đánh giá mô hình dự đoán bất thường dựa trên cảm biến nhiệt độ.

đúng lợn bình thường, nhưng lại có số lượng dự đoán sai khá cao với 145 sai dương tính (FP) và 160 sai âm tính (FN). Trong khi đó, PCA-Decision tree cho thấy sự cải thiện rõ rệt, với số lượng sai dương tính và âm tính giảm đáng kể, chỉ còn 58 FP và 73 FN, cùng với độ chính xác tốt hơn trong phân loại. PCA-Random forest thể hiện hiệu suất cao với 929 trường hợp TN và 571 TP, đồng thời chỉ có 22 FP và 37 FN, cho thấy khả năng phân loại rất đáng tin cậy. Cuối cùng, 1D-CNN là mô hình có hiệu suất tốt nhất, với chỉ 19 FP và 34 FN, phản ánh khả năng phân loại chính xác và ổn định nhất trong số các mô hình được so sánh. Điều này cho thấy, 1D-CNN là lựa chọn tối ưu khi cần giảm thiểu sai sót trong dự đoán bất thường nhiệt độ của lợn. Cụ thể, kết quả đánh giá mô hình dự đoán sự bất thường của lợn dựa trên dữ liệu cảm biến nhiệt độ theo thời gian bằng các thuật toán học máy (hình 11).

Mô hình PCA-SVC đạt độ chính xác 83,1%, với Precision 82,7% và Recall 81,3%. Mặc dù có khả năng phân loại khá, mô hình này vẫn bỏ sót một số trường hợp bất thường do việc giảm chiều dữ liệu bằng PCA có thể làm mất thông tin quan trọng.

Mô hình PCA-Decision tree có độ chính xác tăng lên 92,0%, với Precision 91,3% và Recall 89,3%. Mô hình này cải thiện khả năng phân loại nhờ cấu trúc phân nhánh của cây quyết định, nhưng vẫn có nguy cơ overfitting khi xử lý dữ liệu phức tạp.

Mô hình PCA-Random forest với độ chính xác 93,8%, Precision 93,4% và Recall 91,2%, mô hình này cho thấy hiệu suất tốt hơn nhờ việc tổng hợp kết quả từ nhiều cây quyết định, giảm thiểu overfitting và tăng khả năng tổng quát hóa.

Mô hình 1D-CNN đạt hiệu suất cao nhất với độ chính xác 96,6%, Precision và Recall đều đạt 96,8 và 94,4%. 1D-CNN có khả năng trích xuất các đặc trưng phức tạp từ dữ liệu chuỗi thời gian, giúp phát hiện chính xác các bất thường nhiệt độ.

Mối tương quan giữa các chỉ tiêu cho thấy 1D-CNN vượt trội hơn các mô hình khác trong việc cân bằng giữa độ chính xác và khả năng phát hiện đầy đủ các bất thường, khẳng định tiềm năng lớn của việc ứng dụng mạng nơ-ron tích chập trong giám sát sức khỏe lợn.

Kết quả nghiên cứu phù hợp với xu hướng trong lĩnh vực ứng dụng học sâu vào giám sát sức khỏe vật nuôi. Cụ thể, việc sử dụng mạng nơ-ron tích chập để phân tích dữ liệu cảm biến đã được chứng minh là hiệu quả trong nhiều nghiên cứu trước đây. Ví dụ, nghiên cứu của B. Savoini và cs (2025) [5] đã sử dụng 1D-CNN để phân tích dữ liệu và phát hiện sớm bệnh ở gia súc, đạt độ chính xác 90%. Kết quả 96,6% của nghiên cứu này cao hơn, cho thấy tiềm năng lớn của việc áp dụng 1D-CNN trong giám sát sức khỏe lợn. Ngoài ra, nghiên cứu của Y. Kim và cs (2019) [6] đã áp dụng Random forest để phân loại tình trạng sức khỏe ở lợn dựa trên dữ liệu hành vi, đạt độ chính xác 92%. Kết quả 94,5% của nghiên cứu này với PCA-Random forest cho thấy, việc kết hợp giảm chiều dữ liệu bằng PCA có thể cải thiện hiệu suất mô hình.

Các kết quả cho thấy, mô hình 1D-CNN có hiệu suất vượt trội trong việc dự đoán sự bất thường nhiệt độ ở lợn, nhờ khả năng học các đặc trưng phức tạp từ dữ liệu chuỗi thời gian. Việc phát hiện sớm các bất thường nhiệt độ là rất quan trọng để ngăn chặn dịch bệnh và đảm bảo sức khỏe đàn lợn. So với các mô hình truyền thống, 1D-CNN không chỉ cải thiện độ chính xác mà còn đạt sự cân bằng tốt giữa Precision và Recall.

4. Kết luận

Nghiên cứu này đã phát triển và thử nghiệm hệ thống giám sát nhiệt độ tự động ManTemp dành cho chăn nuôi lợn, sử dụng cảm biến thể tai và kết nối Bluetooth sóng ngắn (BLE). Kết quả cho thấy, hệ thống ManTemp có khả năng đo lường và truyền tải dữ liệu nhiệt độ chính xác, hiệu quả trong việc phát hiện các bất thường về nhiệt độ cơ thể của lợn.

Các thuật toán học máy được áp dụng, bao gồm PCA-SVC, PCA-Decision tree, PCA-Random forest và 1D-CNN, đã được đánh giá dựa trên các chỉ tiêu độ chính xác, precision, recall và F1-score. Trong đó, 1D-CNN đã chứng minh là mô hình hiệu quả nhất với độ chính xác 96,6%, vượt trội so với các mô hình khác. Điều này khẳng định tiềm năng của 1D-CNN trong việc xử lý dữ liệu chuỗi thời gian, giúp phát hiện sớm các bất thường và các dấu hiệu bệnh tật.

Hệ thống ManTemp không chỉ là một bước tiến trong việc ứng dụng công nghệ vào chăn nuôi mà còn mở ra nhiều cơ hội mới để xây dựng một nền nông nghiệp thông minh. Việc kết hợp giữa công nghệ cảm biến và các thuật toán học máy trong giám sát sức khỏe lợn, mở ra hướng đi mới trong việc áp dụng công nghệ số vào nông nghiệp. Các nghiên cứu tiếp theo có thể mở rộng phạm vi ứng dụng sang các loại vật nuôi khác và tiếp tục tối ưu hóa các mô hình học máy để nâng cao hiệu suất dự đoán.

LỜI CẢM ƠN

Nghiên cứu này được hỗ trợ bởi Viện Hàn lâm Khoa học và Công nghệ Việt Nam trong đề tài mã số CN4000.01/21-23. Các tác giả xin chân thành cảm ơn.

TÀI LIỆU THAM KHẢO

- [1] L. Wang, R. Madera, Y. Li, et al. (2020), "Recent advances in the diagnosis of classical swine fever and future perspectives", *Pathogens*, **9**(8), DOI: 10.3390/pathogens9080658.
- [2] M. Stewart, J.R. Webster, G.A. Verkerk, et al. (2007), "Non-invasive measurement of stress in dairy cows using infrared thermography", *Physiol. Behav.*, **92**(3), pp.520-525, DOI: 10.1016/j.physbeh.2007.04.034.
- [3] K.A. Herborn, J.L. Graves, P. Jerem, et al. (2015), "Skin temperature reveals the intensity of acute stress", *Physiol. Behav.*, **152**, pp.225-230, DOI: 10.1016/j.physbeh.2015.09.032.
- [4] R. Heydon (2012), *Bluetooth Low Energy - The Developer's Handbook*, Prentice Hall, 368pp.
- [5] B. Savoini, J. Bertolaccini, S. Montavon, et al. (2025), "Convolutional neural network for early detection of lameness and irregularity in horses using an IMU sensor", *arXiv*, DOI: 10.48550/arXiv.2503.13578.
- [6] Y. Kim, S. Lee, D. Park (2019), "Health status classification of pigs using random forest algorithm", *Animal Bioscience*, **32**(9), pp.1377-1383.

Phòng chống phương tiện bay không người lái dựa trên chiếm quyền điều khiển giao thức: Một số vấn đề kỹ thuật và giải pháp minh họa

Bùi Công Thành*, Phùng Đức Dũng, Đồng Xuân Chinh

Viện Nghiên cứu 486, Bộ Tư lệnh 86, phường Xuân Đình, Hà Nội, Việt Nam

Ngày nhận bài 8/11/2024; ngày gửi phản biện 10/11/2024; ngày nhận phản biện 5/12/2024; ngày chấp nhận đăng 12/12/2024

Tóm tắt:

Phương tiện bay không người lái (UAV) ngày càng đóng vai trò quan trọng trong cả lĩnh vực dân sự và quân sự nhờ tính linh hoạt, chi phí vận hành thấp và khả năng thực hiện nhiệm vụ kéo dài. Tuy nhiên, sự phổ biến rộng rãi này cũng đặt ra nhiều thách thức về an ninh, đặc biệt trong việc quản lý và ngăn chặn các hoạt động UAV trái phép. Trong số các kỹ thuật phòng chống UAV hiện nay, phương pháp chiếm quyền điều khiển bằng cách khai thác lỗ hổng trong giao thức điều khiển đang được đánh giá là hướng tiếp cận tiềm năng và tiết kiệm chi phí, đặc biệt phù hợp với mục tiêu nghiên cứu và ứng dụng trong quốc phòng. Bài báo trình bày một số kết quả nghiên cứu liên quan đến phương án phòng chống UAV thông qua chiếm quyền điều khiển giao thức MAVLink - một giao thức truyền thông nhẹ được sử dụng rộng rãi cho việc điều khiển và truyền thông tin UAV. Kết quả cho thấy, do MAVLink 1.0 thiếu mã hóa và xác thực, kẻ tấn công có thể dễ dàng giành quyền điều khiển UAV. Mô phỏng này góp phần hỗ trợ đào tạo, nghiên cứu trong điều kiện thiếu trang thiết bị thực tế, đồng thời mở ra các hướng nghiên cứu chiến lược phòng chống UAV chủ động và hiệu quả hơn trong tương lai.

Từ khóa: MAVLink, phòng chống phương tiện bay không người lái, phương tiện bay không người lái.

Research on control takeover techniques for counter-UAV (Unmanned Aircraft Vehicle)

Cong Thanh Bui*, Duc Dung Phung, Xuan Chinh Dong

Institute 486, Command 86, Xuan Dinh Ward, Hanoi, Vietnam

Received 8 November 2024; revised 5 December 2024; accepted 12 December 2024

Abstract:

Unmanned aerial vehicles (UAVs) have become increasingly integral in both civilian and military domains due to their flexibility, cost-effectiveness, and mission endurance. However, this widespread adoption introduces complex security challenges, especially in terms of managing and preventing unauthorized UAV activities. Among the emerging counter-UAV techniques, hijacking UAVs by exploiting control protocol vulnerabilities is considered a promising and cost-efficient approach, particularly for research and defense applications. This paper presents technical findings related to hijacking-based countermeasures against UAVs by targeting the MAVLink protocol - a widely adopted lightweight communication protocol for UAV command and telemetry. The research outlines

*Tác giả liên hệ: Email: phungducung@gmail.com

the architecture of UAV control systems, identifies protocol vulnerabilities and demonstrates how spoofed control packets can lead to full UAV takeover. The results show that, due to the lack of encryption and authentication in MAVLink 1.0, an attacker can successfully disrupt and override mission commands, effectively hijacking UAVs. This virtual simulation provides a valuable resource for education, training, and further research on UAV cybersecurity, particularly in environments where access to real UAV systems and field conditions is limited. The study emphasizes the importance of protocol security in UAV systems and opens new directions for proactive defense strategies.

Keywords: Counter-UAV, MAVLink, Unmanned aerial vehicle.

1. Đặt vấn đề

Phương tiện bay không người lái là loại phương tiện hoạt động không cần người lái, được chế tạo dựa trên nhiều công nghệ trong một hệ thống chỉnh thể, tinh vi cả về phần cứng, cơ khí, phần mềm và chiến thuật. Nhờ những ưu điểm như khả năng cơ động độc lập, hoạt động trong điều kiện khắc nghiệt, thời gian dài, chi phí thấp và khả năng che giấu tốt, UAV ngày càng trở nên phổ biến trong mọi lĩnh vực đời sống, xã hội, bao gồm cả nhà máy, nông nghiệp... và quốc phòng, an ninh. Từ đó cũng tạo ra nhiều vấn đề mới về quản lý, phòng chống UAV để đảm bảo quốc phòng, an ninh [1].

Trong quân đội của các nước phát triển, UAV là lực lượng được biên chế chính thức và phổ biến như hầu hết các hệ thống vũ khí thông thường [2, 3].

Trong thập kỷ qua, chúng ta đã chứng kiến sự đầu tư quyết liệt của các nước cho UAV. Năm 2012, Hoa Kỳ chi gần 5 tỷ USD dành cho nghiên cứu, phát triển và mua sắm UAV chiến lược. Tổng thống Nga Putin công bố khoản ngân sách 12,2 tỷ USD đầu tư cho phát triển UAV trong khoảng thời gian 2013-2020; tính đến năm 2010, đã có 42 chính phủ nước ngoài mua UAV của Israel [3]. Theo Bộ Quốc phòng Hoa Kỳ, tính từ năm 2001 đến năm 2023, hơn 23 triệu giờ bay đã được ghi nhận cho các UAV của quân đội Hoa Kỳ, riêng trong năm 2023 đã có hơn 2 triệu giờ bay [4].

Sự đầu tư cho nghiên cứu, phát triển việc quản lý, kiểm soát mà gọi chung là phòng chống UAV cũng được đặc chú trọng, đặc biệt là trong Quân đội. Tính đến 3/2024 trong cuộc xung đột Nga - Ukraine,

Nga đã kiểm soát, tiêu diệt hơn 16 ngàn chiếc UAV; số liệu này từ phía Ukraine là hơn 8 ngàn [5]. Trong nước, một số các sản phẩm phòng chống UAV được biết đến như CA-18 do Học viện Kỹ thuật Quân sự sản xuất, DroneGun do Hãng Drone Shield sản xuất được sử dụng tại một số đơn vị an ninh, dân quân.

Các công nghệ chống UAV thường được áp dụng cho từng trường hợp, từng mục đích và từng loại UAV tương ứng để đạt hiệu quả cao. Với các cuộc xung đột gần đây, ta có thể thấy rằng, những phương thức như gây nhiễu, giả mạo hay tác động mật vật lý đều đã quá phổ dụng ở các nước lớn, chi phí lớn và thường hiệu quả chưa đạt mong muốn. Với việc đa số các UAV đang sử dụng nền tảng TCP/IP cho các kênh điều khiển, việc nghiên cứu phòng chống UAV dựa trên tấn công mạng điều khiển UAV đang được đánh giá là tiềm năng, giúp giảm thiểu kinh phí và tổn thất lực lượng.

Bên cạnh đó, trong điều kiện hạ tầng cho học tập, nghiên cứu UAV còn hết sức khó khăn, đặc biệt là lĩnh vực tác chiến không gian mạng vào UAV. Do các thách thức như: thiếu trang thiết bị, phương tiện cho nghiên cứu, điều kiện không gian cho thử nghiệm UAV, môi trường mạng liên quan đến UAV, quá trình thử nghiệm cần phải tổ chức cả đội hình nên tốn nhiều công sức... [1]. Do vậy, việc nghiên cứu mô phỏng kịch bản phòng chống UAV luôn được quan tâm nghiên cứu [1, 6], kết quả nghiên cứu giúp hỗ trợ cho công tác huấn luyện, nghiên cứu về phòng chống UAV.

Trong nội dung tiếp theo, phần 2 sẽ trình bày một số kiến thức có tính lý thuyết nền tảng liên quan đến nghiên cứu; phần 3 phân tích khả năng tấn công

chiếm quyền điều khiển đối với giao thức MAVLink; phương án mô phỏng kịch bản tấn công mạng vào UAV bằng các công cụ, phần mềm sẵn có được trình bày trong phần 4.

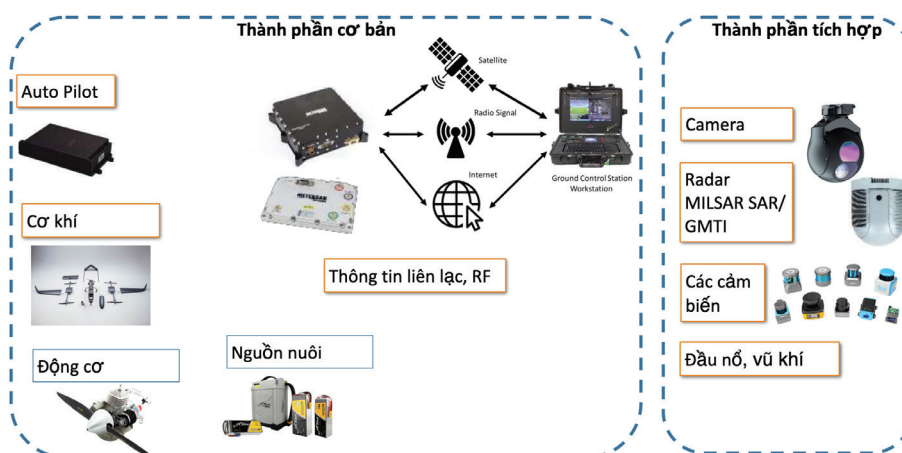
2. Cơ sở lý thuyết liên quan

2.1. Kiến trúc của phương tiện bay không người lái

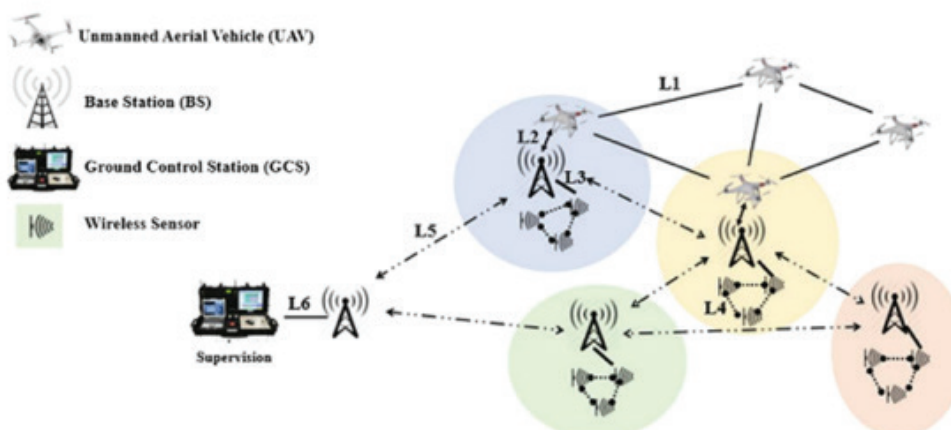
Một hệ thống UAV là tổ hợp nhiều thành phần công nghệ, các khối chính có thể chia làm hai nhóm: Khối thành phần cơ bản, là các thành phần cần thiết cho một UAV ở cả phía thiết bị bay và trạm điều khiển; Khối thành phần tích hợp, là thành phần có tính nâng cao theo từng mục tiêu cụ thể, bao gồm các loại chính như camera, cảm biến, hệ thống radar tích hợp, vũ khí. Mô tả từng thành phần cơ bản như trên hình 1.

Để đảm bảo hoạt động tự hành thông minh, các UAV thường bao gồm rất nhiều kênh thông tin như tại hình 2 [7], trong đó: L1 là liên kết liên lạc giữa các UAV dựa theo chuẩn thông dụng IEEE 802.11; L2 tương ứng là các giao tiếp giữa UAV và trạm phát sóng (BS); L3 là liên kết truyền thông giữa cảm biến không dây (WS) và BS; L4 liên lạc giữa các nút cảm biến không dây WS; L5 liên lạc giữa các BS; L6 liên lạc tương ứng giữa điều khiển qua trạm mặt đất (Ground control station - GCS) và BS.

Liên lạc từ xa với UAV qua sóng vô tuyến là xương sống trong việc điều khiển thành công thiết bị bay. Để liên lạc với UAV, thông tin phải được truyền liên tục từ UAV đến GCS, do đó liên kết hai chiều được thiết lập giữa chúng để đảm bảo gửi lệnh và thực hiện lệnh từ xa theo thời gian thực. Thông tin trả về chứa dữ liệu về vị trí địa lý, tốc độ bay,



Hình 1. Các thành phần liên quan đến phương tiện bay không người lái.



Hình 2. Các kênh liên lạc thông tin liên quan đến phương tiện bay không người lái [7].

các thông số đo được từ cảm biến về môi trường xung quanh... ghi lại trong suốt quá trình bay. Kênh thông tin điều khiển UAV được gửi lên từ GCS chứa các lệnh để điều khiển như hướng bay tiếp, tốc độ bay mới, lên cao hay xuống thấp... Đây cũng là đối tượng chính được bài báo tập trung trình bày, cụ thể hơn là ở lớp giao thức điều khiển UAV.

2.2. Tấn công mạng vào phương tiện bay không người lái

Dưới góc độ tấn công mạng, UAV có các thành phần chính có thể bị khai thác bởi tác chiến mạng như sau:

- Kênh định vị, dẫn đường: UAV có nhiều cơ chế để xác định phương hướng di chuyển như sử dụng hệ thống định vị GNSS, trạm điều khiển mặt đất bay theo bầy đàn, hay tự hành thông qua nhận dạng quang học. Do vậy, có thể tác chiến mạng đến vận hành, di chuyển của UAV [7].

- Kênh điều khiển hoạt động: thường bao gồm kênh điều khiển lệnh và kênh nhận trạng thái UAV. Kênh điều khiển đường lên để truyền lệnh điều khiển bay từ khối điều khiển từ xa đến UAV. Kênh trạng thái đường xuống để truyền thông tin trạng thái bay (vị trí, độ cao, lượng điện, tốc độ...) từ UAV về người điều khiển [7].

- Kênh truyền thông dữ liệu: Với mục tiêu trinh sát thông tin, UAV thu thập đa dạng thông tin, ở các kiểu dữ liệu khác nhau (hình ảnh, âm thanh, nhiệt độ, mức độ bức xạ RF...). Các dữ liệu này sẽ được truyền đến các trạm cố định, di động hay các UAV khác qua kênh thông tin dữ liệu; các UAV thương mại thường phải tuân thủ các chuẩn giao thức điều khiển, kết nối theo quy định. Do vậy, phòng chống UAV thông qua tác chiến mạng đến dữ liệu và các kênh thông tin liên lạc này là hoàn toàn có cơ sở [7].

- Khối điện tử và phần mềm UAV: Bản chất UAV có chứa các thành phần cho phép điều khiển bay, thông tin điều khiển, thông tin liên lạc. Các khối này bản chất là các bo mạch, firmware, phần mềm và hệ điều hành. Do vậy, tiềm ẩn các lỗ hổng bảo mật và có thể là mục tiêu khai thác tác chiến mạng. Ngoài ra, việc UAV thường có tài nguyên hạn chế để giảm tải trọng, năng lượng cũng dẫn đến dễ bị tấn công dạng từ chối dịch vụ.

- Trạm điều khiển mặt đất, trung tâm chỉ huy: Trạm mặt đất và trung tâm chỉ huy thường chứa cơ sở dữ liệu quản lý, điều hành và dữ liệu trinh sát của UAV. Các trạm mặt đất thường kết nối online/offline với các hệ thống mạng công nghệ thông tin chuyên dụng để trao đổi dữ liệu. Do vậy, việc tác chiến vào các trung tâm mặt đất này có thể giúp thu thập được các thông tin cơ mật, qua đó giúp phòng chống UAV hiệu quả.

2.3. Kênh điều khiển hoạt động UAV

Có hai kiểu cơ bản cho điều khiển UAV, điều khiển từ xa trực tiếp và GCS. Trong hệ thống điều khiển từ xa trực tiếp, người dùng quan sát UAV hoặc quan sát camera gắn trên UAV và điều khiển camera bằng bộ điều khiển. UAV được điều khiển trong thời gian thực bằng cách truyền mệnh lệnh qua kênh điều khiển hoạt động.

Cả bộ điều khiển và UAV được kết nối thông qua một mô-đun thực hiện giao tiếp giữa chúng với sự trợ giúp của các giao thức truyền thông. Thông thường là wifi, ZigBee và nhiều thiết bị mạng khác được sử dụng để liên lạc.

Đối với hệ thống trạm điều khiển mặt đất, sử dụng một máy tính để kết nối phần mềm với UAV, sau đó thực hiện lệnh nhiệm vụ do người dùng tải lên. Bằng cách thu thập thông tin từ nhiều cảm biến được cài đặt trên UAV, GCS có thể theo dõi trạng thái UAV, chẳng hạn như độ cao hiện tại, khoảng cách, vị trí bản đồ và trạng thái nhiệm vụ thực tế.

Các UAV được điều khiển hoạt động bay qua các kết nối vô tuyến với công nghệ như trải phổ nhảy tần (FHSS/DSSS) hệ thống dịch vụ dẫn đường toàn cầu vệ tinh (GNSS), wifi, bluetooth, 5G/LTE.

Thu thập và nhận dạng giao thức điều khiển của UAV là quá trình thu thập thông tin về các tín hiệu và giao thức được sử dụng để điều khiển máy bay không người lái, sau đó phân tích dữ liệu điều khiển, nhận dạng giao thức điều khiển để thực hiện các tấn công leo thang tiếp theo.

Bằng cách phân tích dữ liệu đã giải điều chế, ta có thể xác định các trường thông tin quan trọng trong gói tin, bao gồm địa chỉ nguồn, đích và các lệnh điều khiển gửi đến UAV. Sau khi đã hiểu rõ cấu trúc của gói tin và các thông tin quan trọng trong

đó, chúng ta có thể tiến hành giả mạo các gói tin điều khiển. Điều này có thể thực hiện bằng cách tạo ra các gói tin giả mạo có cấu trúc tương tự như gói tin gốc và truyền chúng đến UAV. Khi UAV nhận được các gói tin này, nó có thể bị đánh lừa và thực hiện các hành động không mong muốn, như thay đổi hướng di chuyển hoặc thậm chí là bị kẻ tấn công chiếm toàn quyền điều khiển.

Nghiên cứu này chỉ tập trung vào một số vấn đề kỹ thuật ở lớp giao thức điều khiển UAV, phân tích lỗ hổng giao thức, giả mạo gói tin, chiếm quyền điều khiển UAV và tập trung đề xuất, xây dựng môi trường ảo minh họa kịch bản tấn công mạng chiếm quyền điều khiển UAV.

3. Phân tích khả năng tấn công phương tiện bay không người lái thông qua giả mạo giao thức điều khiển MAVLink

MAVLink (Micro air vehicle Link) là một giao thức truyền thông rất nhẹ, được sử dụng để tương tác với các hệ thống không người lái hoặc các thiết bị khác trên bo mạch [6, 8]. Chẳng hạn, MAVLink cho phép người dùng tải lên các nhiệm vụ bay, lấy dữ liệu chuyển bay hoặc thay đổi các tham số của hệ thống không người lái. Giao thức này được thiết kế dựa trên mô hình kết hợp giữa việc xuất bản - theo dõi hoặc giao tiếp điểm - điểm. Các luồng dữ liệu được gửi hoặc xuất bản dưới dạng các chủ đề, trong khi các tiêu giao thức cấu hình như giao thức nhiệm vụ, hoặc giao thức tham số sử dụng giao tiếp điểm - điểm với việc truyền lại. Tính nhẹ của MAVLink xuất phát từ việc nó sử dụng ít dữ liệu thừa trên mỗi gói tin. MAVLink v1 chỉ có tiêu đề 8 byte cho mỗi thông điệp, trong khi MAVLink v2 có tiêu đề 14 byte nhưng cung cấp bảo mật và khả năng mở rộng tốt hơn.

Về bản chất, UAV sử dụng MAVLink để được giám sát và điều khiển. MAVLink hỗ trợ các loại lớp vận chuyển khác nhau nhờ cấu trúc gọn nhẹ. Nó có thể truyền qua wifi, Ethernet (mạng TCP/IP) hoặc các kênh băng thông thấp hoạt động ở tần số 433, 868 và 915 MHz [8]. Kết nối MAVLink được thiết lập như sau:

Khi UAV khởi động, nó truyền một thông điệp có tên là HEARTBEAT theo diện tin quảng bá hoặc có thể mang địa chỉ chỉ định của GCS. Đồng thời,

GCS lắng nghe bản tin HEARTBEAT trên mặt đất. Khi GCS phát hiện được bản tin của HEARTBEAT từ UAV gửi tới, nếu bản tin chứa địa chỉ của GCS mà nó cần truyền, GCS sẽ gửi lại bản tin HEARTBEAT cho UAV, sau đó cả 2 gửi liên tục bản tin này cho nhau mỗi giây để giữ cho liên lạc này tồn tại. Sau một khoảng thời gian kết nối được thiết lập hoàn toàn, UAV liên tục báo về trạng thái kết nối hiện tại để GCS có thể theo dõi tình trạng của UAV. Tần suất báo về thường khác nhau, phụ thuộc vào loại UAV, băng thông cho phép... GCS gửi bản tin MAVLink đến UAV được kết nối khi người điều khiển thực hiện gửi lệnh mới, cập nhật nhiệm vụ. Trình phân tích bản tin MAVLink được thực hiện sẽ không đề cập tới giá trị nào trong tiêu đề giao thức, ví dụ như địa chỉ IP đích nguồn... Do vậy, MAVLink dễ bị tấn công bằng cách tiêm bản tin sai nếu payload hợp lệ. Ngoài ra, GCS trao đổi dữ liệu với UAV thông qua một kênh không được xác thực và không có mã hóa. Những kết nối này có thể dễ dàng bị tấn công, bị khai thác bởi đối tượng có thiết bị phát phù hợp, có thể giao tiếp với máy bay không người lái, đưa lệnh vào phiên hiện có và dễ dàng khởi chạy tấn công vào UAV.

Sau khi đạt được thông tin về giao thức điều khiển của MAVLink, ta có thể tạo ra các tín hiệu giả (Spoofing Signal) để truyền đến UAV mục tiêu, nhằm khiến UAV hiểu nhầm rằng tín hiệu nhận được từ người tấn công chính là từ GCS gốc. Thuật toán thu thập và giải mã gói tin MAVLink như sau [9]:

Thuật toán: Thu thập và giải mã gói tin dạng bản rõ

Đầu vào: Gói tin thu thập được

Đầu ra: Bản rõ

Thủ tục (Khai thác Mavlink)

1 Tạo Đối Tượng Drone

2 Định Nghĩa Trình Xử Lý Tín Nhấn Mavlink

3 $D0 \leftarrow \text{đọc}(\text{Gói Dữ Liệu})$

4 Trong khi (KếtNối! = 0):

5 GóiTin $\leftarrow \text{Lấy}(\text{độ dài, địa chỉ, cổng})$

6 Phản Hồi (ThuViện_Mavlink) $\leftarrow \text{đọc}(\text{TínNhấn_Mavlink})$

7 Với (i $\leftarrow \text{Đọc}(\text{GóiTin})$):

8 $D0 \leftarrow \text{PhânTích}(\text{GóiTin}) + 0x00ff$ (khung bắt đầu STX bản v1.0)

9 $D1 \leftarrow \text{PhânTích}(\text{KýTự_Hex})$

10 $\text{ĐốiTượngDrone} \leftarrow \text{SẵnSàng}(D0, D1)$

11 nếu (GóiTin! = Null)

12 GiảiNén (GóiTin)

13 DữLiệu_ĐãNhận (DạngNhịPhân, Hex)

14 BảnRõ $\leftarrow \text{ChuyểnĐổi}(\text{DạngNhịPhân, Hex})$

15 ThôngTin_BáoMật $\leftarrow \text{Lấy}(\text{GPS, tài_trọng, độ_cao, kinh_độ, vĩ_độ, v.v.})$

16 Kết thúc thủ tục

Khi UAV thực hiện nhiệm vụ và bắt đầu trình bay, người tấn công gửi một bản tin MISSION_COUNT vừa gửi từ GCS tới UAV và khởi tạo gói tin nhiệm vụ cho UAV. UAV sẽ gửi MISSION_REQUEST tới GCS yêu cầu thông tin nhiệm vụ mà GCS đã gửi trước đó nên nó sẽ không truyền nữa. Vì thế, UAV chuyển sang trạng thái chờ thông tin nhiệm vụ tiếp theo. Khi đó, ta có thể dễ dàng chiếm quyền điều khiển UAV.

4. Mô phỏng tấn công phương tiện bay không người lái thông qua giả mạo giao thức điều khiển

4.1. Cài đặt môi trường thử nghiệm

- Thành phần UAV ảo: sử dụng nền tảng mở jmvmsim kết hợp PX4 (phần mềm điều khiển bay nguồn mở).
- Thành phần GCS: Sử dụng nền tảng mở QgroundControl, là một GCS cho phép kiểm soát và cấu hình chuyển bay đầy đủ cho các phương tiện PX4.
- Kẻ tấn công (Attacker): Sử dụng nền tảng mở GCS dạng dòng lệnh Python script của thư viện

Pymavlink, dùng MAVProxy (phần mềm mã nguồn mở đóng vai trò router) kết hợp các công cụ tấn công mạng có trong bộ Kali Linux.

Các thành phần khác nhau được giả lập trên các máy tính, kết nối nhau thông qua Wifi và sử dụng giao thức MAVLink trong điều khiển giữa GCS và UAV.

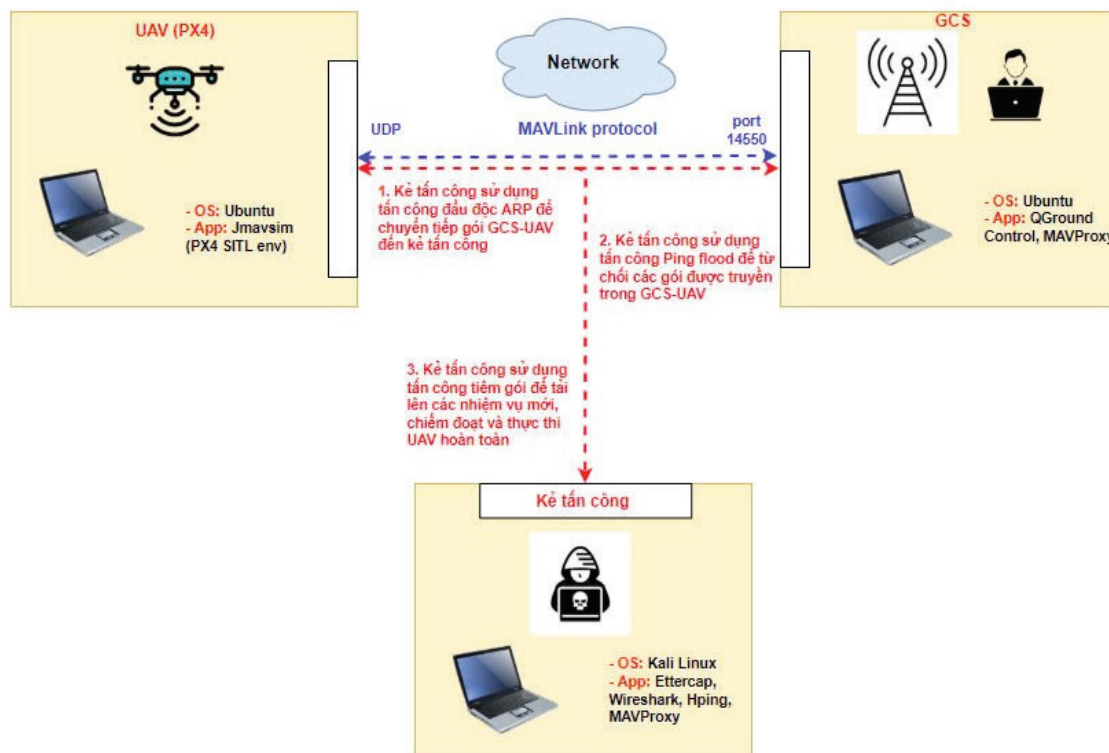
Kịch bản tấn công như mô tả trên hình 3.

4.2. Kịch bản thực hành và kết quả

Sau khi cài đặt xong chương trình, ta tiến hành dựng kịch bản khai thác giao thức MAVLink 1.0 và tấn công chiếm quyền điều khiển UAV. Kịch bản sẽ mô phỏng sau khi đã hoàn tất cài đặt các thành phần và mạng như sau:

Bước 1: Khởi động UAV, sau khi khởi chạy, giao diện cho thấy như trên hình 4.

Bước 2: Điều khiển UAV bằng GCS. Thực hiện đẩy một nhiệm vụ lên UAV bằng QGroundControl, UAV sẽ hoạt động tuân theo nhiệm vụ từ GCS di chuyển theo tuyến được giao (các waypoint).

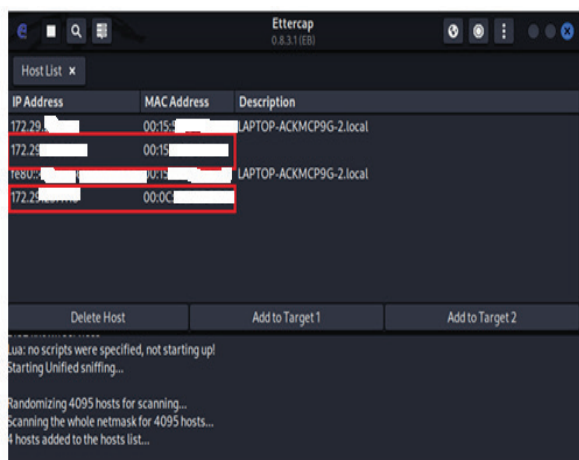


Hình 3. Mô hình kịch bản thực hành tấn công chiếm quyền điều khiển phương tiện bay không người lái.

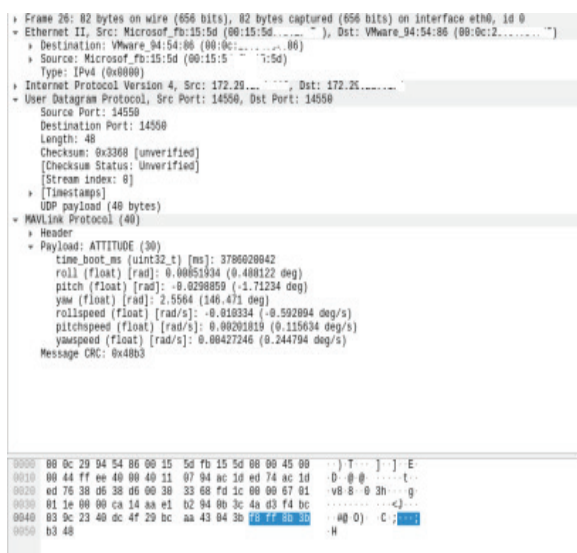


Hình 4. Hình ảnh giả lập.

Bước 3: Trình sát, thu thập và phân tích thông tin. Kẻ tấn công thực sử dụng Ettercap, thu thập, trình sát các thông tin về UAV và GCS (port, protocol...), thông qua Wireshark để phân tích gói tin thu thập được. Kẻ tấn công thu thập, phân tích dựa trên các thông tin như: cổng giao tiếp (thường là các cổng MAVLink phổ biến như 14550 trong phiên UDP); giao thức được sử dụng (MAVLink); cấu trúc của các payload (chẳng hạn như các lệnh nhiệm vụ, heartbeat, dữ liệu viễn thông). Một số kết quả thực hành như trên các hình 5 và 6.



Hình 5. Trình sát các thành phần tham gia mạng kết nối phương tiện bay không người lái - điều khiển qua trạm mặt đất.



Hình 6. Minh họa phân tích giao thức MAVLink.

Bước 4: Khởi tạo tấn công từ chối dịch vụ đến UAV. Trên cơ sở thông tin đã trình sát, thu thập tại bước 3, thực hiện tấn công từ chối dịch vụ làm UAV dừng lại giữa không trung, không thể tiếp tục thực hiện nhiệm vụ đã định hoặc phản hồi các lệnh mới từ GCS, trạng thái bay chuyển sang chờ và treo, tiếp tục đợi nhiệm vụ bay để thực thi.

Bước 5: Kẻ tấn công thực hiện chiếm quyền điều khiển UAV. Sau khi làm gián đoạn giao tiếp giữa GCS gốc và UAV, kẻ tấn công ngừng can thiệp

nhưng lợi dụng việc giao thức MAVLink thiếu xác thực người dùng, không xác thực lệnh đến, điều này cho phép kẻ tấn công tiêm các lệnh mới. Kẻ tấn công sử dụng MAVProxy để: Xóa nhiệm vụ hiện tại đã tải lên bởi phi công hợp pháp trước đó; đẩy lên nhiệm vụ mới là tập hợp waypoint mới theo quỹ đạo của kẻ tấn công. Do vậy, UAV giờ đây tuân theo các lệnh của kẻ tấn công và không còn phản hồi với sự điều khiển của phi công ban đầu. Nhiệm vụ đã bị chiếm quyền hoàn toàn. Kết quả thể hiện trên giao diện dòng lệnh cho thấy như hình 7.

```

kali@kali: ~/Desktop/do_an_mavlink
File Actions Edit View Help
kali@kali: x kali@kali:~/Desktop/do_an_mavlink x root@kali:/home/kali x
"Z
zsh: suspended python attackWP.py

kali@kali:~/Desktop/do_an_mavlink
python attackWP.py
*Test upload mission*
- Checking heartbeat
- Heartbeat from (system : component 0)
- Clearing all waypoints
- Sending message
- Waiting for message: MISSION_REQUEST
+ Message read: MISSION_REQUEST (target_system : 255, target_component : 0, seq : 0, mission_type : 0)
- Creating waypoint 1
- Waiting for message: MISSION_REQUEST
+ Message read: MISSION_REQUEST (target_system : 255, target_component : 0, seq : 1, mission_type : 0)
- Creating waypoint 2
- Waiting for message: MISSION_REQUEST
+ Message read: MISSION_REQUEST (target_system : 255, target_component : 0, seq : 2, mission_type : 0)
- Creating waypoint 3
- Waiting for message: MISSION_REQUEST
+ Message read: MISSION_REQUEST (target_system : 255, target_component : 0, seq : 3, mission_type : 0)
- Creating waypoint 4
- Waiting for message: MISSION_REQUEST
+ Message read: MISSION_REQUEST (target_system : 255, target_component : 0, seq : 4, mission_type : 0)
- Creating waypoint 5
- Waiting for message: MISSION_REQUEST
+ Message read: MISSION_REQUEST (target_system : 255, target_component : 0, seq : 5, mission_type : 0)
- Mission start
- Waiting for message: MISSION_ACK
+ Message read: MISSION_ACK (target_system : 255, target_component : 0, type : 0, mission_type : 0)
- Mission start
- Waiting for message: COMMAND_ACK
+ Message read: COMMAND_ACK (command : 300, result : 0, progress : 0, result_param2 : 0, target_system : 255, target_component : 0)
+ Message read: MISSION_ITEM_REACHED (seq : 0)
+ Message read: MISSION_ITEM_REACHED (seq : 1)
+ Message read: MISSION_ITEM_REACHED (seq : 2)
+ Message read: MISSION_ITEM_REACHED (seq : 3)
+ Message read: MISSION_ITEM_REACHED (seq : 4)
kali@kali:~/Desktop/do_an_mavlink

```

Hình 7. Giao diện dòng lệnh thể hiện kết quả chiếm quyền, chuyển lệnh điều khiển đến phương tiện bay không người lái thành công.

5. Kết luận

Nội dung bài báo đã trình bày một số kết quả nghiên cứu về phòng chống UAV sử dụng phương án chiếm quyền điều khiển giao thức lớp mạng, đã thực hành mô phỏng phương án chiếm quyền điều khiển UAV.

Phương pháp tấn công dựa trên phân tích giao thức điều khiển đối với UAV đang ngày càng được quan tâm nghiên cứu. Tuy vậy, phương pháp này gặp nhiều thách thức lớn như: khó khăn trong việc phân tích đối với dữ liệu hay tín hiệu điều khiển đã được mã hóa; thời gian phân tích thường lớn, do vậy, không phù hợp đối với các bài toán yêu cầu thời gian thực, cần tấn công tức thì. Khi các UAV hoạt động bày đàn, việc phân tích giao thức để chế áp cho từng

thiết bị một là không hiệu quả. Ngoài ra, đối với các UAV thế hệ mới có khả năng tự hành, phương pháp này cũng khó khả thi.

Hướng nghiên cứu này có thể phù hợp với mục tiêu nghiên cứu, phân tích các giao thức điều khiển UAV, từ đó leo thang các chiến lược bắt giữ UAV trình sát để thực hiện các nghiên cứu sâu hơn. Ngoài ra, thực hiện chiếm quyền đối với các UAV dân dụng vi phạm các vùng cấm bay, việc sử dụng phương pháp tấn công thông qua chiếm giao thức điều khiển cho thấy tiềm năng, tính hiệu quả cao.

TÀI LIỆU THAM KHẢO

- [1] Y. Jiang (2021), "Development of a laboratory platform for UAV cybersecurity education", *2021 ASEE Virtual Annual Conference*, 8pp.
- [2] I. Guvenç, F. Koohifar, S. Singh, et al. (2018), "Detection, tracking, and interdiction for amateur drones", *IEEE Communications Magazine*, **56**, pp.75-81.
- [3] R.A. Zitar, M.A. Betar, M. Ryalat, et al. (2022), "A review of UAV visual detection and tracking methods", *Computational Science & Computational Intelligence*, pp.2-10.
- [4] J.H. Kim, N. Kim, C.S. Won (2023), "High-speed drone detection based on yolo-V8", *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, DOI: 10.1109/ICASSP49357.2023.10095516.
- [5] Y. Wang, Y. Chen, J. Choi, et al. (2019), "Towards visible and thermal drone monitoring with convolutional neural networks", *APSIPA Transactions on Signal and Information Processing*, DOI: 10.48550/arXiv.1812.08333.
- [6] Y.M. Kwon, J. Yu, B.M. Cho, et al. (2018), "Empirical analysis of MAVLink protocol vulnerability for attacking unmanned aerial vehicles", *IEEE Access*, **6**, pp.43203-43212, DOI: 10.1109/ACCESS.2018.2863237.
- [7] L. Krichen, F. Mohamed, L.C. Fourati (2018), "Communication architecture for unmanned aerial vehicle system", *International Conference on Ad-hoc Networks and Wireless*, pp.213-225, DOI: 10.1007/978-3-030-00247-3_20.
- [8] A. Koubâa, A. Allouch, M. Alajlan, et al. (2019), "Micro air vehicle link (mavlink) in a nutshell: A survey", *IEEE Access*, **7**, pp.87658-87680, DOI: 10.1109/ACCESS.2019.2924410.
- [9] N.A. Khan, N.Z. Jhanjhi, S.N. Brohi, et al. (2022), "A secure communication protocol for unmanned aerial vehicles", *Comput. Mater. Contin.*, **70**(1), pp.601-618, DOI: 10.32604/cmc.2022.019419.

Phát hiện lỗ hổng phần mềm nhờ học sâu

Bùi Văn Công^{1*}, Vũ Thảo Nguyên², Vũ Đức Minh³, Nguyễn Phương Lan⁴

¹Khoa Công nghệ Thông tin, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp, 456 Minh Khai, phường Vĩnh Tuy, Hà Nội, Việt Nam

²Trường Đại học Công nghệ Nanyang, 50 Đại lộ Nanyang, Singapore

³Trường THPT Cầu Giấy, 8/118 Nguyễn Khánh Toàn, phường Cầu Giấy, Hà Nội, Việt Nam

⁴Trường Quốc tế Nhật Bản, 84A Nguyễn Thanh Bình, phường Hà Đông, Hà Nội, Việt Nam

Ngày nhận bài 21/10/2024; ngày chuyển phản biện 23/10/2024; ngày nhận phản biện 5/11/2024; ngày chấp nhận đăng 29/11/2024

Tóm tắt:

Việc phát triển các dự án phần mềm thành công luôn là mối quan tâm hàng đầu của các tổ chức, doanh nghiệp. Trong đó, việc đảm bảo chất lượng của phần mềm là ưu tiên cao nhất trong toàn bộ quá trình phát triển cũng như vận hành của phần mềm. Bài báo đề cập việc phát hiện các lỗ hổng của mã nguồn và tập trung vào phân tích cú pháp và ngữ nghĩa của các câu lệnh trong mã nguồn. Mô hình phát hiện lỗ hổng mã nguồn được thực hiện theo quy trình: (i) biểu diễn cú pháp và ngữ nghĩa; (ii) trích xuất đặc trưng của mã nguồn; (iii) cân bằng dữ liệu; (iv) phân loại mã nguồn. Đầu ra của mô hình cho biết mã nguồn đó hoặc bình thường, hoặc có lỗ hổng. Mô hình sử dụng bộ dữ liệu SARD để huấn luyện mô hình học sâu. Mạng học sâu sử dụng mô hình BERT (Bidirectional Encoder Representations from Transformers), mô hình Word2Vec kết hợp LSTM (Long Short-Term Memory) và mô hình Word2Vec với BiLSTM (Bidirectional Long Short-Term Memory) theo ba kịch bản. Kết quả phân loại được đưa qua một hàm softmax để trả về một vector chứa dự đoán xác suất xảy ra của từng loại lỗ hổng. Với tỷ lệ phát hiện lỗ hổng mã nguồn cho kết quả chính xác lên đến 82,63%, tương ứng với tỷ lệ bỏ sót chỉ còn 17,37%. Đây là một kết quả có thể chấp nhận được, chứng minh hiệu quả vượt trội đối với bài toán phát hiện lỗ hổng mã nguồn.

Từ khóa: học sâu, học tương phản, lỗ hổng, phần mềm.

Detecting software vulnerabilities using deep learning

Van Cong Bui^{1*}, Thao Nguyen Vu², Duc Minh Vu³, Phuong Lan Nguyen⁴

¹Faculty of Information Technology, University of Economics - Technology for Industrial, 456 Minh Khai Street, Vinh Tuy Ward, Hanoi, Vietnam

²Nanyang Technological University, 50 Nanyang Ave, Singapore

³Cau Giay High School, 8/118 Nguyen Khanh Toan Street, Cau Giay Ward, Hanoi, Vietnam

⁴Japanese International School, 84A Nguyen Thanh Binh Street, Ha Dong Ward, Hanoi, Vietnam

Received 21 October 2024; revised 5 November 2024; accepted 29 November 2024

Abstract:

Developing software projects successful is always a top concern for organizations and enterprises. Among these concerns, ensuring software quality is the highest priority throughout the entire development and operation process. This paper addresses the detection of source code vulnerabilities and focuses on analyzing the syntax and semantics of statements within the source code. The source code vulnerability detection model follows a structured process: (i) syntactic and semantic representation; (ii) feature extraction from source code; (iii) data balancing; and (iv) source code classification. The model's output indicates whether the source code is normal or contains vulnerabilities. The model is

*Tác giả liên hệ: Email: bvcong@uneti.edu.vn

trained using the SARD dataset and incorporates deep learning approaches. Specifically, it employs the BERT (Bidirectional Encoder Representations from Transformers) model, the Word2Vec model combined with LSTM (Long Short-Term Memory), and the Word2Vec model with BiLSTM (Bidirectional Long Short-Term Memory) across three scenarios. Classification results are passed through a softmax function to generate a vector containing the probability predictions for each type of vulnerability. The detection model achieves an accuracy rate of up to 82.63% for identifying source code vulnerabilities, with a corresponding omission rate of only 17.37%. This result is considered acceptable and demonstrates the superior effectiveness of the approach in the task of source code vulnerability detection.

Keywords: contrastive learning, deep learning, software, vulnerabilities.

1. Đặt vấn đề

Số lượng lỗ hổng trên phần mềm máy tính ngày càng gia tăng, dễ dàng trở thành mục tiêu cho các cuộc tấn công cố ý, trong khi các tin tặc lại có vô số phương pháp khai thác các lỗ hổng này. Báo cáo cho thấy mức độ thiệt hại và sự gia tăng đột biến về số lượng lỗ hổng phần mềm ngày càng nghiêm trọng. Khi hệ thống đang tồn tại lỗ hổng, nếu người quản lý không phát hiện ra và cài đặt bản vá trước khi tin tặc biết cách khai thác, chúng có thể bị sử dụng để thực hiện các hành vi nguy hiểm như đánh cắp các dữ liệu quan trọng, gây từ chối dịch vụ... và nguy hiểm nhất là nắm quyền kiểm soát hệ thống và phát tán virus. Để phát hiện lỗ hổng mã nguồn, nghiên cứu chỉ ra có nhiều CVE (Common Vulnerabilities and Exposures) được phát hiện và công bố. Đặc biệt mức độ nghiêm trọng của các CVE là rất lớn. Đồng nghĩa với việc khi các CVE này bị tấn công và khai thác sẽ dẫn đến hậu quả rất nguy hiểm cho các hệ thống của tổ chức.

Vậy nên bài báo đề cập việc phát hiện lỗ hổng trong mã nguồn của phần mềm; gọi là lỗ hổng phần mềm. Có hai phương pháp chính để phát hiện lỗ hổng mã nguồn đó là phương pháp thủ công và phương pháp tự động hóa [1-4]. Đối với hướng tiếp cận thủ công có ưu điểm là hiệu quả cao nhưng nhược điểm là yêu cầu người phân tích phải có trình độ cao đồng thời tốn rất nhiều thời gian. Đối với vấn đề tự động hóa phát hiện lỗ hổng mã nguồn thì đang được nghiên cứu và áp dụng nhiều hiện nay do có nhiều công nghệ tính toán cũng như phương

pháp hỗ trợ mạnh mẽ. Có thể nhận thấy các hướng tiếp cận gần đây thường tập trung vào phân tích mã nguồn theo dạng ngữ nghĩa và cú pháp rồi sử dụng các thuật toán học máy hoặc học sâu để tiếp tục phân tích và dự đoán lỗ hổng. Tuy nhiên, chúng tôi cho rằng các hướng tiếp cận hiện nay đang gặp phải một số vấn đề sau [5-11].

- Thiếu đặc trưng và thuộc tính về lỗ hổng bảo mật: Thực tế cho thấy, một số nghiên cứu đã đề xuất được một số phương pháp mang lại hiệu quả tương đối tốt. Tuy nhiên, khi áp dụng các phương pháp này vào các bộ dữ liệu khác thì tỷ lệ phát hiện nhằm rất cao. Điều này thể hiện, phương pháp đề xuất đó chỉ phù hợp với một số bộ dữ liệu cụ thể nên rất khó áp dụng trong các bộ dữ liệu khác. Hay nói cách khác, các thuộc tính và đặc trưng được các nghiên cứu đề xuất chưa thể hiện được đầy đủ và khác biệt về các lỗ hổng phần mềm

- Bộ dữ liệu thực nghiệm thiếu thực tế: Các nghiên cứu trước đây thường lựa chọn các bộ dữ liệu có sẵn để đánh giá hiệu quả của mô hình đề xuất. Bên cạnh đó, các bộ dữ liệu này thường có đặc điểm là có sự cân bằng giữa các nhãn lỗ hổng và nhãn bình thường. Điều này dẫn đến các nhà nghiên cứu đã không tính toán đến trường hợp làm sao để phát hiện được lỗ hổng của mã nguồn trên các bộ dữ liệu mất cân bằng. Đây là vấn đề rất quan trọng vì trong thực tế, khi kiểm tra và đánh giá lỗ hổng phần mềm thì có sự chênh lệch rất lớn giữa về số lượng bản ghi của các mã nguồn chứa lỗ hổng bảo mật và mã nguồn không chứa lỗ hổng bảo mật. Chính vì

vậy, khi áp dụng các phương pháp này vào thực tế thì tỷ lệ phát hiện sai rất cao về cả khả năng dự đoán chính xác lỗ hổng mã nguồn cũng như mã nguồn bình thường. Cá biệt có trường hợp không thể phát hiện được lỗ hổng phần mềm nào do mô hình bị overfitting.

Để giải quyết các vấn đề trên, các nghiên cứu gần đây thường tập trung áp dụng các kỹ thuật phân tích đồ thị kết hợp với các thuật toán học máy hoặc học sâu [7, 8, 12-15]. Theo đó, các hướng tiếp cận thường phân tích mã nguồn thành các dạng đồ thị như AST (Abstract syntax tree) [16], CFG (Control Flow Graph) [17], PDG (Program Dependence Graph) [18] sau đó dùng các kỹ thuật học máy hoặc học sâu để phân loại mã nguồn bình thường và mã nguồn chứa lỗ hổng bảo mật. Khi quan sát và phân tích về các hướng tiếp cận này, chúng tôi nhận thấy một số vấn đề cần cải thiện như sau:

- Vấn đề biểu diễn mã nguồn: Chúng tôi cho rằng việc phân tích và biểu diễn mã nguồn thành một trong các kiểu như AST, CFG, PDG sẽ không thể hiện được đầy đủ mối quan hệ giữa các thành phần trong hàm và các biến của mã nguồn. Nguyên nhân của vấn đề này là do các cách biểu diễn AST, CFG, PDG sẽ chỉ có khả năng biểu diễn được một vài mối quan hệ giữa thành phần của mã nguồn. Do đó, việc chỉ áp dụng một trong các cách biểu diễn này sẽ dẫn đến tình trạng mã nguồn không được biểu diễn một cách đầy đủ và rõ ràng. Từ đó dẫn đến thông tin được trích xuất kém hiệu quả.

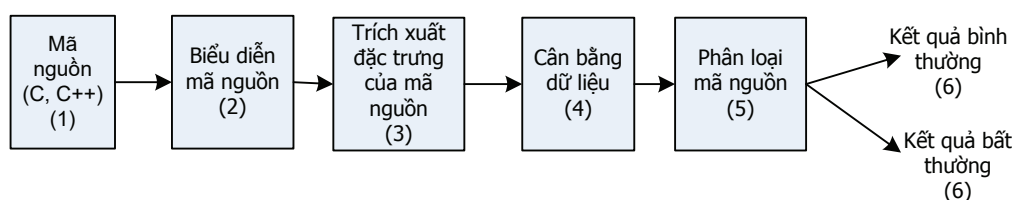
- Vấn đề phân loại lỗ hổng mã nguồn trong các bộ dữ liệu mất cân bằng: để giải quyết các vấn đề liên quan đến mất cân bằng dữ liệu, các hướng tiếp cận thường đề xuất 2 giải pháp: i) lựa chọn các bộ dữ liệu có sự cân bằng về mặt định lượng giữa các nhãn;

- ii) tiến hành phân loại mà không áp dụng kỹ thuật hoặc phương pháp cân bằng dữ liệu. Cả 2 giải pháp này sẽ làm cho hiệu quả của mô hình phân loại không đạt kết quả tốt.

- Phương pháp phát hiện lỗ hổng mã nguồn: Theo đó, ngay sau khi kết thúc quá trình phân tích và trích xuất đặc trưng của mã nguồn thì các đặc trưng này thường được đưa vào mô hình phân loại nhằm phát hiện ra các mã nguồn bình thường và bất thường. Chúng tôi nhận thấy với cách tiếp cận truyền thống như vậy mặc dù sẽ mang lại hiệu quả tốt, nhưng cũng cần khắc phục vì các bộ dữ liệu mất cân bằng sẽ dẫn đến tình trạng mô hình phát hiện dễ bị phát hiện nhầm. Từ đó dẫn đến hiệu quả của các mô hình phát hiện sẽ không được cao.

Để phát hiện lỗ hổng, bài báo đề xuất mô hình sử dụng phân tích học sâu dựa trên luồng xử lý nhằm bổ sung thêm cho phân tích theo cú pháp còn được trích xuất qua biểu đồ luồng điều khiển CFG, biểu diễn được logic cũng như cả mối quan hệ giữa các đỉnh với nhau thông qua sự kết hợp của 3 kỹ thuật xử lý chính: Word2Vec kết hợp LSTM, Word2Vec kết hợp BiLSTM, phát hiện lỗ hổng phần mềm sử dụng BERT.

Tiếp theo, mô hình Word2Vec + LSTM và Word2Vec + BiLSTM được đề xuất nhằm tính toán và trích xuất các đặc trưng của mã nguồn. Trong mô hình này, các thành phần xử lý chính bao gồm tiền xử lý, sử dụng Word2Vec để chuyển chuỗi mã đã được xử lý thành các vector nhúng. Sau đó, các vector nhúng được đưa vào mạng LSTM và BiLSTM để tiến hành học và phân loại lỗ hổng. Kết quả phân loại sẽ được đi qua một hàm softmax để trả về một vector chứa dự đoán xác suất xảy ra của từng loại lỗ hổng đã cho (hình 1).



Hình 1. Mô hình tổng quan về phát hiện lỗ hổng mã nguồn.

2. Phần mềm và lỗ hổng phần mềm

2.1. Phần mềm

Phần mềm máy tính là tập hợp các chương trình, dữ liệu và chỉ dẫn mà máy tính sử dụng để thực hiện các nhiệm vụ [19].

Phần mềm có thể chia thành nhiều loại khác nhau, dựa trên chức năng và mục đích sử dụng: Phần mềm hệ thống; Phần mềm ứng dụng; Phần mềm tiện ích; Phần mềm lập trình; Phần mềm nhúng; Phần mềm mã nguồn mở và thương mại.

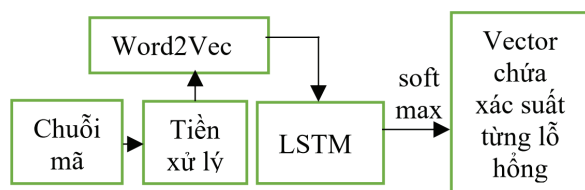
2.2. Lỗ hổng phần mềm

Có các loại lỗ hổng như: lỗi tràn bộ đệm là khi bộ nhớ bị ghi đè nhiều lần trên ngăn xếp; Lưu trữ dữ liệu không an toàn; thiếu chế độ bảo vệ tầng giao vận; rò rỉ dữ liệu; phân quyền và xác thực yếu; mã hóa bị bẻ khóa, không mã hóa; đầu vào không tin cậy; xử lý phiên không đúng, quản lý phiên yếu; thiếu bảo mật nhị phân; thiếu mã hóa dữ liệu; Không xác thực dữ liệu đầu vào; lỗi trong phần mềm; sử dụng mật khẩu yếu; lỗ hổng trong định dạng chuẩn đầu ra cho dữ liệu; lỗi thiết kế không bảo mật, sử dụng thuật toán mật mã yếu; lỗ hổng không đồng bộ trong đa luồng; sử dụng dữ liệu hoặc tài nguyên sai cách [19]. Kết quả đúng trên một dữ liệu cụ thể, nhưng khi có dữ liệu khác, bị xung đột; truy cập vào các tài nguyên hoặc chức năng không cho phép.

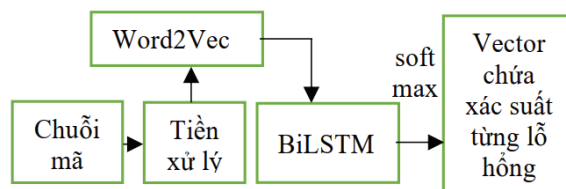
3. Mô hình phát hiện lỗ hổng phần mềm

3.1. Kiến trúc mô hình đề xuất

Trong hình 2 và 3 là mô hình Word2Vec kết hợp với LSTM:



Hình 2. Mô hình Word2Vec kết hợp LSTM tiếp theo là mô hình Word2Vec kết hợp với BiLSTM.



Hình 3. Mô hình Word2Vec kết hợp BiLSTM.

Sau cùng là mô hình phát hiện lỗ hổng phần mềm sử dụng BERT (hình 4).



Hình 4. Các thành phần chính mô hình BERT.

Chức năng và nhiệm vụ các khối con trong mô hình đề xuất thể hiện qua các hình như sau:

- Chuỗi mã: là dữ liệu phục vụ quá trình huấn luyện và phát hiện lỗ hổng. Mã nguồn sẽ bao gồm nhiều dạng ngôn ngữ và lỗ hổng khác nhau. Chúng tôi sử dụng bộ dữ liệu tham chiếu về bảo hiểm phần mềm (SARD) do Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST) [20] tạo ra để thử nghiệm các mô hình đề xuất.

- Khối tiền xử lý: Với bộ dữ liệu tạo thành từ các tệp văn bản chứa mã thô gồm các tệp chứa lỗ hổng và các tệp không chứa lỗ hổng, tiến hành gắn nhãn để phân loại. Sau đó tiến hành loại bỏ các thành phần nhiễu khỏi dữ liệu bằng cách xóa những chú thích, đặt lại tên biến, tên hàm để tạo nên một cấu trúc mã nhất quán, không chứa những thông tin ảnh hưởng tới quá trình đào tạo mô hình.

- Khối Word2Vec: là mô hình giúp tạo ra các biểu diễn embedding của từ trong một không gian có số chiều thấp hơn nhiều lần so với số từ trong từ điển. Ý tưởng đó là hai từ (từ con) xuất hiện trong những văn cảnh giống nhau thường có ý nghĩa gần với nhau. Hay có thể đoán được một từ (từ con) nếu biết các từ xung quanh nó trong câu. Như vậy, có hai cách khác nhau để xây dựng mô hình Word2Vec đó là dự đoán những từ ngữ cảnh nếu biết trước từ

đích (Skip-gram) hoặc CBOW (Continuous Bag of Words) dựa vào những từ ngữ cảnh để dự đoán từ đích.

- Khối BERT: Khối BERT sử dụng Transformer là một mô hình attention học mối tương quan giữa các từ (hoặc một phần của từ) trong một văn bản [3].

- Khối LSTM: Khối này được đề xuất là một trong những giải pháp cho vấn đề phát hiện lỗi hỏng phần mềm từ mã nguồn. Điểm chính trong kiến trúc LSTM chính là các ô nhớ với các cổng cho phép lưu trữ hoặc truy xuất thông tin. LSTM chỉ có thể dự đoán nhãn của từ hiện tại dựa trên thông tin có được từ các từ nằm trước đó.

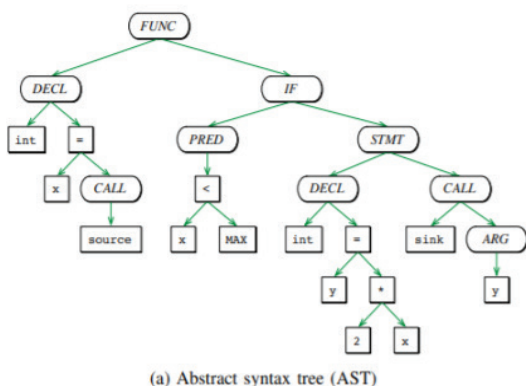
- Khối BiLSTM: Như đã nêu, điểm hạn chế của LSTM chính là chỉ dự đoán nhãn của từ hiện tại dựa trên thông tin có được từ các từ nằm trước đó. Trong khi, việc nhận dạng chính xác cấu trúc và ngữ cảnh của một đoạn mã nguồn không chỉ cần thông tin ở

phía trước của từ đang xét mà còn cả các thông tin phía sau. Điều này LSTM lại không thể nhận dạng được. Chính vì thế, BiLSTM đã được tạo ra để khắc phục điểm yếu trên. Một kiến trúc BiLSTM thường chứa 2 mạng LSTM đơn được sử dụng đồng thời và độc lập để mô hình hóa chuỗi đầu vào theo 2 hướng: từ trái sang phải và từ phải sang trái.

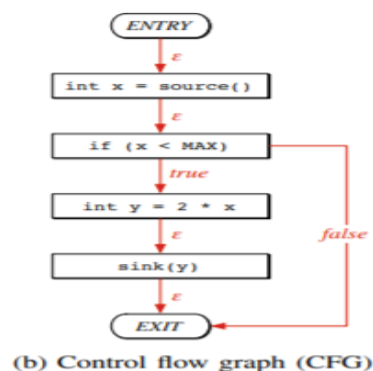
3.2. Biểu diễn mã nguồn phần mềm

Mô hình trong bài báo sử dụng biểu đồ luồng điều khiển CFG [17] là dạng biểu đồ, trong đó, mỗi nút đại diện cho một lệnh hay nói cách khác CFG là một đồ thị biểu diễn đồ họa của luồng điều khiển hoặc tính toán trong quá trình thực thi các chương trình hoặc ứng dụng của một chương trình. Các nút trong CFG là các khối cơ bản, nhóm các lệnh có thể thực thi tuần tự mà không có bất kỳ chuyển hướng điều khiển nào và các cạnh đại diện cho các luồng điều khiển giữa các khối (hình 5).

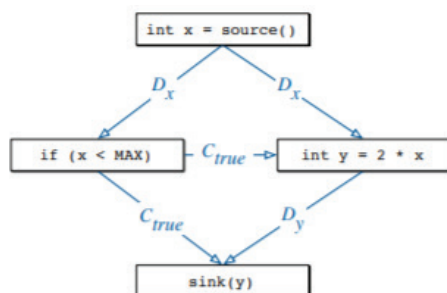
(A)



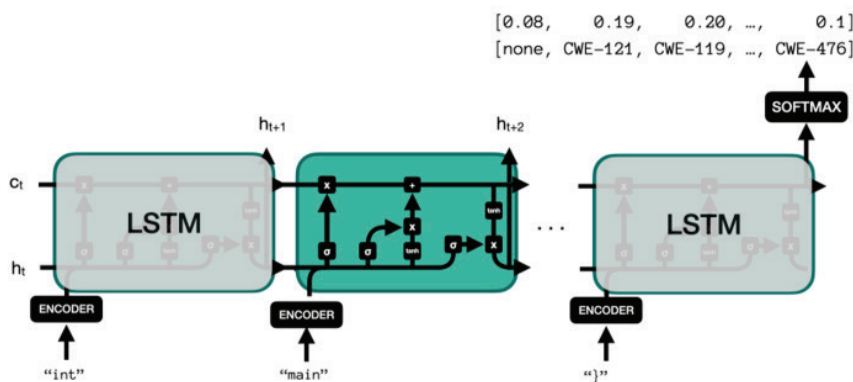
(B)



(C)



Hình 5. Biểu diễn mã nguồn dưới một số dạng: AST (A), CFG (B) và PDG (C).



Hình 6. Mô hình phát hiện lỗ hổng sử dụng LSTM.

3.3. Trích xuất đặc trưng của mã nguồn phần mềm

Các nghiên cứu cho thấy đặc trưng của đỉnh và cạnh trong CPG (Code Property Graph) [4, 21] là một biểu diễn quan trọng của mã nguồn. CPG kết hợp các phần tử từ luồng điều khiển, biểu đồ luồng dữ liệu, cùng với AST và PDG cho phép biểu diễn tổng hợp ngữ nghĩa của mã nguồn. Về mặt hình thức, CPG được ký hiệu là $G = (V, E)$, trong đó V là tập hợp các đỉnh (nút) và E là tập hợp các cạnh trong đồ thị. Mỗi đỉnh chứa thông tin về loại đỉnh.

3.4. Cân bằng dữ liệu

Sau khi thu được hồ sơ đặc trưng của mã nguồn, chúng tôi dùng nó để phân loại nhằm phát hiện các mã nguồn bình thường và mã nguồn chứa lỗ hổng. Từ hồ sơ thu được, chúng tôi nhận thấy có sự chênh lệch rất lớn giữa mã nguồn chứa lỗ hổng và mã nguồn bình thường. Nếu vấn đề này không được giải quyết, sẽ dẫn đến một sai lệch không mong muốn trong mô hình làm hạn chế hiệu suất dự đoán của nó. Hơn nữa, các vector đặc trưng được trích xuất của các mẫu có lỗ hổng và các mẫu không có lỗ hổng thể hiện sự chong chéo đáng kể trong không gian tính năng. Điều này gây khó khăn cho việc phân định các mẫu có lỗ hổng và các mẫu không có lỗ hổng. Để giải quyết tình trạng mất cân bằng dữ liệu trong quá trình huấn luyện và giúp tăng tính đa dạng của dữ liệu chúng tôi sử dụng thuật toán SMOTE (Synthetic Minority Oversampling Technique) [22].

3.5. Phân loại lỗ hổng phần mềm

Dựa trên các đặc trưng của mã nguồn được trích xuất, kết hợp với các vector đặc trưng được sinh ra từ thuật toán SMOTE bài báo sẽ tiến hành phân loại các vector đặc trưng này để kết luận về lỗ hổng của mã nguồn. Để thực hiện mục tiêu này, bài báo sử dụng 2 lớp đơn giản là Fully Connected và Softmax [23]. Trong đó Fully Connected là một cách đơn giản hóa để học các tổ hợp phi tuyến tính của các đối tượng. Lớp Fully Connected đang học một hàm có thể phi tuyến tính trong không gian. Đầu ra được làm phẳng này được cung cấp cho một mạng nơ-ron truyền thẳng và mạng nơ-ron truyền ngược. Qua một loạt các epoch, mô hình có thể phân biệt các đặc điểm nổi bật trong dữ liệu và phân loại chúng bằng cách sử dụng kỹ thuật hàm Softmax [24].

4. Thử nghiệm

4.1. Dữ liệu thử nghiệm

Bài báo này sử dụng SARD [20] do Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST) (Hoa Kỳ) tạo ra. SARD đặc biệt hấp dẫn vì nó không chỉ chứa các lỗ hổng bảo mật mà còn chứa các bản sao đã không tồn tại lỗ hổng của chúng, cho phép tìm hiểu chính xác sự khác biệt giữa lỗ hổng bảo mật và các giải pháp thay thế an toàn của nó. Bộ dữ liệu được xử lý trước khi đào tạo bằng cách loại bỏ tất cả các phần không mong muốn có thể khiến các mô hình bị quá tải và mất cân bằng. Và cuối cùng, bộ dữ liệu còn lại hơn 100.000 tệp, với một nửa chứa một trong

118 lỗi hỏng bảo mật và nửa còn lại chứa các lựa chọn thay thế không dễ bị tấn công. Tiến hành chia bộ dữ liệu thành các thành phần khác nhau, sau đó tiến hành thực nghiệm và đánh giá độ chính xác của các mô hình đề xuất. Toàn bộ quá trình chia tập dữ liệu thực nghiệm cho các kịch bản sẽ được chọn một cách ngẫu nhiên, trong đó 75% của tập dữ liệu sẽ được dùng trong quá trình huấn luyện, 25% còn lại sẽ được dùng trong quá trình kiểm tra.

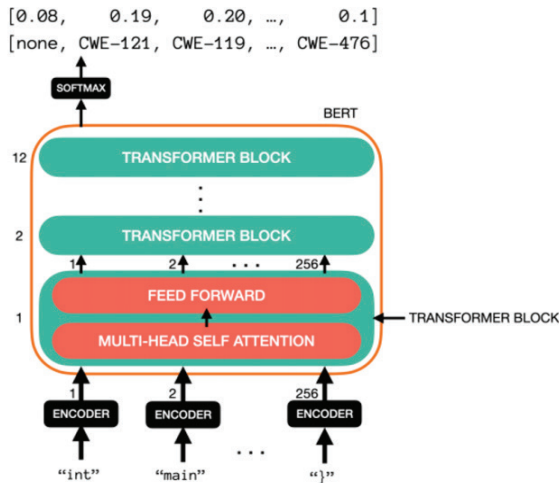
4.2. Kịch bản đánh giá

Để đánh giá được kết quả thử nghiệm tác giả tiến hành ba kịch bản thực nghiệm như sau:

Kịch bản 1: Phát hiện lỗi hỏng phần mềm sử dụng Word2Vec kết hợp với LSTM: Thử nghiệm trên mô hình LSTM với số lượng các nút ẩn khác nhau (hình 6).

Kịch bản 2: Phát hiện lỗi hỏng phần mềm sử dụng Word2Vec kết hợp với BiLSTM: số lượng các nút ẩn khác nhau.

Kịch bản 3: Phát hiện lỗi hỏng phần mềm sử dụng BERT: số lượng tầng khác nhau (hình 7).



Hình 7. Mô hình phát hiện lỗi hỏng sử dụng BERT.

4.3. Tiêu chí đánh giá mô hình

Accuracy: Tỷ lệ giữa số lượng mẫu được phân loại đúng và tổng số mẫu. Độ chính xác được tính theo công thức sau:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

trong đó: TP - True positive: số lượng mẫu có lỗi hỏng được phân loại chính xác; FN - False negative: số lượng mẫu có lỗi hỏng được phân loại là không có lỗi hỏng; TN - True negative: số lượng mẫu không có lỗi hỏng được phân loại chính xác; FP - False positive: số lượng mẫu không có lỗi hỏng được phân loại thành có lỗi hỏng.

Precision: là tỷ lệ của true positive points trên tổng số điểm được phân loại là positive (TP + FP). Độ chính xác cao có nghĩa là độ chính xác cao của việc xác định points.

$$precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

Recall: là tỷ lệ của các true positive points trên tổng số các real positive points (TP + FN). Recall cao có nghĩa là TP cao và tỷ lệ bỏ sót positive points là thấp.

$$recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

F1-score: là trung bình hài hòa của độ chính xác và thu hồi. F1 càng cao, bộ phân loại càng tốt.

$$recall = \frac{TP}{TP+FN} \times 100\% \quad (4)$$

4.4. Kết quả thực nghiệm

Kết quả thực nghiệm theo kịch bản 1: trong kịch bản thực nghiệm 1, phát hiện lỗi hỏng phần mềm sử dụng kiến trúc mô hình Word2Vec kết hợp với LSTM trên bộ dữ liệu SARD cho kết quả theo bảng 1, tác giả thấy được độ chính xác của quá trình phân loại tiến trình cho kết quả khác nhau khi thay đổi số lượng các nút ẩn. Với số lượng các nút ẩn là 400 thì thuật toán LSTM mang lại hiệu quả cao nhất trên tất cả các độ đo. Mặc dù số lượng các nút ẩn thay đổi từ 200 đến 1000 nhưng độ chính xác của thuật toán lại không có nhiều thay đổi. Kết quả cao nhất và thấp nhất chỉ chênh lệch nhau: accuracy tăng 1,64%, precision tăng 1,82%, recall tăng 1,94% và F1 tăng 1,89%. Như vậy, mặc dù số lượng các nút ẩn tăng hay giảm cũng không làm cho độ chính xác trong quá trình phân loại lại thay đổi một cách đáng kể. Điều này cho thấy thuật toán LSTM cung cấp khả năng phân loại tốt và ổn định khi thay đổi số lượng các nút ẩn.

Bảng 1. Kết quả thực nghiệm phát hiện lỗi hồng phần mềm sử dụng kiến trúc mô hình Word2Vec kết hợp với LSTM.

Số các nút ẩn	Chính xác	Độ phủ	F1	Độ chính xác
1000	71,15	67,89	69,48	71,53
400	72,97	69,83	71,37	73,17
200	71,70	68,47	70,05 %	72,02

Kết quả thực nghiệm kịch bản 2: Kịch bản 2 sử dụng kiến trúc mô hình Word2Vec kết hợp với BiLSTM trên bộ dữ liệu SARD cho kết quả từ bảng 1 và 2 thấy được khi thay đổi số lượng các các nút ẩn thì LSTM và BiLSTM cho ra các kết quả tương tự nhau. Dựa trên kết quả kịch bản 1 và 2 đã chứng minh được mô hình NLP có ý nghĩa đối với bài toán phân loại lỗi hồng. Điều này được thể hiện trên độ chính xác tổng quát tối thiểu đều khoảng trên 71% và đối với giá trị recall tối thiểu là 67%. Nói chung, kết quả phân loại đối với kịch bản chỉ dừng ở mức đạt mức chấp nhận được và cho kết quả ở mức trung bình còn kịch bản 2 thì tương đối ổn. Kịch bản 2 sử dụng BiLSTM đã mang lại kết quả tốt hơn LSTM tới 6% cho thấy tầm quan trọng của việc lưu giữ thông tin theo ngữ cảnh trên từng phân đoạn. Bộ nhớ hai chiều trong BiLSTM có thể cải thiện đáng kể hiệu suất chính xác của việc phát hiện lỗi hồng phần mềm theo cách NLP vì thông tin theo ngữ cảnh được xem xét một cách toàn diện.

Bảng 2. Kết quả thực nghiệm phát hiện lỗi hồng phần mềm sử dụng kiến trúc mô hình Word2Vec kết hợp với BiLSTM.

Số các nút ẩn	Chính xác	Độ phủ	F1	Độ chính xác
1000	78,57	75,86	77,19	78,67
400	80,13	77,56	78,83	79,83
200	79,62	77,43	78,51	79,17

Kết quả thực nghiệm kịch bản 3: Bảng 3 cho thấy số lượng tầng tăng thì độ chính xác của mô hình cũng tăng theo. Tuy nhiên, việc tăng số tầng

làm tăng hiệu suất mô hình lên khoảng 2% nhưng đồng thời cũng tăng thời gian xử lý của mô hình lên. Trong thực tế, để cân bằng giữa hiệu suất và thời gian thì cần phải cân nhắc lựa chọn được các tham số phù hợp để cho ra được kết quả như mong đợi. Đối với kịch bản 3 sử dụng mô hình BERT với cơ chế hai chiều và cơ chế chú ý (attention) cho kết quả phát hiện lỗi hồng tốt hơn hẳn. Điều này được thể hiện trên kết quả recall cao nhất trong 3 kịch bản là 82% tương ứng với tỷ lệ bỏ sót là 18%. Về độ chính xác, mô hình BERT và các biến thể của nó đều hoạt động tốt hơn các mô hình học sâu NLP truyền thống đã được thử nghiệm. Điều này là do cơ chế chú ý trong các mô hình BERT có thể trích xuất thông tin nhận biết ngữ cảnh theo cách tốt hơn nhiều so với LSTM hoặc RNN. Kết quả này chứng minh được mô hình đề xuất có giá trị thực tiễn đối với bài toán phát hiện lỗi hồng mã nguồn.

Bảng 3. Kết quả thực nghiệm phát hiện lỗi hồng phần mềm sử dụng mô hình BERT.

Số các tầng	Chính xác	Độ phủ	F1	Độ chính xác
4	82,89	80,97	81,92	84,08
8	83,58	81,72	82,64	84,64
12	84,41	82,63	83,51	85,33

Chúng tôi đã sử dụng bộ dữ liệu FFmpeg+Qume để so sánh việc phát hiện lỗi hồng mã nguồn theo mô hình BERT, như kịch bản 3 nêu trên với các mô hình Russell [13], VulDeePecker [24], SySeVR [25]. Kết quả thể hiện trong bảng 4.

Các kết quả thực nghiệm trên bảng 4 cho thấy tiếp cận của chúng tôi được đề xuất trong bài báo không chỉ mang lại hiệu quả tốt trong nhiệm vụ phân loại lỗi hồng mã nguồn mà còn tối ưu hơn so với các mô hình, hướng tiếp cận khác. Cụ thể đối với bộ dữ liệu FFmpeg+Qume, dựa trên kết quả thực nghiệm thì thấy được rằng với độ đo accuracy thì tiếp cận của chúng tôi hiệu quả hơn tất cả các nghiên cứu khác từ 0,2 đến 10%. Tương tự như vậy đối với độ đo precision, tiếp cận này cũng tốt hơn từ 0,3 đến 9%

Bảng 4. So sánh mô hình đề xuất với một số mô hình của các tác giả khác.

Bộ dữ liệu	Tiếp cận	Acc (độ chính xác)	Pre (chính xác)	Rec (độ phủ)	F1 (chỉ số F1)
FFmpeg+Qume	Kịch bản 3 theo mô hình BERT	62,78	57,15	72,98	64,1
	Russell [13]	58,13	54,04	39,50	45,62
	VulDeePecker [25]	53,58	47,36	28,70	35,20
	SySeVR [25]	52,52	48,34	65,96	56,03

so với các hướng tiếp cận khác. Còn đối với độ đo recall thì tiếp cận của chúng tôi đề xuất cao hơn tất cả các tiếp cận còn lại từ 10 đến hơn 31%. Tương tự như độ đo recall, độ đo F1_score theo tiếp cận của chúng tôi cũng cao hơn các nghiên cứu khác từ 7 đến 30%.

5. Kết luận

Nghiên cứu đã đề xuất sử dụng mô hình BERT với cơ chế hai chiều và chú ý cho kết quả phát hiện lỗi hồng với độ chính xác theo tỷ lệ phát hiện lỗi hồng mã nguồn cho kết quả chính xác lên đến 82,63% tương ứng với tỷ lệ bỏ sót chỉ còn 17,37%. Các kết quả thực nghiệm trong mục 4 của bài báo đã chứng minh được hiệu quả vượt trội đối với bài toán phát hiện lỗi hồng mã nguồn.

Thực nghiệm so sánh và đánh giá hiệu quả trong việc phát hiện lỗi hồng của mã nguồn trong bảng 3 cho thấy, số lượng tầng tăng từ 4 đến 8 thì F1-score tăng từ 81,92% đến 82,64%, tức là độ chính xác của mô hình cũng tăng theo. Tuy nhiên, việc tăng số tầng làm tăng hiệu suất mô hình lên khoảng 2% đồng thời cũng làm tăng thời gian xử lý của mô hình lên. Trong thực tế, để cân bằng giữa hiệu suất và thời gian cần lựa chọn các tham số phù hợp để cho ra được kết quả như mong đợi. Các kết quả thử nghiệm để lựa chọn các tham số phù hợp cũng như mô hình đã được trình bày.

TÀI LIỆU THAM KHẢO

[1] B. Chernis, R. Verma (2018), "Machine learning methods for software vulnerability detection," *Proceedings of The Fourth ACM International Workshop on Security and Privacy Analytics*, pp.31-39, DOI: 10.1145/3180445.3180453.

[2] B. Liu, W. Guan, C. Yang, et al. (2023), "Transformer and graph convolutional network for text classification", *International Journal of Computational Intelligence Systems*, **16**, DOI: 10.1007/s44196-023-00337-z.

[3] J. Devlin, M.W. Chang, K. Lee, et al. (2019), "Bert: Pre-training of deep bidirectional transformers for language understanding", *The 2019 Conference of The North American Chapter of The Association for Computational Linguistics: Human Language Technologies*, **1**, pp.4171-4186, DOI: 10.18653/v1/N19-1423.

[4] D.X. Cho, H.M. Dao, M.T. Cong, et al. (2023), "A novel approach for software vulnerability detection based on intelligent cognitive computing", *The Journal of Supercomputing*, **79**, pp.17042-17078, DOI: 10.1007/s11227-023-05282-4.

[5] G. Lin, S. Wen, Q.L. Han, et al. (2020), "Software vulnerability detection using deep neural networks: A survey", *Proceedings of The IEEE*, **108(10)**, pp.1825-1848, DOI: 10.1109/JPROC.2020.2993293.

[6] P. Zeng, G. Lin, L. Pan, et al. (2020), "Software vulnerability analysis and discovery using deep learning techniques: A survey", *IEEE Access*, **8**, pp.197158-197172, DOI: 10.1109/ACCESS.2020.3034766.

[7] H. Wang, G. Ye, Z. Tang, et al. (2021), "Combining graph-based learning with automated data collection for code vulnerability detection", *IEEE Transactions on Information Forensics and Security*, **16**, pp.1943-1958, DOI: 10.1109/TIFS.2020.3044773.

[8] X. Li, L. Wang, Y. Xin, et al. (2021), "Automated software vulnerability detection based on hybrid neural network", *Applied Sciences*, **11(7)**, DOI: 10.3390/app11073201.

[9] H. Wei, M. Li (2017), "Supervised deep features for software functional clone detection by exploiting lexical and syntactical information in source code", *Proceedings of The TwentySixth International Joint Conference on Artificial Intelligence*, pp.3034-3040.

[10] G. Siewruk, W. Mazurczyk (2021), "Context-aware software vulnerability classification using machine learning", *IEEE Access*, **9**, pp.88852-88867, DOI: 10.1109/ACCESS.2021.3075385.

- [11] X. Li, L. Wang, Y. Xin, et al. (2020), “Automated vulnerability detection in source code using minimum intermediate representation learning”, *Appl. Sci.*, **10**, DOI: 10.3390/app10051692.
- [12] W. Zheng, J. Gao, X. Wu, et al. (2020), “The impact factors on the performance of machine learning-based vulnerability detection: A comparative study”, *The Journal of Systems & Software*, **168(7)**, DOI: 10.1016/j.jss.2020.110659.
- [13] R.L. Russell, L. Kim, L.H. Hamilton, et al. (2018), “Automated vulnerability detection in source code using deep representation learning”, *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp.757-762, DOI: 10.1109/ICMLA.2018.00120.
- [14] P. Haridas, G. Chennupati, N. Santhi, et al. (2020), “Code characterization with graph convolutions and capsule networks”, *IEEE Access*, **8**, pp.136307-136315, DOI: 10.1109/ACCESS.2020.3011909.
- [15] Z. Li, D. Zou, J. Tang, et al. (2019), “A comparative study of deep learning-based vulnerability detection system”, *IEEE Access*, **7**, pp.103184-103197, DOI: 10.1109/ACCESS.2019.2930578.
- [16] F. Yamaguchi, M. Lottmann, K. Rieck (2012), “Generalized vulnerability extrapolation using abstract syntax trees”, *Annual Computer Security Applications Conference*, **28**, pp.358-368.
- [17] H. Gascon, F. Yamaguchi, D. Arp, et al. (2013), “Structural detection of android malware using embedded call graphs”, *ACM Workshop on Artificial Intelligence and Security*, pp.45-54.
- [18] K. Ferrante, J. Ottenstein, D. Warren (1989), “The program dependence graph and its use in optimization”, *ACM Transactions on Programming Languages and Systems*, **9(3)**, pp.319-349.
- [19] D.X. Cho (2023), “A new approach to software vulnerability detection based on CPG analysis”, *Cogent Engineering*, **10(1)**, DOI: 10.1080/23311916.2023.2221962.
- [20] R. Krishna, Y. Ding, B. Ray (2022), “Deep learning based vulnerability detection: Are we there yet?”, *IEEE Transactions on Software Engineering*, **48(9)**, pp.3280-3296, DOI: 10.1109/TSE.2021.3087402.
- [21] F. Yamaguchi, N. Golde, D. Arp, et al. (2014), “Modeling and discovering vulnerabilities with code property graphs”, *IEEE Symposium on Security and Privacy*, pp.590-604, DOI: 10.1109/SP.2014.44.
- [22] N.V. Chawla, K.W. Bowyer, L.O. Hall, et al. (2002), “SMOTE: Synthetic minority over-sampling technique”, *Journal of Artificial Intelligence Research*, **16**, pp.321-357.
- [23] E. Hoffer, N. Ailon (2015), “Deep metric learning using triplet network”, *International Workshop on Similarity-Based Pattern Recognition*, pp.84-92.
- [24] Z. Li, D. Zou, S. Xu, et al. (2018), “VulDeePecker: A deep learning-based system for vulnerability detection”, arXiv, DOI: 10.14722/ndss.2018.23158.
- [25] Z. Li, D. Zou, S. Xu, et al. (2018), “SySeVR: A framework for using deep learning to detect software vulnerabilities”, *IEEE Transactions on Dependable and Secure Computing*, **19(4)**, DOI: 10.1109/TDSC.2021.3051525.

Áp dụng mạng CNN trong phân loại quang phổ Raman của dung dịch glucose tinh khiết với bộ dữ liệu tăng cường

Nguyễn Thanh Tùng^{1,2*}, Ngô Quang Trí³

¹Trường Quốc tế, Đại học Quốc gia Hà Nội, 144 Xuân Thủy, phường Cầu Giấy, Hà Nội, Việt Nam

²Khoa Công nghệ thông tin, Trường Đại học Nguyễn Tất Thành, 300A Nguyễn Tất Thành, phường Xóm Chiếu, TP. Hồ Chí Minh, Việt Nam

³Trường Đại học Kinh tế - Kỹ thuật Công nghiệp, 456 Minh Khai, phường Vĩnh Tuy, Hà Nội, Việt Nam

Ngày nhận bài 16/5/2025; ngày gửi phản biện 19/5/2025; ngày nhận phản biện 9/6/2025; ngày chấp nhận đăng 13/6/2025

Tóm tắt:

Trong những năm gần đây, ứng dụng công nghệ học máy trong chẩn đoán tiểu đường theo phương pháp xâm lấn trở thành một lĩnh vực nghiên cứu quan trọng trong y học. Trong giai đoạn nghiên cứu hiện tại, chúng tôi tạo ra các mẫu dung dịch glucose với 14 mức khác nhau dựa trên nồng độ mol của glucose trong dung dịch, trước khi chiếu tia laser để thu thập quang phổ Raman. Dựa trên kết quả tích cực từ nghiên cứu trước, chúng tôi tái sử dụng phương pháp tăng quy mô bộ dữ liệu bằng kỹ thuật tạo dữ liệu ảo thêm nhiễu jitter. Từ đó một bộ dữ liệu mới được tạo ra, gồm dữ liệu thật và dữ liệu ảo có số mẫu lớn hơn gấp 100 lần. Mô hình mạng nơ-ron được sử dụng là mô hình tích chập một chiều với mục đích xác định mức glucose trong các mẫu. Kết quả cho thấy, kỹ thuật tạo mẫu giả thông qua phương pháp thêm nhiễu jitter nâng cao độ chính xác rõ rệt trong việc giảm thiểu hiện tượng quá khớp của mô hình học sâu, đồng thời đạt độ chính xác cao trong việc phân loại tín hiệu tán xạ Raman.

Từ khóa: chẩn đoán tiểu đường, mạng nơ-ron tích chập, nhân bản dữ liệu, nồng độ glucose, quang phổ Raman.

Application of jitter augmentation in classification of Raman scattering from glucose liquid with several machine learning models

Thanh Tung Nguyen^{1,2*}, Quang Tri Ngo³

¹International School, Vietnam National University - Hanoi, 144 Xuan Thuy Street, Cau Giay Ward, Hanoi, Vietnam

²300A Nguyen Tat Thanh Street, Xom Chieu Ward, Ho Chi Minh City, Vietnam

³University of Economics - Technology for Industrial, 456 Minh Khai Street, Vinh Tuy Ward, Hanoi, Vietnam

Received 16 May 2025; revised 9 June 2025; accepted 13 June 2025

Abstract:

The utilisation of machine learning technology for invasive diabetes diagnosis has emerged as a prominent research area in the medical field in recent years. The study employs Raman spectroscopy of a glucose fluid sample to ascertain glucose levels. We produce glucose-liquid samples with 14 varying ratios of glucose to pure water to replicate the 14 glucose concentrations

*Tác giả liên hệ: Email: tung_nt@vnu.edu.vn

found in human blood and acquire the corresponding Raman spectra. With the optimistic result of the previous study, we also design the jittering augmentation approach to enhance the dataset and the dataset size was increased by a factor of 100. We employ a 1-D Convolutional Neural Network to ascertain glucose levels in samples. The result proves the positive effect of jitter augmentation for improving the performance of the deep learning model as well as high accuracy of these models in classification of Raman scattering.

Keywords: artificial intelligence, convolutional neural network, glucose concentration, jitter augmentation, Raman scattering.

1. Đặt vấn đề

Việc phát hiện kịp thời bệnh tiểu đường có thể được thực hiện thông qua các phương pháp theo dõi glucose máu chỉ phí thấp. Hiện nay, phương pháp phát hiện đường huyết xâm lấn vẫn phổ biến do tính đơn giản, liên quan đến việc thu mẫu máu để phân tích in vitro. Các thiết bị theo dõi glucose truyền thống sử dụng phương pháp điện hóa, yêu cầu lấy một lượng máu xác định từ bệnh nhân thông qua việc chích ngón tay hoặc sử dụng kim cấy dưới da. Hầu hết những người mắc bệnh tiểu đường cảm thấy không thoải mái khi phải kiểm tra mức đường huyết thường xuyên, do cần lấy máu nhiều lần mỗi ngày để xét nghiệm.

Nghiên cứu này áp dụng chiến lược học sâu nhằm giải quyết khó khăn đó, kế thừa từ các nghiên cứu trước của chúng tôi. Mạng nơ-ron tích chập một chiều (viết tắt là 1-D CNN hoặc CNN) được sử dụng để phân tích mối tương quan giữa dữ liệu quang phổ Raman và mức glucose. Để giảm thiểu hiện tượng quá khớp (overfitting) trong quá trình huấn luyện, chúng tôi áp dụng chiến lược tăng cường dữ liệu bằng cách thêm nhiễu jitter nhằm tạo ra các mẫu tương đương, góp phần mở rộng tập dữ liệu. Kết quả thu được cho thấy hiệu quả rõ rệt, độ chính xác đạt trên 96%.

Bài báo này được tổ chức thành bốn phần chính: Giới thiệu, phương pháp, kết quả thực nghiệm và kết luận. Phần giới thiệu trình bày tổng quan về vấn đề nghiên cứu và tầm quan trọng của nó. Phần phương pháp mô tả các chiến lược và quy trình được sử dụng để xây dựng mô hình 1-D CNN và xử lý tập dữ liệu. Trong phần kết quả thực nghiệm, chúng tôi phân tích và thảo luận các

phát hiện của nghiên cứu, bao gồm hiệu quả của mô hình CNN. Phần kết luận tổng hợp các kết quả chính, đánh giá tác động của chúng và đề xuất các hướng nghiên cứu trong tương lai.

2. Nghiên cứu liên quan

Do những hạn chế của các phương pháp xét nghiệm đường huyết truyền thống, nhiều phương pháp không xâm lấn đã được phát triển nhằm cung cấp đánh giá nồng độ glucose một cách chính xác. Gần đây, quang phổ Raman đã được nghiên cứu như một công cụ để đánh giá mức glucose. Quang phổ Raman, dựa trên hiện tượng Raman được phát hiện vào năm 1928 [1], hoạt động như một kỹ thuật tán xạ. Tán xạ ánh sáng xảy ra do sự tương tác giữa các phân tử trong mẫu với tia laser đơn sắc. Ánh sáng bị tán xạ ở một tần số khác với tán xạ đàn hồi sẽ tạo nên phổ Raman, phản ánh sự tương tác không đàn hồi giữa phân tử mẫu và ánh sáng đơn sắc. Thiết bị sử dụng trong quang phổ Raman, gọi là máy quang phổ Raman, thường bao gồm bốn thành phần chính: nguồn phát laser, buồng chứa mẫu, buồng quang phổ sử dụng cách tử nhiễu xạ và bộ phận thu nhận tín hiệu. Quang phổ Raman có thể được ứng dụng để đo nhiều loại hợp chất hóa học khác nhau [2]. Một phát hiện đáng chú ý về mối liên hệ giữa dữ liệu phổ Raman thu được từ cơ thể người và mức đường huyết đã được trình bày trong [3]. Một số nghiên cứu trong những năm gần đây đã xây dựng chiến lược hoặc phương thức đo glucose không xâm lấn bằng phổ Raman trên người đã cho thấy tiềm năng ứng dụng. Nghiên cứu của I. Rehan và cs (2024) [4] đã áp dụng kỹ thuật học máy cơ bản là máy vector hỗ trợ (viết tắt là SVM) để phân tích dữ liệu phổ Raman từ nước tiểu để đánh giá xem người đó có

bị tiểu đường không, kết quả đạt được đáng kể là tỷ lệ chính xác lên tới 94%. Trong nghiên cứu của L.A. Duc và cs (2023) [5], bệnh tiểu đường tuýp 2 cũng đã được phát hiện thông qua sự kết hợp giữa phổ Raman và các kỹ thuật học máy. Tuy nhiên, thách thức trong việc sử dụng phổ Raman là đảm bảo hiệu quả dự đoán mức glucose một cách ổn định. Nhiều yếu tố có thể ảnh hưởng đến kết quả, chẳng hạn như cường độ tín hiệu Raman yếu và đặc điểm phức tạp của cơ thể người khi có cấu trúc dày đặc gồm các mô và chất lỏng [6]. Trước tình hình đó, một số nghiên cứu đã có những cách tiếp cận đơn giản hơn khi thu thập Raman từ dung dịch glucose thay vì thu trực tiếp trên da người, nhằm đánh giá hiệu quả học máy khi trích xuất glucose. Nghiên cứu của González-Viveros đề xuất giải pháp áp dụng Mạng nơ-ron tích chập để phân loại nồng độ đường trong dung dịch nước đường với độ chính xác cao [7].

Từ các nghiên cứu trước, nhóm nghiên cứu đã thực hiện toàn bộ các bước, từ thu thập dữ liệu đến đề xuất các mô hình học máy để giải quyết bài toán phân loại dựa trên dữ liệu phổ Raman có đặc trưng nồng độ glucose. Theo T.N. Quang và cs (2024) [8], các mẫu giả lập được tạo ra với giá trị mong muốn thay vì sử dụng mẫu sinh học, do gặp nhiều hạn chế trong việc thu thập mẫu theo cách không xâm lấn hoặc xâm lấn ở các vùng khác nhau trên cơ thể người [9], cũng như các vấn đề đạo đức liên quan đến thử nghiệm trên người.

Trong nghiên cứu trước, chúng tôi đã tiến hành toàn bộ quá trình từ thu thập dữ liệu đến đề xuất mô hình học máy để phân loại dữ liệu phổ Raman với đặc trưng nồng độ glucose [8]. Một máy quang phổ Raman kích thước lớn được sử dụng để thu thập dữ liệu từ vật liệu. Các mẫu được tổng hợp bằng cách pha trộn glucose tinh khiết với nước khử ion theo tỷ lệ chính xác nhằm mô phỏng mẫu máu có các mức glucose cụ thể trong môi trường phòng thí nghiệm. Chúng tôi thiết lập công suất laser ở mức 115 mW và cấu hình phạm vi đo để bao phủ bước sóng 792,3 nm trong 300 giây. Sau đó, chúng tôi nhỏ 3 ml dung dịch glucose vào vị trí cần đo để thu nhận phổ Raman. Tín hiệu tán xạ này được số hóa thành dữ liệu có thể tiền xử

lý và đưa vào mô hình học máy. Độ chính xác ở cấp mẫu đóng vai trò then chốt trong việc đánh giá khả năng dự đoán mức glucose của mô hình trong nghiên cứu này. Chúng tôi cũng sử dụng mô hình CNN 4 lớp cùng với các mô hình học máy đơn giản như SVM, Random Forest và Extra Trees. Tuy nhiên, do số lượng dữ liệu vẫn còn hạn chế, chúng tôi đã áp dụng kỹ thuật tăng cường dữ liệu jitter để tăng số lượng mẫu, qua đó tránh hiện tượng quá khớp trong các mô hình học máy phức tạp như CNN [10]. Kết quả cho thấy hiệu quả rõ rệt với độ chính xác cao trên tất cả các mô hình học máy, đặc biệt là mô hình CNN. Đáng chú ý, hiệu suất của mô hình khi sử dụng tập dữ liệu đã tăng cường cao hơn so với tập gốc, điều này cho thấy hiệu quả của kỹ thuật jitter augmentation đối với dữ liệu Raman từ mẫu tinh khiết. Do đó, trong bài báo này, chúng tôi áp dụng kỹ thuật jitter augmentation với quy mô lớn hơn và đề xuất một mô hình CNN có kích thước lớn.

3. Tạo dữ liệu ảo với thêm nhiễu jitter

Chúng tôi sử dụng tập dữ liệu Raman từ nghiên cứu trước được thu thập bằng thiết bị phổ kế quy mô lớn [8]. Tập dữ liệu gốc, được tạo ra từ các phép đo thực nghiệm, bao gồm 14 nhãn tương ứng với 14 mức nồng độ glucose từ 5,0 đến 7,5 mmol/l. Mặc dù khoảng giá trị glucose này không bao phủ toàn bộ dải chỉ số glucose của bệnh tiểu đường, mục tiêu của bài báo này cũng như nghiên cứu trước [8] là nhằm chứng minh tính khả thi của mô hình CNN và việc sử dụng các nguồn dữ liệu không lấy từ người. Do đó, phạm vi glucose giới hạn này không làm ảnh hưởng đến tính toàn vẹn của nghiên cứu. Các khoảng nồng độ glucose được gán nhãn một cách đều đặn, trung bình từ 0,1 đến 0,5 mmol/l. Mỗi mẫu chứa nhiều cặp dữ liệu bước sóng - cường độ trong phổ Raman, các cặp này được đánh chỉ số và sắp xếp theo thứ tự tăng dần của độ dịch bước sóng.

Trong nghiên cứu trước của T.N. Quang và cs (2024) [8], thuật toán tăng cường jitter được sử dụng để tạo ra các mẫu dữ liệu mới, dựa trên sự tương đồng giữa phổ tán xạ Raman và dữ liệu chuỗi thời gian mà đặc trưng của các nhãn glucose

được xác định bởi mối tương quan tỷ lệ giữa các giá trị cường độ liên kề [11]. Trong bài báo này, công thức tổng quát để tạo dữ liệu chuỗi thời gian mới bằng phương pháp jitter [12] được áp dụng cho phổ Raman như công thức (1), với một mẫu Raman thực $I = \{I_1, I_2, \dots, I_j, \dots, I_n\}$:

$$I(p_i) = \{I_1 + \epsilon_1, I_2 + \epsilon_2, \dots, I_j + \epsilon_j, \dots, I_n + \epsilon_n\} \quad (1)$$

trong đó: $I(p_i)$: mẫu ảo thứ i được tạo từ mẫu thật I ; I_j : Cường độ ở bước sóng thứ j ; n : Số bước sóng trong quang phổ Raman; ϵ_j : Giá trị nhiễu jitter ngẫu nhiên ở cường độ được thêm vào bước sóng thứ j .

Trong nghiên cứu của T.N. Quang và cs (2024) [8], kích thước dữ liệu sau khi sao chép bằng kỹ thuật tăng cường jitter đã tăng lên gấp 20 lần, từ 70 mẫu ban đầu lên thành 1400 mẫu. Tương ứng, mỗi mức glucose trong tập dữ liệu gốc bao gồm 5 mẫu, trong khi tập dữ liệu đã tăng cường chứa 100 mẫu gồm 5 mẫu gốc và 95 mẫu giả lập. Trong nghiên cứu này, kích thước dữ liệu được nhân lên 100 lần (từ 70 lên 7000 mẫu). Cụ thể, mỗi mức glucose trong tập dữ liệu tăng cường bao gồm 500 mẫu,

Bảng 1. Số lượng mẫu tương ứng với mỗi mức glucose với mỗi bộ dữ liệu.

Mức đường huyết	Số mẫu mỗi nhãn mức glucose		
	Bộ dữ liệu gốc	Bộ dữ liệu tăng cường	Bộ dữ liệu tăng cường ở nghiên cứu trước [8]
5,0	5	500	100
5,2	5	500	100
5,4	5	500	100
5,5	5	500	100
5,6	5	500	100
5,8	5	500	100
6,0	5	500	100
6,2	5	500	100
6,4	5	500	100
6,5	5	500	100
6,6	5	500	100
6,8	5	500	100
7,0	5	500	100
7,5	5	500	100

trong đó có 5 mẫu gốc và 495 mẫu giả lập. Bảng 1 trình bày số lượng mẫu tương ứng ở mỗi mức glucose trong từng tập dữ liệu.

Chúng tôi cũng lưu ý trong bảng 1 rằng, mã nhãn được sử dụng như một định danh trong quá trình huấn luyện và kiểm thử của mô hình học sâu. Do đặc trưng của bài toán phân loại, mức glucose của mỗi mẫu được xác định dựa trên mã định danh này thay vì giá trị thực tế của nó.

4. Thử nghiệm với mô hình học máy

4.1. Đề xuất mô hình CNN

Trong nghiên cứu trước [8], chúng tôi đã đề xuất một mô hình CNN gồm 4 lớp và kết quả thu được khá khả quan với tập dữ liệu đã được tăng cường. Trong nghiên cứu này, kích thước tập dữ liệu được tăng lên gấp 100 lần và dữ liệu của mỗi mẫu có kích thước rất lớn với 2030 giá trị, do đó, một mô hình CNN lớn hơn cũng được đề xuất. Cụ thể, một mạng 1-D CNN với 7 lớp nơ-ron được xây dựng, bao gồm 5 lớp tích chập, 1 lớp đầu vào và 1 lớp đầu ra.

Lớp đầu vào có nhiệm vụ tiếp nhận giá trị của mẫu dưới dạng mảng gồm 2030 phần tử, tương ứng với số lượng giá trị cường độ của phổ Raman. Lớp đầu ra có nhiệm vụ làm phẳng và trả về nhãn thuộc tập 14 nhãn. Các đặc tính của từng lớp nơ-ron được mô tả trong bảng 2.

Bảng 2. Mô hình CNN đề xuất.

Thứ tự lớp	Kích thước nhân	Số bộ lọc
Lớp 1	48	1152
Lớp 2	24	576
Lớp 3	12	144
Lớp 4	6	36
Lớp 5	3	9

Tương tự như trong nghiên cứu trước [8], lớp liên kế lớp đầu ra có kích thước kernel nhỏ hơn một cách tỷ lệ so với các lớp ở xa hơn, điều này là do cơ chế pooling vốn có trong mạng 1-D CNN. Giá trị của lớp trước đó (gần lớp đầu vào hơn) có kích thước kernel gấp đôi so với lớp liên kế phía sau, tức lớp ở gần lớp đầu ra hơn. Ngoài ra, để

nâng cao hiệu suất của mô hình CNN, số lượng bộ lọc tại mỗi lớp được xác định bằng bình phương của giá trị kích thước kernel tương ứng. Chúng tôi sử dụng hàm kích hoạt ReLU và áp dụng phương pháp chuẩn hóa L2 với hệ số điều chỉnh là 0,05 ở tất cả các lớp.

Chương này trình bày cấu trúc và nội dung của tập dữ liệu được sử dụng cho mô hình CNN, cùng với kiến trúc của chính mô hình CNN. Trong chương tiếp theo, chúng tôi sẽ trình bày quá trình triển khai mô hình CNN, quy trình huấn luyện và mô tả thiết lập thực nghiệm.

4.2. Quá trình huấn luyện và kiểm thử

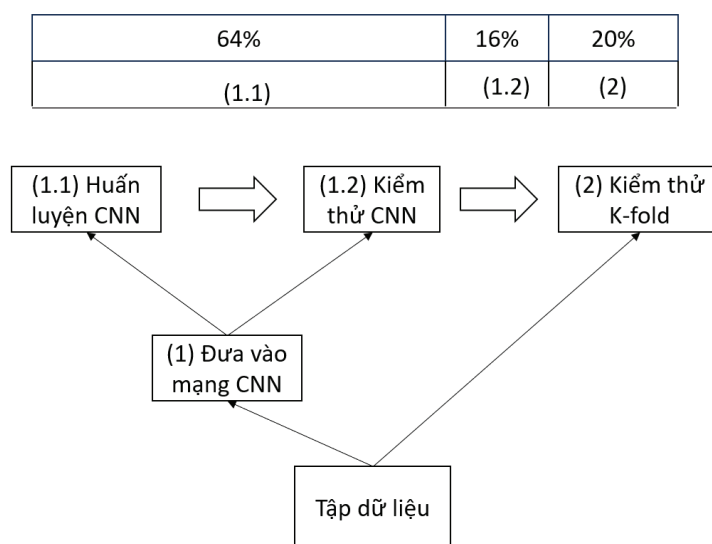
Yếu tố cốt lõi trong việc đánh giá mô hình CNN là xác thực cách mô hình học trong quá trình huấn luyện và hoạt động như thế nào trong quá trình kiểm thử. Ngoài ra, chúng tôi cũng so sánh hiệu suất của mô hình CNN. Do đó, một loạt thí nghiệm được thiết kế với phương pháp tạo tập dữ liệu riêng biệt cho huấn luyện và kiểm thử, lịch trình huấn luyện và kết quả từ quá trình kiểm thử.

Trong nghiên cứu trước [8], kỹ thuật kiểm định chéo K-fold được sử dụng, theo đó tập dữ liệu được chia thành 5 phần bằng nhau, trong đó một phần được chọn làm tập kiểm thử, bốn phần còn lại được dùng cho huấn luyện.

Với cách phân chia này, tập kiểm thử bao gồm 1330 mẫu được chia đều cho 14 mã nhãn, tức mỗi nhãn có 95 mẫu. Tương tự, tập huấn luyện có tổng

cộng 5320 mẫu, trong đó mỗi nhãn có 380 mẫu. Số lần lặp lại quá trình là 50. Để giảm thiểu hiện tượng quá khớp, chúng tôi áp dụng kiểm định chéo K-fold với thư viện Sklearn, một kỹ thuật trong đó mỗi phần được luân phiên làm tập kiểm thử, trong khi các phần còn lại được sử dụng cho huấn luyện. Ngoài ra, sau khi đưa vào mô hình học máy, quá trình chia dữ liệu cũng tự động tiến hành với thư viện Tensorflow. Hình 1 mô tả các bước thực hiện thí nghiệm gồm hai bước là: bước (1) đưa dữ liệu vào mạng 1-D CNN phục vụ việc huấn luyện, điều chỉnh trọng số các lớp tích chập và bước (2) là cho mạng CNN đã “được học” thông qua tối ưu hóa trọng số các lớp tích chập thí nghiệm với dữ liệu độc lập. Bản thân bước (1) cũng bao gồm nhiều đợt với mỗi đợt được thực hiện theo chu trình tuần hoàn là (1.1) huấn luyện và (1.2) kiểm thử. Lưu ý rằng kiểm thử ở bước (1.2) khác với kiểm thử K-fold ở bước (2). Kiểm thử bước (1.2) được thực hiện sau mỗi lần huấn luyện ở bước (1.1) và thực hiện nhiều lần theo từng đợt với mục đích tối ưu hóa trọng số lớp tích chập trong khi kiểm thử K-fold được thực hiện duy nhất một lần sau khi mô hình đã được tối ưu.

Như vậy, có thể thấy việc phân chia tập dữ liệu diễn ra hai lần, một lần thực hiện để phục vụ việc đánh giá hiệu năng theo cơ chế K-fold, một lần để mạng 1-D CNN điều chỉnh tham số sau huấn luyện. Với cơ chế K-fold, tập dữ liệu được chia thành 5 phần bằng nhau, trong đó một phần được



Hình 1. Phân bổ tập dữ liệu vào các bước trong mỗi lần vận hành mạng 1-D CNN.

chọn làm tập kiểm thử và bốn phần còn lại dùng cho huấn luyện. Như vậy dữ liệu phục vụ mạng 1-D CNN là 80%. Một lần nữa, tỷ lệ chia trong mạng 1-D CNN được thiết lập theo chuẩn là 0,2, tương tự tỷ lệ với cơ chế K-fold. Việc phân chia tập dữ liệu sẽ được trực quan hóa ở hình 1, với tỷ lệ 64% phục vụ huấn luyện CNN, 16% phục vụ kiểm thử CNN và 20% phục vụ kiểm thử K-fold. Lưu ý rằng, do kiểm thử và huấn luyện CNN được thực hiện nhiều lần trong một lần vận hành, dữ liệu phân chia vào hai tập kiểm thử và huấn luyện trong CNN sẽ được tự động sắp xếp, phân chia liên tục trong tập dữ liệu đưa vào mạng CNN (chiếm 80%) còn dữ liệu phục vụ kiểm thử K-fold sẽ là cố định.

4.3. Tiêu chí đo đạc

Trong phạm vi nghiên cứu, một số tiêu chí phổ biến trong lĩnh vực học máy được sử dụng để đánh giá như: độ chính xác (accuracy) và độ sai số. Trong các chỉ số này, độ chính xác là giá trị quan trọng nhất phản ánh hiệu suất của mô hình học máy với một tập dữ liệu cụ thể.

Trước khi tính toán độ chính xác, chúng tôi xác định cách thức đánh giá hiệu suất của mô hình. Trong bài toán phân loại, nhiệm vụ chính là mô hình sẽ dự đoán một nhãn cho một mẫu phổ Raman, sau đó kiểm tra kết quả bằng cách so sánh với mã nhãn thực tế. Do đó, kết quả đầu ra của mô hình sẽ có hai trạng thái: đúng hoặc sai.

Độ chính xác là tỷ lệ giữa số lần dự đoán đúng trên tổng số lần phân loại, được tính theo công thức (2):

$$Accuracy (\%) = \frac{N_T \times 100\%}{N_T + N_F} \quad (2)$$

trong đó: N_T : Số lần phân loại đúng; N_F : Số lần phân loại sai.

Với tiêu chí hàm sai số, chúng tôi sử dụng hàm mất mát entropy dạng Categorical để đo độ sai khác giữa phân phối xác suất dự đoán của mô hình đối với các lớp và phân phối thực tế của nhãn. Hàm mất mát này được tính theo công thức (3), nhằm đo lường mức độ khác biệt giữa hai phân phối p và q [13]:

$$H(p, q) = - \sum_i p_i \log(q_i) \quad (3)$$

Trong phạm vi nghiên cứu, độ chính xác được tính theo công thức (2) và (3) cho từng vòng lặp trong tổng số 50 vòng lặp, trước khi tính giá trị trung bình cuối cùng.

4.4. Kết quả

Bảng 3 trình bày giá trị trung bình của hai tiêu chí đánh giá: tỷ lệ chính xác tính theo công thức (2) và độ sai số tính theo công thức (3) của mô hình CNN với 5 lớp tích chập trên ba loại tập dữ liệu: tập dữ liệu gốc, tập dữ liệu đã tăng cường trong nghiên cứu trước [8] và tập dữ liệu đã tăng cường trong nghiên cứu này.

Bảng 3. Kết quả đo đạc và đối sánh.

CNN	Tập dữ liệu	Độ chính xác (%)	Độ mất mát
4 lớp tích chập [8]	Raw [8]	90,08	9,53
	Modest Enriched [8]	95,03	8,51
5 lớp tích chập	Tập dữ liệu thô	88,73	11,02
	Modest Enriched [8]	95,92	8,16
	Large Enriched	96,81	7,87

Kết quả bảng 3 cho thấy, mô hình CNN với 5 lớp tích chập gặp phải hiện tượng underfitting khi xử lý dữ liệu quy mô rất nhỏ như tập dữ liệu gốc. Khi áp dụng mô hình CNN lớn hơn lên tập dữ liệu có quy mô trung bình được sử dụng trong nghiên cứu trước [8], độ chính xác của mô hình có cải thiện nhẹ so với mô hình CNN nhỏ hơn trong nghiên cứu đó, song song với đó là độ sai số thấp hơn. Đối với tập dữ liệu đã được tăng cường ở quy mô lớn trong nghiên cứu này, kết quả thu được rất tích cực, với độ chính xác ở mức cao nhất là 96,81% và độ sai số ở mức thấp nhất là 7,87.

5. Kết luận

Chúng tôi đánh giá tính khả thi của mô hình mạng nơ-ron tích chập một chiều (1-D CNN) trong việc phân loại các mức nồng độ glucose cụ thể dựa trên dữ liệu phổ Raman thu được từ dung dịch glucose tinh khiết.

Trong quá trình thu thập dữ liệu, chúng tôi tạo ra một tập gồm năm mẫu, mỗi mẫu chứa dữ liệu của 14 mức nồng độ glucose, dao động từ 5,0 đến 7,5 mmol/l. Tuy nhiên, kích thước mẫu như vậy

là chưa đủ để giảm thiểu hiện tượng quá khớp thường gặp trong các mô hình học sâu. Vì vậy, chúng tôi áp dụng kỹ thuật tăng cường dữ liệu bằng jitter để tạo ra các mẫu mô phỏng. Tổng số mẫu được tạo ra sau quá trình tăng cường dữ liệu là 7000 mẫu, với mỗi mức glucose được đại diện bởi 500 mẫu.

Trong nghiên cứu này, chúng tôi hoàn toàn kiểm soát toàn bộ quy trình tạo tập dữ liệu bao gồm việc tạo mẫu vật liệu, thu thập phổ Raman thông qua tương tác giữa ánh sáng và vật liệu, cũng như thiết kế và triển khai thuật toán tăng cường dữ liệu jitter.

Về thiết kế mô hình học sâu, chúng tôi triển khai mạng CNN một chiều với bốn lớp và sử dụng hàm mất mát dạng entropy phân loại (categorical entropy). Kỹ thuật kiểm định chéo K-fold được sử dụng để chia tập dữ liệu đã tăng cường thành các phần huấn luyện và kiểm thử. Mô hình được huấn luyện trên tập huấn luyện và đánh giá trên tập kiểm thử qua 50 lần lặp lại.

Chúng tôi sử dụng thư viện TensorFlow với ngôn ngữ Python để xây dựng mô hình CNN, và sử dụng thư viện scikit-learn (sklearn) với Python để triển khai kiểm định chéo K-fold. Việc đánh giá hiệu suất mô hình được thực hiện dựa trên bốn tiêu chí, trong đó có tỷ lệ chính xác và giá trị hàm mất mát. Mặc dù có một vài bất ổn trong một số vòng lặp, chúng tôi vẫn tiếp tục thử nghiệm mô hình CNN với tập dữ liệu thu được từ người thật vốn đặc trưng bởi mức nhiễu cao và không có giá trị glucose xác định rõ ràng.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Bộ Khoa học và Công nghệ theo mã số đề tài DTDL.CN-25/23. Các tác giả xin chân thành cảm ơn.

TÀI LIỆU THAM KHẢO

[1] C.V. Raman, K.S. Krishnan (1928), "A new type of secondary radiation", *Nature*, **121**(3048), pp.501-502, DOI: 10.1038/121501c0.

[2] J. Xu, Q. He, Z. Xiong, et al. (2021), "Raman spectroscopy as a versatile tool for investigating thermochemical processing of coal, biomass, and wastes: Recent advances and future perspectives", *Energy & Fuels*, **35**(4), pp.2870-2913, DOI: 10.1021/acs.energyfuels.0c03975.

[3] J. Shao, M. Lin, Y. Li, et al. (2012), "In vivo blood glucose quantification using Raman spectroscopy", *PLOS ONE*, **7**(10), DOI: 10.1371/journal.pone.0048127.

[4] I. Rehan, S. Khan, R. Ullah (2024), "Raman spectroscopy assisted support vector machine: A steadfast tool for noninvasive classification of urinary glucose of diabetes mellitus", *Physica Scripta*, **99**(2), DOI: 10.1088/1402-4896/ad1da8.

[5] L.A. Duc, N.T. Tung, T.T. Oanh, et al. (2023), "Non-invasive in vivo type 2 diabetes mellitus diagnosis using Raman spectroscopy in combination with machine learning", *Mobile Networks and Applications*, DOI: 10.1007/s11036-023-02220-9.

[6] J. Smulko, M.S. Wróbel, I. Barman (2015), *Noise in Biological Raman Spectroscopy*, Book Noise in Biological Raman Spectroscopy, pp.1-6.

[7] N.G. Viveros, P.G. Gil, J.C. Ramos, et al. (2021), "On the estimation of sugars concentrations using Raman spectroscopy and artificial neural networks", *Food Chemistry*, **352**, DOI: 10.1016/j.foodchem.2021.129375.

[8] T.N. Quang, T.T. Nguyen, H.P.T. Viet (2024), "Machine learning approach for early detection of diabetes using Raman spectroscopy", *Mobile Networks and Applications*, **29**, pp.294-305, DOI: 10.1007/s11036-024-02340-w.

[9] M. Shokrehodaie, D.P. Cistola, R.C. Roberts, et al. (2021), "Non-invasive glucose monitoring using optical sensor and machine learning techniques for diabetes applications", *Ieee Access: Practical Innovations, Open Solutions*, **9**, pp.73029-73045, DOI: 10.1109/access.2021.3079182.

[10] A. Gavrilov, A. Jordache, M. Vasdani, et al. (2018), "Preventing model overfitting and underfitting in convolutional neural networks", *International Journal of Software Science and Computational Intelligence*, **10**(4), pp.19-28, DOI: 10.4018/IJSSCI.2018100102.

[11] P. Esling, C. Agon (2012), "Time-series data mining", *ACM Computing Surveys (CSUR)*, **45**(1), DOI: 10.1145/2379776.2379788.

[12] G. Iglesias, E. Talavera, A.G. Prieto, et al. (2023), "Data augmentation techniques in time series domain: A survey and taxonomy", *Neural Computing and Applications*, **35**(14), pp.10123-10145, DOI: 10.1007/s00521-022-08122-y.

[13] C.H. Chen, P.H. Lin, J.G. Hsieh, et al. (2020), "Robust multi-class classification using linearly scored categorical cross-entropy", *2020 International Joint Conference on Neural Networks (IJCNN)*, DOI: 10.1109/IJCNN48605.2020.9207212.

Chiến lược vùng đệm thích nghi cho khai thác mẫu tuần tự lợi ích cao đều đặn trên dữ liệu tăng trưởng

Trần Minh Thái*

Khoa Công nghệ Thông tin, Trường Đại học Ngoại ngữ - Tin học TP. Hồ Chí Minh
228 Sư Vạn Hạnh, phường Hòa Hưng, TP. Hồ Chí Minh, Việt Nam

Ngày nhận bài 2/12/2025, ngày gửi phản biện 22/12/2025, ngày nhận phản biện 31/12/2025

Tóm tắt:

Khai thác mẫu tuần tự lợi ích cao đều đặn (RHUSP) trên dữ liệu tăng trưởng đối mặt với thách thức lớn trong việc cân bằng giữa hiệu suất tính toán và độ chính xác. Các thuật toán hiện tại sử dụng tỷ lệ vùng đệm cố định (μ) thường gặp hạn chế: giá trị μ thấp gây lãng phí tài nguyên bộ nhớ, trong khi μ cao dẫn đến rủi ro bỏ sót các mẫu quan trọng. Bài báo này đề xuất thuật toán Adaptive-RIncHusp với chiến lược vùng đệm thích nghi, cho phép tự động điều chỉnh tham số μ tại mỗi lô dữ liệu cập nhật. Phương pháp này tích hợp ba quy tắc suy diễn hỗ trợ gồm: tỷ lệ lợi ích, độ khó ngưỡng và tốc độ tăng trưởng, kết hợp với hàm rủi ro độ nhạy cao ($f(x)=x^{0.25}$) để phát hiện sớm các biến động dữ liệu. Thử nghiệm trên 5 tập dữ liệu chuẩn chứng minh Adaptive-RIncHusp nhanh hơn 8-15% và tiết kiệm 20-40% bộ nhớ so với chiến lược cố định an toàn ($\mu=0.4$) mà vẫn duy trì độ phủ trên 97%. Đồng thời, thuật toán đạt tốc độ tương đương với chiến lược cố định tích cực ($\mu=0.9$) nhưng khắc phục hoàn toàn vấn đề mất mẫu, khẳng định tính hiệu quả trong môi trường dữ liệu biến động.

Từ khóa: Khai thác dữ liệu, mẫu tuần tự lợi ích cao, mẫu đều đặn, khai thác tăng trưởng, vùng đệm thích nghi, tinh chỉnh tham số động

Adaptive buffer strategy for regular High-Utility sequential pattern Mining on incremental data

Minh Thai Tran*

Faculty of Information Technology, University of Foreign Languages - Information Technology
228 Su Van Hanh Street, Hoa Hung Ward, Ho Chi Minh City, Vietnam

Received 2 December 2025; revised 24 December 2025; accepted 29 February 2025

Abstract:

Regular High-Utility Sequential Pattern Mining (RHUSP) from incremental databases poses a significant challenge in balancing computational efficiency and accuracy. Current algorithms using a fixed buffer ratio (μ) often suffer from limitations: a low μ value results in wasted memory resources, while a high μ carries the risk of missing important patterns. This paper proposes Adaptive-RIncHusp, an algorithm featuring an adaptive buffer strategy that automatically adjusts the parameter μ for each data batch update. The method integrates three complementary heuristics: utility Ratio, Threshold difficulty, and growth rate, combined with a high-sensitivity risk function ($f(x)=x^{0.25}$) for early detection of data fluctuations. Experiments on 5 benchmark datasets demonstrate that Adaptive-RIncHusp is 8-15% faster and saves 20-40% memory compared to the conservative fixed strategy ($\mu=0.4$) while maintaining recall $> 97\%$. Furthermore, the algorithm achieves speeds comparable to the aggressive fixed strategy ($\mu=0.9$) while completely eliminating pattern loss, confirming its effectiveness in volatile data environments.

Keywords: data mining, high-utility sequential patterns, regular patterns, incremental mining, adaptive buffer, dynamic parameter tuning.

Email: thaitm@hufit.edu.vn

I. Giới thiệu

Khai thác mẫu tuần tự lợi ích cao đều đặn (Regular High-Utility Sequential Patterns - RHUSP) là một bài toán quan trọng trong lĩnh vực khai thác dữ liệu, với nhiều ứng dụng thực tế trong phân tích hành vi khách hàng, dự đoán xu hướng thị trường, và hệ thống khuyến nghị [1, 2]. Khác với khai thác mẫu tuần tự thông thường chỉ dựa trên tần suất xuất hiện, RHUSP xem xét đồng thời ba yếu tố quan trọng: (i) thứ tự thời gian của các sự kiện, (ii) giá trị lợi ích (utility) của mỗi mẫu phản ánh lợi nhuận thực tế và (iii) tính đều đặn (regularity) đo lường sự ổn định xuất hiện của mẫu trong toàn bộ cơ sở dữ liệu [3].

1.1. Vấn đề và thách thức

Trong môi trường dữ liệu tăng trưởng liên tục như thương mại điện tử, mạng xã hội và IoT, việc khai thác lại toàn bộ cơ sở dữ liệu sau mỗi lần cập nhật là không khả thi về mặt tính toán. Các thuật toán khai thác tăng trưởng (incremental mining) hiện đại giải quyết vấn đề này bằng kỹ thuật lưu đệm (buffering): lưu trữ các mẫu bán lợi ích (Semi-High-Utility Sequences - SHS), tức những mẫu chưa đủ điều kiện là lợi ích cao (High-Utility) nhưng có tiềm năng trở thành lợi ích cao sau các lần cập nhật tiếp theo.

Tham số quan trọng nhất của kỹ thuật lưu đệm là tỷ lệ vùng đệm μ (với $0 < \mu \leq 1$), quyết định ngưỡng để một mẫu được lưu trữ. Một mẫu P được lưu đệm nếu $u(P) \geq \mu \times \text{minUtility}$ và $\text{reg}(P) \leq \delta \text{maxReg}$. Tuy nhiên, các phương pháp hiện tại [4] sử dụng giá trị μ cố định trong suốt quá trình khai thác, dẫn đến hai vấn đề nghiêm trọng.

Vấn đề 1 (Cơ chế vùng đệm có chọn lọc - Conservative): Nếu μ thấp (ví dụ: 0.4), quá nhiều SHS được lưu trữ. Điều này đảm bảo độ phủ (recall) tối đa nhưng gây tốn kém bộ nhớ (tăng 40-60%) và thời gian tính toán do phải xử lý nhiều mẫu không cần thiết.

Vấn đề 2 (Cơ chế đệm tích cực - Aggressive): Nếu μ cao (ví dụ: 0.9), rất ít SHS được lưu trữ, tiết kiệm tài nguyên nhưng dẫn đến mất mẫu thực (false negatives) nghiêm trọng khi dữ liệu tăng trưởng không đồng đều.

Thực tế, không tồn tại một giá trị μ cố định tối ưu cho mọi tập dữ liệu và mọi giai đoạn tăng trưởng. Các yếu tố ảnh hưởng bao gồm: kích thước lô dữ liệu mới ($|\Delta S D|$) so với cơ sở dữ liệu hiện tại, phân phối lợi ích (utility distribution) trong lô mới, mức độ thay đổi của ngưỡng minUtility (tăng, giảm, đột ngột hay ổn định) và đặc điểm dữ liệu (dày đặc/thưa, phân bố đều/lệch).

1.2. Đóng góp chính

Để giải quyết vấn đề sự đánh đổi giữa tốc độ và độ chính xác trong khai thác tăng trưởng RHUSP, bài báo này đề xuất Adaptive-RIncHusp với chiến lược vùng đệm thích nghi. Các đóng góp chính bao gồm:

Đề xuất ba quy tắc suy diễn (heuristic) hỗ trợ để tính toán μ thích nghi. Quy tắc Tỷ lệ lợi ích (utility ratio) đo tỷ lệ lợi ích của lô dữ liệu mới so với tổng lợi ích hiện tại, phát hiện các lô có tác động lớn. Quy tắc độ khó ngưỡng (threshold Difficulty) đánh giá mức độ thay đổi của ngưỡng minUtility , phản ánh độ khó của bài toán. Quy tắc Tốc độ tăng trưởng (growth rate) đo tốc độ tăng trưởng của cơ sở dữ liệu, tối ưu hóa hiệu suất tính toán.

Thiết kế hàm rủi ro với độ nhạy cao, sử dụng $f(x) = x \cdot 0.25$ thay vì tuyến tính để phát hiện sớm các thay đổi nhỏ trong dữ liệu, cho phép thuật toán phản ứng kịp thời.

Đề xuất chiến lược kết hợp (COMBINED) với cơ chế an toàn, lấy giá trị μ nhỏ nhất từ ba heuristic, đảm bảo lưu đệm đủ mẫu trong mọi trường hợp biên.

1.3. Bố cục bài báo

Phần còn lại của bài báo được tổ chức như sau: Mục 2 trình bày bảng tóm tắt các ký hiệu được dùng trong bài báo. Mục 3 trình bày nghiên cứu liên quan về khai thác mẫu tuần tự và các phương pháp tăng trưởng. Mục 4 giới thiệu các định nghĩa hình thức, tính chất lý thuyết và phát biểu bài toán. Mục 5 mô tả chi tiết thuật toán Adaptive-RIncHusp với ba heuristic thích nghi. Mục 6 trình bày thiết kế thực nghiệm, kết quả đánh giá và phân tích so sánh. Cuối cùng, mục 7 tóm tắt các kết quả chính và đề xuất hướng nghiên cứu tương lai.

2. KÝ HIỆU VÀ THUẬT NGỮ

Bảng 1 tóm tắt các ký hiệu được sử dụng trong nội dung của bài báo.

Bảng 1. Bảng ký hiệu.

Ký hiệu	Ý nghĩa
<i>Dữ liệu và Mẫu</i>	
D	Cơ sở dữ liệu hiện tại
D_0	Cơ sở dữ liệu ban đầu
ΔD	Lô dữ liệu tăng trưởng
$ D $	Số lượng chuỗi trong cơ sở dữ liệu
$ \Delta D $	Kích thước lô dữ liệu
I	Tập tất cả các mục (items)
HS	Tập mẫu tuần tự lợi ích cao
SHS	Tập mẫu tuần tự bán lợi ích
<i>Lợi ích và Ngưỡng</i>	
$U(D)$	Tổng lợi ích của cơ sở dữ liệu D
$U(\Delta D)$	Tổng lợi ích của lô ΔD
$u(P, S)$	Lợi ích của mẫu P trong chuỗi S
$u(P)$	Tổng lợi ích của mẫu P
θ	Ngưỡng <i>minUtility</i> (tỷ lệ %)
θ_{abs}	Ngưỡng <i>minUtility</i> tuyệt đối ($= \theta \times U(D)$)
τ	Ngưỡng vùng đệm ($= \mu \times \theta_{abs}$)
ρ	Ngưỡng <i>maxRegularity</i> (tỷ lệ %)
ρ_{abs}	Ngưỡng <i>maxRegularity</i> tuyệt đối ($= \rho \times D $)
<i>Vùng đệm</i>	
μ	Tỉ lệ vùng đệm (buffer ratio)
μ_{init}	Giá trị khởi tạo của μ
μ_{min}, μ_{max}	Phạm vi của μ (mặc định: [0.4, 0.9])
μ_a	Tỉ lệ vùng đệm thích nghi (COMBINED)
μ_u	Tỉ lệ vùng đệm dựa trên Tỷ lệ lợi ích
μ_b	Tỉ lệ vùng đệm dựa trên Độ khó ngưỡng
μ_g	Tỉ lệ vùng đệm dựa trên Tốc độ tăng trưởng
<i>Tỷ lệ và Hệ số</i>	
r_u	Tỷ lệ lợi ích: $U(\Delta D)/U(D)$
r_g	Tỷ lệ tăng trưởng: $ \Delta D / D $
f_u, f_b, f_g	Hệ số rủi ro (risk factors) từ 3 heuristic
<i>Hàm</i>	
$\mathcal{R}(x, l, h)$	Hàm rủi ro: $\text{HighSensitivityRisk}(x, l, h)$

3. NGHIÊN CỨU LIÊN QUAN

Sự phát triển của lĩnh vực khai thác dữ liệu trong những năm gần đây chứng kiến sự chuyển dịch mạnh mẽ từ các kỹ thuật khai thác tĩnh sang các phương pháp thích nghi (adaptive), nhằm đáp ứng yêu cầu xử lý dữ liệu dòng và dữ liệu tăng trưởng. Phần này tổng quan các tiến bộ cốt lõi trong khai thác mẫu tuần tự lợi ích cao (HUSPM) và các chiến lược cập nhật dữ liệu hiện đại để làm rõ bối cảnh ra đời của thuật toán đề xuất.

3.1. Từ khai thác vét cạn đến khai thác có mục tiêu

Các thuật toán khai thác mẫu tuần tự (SPM) truyền thống như PrefixSpan [5] hay GSP [6] thường đối mặt với hạn chế lớn là sinh ra số lượng mẫu khổng lồ nhưng thiếu giá trị thực tiễn. Để khắc phục điều này, HUSPM ra đời với mục tiêu tích hợp độ đo lợi ích (utility) vào quá trình khai thác. Theo bài khảo sát toàn diện của R. Zhang và cs [7], xu hướng chủ đạo của các thuật toán HUSPM hiện đại là chuyển dịch từ cấu trúc cây (tree-based) sang cấu trúc danh sách (utility-list) nhằm tối ưu hóa hiệu suất bộ nhớ.

Đáng chú ý, trong giai đoạn 2023-2024, cộng đồng nghiên cứu bắt đầu thay đổi cách tiếp cận từ “khai thác toàn bộ” (exhaustive mining) sang “khai thác có mục tiêu” (targeted mining). Huang và cs [8] đã đề xuất thuật toán TaSPM, cho phép truy vấn trực tiếp các mẫu quan tâm mà không cần quét toàn bộ không gian tìm kiếm. Xu hướng này đặt ra thách thức cho các thuật toán tổng quát như RIncHusp là phải cải thiện khả năng lọc nhiễu và tập trung vào các mẫu thực sự có ý nghĩa.

3.2. Mở rộng khái niệm: tính đều đặn và ổn định

Bên cạnh lợi ích, tính đều đặn là thước đo quan trọng để đánh giá độ tin cậy của mẫu trong môi trường biến động.

Tính đều đặn (Regularity): Thuật toán RIncHusp [4] được xem là công trình nền tảng định nghĩa tính đều đặn dựa trên khoảng cách tối đa giữa các lần xuất hiện. Tuy nhiên, hạn chế chính của RIncHusp là việc sử dụng vùng đệm (buffer) với tỷ lệ cố định, thiếu linh hoạt khi đối mặt với dữ liệu bùng nổ.

Tính ổn định (Stability): Năm 2024, các khái niệm này được nâng cấp phức tạp hơn. G. Huang và cộng sự [9] giới thiệu IPHM trên tạp chí Heliyon, sử dụng cấu trúc danh sách lợi ích tăng trưởng để duy trì mẫu có tính chu kỳ.

Song song đó, C. Xie và cộng sự [10] đề xuất khái niệm “Stable Periodic”, lập luận rằng một mẫu chỉ có giá trị thực tiễn nếu chu kỳ xuất hiện của nó duy trì sự ổn định theo thời gian.

3.3. Chiến lược cập nhật: pre-large và vùng đệm thích nghi

Trong môi trường tăng trưởng, việc hạn chế quét lại cơ sở dữ liệu là then chốt.

Chiến lược Pre-large được H. Yan và cs (2024) [11] công bố qua thuật toán Pre-HUSPM, sử dụng khái niệm “Pre-large” với hai ngưỡng hỗ trợ để tạo vùng an toàn. Hệ thống chỉ quét lại khi dữ liệu mới vượt quá ngưỡng an toàn. Tuy nhiên, đây vẫn là chiến lược tĩnh do các ngưỡng được định nghĩa trước.

Chiến lược vùng đệm thích nghi (Adaptive Buffer) được B.S. Prashanth và cs (2025) [12] công bố trên IEEE Access. Ngược lại với vùng an toàn tĩnh, nghiên cứu này đã chứng minh rằng việc điều chỉnh kích thước bộ đệm dựa trên độ biến động (concept drift) giúp duy trì độ chính xác tốt hơn phương pháp cố định. Bài báo này áp dụng tư duy “Adaptive Buffering” [12] vào bài toán HUSPM để khắc phục sự cứng nhắc của chiến lược “Pre-large” [11].

3.4. Tối ưu tham số: quy tắc suy diễn so với AutoML

Một hướng đi khác là sử dụng AI để tự động tinh chỉnh tham số, ví dụ như framework NiaAutoARM [13] sử dụng Automated Machine Learning (AutoML). Tuy nhiên, chi phí tính toán cho AutoML thường quá lớn cho ứng dụng thời gian thực. Do đó, cách tiếp cận sử dụng quy tắc suy diễn thích nghi (Adaptive Heuristics) như đề xuất trong bài báo này là giải pháp cân bằng (trade-off) tối ưu: chi phí tính toán thấp nhưng vẫn đạt khả năng thích nghi cao.

Tóm lại, bức tranh nghiên cứu giai đoạn 2024-2025 cho thấy sự thiếu vắng của một thuật toán khai thác mẫu tuần tự vừa đảm bảo tính lợi ích và đều đặn (như RIncHusp), vừa có khả năng thích nghi tham số thời gian thực (như Adaptive Buffering) mà không tổn kém chi phí tính toán như AutoML. Adaptive-RIncHusp được đề xuất để lấp đầy khoảng trống này.

4. CƠ SỞ LÝ THUYẾT VÀ PHÁT BIỂU BÀI TOÁN

Phần này trình bày các định nghĩa hình thức về khai thác mẫu tuần tự lợi ích cao đều đặn (RHUSP), các tính chất lý thuyết làm cơ sở cho

việc cắt tỉa không gian tìm kiếm, và phát biểu bài toán khai thác trong môi trường tăng trưởng.

4.1. Các định nghĩa

Trong bối cảnh đó, Adaptive-RIncHusp kế thừa tính đều đặn của RIncHusp nhưng bổ sung khả năng “thích nghi” để giải quyết bài toán mà tham số tĩnh không thể xử lý. Cho tập hợp các mục (items) $I = \{i_1, i_2, \dots, i_m\}$. Mỗi mục i_j được gán một giá trị lợi nhuận đơn vị, gọi là lợi ích ngoại (external utility), ký hiệu là $eu(i_j)$.

Định nghĩa 4.1 (Cơ sở dữ liệu chuỗi): Một cơ sở dữ liệu chuỗi SD là tập hợp các bộ $\langle sid, S \rangle$, trong đó sid là định danh duy nhất và S là một chuỗi. Một chuỗi $S = \langle e_1, e_2, \dots, e_l \rangle$ là một danh sách có thứ tự của các sự kiện (events), trong đó mỗi sự kiện e_k là một tập hợp các mục. Trong mỗi sự kiện, mỗi mục i có một số lượng, gọi là lợi ích nội tại (internal utility), ký hiệu $iu(i, e_k)$.

Định nghĩa 4.2 (Lợi ích): Lợi ích được xác định theo nhiều cấp độ. lợi ích của mục i trong sự kiện e được tính bằng $u(i, e) = iu(i, e) \times eu(i)$. Lợi ích của sự kiện e là tổng lợi ích của các mục trong sự kiện đó: $u(e) = \sum_{i \in e} u(i, e)$. Lợi ích của mẫu P trong chuỗi S , ký hiệu $u(P, S)$, được tính khi mẫu P xuất hiện trong S qua các sự kiện $ek_1, \dots, ek_p$, là tổng lợi ích của các mục tương ứng trong lần xuất hiện đó; nếu P xuất hiện nhiều lần, ta chọn lần xuất hiện có lợi ích lớn nhất. lợi ích của mẫu P trong SD được tính bằng $u(P) = \sum_{S \in SD} u(P, S)$.

Định nghĩa 4.3 (Tính đều đặn): Cho một mẫu P xuất hiện trong các chuỗi có sid là $\{sid_1, sid_2, \dots, sid_k\}$ được sắp xếp tăng dần. Để tính toán đầy đủ, ta quy ước $sid_0 = 0$ và $sid_{k+1} = |SD| + 1$. Tính đều đặn của P , ký hiệu $reg(P)$, được định nghĩa theo công thức (1):

$$reg(P) = \max_{1 \leq j \leq k+1} (sid_j - sid_{j-1}) \quad (1)$$

Một mẫu được gọi là đều đặn nếu $reg(P) \leq \delta_{maxReg}$, trong đó δ_{maxReg} là ngưỡng do người dùng quy định.

Định nghĩa 4.4 (Mẫu RHUSP): Một mẫu tuần tự P được gọi là mẫu tuần tự lợi ích cao đều đặn (RHUSP) nếu nó thỏa mãn đồng thời hai điều kiện: (1) lợi ích cao, tức $u(P) \geq minUtility$, và (2) tính đều đặn, tức $reg(P) \leq \delta_{maxReg}$.

4.2. Tính chất cắt tỉa

Một thách thức lớn trong HUSPM là lợi ích không tuân theo tính chất đơn điệu giảm (Antimonotonicity), nghĩa là một mẫu con có thể có lợi ích thấp hơn mẫu cha. Do đó, không thể dùng $u(P)$ để cắt tỉa trực tiếp. Ta cần một cận trên (upper-bound).

Định nghĩa 4.5 (lợi ích trọng số chuỗi - SWU [14]): lợi ích trọng số chuỗi của một mẫu P , ký hiệu $SWU(P)$, là tổng lợi ích của tất cả các chuỗi trong SD mà có chứa P , được định nghĩa theo công thức (2):

$$SWU(P) = \sum_{S \in SD, P \subseteq S} u(S) \quad (2)$$

trong đó $u(S)$ là tổng lợi ích của toàn bộ chuỗi S .

Định lý 4.1 (Tính đơn điệu giảm của SWU [14]): Đối với bất kỳ mẫu P nào, ta luôn có $SWU(P) \geq u(P)$. Quan trọng hơn, SWU tuân theo tính chất đơn điệu giảm: Nếu $SWU(P) < \minUtility$, thì mọi mẫu cha (super-sequence) của P cũng sẽ có $SWU < \minUtility$ và chắc chắn không phải là mẫu lợi ích cao.

Chứng minh: Chứng minh được thực hiện qua hai bước như sau.

Bước 1: Chứng minh $SWU(P) \geq u(P)$.

$$SWU(P) = \sum_{S \in SD, P \subseteq S} u(S) \quad (3)$$

$$= \sum_{S \in SD, P \subseteq S} \sum_{i \in S} u(i, S) \quad (4)$$

$$\geq \sum_{S \in SD, P \subseteq S} u(P, S) = u(P) \quad (5)$$

Bất đẳng thức được chứng minh vì $u(P, S) \leq u(S)$ với mọi $P \subseteq S$.

Bước 2: Chứng minh tính đơn điệu giảm.

Cho $P' \supseteq P$ là mẫu cha của P . Tập các chuỗi chứa P' là tập con của tập các chuỗi chứa P :

$$\{S \in SD : P' \subseteq S\} \subseteq \{S \in SD : P \subseteq S\}$$

Do đó, $SWU(P') \leq SWU(P)$. Nếu $SWU(P) < \minUtility$, thì $SWU(P') < \minUtility$, suy ra $u(P') < \minUtility$ theo bất đẳng thức đã chứng minh ở Bước 1.

Định lý 4.1 là cơ sở lý thuyết cho phép cắt tỉa không gian tìm kiếm trong cấu trúc cây DETRHUS

4.3. Cấu trúc cây DETRHUS

Cấu trúc DETRHUS (Dynamic Extended Trie for Regular High-Utility Sequences) là cấu trúc dữ liệu cốt lõi trong thuật toán RIncHusp [4], được thiết kế để lưu trữ và cập nhật hiệu quả các mẫu tuần tự trong môi trường tăng trưởng.

DETRHUS là một cây có gốc (rooted tree) trong đó mỗi bộ nút v lưu trữ bộ thông tin $\langle item, u, reg, sidlast, sChildren, iChildren \rangle$, ý nghĩa như sau: $item$ là mục tương ứng với nút; u là tổng lợi ích tích lũy của mẫu từ gốc đến nút hiện tại; reg là khoảng cách lớn nhất (maxReg) giữa các lần xuất hiện liên tiếp; $sidlast$ là định danh của chuỗi cuối cùng chứa mẫu; $sChildren$ là danh sách con theo mở rộng tuần tự (s-extension, mẫu con xuất hiện ở sự kiện tiếp theo); và $iChildren$ là danh sách con theo mở rộng đồng thời (i-extension, mẫu con xuất hiện cùng sự kiện).

Cấu trúc DETRHUS hỗ trợ ba thao tác chính để duy trì và cập nhật thông tin mẫu tuần tự. Thao tác *UpdateDETRHUS* được thực hiện khi có chuỗi mới S với định danh sid được thêm vào cơ sở dữ liệu. Thuật toán duyệt qua các nút thông tin theo công thức: lợi ích mới $u_{new} = u_{old} + u(P, S)$, tính đều đặn mới $reg_{new} = \max(reg_{old}, sid - sid_{last})$, và cập nhật định danh chuỗi cuối $sid_{last} = sid$. Thao tác *PruneTree* thực hiện cắt tỉa cây bằng cách loại bỏ các nút v thỏa mãn một trong trong cây và cập nhật hai điều kiện: nút có lợi ích thấp hơn ngưỡng vùng đệm ($v.u < \tau$) và là nút lá, hoặc nút có tính đều đặn vượt quá ngưỡng cho phép ($v.reg > \rho_{abs}$). Thao tác *ClassifyPatterns* duyệt qua toàn bộ cây để phân loại các mẫu: nếu $v.u \geq \theta_{abs}$ và $v.reg \leq \rho_{abs}$ thì mẫu được xếp vào tập HS ; nếu $v.u \geq \tau$ và $v.reg \leq \rho_{abs}$ thì mẫu được xếp vào tập SHS .

4.4. Bài toán khai thác tăng trưởng và vùng đệm

Trong môi trường dữ liệu tăng trưởng, cơ sở dữ liệu SD thay đổi liên tục: $SD_{new} = SD_{old} \cup \Delta SD$.

Định nghĩa 4.6 (Mẫu bán lợi ích - SHS): Để tránh việc quét lại S_{Dold} , các thuật toán cần lưu trữ các mẫu tiềm năng. Một mẫu P là SHS nếu:

$$u(P) \geq \mu \times \minUtility \text{ và } reg(P) \leq \delta_{maxReg}$$

trong đó $\mu \in (0, 1]$ là tỷ lệ vùng đệm (buffer ratio).

Bài toán khai thác tăng trưởng RHUSP [4]: Cho cơ sở dữ liệu cũ SD_{old} , tập hợp các mẫu HS và SHS đã khai thác từ SD_{old} , lô dữ liệu mới ΔSD (increment batch), ngưỡng $minUtility$ và ngưỡng tính đều đặn $\delta maxReg$. Kết quả là cập nhật tập HS' và SHS' trên SD_{new} với ràng buộc là Tối thiểu hóa việc truy cập lại dữ liệu trong SD_{old}

4.5. Ví dụ minh họa

Để làm rõ các khái niệm trên, xét cơ sở dữ liệu SD_0 và bảng lợi ích ngoại như Bảng 2 và Bảng 3.

Bảng 2. Cơ sở dữ liệu mẫu SD_0 .

SID Chuỗi (Sequence)
$S1 \langle (a:1)(c:2, d:6)(e:2) \rangle$
$S2 \langle (b:3)(f:5)(d:5, e:1) \rangle$
$S3 \langle (a:3)(c:3, d:5) \rangle$
$S4 \langle (b:3)(e:3)(f:6) \rangle$
$S5 \langle (a:1)(b:4, e:2) \rangle$

Bảng 3. Lợi ích ngoại.

Mục	a	b	c	d	e	f
Giá trị	4	3	7	2	5	1

Giả sử $minUtility = 34$ (khoảng 20% tổng lợi ích) và $\delta maxReg = 3$.

Phân loại mẫu trên SD_0 : Xét mẫu $\langle c \rangle$, ta có $u(\langle c \rangle) = 14(S1) + 21(S3) = 35 \geq 34$, và $Regularity = \max(1, 2, 2) = 2 \leq 3$, do đó $\langle c \rangle$ là RHUSP. Ngược lại, mẫu $\langle b f \rangle$ có $u(\langle b f \rangle) = 29 < 34$, do đó không phải RHUSP.

Kịch bản cập nhật (ΔSD_1): Giả sử có thêm 2 chuỗi mới $S6, S7$. Tổng lợi ích toàn hệ thống tăng lên, dẫn đến ngưỡng $minUtility$ mới có thể tăng lên (ví dụ: 52). Vấn đề nảy sinh với mẫu $\langle b f \rangle$ có lợi ích 29: Nếu dùng $\mu = 0,9$ (ngưỡng đệm $0,9 \times 34 = 30,6$), mẫu $\langle b f \rangle$ (29) đã bị loại bỏ từ trước, và nếu trong $S6, S7$, $\langle b f \rangle$ xuất hiện nhiều, ta sẽ bị mất mẫu này vĩnh viễn (sót mẫu - false Negative). Ngược lại, nếu dùng $\mu = 0,4$ (ngưỡng đệm $0,4 \times 34 = 13,6$), mẫu $\langle b f \rangle$ được giữ lại, nhưng hàng ngàn mẫu rác khác cũng được giữ lại, gây tràn bộ nhớ.

Điều này cho thấy sự cần thiết của một cơ chế xác định μ thông minh hơn thay vì cố định, sẽ được trình bày trong Mục V.

5. PHƯƠNG PHÁP ĐỀ XUẤT: ADAPTIVE-RINCHUSP

5.1. Tổng quan chiến lược

Chiến lược vùng đệm thích nghi được thiết kế dựa trên các giả định và có phạm vi áp dụng cụ thể như sau:

Giả định 1 (Tính liên tục của phân phối lợi ích): Phân phối lợi ích giữa các lô dữ liệu liên tiếp không thay đổi đột ngột một cách cực đoan. Điều này cho phép các quy tắc suy diễn dựa trên lô trước để dự đoán hành vi của lô sau.

Trong trường hợp dữ liệu có trôi dạt khái niệm (concept drift) nghiêm trọng, cần kết hợp thêm các kỹ thuật phát hiện trôi dạt.

Giả định 2 (Tính đại diện của các chỉ số): Ba chỉ số (tỷ lệ lợi ích, độ khó ngưỡng, tốc độ tăng trưởng) phản ánh đầy đủ các yếu tố chính ảnh hưởng đến quyết định lưu đệm. Trong các miền ứng dụng đặc thù, có thể bổ sung thêm các quy tắc suy diễn chuyên biệt.

Phạm vi áp dụng: Chiến lược này phù hợp nhất với các hệ thống có dữ liệu tăng trưởng theo lô (batch incremental), trong đó kích thước lô từ 5% đến 30% cơ sở dữ liệu hiện tại. Đối với dữ liệu dòng (streaming) với cập nhật liên tục từng bản ghi, cần điều chỉnh cơ chế tính toán để giảm chi phí phụ trội (overhead).

Sự đánh đổi giữa hiệu quả và độ chính xác: Ba quy tắc suy diễn được thiết kế theo nguyên tắc “an toàn trước” (safety-first), ưu tiên duy trì độ phủ cao hơn tối ưu hóa tốc độ. Việc chọn giá trị μ nhỏ nhất trong chiến lược COMBINED đảm bảo nếu bất kỳ quy tắc nào phát hiện rủi ro, hệ thống sẽ tự động tăng vùng đệm. Điều này có thể dẫn đến việc lưu đệm dư thừa trong một số trường hợp, nhưng đổi lại là độ phủ cao (>97% trong thực nghiệm).

Chi phí tính toán bổ sung: Việc tính toán μ thích nghi chỉ yêu cầu $O(1)$ thời gian cho mỗi lô dữ liệu, không ảnh hưởng đáng kể đến hiệu năng tổng thể của thuật toán.

5.2. Hàm rủi ro với độ nhạy cao

Để phát hiện sớm các thay đổi nhỏ trong dữ liệu, chúng tôi sử dụng hàm rủi ro phi tuyến $R(x, \alpha, \beta)$ được định nghĩa như sau:

$$R(x, \alpha, \beta) = \begin{cases} 0, & x < \alpha \\ \left(\frac{x-\alpha}{\beta-\alpha}\right)^{0.25}, & \alpha \leq x \leq \beta \\ 1, & x > \beta \end{cases} \quad (6)$$

Trong công thức (6), x là giá trị đầu vào cần đánh giá (ví dụ: *util Ratio*, *growthRate*), α là ngưỡng dưới (lower bound) và β là ngưỡng trên (upper bound) để kích hoạt tính toán rủi ro.

Hàm số sử dụng lũy thừa bậc 0,25 (căn bậc 4) thay vì hàm tuyến tính. Điều này giúp đạo hàm của hàm số trở nên rất lớn khi x vừa vượt qua ngưỡng α , cho phép thuật toán phản ứng mạnh mẽ ngay cả với những biến động nhỏ ban đầu (ví dụ: x chỉ vượt α 4% nhưng mức rủi ro được đánh giá lên tới 44,7%).

5.3. Ba quy tắc suy diễn thích nghi

a) Quy tắc suy diễn 1: Tỷ lệ lợi ích

Quy tắc này đo tỷ lệ lợi ích của lô dữ liệu mới so với tổng lợi ích hiện tại theo công thức (7):

$$utilRatio = \frac{u(\Delta SD)}{u(SD_{old})} \quad (7)$$

Nếu *util Ratio* cao (ví dụ: $\geq 20\%$), lô dữ liệu mới có tác động lớn và có thể thay đổi đáng kể tập RHUSP. Do đó, cần giảm μ để lưu đệm nhiều mẫu hơn theo công thức (8):

$$\mu_{util} = \mu_{max} - R(utilRatio, 0,01, 0,20) \times (\mu_{max} - \mu_{min}) \quad (8)$$

Ngưỡng: *thresholdlow* = 0,01 (1%), *thresholdhigh* = 0,20 (20%) Ví dụ: Nếu lô mới đóng góp 15% lợi ích, rủi ro = $(0,15/0,20)0,25 = 0,922$, dẫn đến μ giảm mạnh để lưu đệm nhiều mẫu.

b) Quy tắc suy diễn 2: Độ khó ngưỡng

Quy tắc này đánh giá độ khó của bài toán qua tỷ lệ *minUtility* so với tổng lợi ích theo công thức (9):

$$minUtilityRatio = \frac{minUtility}{u(SD_{new})} \quad (9)$$

Nếu *minUtilityRatio* thấp (bài toán khó, nhiều mẫu thỏa mãn), cần lưu đệm nhiều hơn theo công thức (10):

$$\mu_{base} = \mu_{max} - R(minUtilityRatio, 0,001, 0,005) \times (\mu_{max} - \mu_{min}) \quad (10)$$

Ngưỡng: *thresholdlow* = 0,001 (0,1%), *thresholdhigh* = 0,005 (0,5%). Lý do: Ngưỡng rất thấp vì *minUtilityRatio* thường ở khoảng 0,1-0,5% trong thực tế.

c) Quy tắc suy diễn 3: Tốc độ tăng trưởng

Quy tắc này tối ưu hóa hiệu suất dựa trên tốc độ tăng cơ sở dữ liệu theo công thức (11):

$$growthRate = \frac{|\Delta SD|}{|SD_{old}|} \quad (11)$$

Nếu *growthRate* cao (cơ sở dữ liệu tăng nhanh), cần cân bằng giữa việc lưu đệm và tốc độ theo công thức (12):

$$\mu_{growth} = \mu_{max} - R(growthRate, 0,05, 0,25) \times (\mu_{max} - \mu_{min}) \quad (12)$$

Ngưỡng: *thresholdlow* = 0,05 (5%), *thresholdhigh* = 0,25 (25%) Ví dụ: Nếu lô mới chiếm 20% cơ sở dữ liệu, rủi ro = $(0,20/0,25)0,25 = 0,946$, μ giảm để xử lý tăng trưởng nhanh.

5.4. Chiến lược kết hợp

Chiến lược kết hợp (COMBINED) lấy giá trị μ nhỏ nhất theo công thức (13):

$$\mu_{adaptive} = \min(\mu_{util}, \mu_{base}, \mu_{growth}) \quad (13)$$

Cơ sở lý thuyết: Việc chọn giá trị nhỏ nhất (minimum) đảm bảo vùng đệm đủ mẫu trong mọi trường hợp biên. Nếu một trong ba heuristic phát hiện rủi ro cao, thuật toán sẽ lưu đệm nhiều hơn để tránh mất mẫu.

5.5. Thuật toán Adaptive-RIncHusp

Thuật toán 1 mô tả chi tiết quá trình khai thác RHUSP với chiến lược vùng đệm động. Đây là mở rộng của RIncHusp [4] với điểm khác biệt chính là việc tính toán μ thích nghi tại mỗi lô dữ liệu.

So sánh với RIncHusp [4]: Điểm khác biệt là dòng 16 và 19, nơi τ được tính bởi μ_a (thích nghi) thay vì μ cố định. Tất cả các bước khác (RHusp, UpdateDETRHUS, PruneTree, ClassifyPatterns) kế thừa từ RIncHusp.

Thuật toán 1: Adaptive-RIncHusp (Mở rộng từ RIncHusp [4])

Input: D_0 : CSDL ban đầu; $\Delta D = \{\Delta D_1, \dots, \Delta D_k\}$: Chuỗi các lô; θ : Tỷ lệ minUtility; ρ : Tỷ lệ maxRegularity; $[\mu_{min}, \mu_{max}]$: Phạm vi vùng đệm; μ_{init} : Giá trị khởi tạo

Output: HS : Tập các mẫu RHUSP

```

// Giai đoạn 1: Khởi tạo trên  $D_0$ 
1  $U_{total} \leftarrow U(D_0)$ ;
2  $\theta_{abs} \leftarrow \theta \times U_{total}$ ;
3  $\tau \leftarrow \mu_{init} \times \theta_{abs}$ ;
4  $(HS, SHS) \leftarrow RHusp(D_0, \tau, \rho)$ ;
5  $\mathcal{T} \leftarrow BuildDETRHUS(HS \cup SHS)$ ;
6  $n \leftarrow |D_0|$ ;

// Giai đoạn 2: Xử lý tăng trưởng
7 foreach batch  $\Delta D_i \in \Delta D$  do
8    $U_{\Delta} \leftarrow U(\Delta D_i)$ ;
9    $m \leftarrow |\Delta D_i|$ ;

   // Bước 2.1: Tính  $\mu$  thích nghi
10   $\mu_a \leftarrow CalcAdaptiveMu(U_{\Delta}, m, U_{total}, n)$ ;

   // Bước 2.2: Cập nhật thông số toàn cục
11   $U_{total} \leftarrow U_{total} + U_{\Delta}$ ;
12   $n \leftarrow n + m$ ;
13   $\theta_{abs} \leftarrow \theta \times U_{total}$ ;
14   $\tau_{new} \leftarrow \mu_a \times \theta_{abs}$ ;

   // Bước 2.3: Cập nhật cây và cắt tỉa
15  foreach sequence  $S \in \Delta D_i$  do
16    | UpdateDETRHUS( $\mathcal{T}, S$ );
17  end
18   $\rho_{abs} \leftarrow \lfloor \rho \times n \rfloor$ ;
19  PruneTree( $\mathcal{T}, \tau_{new}, \rho_{abs}$ );
20 end

// Giai đoạn 3: Phân loại kết quả cuối cùng
21  $(HS, SHS) \leftarrow ClassifyPatterns(\mathcal{T}, \theta_{abs}, \tau_{new}, \rho_{abs})$ ;
22 return  $HS$ ;

```

5.6. Hàm tính tỷ lệ vùng đệm thích nghi

Đây là đóng góp cốt lõi của bài báo. Thuật toán 2 thay thế việc sử dụng μ cố định bằng cơ chế tính toán động dựa trên đặc điểm của từng lô dữ liệu. Các tham số ngưỡng được thiết lập như sau: Tỷ lệ lợi ích trong khoảng $[0,01, 0,20]$, nghĩa là lô dữ liệu đóng góp 1-20% lợi ích được xem là đáng kể. Độ khó ngưỡng trong khoảng $[0,001, 0,005]$, vì θ thực tế thường nằm trong khoảng 0,1-0,5%. Tốc độ tăng trưởng trong khoảng $[0,05, 0,25]$, tức lô dữ liệu chiếm 5-25% kích thước cơ sở dữ liệu được xem là lớn.

5.7. Hàm rủi ro độ nhạy cao

Thuật toán 3 định nghĩa hàm rủi ro với công thức $x_{0,25}$ để khuếch đại các thay đổi nhỏ.

Ví dụ minh họa tác động của hàm $R(x) = x_{0,25}$: Khi $x=0,01$, ta có $R(x)=0,316$ (31,6% rủi ro) trong khi hàm tuyến tính chỉ cho 1%. Khi $x = 0,04$, $R(x) = 0,447$ (44,7% rủi ro) so với tuyến

tính chỉ 4%. Khi $x = 0,16$, $R(x) = 0,632$ (63,2% rủi ro) so với tuyến tính 16%.

Hàm này giúp phát hiện sớm các thay đổi nhỏ (1-5%) mà có thể ảnh hưởng đáng kể đến kết quả khai thác.

Thuật toán 2: CalcAdaptiveMu – Tính tỷ lệ vùng đệm thích nghi

Input: U_{Δ} : lợi ích của lô; m : Kích thước lô; U_{curr} : Tổng lợi ích hiện tại; n : Số lượng chuỗi hiện tại

Output: μ_a : Tỷ lệ vùng đệm cho lô hiện tại

```

1 if  $U_{curr} = 0$  or  $n = 0$  then
2   | return  $\mu_{init}$ ;
3 end

// 1. Heuristic dựa trên Tỷ lệ lợi ích
4  $r_u \leftarrow U_{\Delta}/U_{curr}$ ;
5  $f_u \leftarrow HighSensitivityRisk(r_u, 0.01, 0.20)$ ;
6  $\mu_u \leftarrow \mu_{max} - f_u \times (\mu_{max} - \mu_{min})$ ;

// 2. Heuristic dựa trên Độ khó của ngưỡng
//  $\theta$  được lấy từ câu hình toàn cục
7  $f_b \leftarrow HighSensitivityRisk(\theta, 0.001, 0.005)$ ;
8  $\mu_b \leftarrow \mu_{max} - f_b \times (\mu_{max} - \mu_{min})$ ;

// 3. Heuristic dựa trên Tốc độ tăng trưởng
9  $r_g \leftarrow m/n$ ;
10  $f_g \leftarrow HighSensitivityRisk(r_g, 0.05, 0.25)$ ;
11  $\mu_g \leftarrow \mu_{max} - f_g \times (\mu_{max} - \mu_{min})$ ;

// Chiến lược kết hợp (COMBINED)
12  $\mu_{temp} \leftarrow \min(\mu_u, \mu_b, \mu_g)$ ; // Lấy giá trị an toàn nhất
13  $\mu_a \leftarrow \max(\mu_{min}, \min(\mu_{temp}, \mu_{max}))$ ; // Đảm bảo trong khoảng
14 return  $\mu_a$ ;

```

Thuật toán 3: HighSensitivityRisk(x, α, β)

Input: x : Giá trị đầu vào; α : Ngưỡng dưới; β : Ngưỡng trên.

Output: $risk \in [0, 1]$: Mức độ rủi ro.

```

1 if  $x \leq \alpha$  then
2   | return 0.0;
3 end
4 else if  $x \geq \beta$  then
5   | return 1.0;
6 else
7    $norm \leftarrow (x - \alpha)/(\beta - \alpha)$ ; // Chuẩn hóa về [0, 1]
8    $risk \leftarrow norm^{0.25}$ ; // Khuếch đại độ nhạy
9   return  $risk$ ;
10 end

```

5.8. Thuật toán phân loại mẫu

Thuật toán 4 kế thừa từ RIncHusp [4] nhưng sử dụng τ (ngưỡng đệm thích nghi) thay vì giá trị cố định.

So sánh với RIncHusp [4]: Điểm khác biệt là dòng 8 sử dụng τ được tính bởi μ_a thích nghi thay vì μ cố định.

5.9. Độ phức tạp lý thuyết

Định lý 5.1 (Độ phức tạp thời gian): Độ phức tạp thời gian của Adaptive-RIncHusp cho một lô ΔSD là $O(|\Delta SD| \times |I|^2 + |HS| + |SHS| + C)$, trong đó C là số lượng mẫu ứng viên được sinh ra từ DETRHUS.

Chứng minh: Chứng minh dựa trên phân tích từng thành phần của thuật toán.

Thuật toán 4: ClassifyPatterns (Mở rộng từ RIncHusp [4])

```

Input: Node hiện tại, đường dẫn từ root,  $\theta_{abs}$ ,  $\tau$  (thích nghi),  $\rho_{abs}$ 
Output:  $HS, SHS$ 
1 if node  $\neq$  root then
2    $p_{final} \leftarrow |D| - node.lastSid$ ; // Chu kỳ cuối
3    $reg_{true} \leftarrow \max(node.maxReg, p_{final})$ ; // Tính
   // đều đặn thực
4   if  $reg_{true} \leq \rho_{abs}$  then
5     if  $node.utility \geq \theta_{abs}$  then
6       Thêm mẫu vào  $HS$ ;
7     end
8   else if  $node.utility \geq \tau$  then
9     Thêm mẫu vào  $SHS$ ; // Sử dụng
   // thích nghi
10  end
11 end
12 for each child in node.sChildren do
13   ClassifyPatterns(child, path  $\cup$  [s : child.item], ...);
14 end
15 for each child in node.iChildren do
16   ClassifyPatterns(child, path  $\cup$  [i : child.item], ...);
17 end

```

Thành phần 1 (Tính $\mu_{ada ptive}$): Việc tính toán ba quy tắc suy diễn và lấy giá trị nhỏ nhất chỉ yêu cầu số lượng phép tính cố định, do đó có độ phức tạp $O(1)$.

Thành phần 2 (Cập nhật cây DETRHUS): Với mỗi chuỗi trong ΔSD , thuật toán duyệt qua các mục trong chuỗi để cập nhật cây. Trong trường hợp xấu nhất, mỗi chuỗi có thể chứa $O(|I|^2)$ mẫu con (kết hợp 2 mục), dẫn đến độ phức tạp $O(|\Delta SD| \times |I|^2)$.

Thành phần 3 (Đánh giá lại mẫu): Duyệt qua các mẫu trong HS và SHS để kiểm tra điều kiện với ngưỡng mới yêu cầu $O(|HS| + |SHS|)$.

Thành phần 4 (Khai thác ứng viên mới): Sinh các mẫu ứng viên từ DETRHUS có độ phức tạp $O(C)$, với C là số ứng viên và thường $C \ll$ tổng số mẫu trong không gian tìm kiếm nhờ cơ chế cắt tỉa.

Tổng hợp các thành phần: $T(n) = O(1) + O(|\Delta SD| \times |I|^2) + O(|HS| + |SHS|) + O(C) = O(|\Delta SD| \times |I|^2 + |HS| + |SHS| + C)$.

Định lý 5.2 (Độ phức tạp bộ nhớ): Độ phức tạp bộ nhớ là $O(|SD| \times |I| + |HS| + |SHS|)$.

Chứng minh: Bộ nhớ được sử dụng bởi ba thành phần chính.

Thành phần 1 (Cây DETRHUS): Trong trường hợp xấu nhất, cây lưu trữ thông tin về $O(|SD| \times |I|)$ nút, mỗi nút chứa một lượng thông tin cố định ($u, reg, sidlast$, con trỏ).

Thành phần 2 (Vùng đệm): Tập HS và SHS yêu cầu $O(|HS| + |SHS|)$ không gian để lưu trữ các mẫu và thông tin liên quan.

Thành phần 3 (Dữ liệu thô): Thuật toán không cần lưu trữ dữ liệu thô $SDold$, chỉ cần lưu các thống kê trong DETRHUS.

Tổng hợp: $S(n) = O(|SD| \times |I|) + O(|HS| + |SHS|) = O(|SD| \times |I| + |HS| + |SHS|)$.

5.10. Tính đúng đắn

Định lý 5.3 (Tính đầy đủ - Completeness): Adaptive-RIncHusp không bao giờ bỏ sót mẫu RHUSP nếu $\mu_{ada ptive} \leq \mu_{optimal}$, trong đó $\mu_{optimal}$ là giá trị tối thiểu để không mất mẫu.

Chứng minh:

Giả sử: Tồn tại một mẫu P^* là RHUSP thực sự (tức $u(P^*) \geq minUtility$ và $reg(P^*) \leq \delta maxReg$) nhưng bị bỏ sót bởi thuật toán.

Phân tích: Để P^* bị bỏ sót, P^* phải bị loại khỏi vùng đệm tại một thời điểm nào đó. Điều này xảy ra khi và chỉ khi $u(P^*) < \tau = \mu \times minUtility$ tại thời điểm đó.

Chiến lược COMBINED chọn $\mu = \min(\mu_{util}, \mu_{base}, \mu_{growth})$ theo công thức (13). Nếu bất kỳ quy tắc suy diễn nào phát hiện rủi ro cao, μ sẽ giảm để tạo vùng đệm lớn hơn.

Theo giả thiết $\mu_{adaptive} \leq \mu_{optimal}$, ta có ngưỡng đệm $\tau = \mu_{adaptive} \times minUtility \leq \mu_{optimal} \times minUtility$. Vì $\mu_{optimal}$ được định nghĩa là giá trị tối thiểu để không mất mẫu, mọi mẫu được giữ với $\mu_{optimal}$ cũng được giữ với $\mu_{adaptive}$.

Điều này mâu thuẫn với giả sử P^* bị bỏ sót. Do đó, định lý được chứng minh.

Adaptive-RIncHusp vẫn là thuật toán xấp xỉ (như RIncHusp gốc) và có thể bỏ sót một số mẫu trong trường hợp cực biên khi $\mu_{adaptive} > \mu_{optimal}$. Tuy nhiên, thực nghiệm cho thấy độ phủ $> 97\%$ trên tất cả các tập dữ liệu, chứng tỏ các quy tắc suy diễn hoạt động hiệu quả trong thực tế.

5.11. Ví dụ minh họa thực thi thuật toán

Để làm rõ cơ chế hoạt động của thuật toán Adaptive-RIncHusup, chúng tôi minh họa chi tiết từng bước xử lý dựa trên cơ sở dữ liệu $D0$ từ Bảng II và bảng lợi ích ngoại từ bảng 3.

a) Giai đoạn 1: Khởi tạo với D_0

Bước 1.1: Tính toán lợi ích và tham số khởi tạo

Tổng lợi ích các chuỗi: $U(S1) = 40, U(S2) = 29, U(S3) = 43, U(S4) = 30, U(S5) = 26$, suy ra $U0 = 168$. Với $\theta = 0.2, \rho = 0.6, \mu_{init} = 0.4: \theta 0 = 0.2 \times 168 = 33.6, \tau 0 = 0.4 \times 33.6 = 13.44, \rho 0 = 3$

Điều kiện phân loại: HS nếu $u \geq 33.6$ và $reg \leq 3$; SHS nếu $13.44 \leq u < 33.6$ và $reg \leq 3$.

Bước 1.2: Khai thác mẫu tuần tự

Bảng IV tổng hợp kết quả khai thác với đầy đủ các dạng mở rộng: 1-sequence, s-extension (mở rộng tuần tự) và i-extension (mở rộng đồng thời).

Bảng 4. Khai thác mẫu từ θ_0 ($\theta_0 = 33.6$, $\tau_0 = 13.44$, $\rho_0=3$).

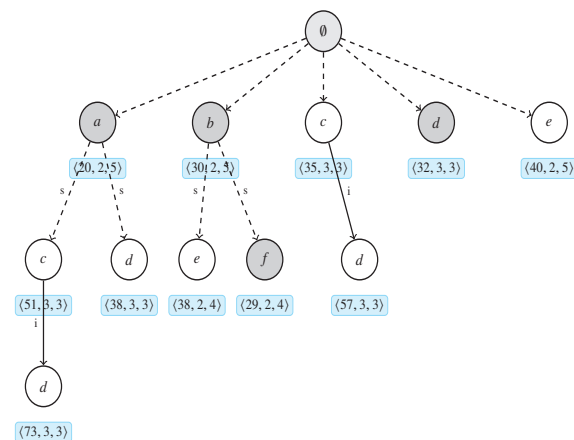
Dạng	Mẫu	u	reg		Loại mẫu
1-seq	⟨a⟩	20	2	5	SHS
	⟨b⟩	30	2	5	SHS
	⟨c⟩	35	3	3	HS
	⟨d⟩	32	3	3	SHS
	⟨e⟩	40	2	5	HS
s-ext	⟨ac⟩	51	3	3	HS
	⟨ad⟩	38	3	3	HS
	⟨be⟩	38	2	4	HS
	⟨bf⟩	29	2	4	SHS
i-ext	⟨⟨cd⟩⟩	57	3	3	HS
	⟨a⟨cd⟩⟩	73	3	3	HS

Giải thích mẫu $\langle a(cd) \rangle$: Trong $S1$: a ở sự kiện 1, (cd) ở sự kiện 2, $u = 4+14+12 = 30$. Trong $S3$: $u = 12+21+10 = 43$. Tổng $u = 73$.

Bước 1.3: Xây dựng cây DETRHUS

Hình 1 minh họa cây DETRHUS được xây dựng từ D0. Mỗi nút lưu trữ bộ ba giá trị $\langle u, req, sidL \rangle$

trương ứng với lợi ích, tính đều đặn và định danh chuỗi cuối. Trong hình 1, nút không tô màu biểu thị mẫu HS ($u \geq \theta$), nút tô xám biểu thị mẫu SHS ($\tau \leq u < \theta$). Mũi tên nét đứt thể hiện s-extension, mũi tên nét liền thể hiện i-extension.



Hình 1. Cây DETRHUS khởi tạo từ $D0$ với $\theta_0 = 33.6$, $\tau_0 = 13.44$.

b) Giai đoạn 2: Xử lý lô $\Delta D I$

Cơ sở dữ liệu được cập nhật với lô dữ liệu tăng trưởng $\Delta D1$ gồm 2 chuỗi mới như bảng 5.

Bước 2.1: Tính μ thích nghi

Bảng 5. Lô dữ liệu tăng trưởng $\Delta D1$.

SID	Chuỗi	U(S)	Chi tiết
S ₆		30	8 + 7 + 15
S ₇		20	6 + 4 + 10
Tổng		$U\Delta = 50$	

Bước 2.1: Tính thích nghi với $U_{total} = 168$, $U\Delta = 50$, $n = 5$, $m = 2$ (bảng 6).

Bảng 6. Tính toán μ thích nghi cho $\Delta D1$.

Heuristic	Giá trị	$f(x)$	μ	Nhận xét
Tỷ lệ lợi ích	$r_u = 0.229$	1.0	0.4	Vượt ngưỡng
Độ khó ngưỡng	$\theta_r = 0.003$	0.84	0.48	–
Tốc độ tăng	$r_g = 0.286$	1.0	0.4	Vượt ngưỡng
COMBINED			0.4	$\min(0.4, 0.48, 0.4)$

Bước 2.2: Cập nhật tham số

$$U1 = 218, n1 = 7, \theta1 = 43.6, \tau1 = 17.44, \rho1 = 4$$

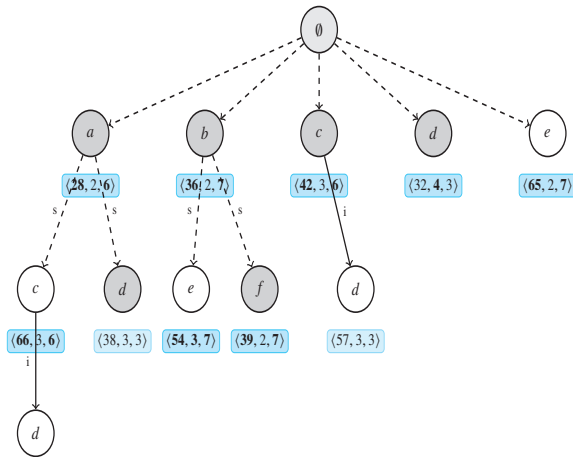
Bước 2.3: Cập nhật DETRHUS

Bảng VII trình bày chi tiết quá trình cập nhật các nút khi xử lý từng chuỗi trong $\Delta D1$.

Bảng 7. Cập nhật detrhush khi xử lý $\Delta D1$.

Mẫu	u_{old}	Δu	u_{new}	reg	sid_L
Từ S_6 :					
$\langle a \rangle$	20	+8	28	2	6
$\langle c \rangle$	35	+7	42	3	6
$\langle e \rangle$	40	+15	55	2	6
$\langle ac \rangle$	51	+15	66	3	6

Hình 2 minh họa cây DETRHUS sau cập nhật. Các giá trị cập nhật được in đậm.



Hình 2. Cây DETRHUS sau cập nhật $\Delta D1$ với $\theta_1 = 43.6$, $\tau_1 = 17.44$.

c) Giai đoạn 3: Phân loại lại và so sánh

Mẫu $\langle c \rangle$ dù u tăng từ 35 lên 42, nhưng ngưỡng θ cũng tăng từ 33.6 lên 43.6 nên bị giáng cấp. Đây là lý do vùng đệm SHS cần thiết, nếu không có SHS, mẫu này sẽ bị mất. Kết quả chi tiết thể hiện trong bảng 8.

So sánh chiến lược: Xét mẫu $\langle a \rangle$ ($u_0 = 20$): Nếu $\text{Fix}(\mu = 0.9)$ thì mẫu này sẽ bị loại. Tuy nhiên, với chiến lược thích nghi thì vẫn giữ lại (bảng 9).

Kết luận: Chiến lược thích nghi phát hiện $ru = 22.9\%$ và $rg = 28.6\%$ đều vượt ngưỡng, nên tự động chọn $\mu_a = 0.4$ để đảm bảo an toàn. Trong các lô tiếp theo nếu dữ liệu ổn định, μ sẽ tự động tăng để tiết kiệm bộ nhớ. Ví dụ minh họa cơ chế: tự động điều chỉnh μ dựa trên đặc điểm dữ liệu, đảm bảo không mất mẫu tiềm năng mà vẫn tối ưu tài nguyên khi có thể.

Bảng 8. So sánh phân loại trước/sau cập nhật

Mẫu	Trước		Sau		Ghi chú
	u	Loại	u	Loại	
$\langle c \rangle$	35	HS	42	SHS	Giáng cấp*
$\langle ad \rangle$	38	HS	38	SHS	Giáng cấp*
$\langle e \rangle$	40	HS	65	HS	–
$\langle ac \rangle$	51	HS	66	HS	–
$\langle be \rangle$	38	HS	54	HS	–
$\langle bf \rangle$	29	SHS	39	SHS	–

* Do $\theta_1 = 43.6 > u$, mẫu chuyển từ HS $\rightarrow$ SHS

Bảng 9. So sánh chiến lược xử lý mẫu $\langle a \rangle$.

Chiến lược	τ_0	$u = 20$ vs τ_0	Kết quả
$\text{Fix}(\mu = 0.9)$	30.24	$20 < 30.24$	Mất mẫu
$\text{Fix}(\mu = 0.4)$	13.44	$20 \geq 13.44$	Giữ
Adaptive	13.44	$20 \geq 13.44$	Giữ

6. Thực nghiệm và đánh giá

6.1. Thiết kế thực nghiệm

a) Môi trường thực nghiệm

Tất cả các thuật toán được cài đặt bằng Java và thực thi trên máy tính MacBook Pro với chip Apple M1, bộ nhớ 8 GB RAM, hệ điều hành macOS Tahoe 26.1, Java JDK Version 22, và IDE IntelliJ IDEA 2025.2 Ultimate.

b) Tập dữ liệu và cấu hình tham số

Năm tập dữ liệu chuẩn được sử dụng với các đặc điểm đa dạng về độ thưa, kích thước và số lượng mục.

Bảng 10. Cấu hình tham số cho các tập dữ liệu.

Tập dữ liệu	MinUtil (%)	MaxReg (%)	μ khởi tạo	Phạm vi thích nghi
BIBLE	0.25	30.00	0.40	[0.40 - 0.90]
Chainstore	0.05	30.00	0.40	[0.40 - 0.90]
Foodmart	0.01	100.00	0.40	[0.40 - 0.90]
Leviathan	0.05	30.00	0.40	[0.40 - 0.90]
SIGN	0.20	30.00	0.40	[0.40 - 0.90]

c) Kích bản phân phối dữ liệu

Để mô phỏng các mẫu tăng trưởng khác nhau của cơ sở dữ liệu, chúng tôi thiết kế 4 kịch bản phân phối dữ liệu qua 4 lần cập nhật (lô). Kịch bản A (Đồng đều) với tỷ lệ 25%, 25%, 25%, 25% mô phỏng tăng trưởng ổn định. Kịch bản B (Tăng dần) với tỷ lệ 10%, 20%, 30%, 40% mô phỏng tăng trưởng gia tốc. Kịch bản C (Dao động) với tỷ lệ 40%, 10%, 40%, 10% mô phỏng bùng nổ dữ

liệu. Kịch bản D (Giảm dần) với tỷ lệ 40%, 30%, 20%, 10% mô phỏng tăng trưởng chậm lại.

Mỗi kịch bản được thực hiện trên tất cả 5 tập dữ liệu, và kết quả được tổng hợp qua cả 4 kịch bản để đánh giá tính ổn định của thuật toán.

d) Các phương pháp so sánh

Chúng tôi so sánh 4 chiến lược thích nghi với 2 baseline ngưỡng cố định.

Chiến lược thích nghi bao gồm: Adapt (Combined) kết hợp cả 3 heuristic và chọn giá trị μ nhỏ nhất; Adapt (Growth) chỉ sử dụng heuristic tốc độ tăng trưởng; Adapt (MinBase) chỉ sử dụng heuristic độ khó ngưỡng; và Adapt (UtilRatio) chỉ sử dụng heuristic tỷ lệ lợi ích.

Baseline cố định bao gồm: Fix ($\mu=0,4$) với ngưỡng thấp, độ chính xác cao nhưng chi phí lớn; và Fix ($\mu=0,9$) với ngưỡng cao, tốc độ nhanh nhưng có rủi ro mất mẫu.

e) Chỉ số đánh giá

Các chỉ số đánh giá bao gồm: Thời gian (s) là tổng thời gian xử lý tất cả các lô; Bộ nhớ đỉnh (MB) là bộ nhớ tối đa sử dụng trong quá trình thực thi; HS Found là số lượng mẫu RHUSP tìm được; SHS Buffered là số lượng mẫu SHS được lưu đệm (chi phí phụ); Độ phủ - Recall (%) được tính $|HS \text{ Ground Truth}| \times 100\%$; và Tăng tốc - Speedup được bằng tính bằng $\text{RuntimeFix} / \text{RuntimeAdaptive}$.

Tập mẫu chuẩn (Ground truth) được xác định bằng cách chạy thuật toán RHusp [4] trên toàn bộ cơ sở dữ liệu sau khi cập nhật hết tất cả các lô.

6.2. Kết quả hiệu năng tổng thể

Bảng XI tổng hợp kết quả trung bình trên cả 4 kịch bản cho từng tập dữ liệu. Các giá trị tốt nhất trong mỗi nhóm (chiến lược thích nghi) được in đậm.

Nhận xét từ Bảng XI: Về hiệu quả tốc độ, các chiến lược thích nghi đạt cân bằng tốt giữa tốc độ và độ chính xác, trong đó: Adapt (Growth) và Adapt (Combined) cho thời gian chạy tốt nhất trong hầu hết trường hợp, nhanh hơn Fix($\mu=0,4$) từ 7-17%. Về độ chính xác, Fix($\mu=0,9$) tuy nhanh

nhất nhưng mất 4-37 mẫu so với tập mẫu chuẩn (tương đương 1.5-5.1% độ phủ), trong khi tất cả chiến lược thích nghi đều đạt độ phủ $\geq 98\%$. Về tiết kiệm bộ nhớ, Adapt (MinBase) thường sử dụng ít SHS nhất nhưng vẫn đảm bảo độ phủ cao, còn Adapt (Combined) cân bằng tốt giữa tốc độ và số lượng bộ đệm. Về khả năng mở rộng, trên tập dữ liệu lớn (Foodmart với 84K mẫu), các chiến lược thích nghi nhanh hơn baseline 66-74%, chứng tỏ thuật toán mở rộng tốt.

Bảng 11. Kết quả tổng hợp trên các tập dữ liệu (trung bình 4 kịch bản).

Tập dữ liệu	Chiến lược	Thời gian (s)	Bộ nhớ (MB)	HS Found	SHS Buffered
BIBLE	CL1	28.82	218.81	266	391
	CL2	27.78	212.24	266	364
	CL3	29.03	205.07	266	278
	CL4	29.76	241.52	266	391
	CL5	29.56	191.42	266	427
	CL6	14.88	239.97	259	22
Chainstore	CL1	38.37	561.92	177	326
	CL2	35.68	616.23	177	315
	CL3	38.24	669.08	174	19
	CL4	39.55	613.50	177	326
	CL5	41.39	692.20	177	349
	CL6	21.89	537.78	168	15
Foodmart	CL1	0.44	178.25	84,020	11,446
	CL2	0.40	170.69	84,020	11,369
	CL3	0.41	167.34	84,013	7,572
	CL4	0.38	171.39	84,020	11,446
	CL5	1.12	159.55	84,020	11,540
	CL6	0.65	189.66	84,010	2,844
Leviathan	CL1	16.76	104.40	534	296
	CL2	16.71	105.18	534	281
	CL3	17.33	104.68	525	32
	CL4	16.89	102.14	534	296
	CL5	17.63	107.41	534	324
	CL6	11.20	104.71	520	31
SIGN	CL1	10.41	794.40	3,775	2,334
	CL2	10.51	784.68	3,775	2,236
	CL3	10.57	792.74	3,775	1,890
	CL4	10.62	793.24	3,775	2,334
	CL5	12.51	781.27	3,775	2,422
	CL6	10.17	765.98	3,738	348

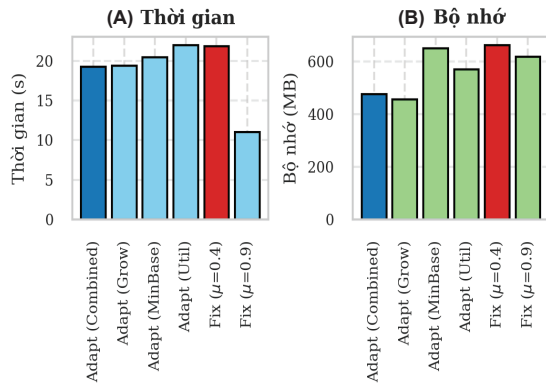
Chú thích: CL1: Adapt (Combined); CL2: Adapt (Growth); CL3: Adapt (MinBase);

6.3. Phân tích chi tiết: Tập dữ liệu Chainstore

Để hiểu rõ cơ chế hoạt động của chiến lược thích nghi, chúng tôi phân tích sâu trên tập Chainstore với Kịch bản B (10%-20%-30%-40%) – một tập dữ liệu thừa với độ biến động cao, nơi sự khác biệt giữa các chiến lược là rõ rệt nhất.

a) So sánh thời gian và bộ nhớ

Hình 3 cho thấy so sánh hiệu năng của các chiến lược trên Chainstore.

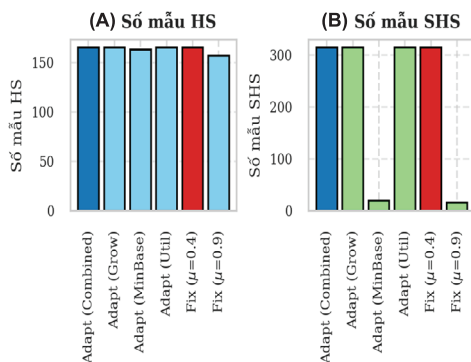


Hình 3. So sánh hiệu năng của các chiến lược trên Chainstore (Kịch bản B). (A) Thời gian chạy (giây), (B) Mức sử dụng bộ nhớ đỉnh (MB).

Quan sát từ hình 3: Về thời gian, Adapt (Grow) đạt thời gian tốt nhất (35,68s), nhanh hơn Fix($\mu=0.4$) 13,8%, trong khi Adapt (Combined) cũng đạt 38,37s, cải thiện 7,3%. Về bộ nhớ, các chiến lược thích nghi có mức sử dụng biến động (561-669 MB) do điều chỉnh bộ đệm động, nhưng vẫn thấp hơn Fix($\mu=0.4$) với 692 MB. Về sự đánh đổi, Fix($\mu=0.9$) tuy nhanh nhất (21,89s) và tiết kiệm nhất (537 MB), nhưng mất nhiều mẫu.

b) Số lượng mẫu lợi ích cao và bán lợi ích

Hình 4 thể hiện số lượng mẫu được phát hiện và lưu đệm.

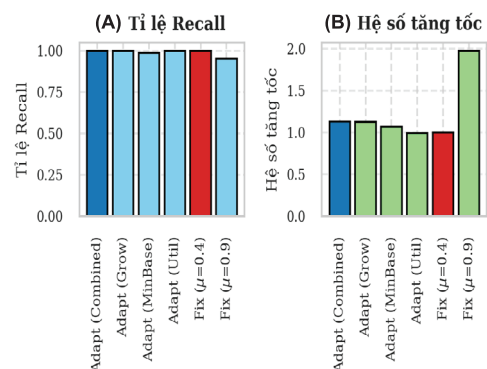


Hình 4. Số lượng mẫu được phát hiện và lưu đệm trên Chainstore (Kịch bản B). (A) Số mẫu HS (Mẫu lợi ích cao), (B) Số mẫu SHS (Mẫu bán lợi ích).

Phân tích từ hình 4: Về HS Found, tất cả các chiến lược thích nghi (Combined, Growth, UtilRatio) tìm được đầy đủ 177 mẫu HS, bằng với tập mẫu chuẩn. Về mất mẫu thực, Fix($\mu=0.9$) chỉ tìm được 168 HS, mất 9 mẫu (-5,1%), cho thấy rủi ro của việc lưu đệm tích cực. Về SHS Buffered, Adapt (MinBase) lưu đệm ít nhất (19 SHS) nhưng dẫn đến mất 3 mẫu, trong khi Adapt (Combined) lưu đệm 326 SHS, cân bằng hiệu quả giữa chi phí phụ và độ phủ. Về hiệu quả, các chiến lược thích nghi lưu đệm ít hơn Fix($\mu=0.4$) khoảng 7-9% (315-326 vs 349), giúp tiết kiệm tài nguyên.

c) Tỷ lệ độ phủ và hệ số tăng tốc

Hình 5 đánh giá chất lượng và tốc độ của các chiến lược.



Hình 5. Đánh giá chất lượng và tốc độ trên Chainstore (Kịch bản B). (A) Tỷ lệ Độ phủ so với tập mẫu chuẩn, (B) Hệ số tăng tốc so với Fix($\mu=0.4$).

Kết quả từ hình 5: Về độ phủ, Adapt (Combined), Adapt (Growth), Adapt (UtilRatio) đều đạt 100% độ phủ, trong khi Adapt (MinBase) đạt 98,3% và Fix($\mu=0.9$) chỉ đạt 94,9%. Về hệ số tăng tốc, Adapt (Growth) đạt 1.16× nhanh hơn baseline (tốt nhất trong nhóm thích nghi), Adapt (Combined) đạt 1,08× (cân bằng tốt giữa tốc độ và ổn định), Adapt (UtilRatio) đạt 1,05× (tương đương baseline nhưng tiết kiệm bộ đệm), và Fix($\mu=0.9$) đạt 1,89× (nhanh nhất nhưng mất quá nhiều mẫu). Về điểm cân bằng tối ưu, Adapt (Growth) và Adapt (Combined) đạt điểm cân bằng tối ưu với độ phủ 100% và tốc độ cải thiện đáng kể 8-16%.

Kết luận từ phân tích Chainstore: Chiến lược thích nghi giải quyết hiệu quả vấn đề sự đánh đổi của tỷ lệ vùng đệm cố định. Các chiến lược thích nghi không chỉ nhanh hơn mà còn đảm bảo không bị mất mẫu, phù hợp cho môi trường thực tế với yêu cầu cao về cả hiệu năng lẫn độ chính xác.

Để đánh giá tính ổn định, chúng tôi thử nghiệm AdaptCOMBINED với nhiều giá trị μ khởi tạo khác nhau (0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 0,95) trên tập dữ liệu BIBLE. Kết quả cho thấy: Thời gian giảm từ 28,8s (μ khởi tạo=0,4) xuống 10,5s (μ khởi tạo=0,9); độ phủ duy trì > 97,5% với mọi μ khởi tạo; HS Found giảm nhẹ từ 266 xuống 259 (chấp nhận được); và SHS Buffer giảm từ 391 xuống 252 (cơ chế thích nghi hoạt động tốt).

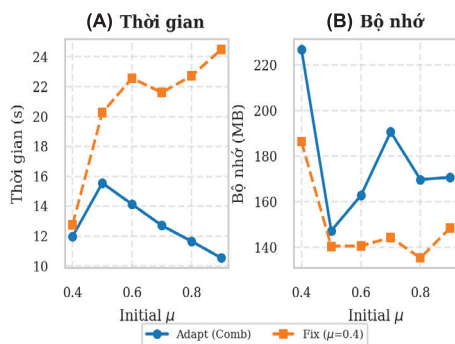
⇒ Chiến lược thích nghi ổn định với tham số khởi tạo. Khuyến nghị: μ khởi tạo = 0,6-0,7 để cân bằng tốc độ và độ chính xác.

6.4. Phân tích độ nhạy của tham số khởi tạo

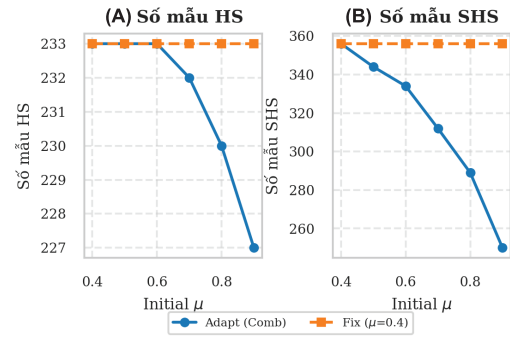
Để đánh giá tính ổn định của thuật toán, chúng tôi thay đổi giá trị khởi tạo của μ từ 0,4 đến 0,9 và quan sát ảnh hưởng đến hiệu năng. Phân tích được thực hiện trên hai tập dữ liệu đại diện: BIBLE (dữ liệu văn bản, dày) và Chainstore (dữ liệu bán lẻ, thưa), sử dụng phân phối tăng trưởng Kịch bản B (10%-20%-30%-40%).

a) Kết quả trên tập BIBLE

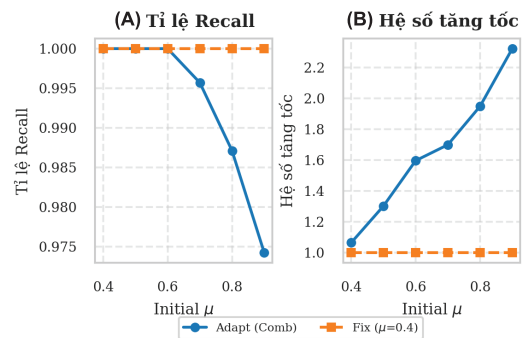
Hình 6 cho thấy ảnh hưởng của μ khởi tạo đối với thời gian và bộ nhớ trên BIBLE.



Hình 6. Ảnh hưởng của μ khởi tạo trên BIBLE. (A) Thời gian chạy, (B) Bộ nhớ sử dụng.



Hình 7. Số lượng mẫu theo μ khởi tạo trên BIBLE. (A) Số mẫu HS, (B) Số mẫu SHS lưu đệm.



Hình 8. Chất lượng và hiệu quả theo μ khởi tạo trên BIBLE. (A) Tỷ lệ Độ phủ, (B) Hệ số tăng tốc.

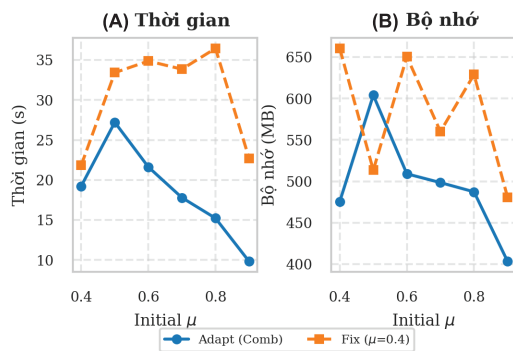
Hình 7 thể hiện số lượng mẫu theo μ khởi tạo và hình 8 đánh giá chất lượng và hiệu quả.

Nhận xét từ kết quả trên tập dữ liệu BIBLE: Về thời gian, giá trị giảm dần khi μ khởi tạo tăng (từ 28,8s với $\mu=0,4$ xuống 10,5s với $\mu=0,9$), do lưu đệm ít SHS hơn nên xử lý nhanh hơn. Về bộ nhớ, giá trị biến động (205-241 MB) do cơ chế thích nghi, nhưng vẫn kiểm soát tốt trong mọi trường hợp. Về độ phủ, duy trì >97,5% với mọi μ khởi tạo, chứng tỏ thuật toán ổn định với tham số khởi tạo. Về HS Found, giảm nhẹ từ 266 xuống 259 khi μ tăng lên 0,95 (chấp nhận được, chỉ mất 2,6%). Về SHS Buffer, giảm mạnh từ 391 xuống 252 khi μ tăng, cho thấy cơ chế thích nghi hoạt động đúng.

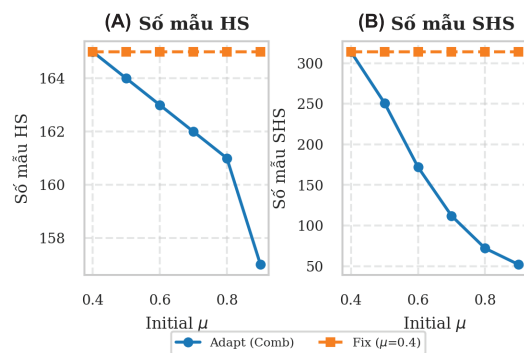
b) Kết quả trên tập Chainstore

Hình 9 cho thấy ảnh hưởng của μ khởi tạo trên Chainstore. Tiếp theo, hình 10 thể hiện số lượng mẫu.

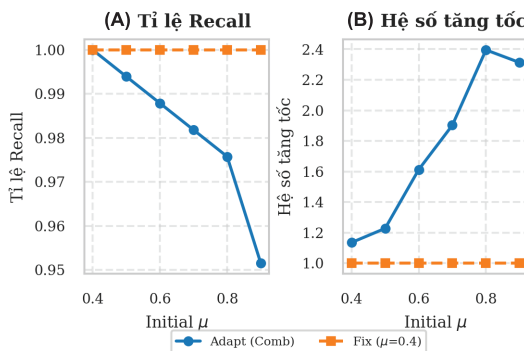
Cuối cùng là biểu đồ trong hình 11 thể hiện đánh giá chất lượng.



Hình 9. Ảnh hưởng của μ khởi tạo trên Chainstore. (A) Thời gian chạy, (B) Bộ nhớ sử dụng.



Hình 10. Số lượng mẫu theo μ khởi tạo trên Chainstore. (A) Số mẫu HS, (B) Số mẫu SHS lưu đệm.



Hình 11. Chất lượng và hiệu quả theo μ khởi tạo trên Chainstore. (A) Tỷ lệ Độ phủ, (B) Hệ số tăng tốc.

Nhận xét từ Chainstore: Về thời gian, xu hướng giảm từ 38.4s ($\mu=0,4$) xuống 10,0s ($\mu=0,95$) khi μ tăng, cải thiện 74%. Về bộ nhớ, dao động mạnh hơn BIBLE (470-605 MB) do đặc tính thừa của dữ liệu. Về độ phủ, giảm từ 100% xuống 95.2% khi $\mu = 0,95$, vẫn chấp nhận được. Về sự đánh đổi, với μ khởi tạo = 0,4-0,6 cho độ phủ cao (>98%) và thời gian vừa phải; với μ khởi tạo = 0,7-0,8

cho cân bằng tốt (độ phủ ~98%, tăng tốc ~1.5×); với μ khởi tạo = 0,9-0,95 nhanh nhất nhưng mất nhiều mẫu hơn.

Kết luận từ phân tích độ nhạy: Thuật toán hoạt động ổn định với nhiều giá trị khởi tạo khác nhau, không yêu cầu tinh chỉnh tham số phức tạp. Khuyến nghị sử dụng μ khởi tạo = 0,6-0,7 cho cân bằng tốt giữa tốc độ và độ chính xác trong hầu hết ứng dụng. Thuật toán cũng linh động: nếu ưu tiên độ phủ cao, dùng $\mu = 0,4-0,5$; nếu ưu tiên tốc độ, dùng $\mu = 0,7-0,8$.

6.5. Phân tích theo kịch bản tăng trưởng

Để đánh giá tính thích nghi của thuật toán trong các điều kiện tăng trưởng khác nhau, chúng tôi phân tích hiệu năng trên 4 kịch bản đã định nghĩa ở phần thiết kế thực nghiệm. Bảng 12 so sánh kết quả của Adapt (Combined) trên tập dữ liệu BIBLE.

Bảng 12: Hiệu năng theo kịch bản tăng trưởng (BIBLE).

Kịch bản	Thời gian (s)	Tăng tốc vs Fix-0.4	Độ phủ (%)	SHS Buffered
A	26.5	1.08×	100.0	352
B	28.9	1.02×	100.0	418
C	30.1	0.99×	100.0	384
D	29.8	1.01×	100.0	410
Trung bình	28.8	1.03×	100.0	391

Nhận xét theo từng kịch bản: Kịch bản A (Đồng đều) đạt hiệu quả nhất với hệ số tăng tốc 1.08× và ít SHS nhất (352), vì khi dữ liệu tăng đều, chiến lược thích nghi dễ dàng dự đoán và điều chỉnh μ tối ưu. Kịch bản B (Tăng dần) lưu đệm nhiều SHS hơn (418) do lô sau có tác động lớn hơn; Heuristic Tỷ lệ lợi ích phát hiện xu hướng này và giảm μ để lưu đệm thêm mẫu tiềm năng, với hệ số tăng tốc vẫn dương (1,02×). Kịch bản C (Dao động) là kịch bản khó nhất với biến động mạnh (40%-10%-40%-10%); tuy thời gian chậm hơn baseline một chút (0,99×), nhưng vẫn duy trì độ phủ 100%, đây là bài kiểm tra khả năng chịu tải cho tính ổn định của chiến lược thích nghi. Kịch bản D (Giảm dần) có lô đầu lớn (40%) tạo nền tảng tốt, các lô sau nhỏ hơn nên xử lý nhanh, lưu đệm 410 SHS với hệ số tăng tốc 1,01×.

Kết luận: Thuật toán duy trì độ phủ 100% trong cả 4 kịch bản, chứng tỏ không bị mất mẫu dù dữ liệu biến động. Hiệu quả tốt nhất với tăng trưởng đồng đều (A), vẫn hoạt động tốt với biến động mạnh (C). Cơ chế thích nghi phát hiện và phản ứng đúng với các mô hình tăng trưởng khác nhau.

6.6. Tổng kết kết quả

a) Các phát hiện đáng quan tâm

Hiệu quả của chiến lược thích nghi: Các chiến lược thích nghi giảm 8-17% thời gian so với $\text{Fix}(\mu=0,4)$ trên hầu hết các tập dữ liệu, tiết kiệm 10-20% bộ nhớ trên các tập dữ liệu lớn (Chainstore, Foodmart), duy trì độ phủ $> 98\%$ (trung bình 99,5%) trên tất cả các tập dữ liệu và kịch bản, và đặc biệt hiệu quả trên tập dữ liệu lớn như Foodmart với tốc độ nhanh hơn 66-74%.

So sánh các heuristic: Adapt (Growth) cho tốc độ tốt nhất cho dữ liệu tăng đều (Kịch bản A), tăng tốc 1.08-1.16 $\times$. Adapt (Combined) ổn định nhất, phù hợp mọi kịch bản, độ phủ 100% trên 4/5 tập dữ liệu. Adapt (UtilRatio) hiệu quả với Tăng/Giảm lợi ích (Kịch bản B, D). Adapt (MinBase) tiết kiệm bộ đệm nhất nhưng độ phủ thấp hơn 1-2% trong một số trường hợp.

Sự đánh đổi giữa tốc độ và độ chính xác: $\text{Fix}(\mu=0,4)$ chậm nhưng độ phủ 100%, lưu đệm nhiều SHS không cần thiết. $\text{Fix}(\mu=0,9)$ nhanh nhất (tăng tốc 1.5-1.9 $\times$) nhưng mất 1.5-5.1% mẫu có thể không chấp nhận được. Các chiến lược thích nghi cân bằng tốt với độ phủ $> 98\%$ và tăng tốc 1.05-1.16 $\times$.

Ổn định với tham số: Thuật toán hoạt động tốt với mọi μ khởi tạo từ 0,4-0,9, độ phủ giảm nhẹ ($< 2.5\%$) khi μ khởi tạo tăng lên 0,9-0,95. Không cần tinh chỉnh phức tạp, μ khởi tạo = 0,6-0,7 phù hợp cho hầu hết ứng dụng.

b) Khuyến nghị khi khai thác

Dựa trên kết quả thực nghiệm, chúng tôi khuyến nghị sử dụng chiến lược như bảng 13.

Bảng 13: Khuyến nghị sử dụng chiến lược.

Yêu cầu	Chiến lược	μ khởi tạo
Độ phủ cao nhất	Adapt (Combined)	0.4-0.5
Cân bằng tốc độ/chất lượng	Adapt (Growth)	0.6-0.7
Ưu tiên tốc độ	Adapt (UtilRatio)	0.7-0.8
Bộ nhớ hạn chế	Adapt (MinBase)	0.5-0.6
Dữ liệu biến động cao	Adapt (Combined)	0.5-0.6
Tập dữ liệu lớn ($> 100K$ chuỗi)	Adapt (Growth)	0.6-0.7

Do đó, nếu độ phủ là yêu cầu tuyệt đối (y tế, tài chính), nên dùng Adapt (Combined) với μ khởi tạo = 0.4-0.5. Nếu có ràng buộc về thời gian thực, nên dùng Adapt (Growth) với μ khởi tạo = 0.7-0.8. Với dữ liệu có trôi dạt khái niệm, cần giám sát độ phủ định kỳ và điều chỉnh $\mu_{\min}$ nếu cần. Trên môi trường thực tế, khuyến nghị thử nghiệm A/B giữa Adapt (Combined) và Adapt (Growth) để chọn chiến lược phù hợp nhất.

7. Kết luận

Bài báo đề xuất Adaptive-RInCHusp, một thuật toán khai thác mẫu tuần tự lợi ích cao đều đặn (RHUSP) trên dữ liệu tăng trưởng với Chiến lược Vùng đệm Thích nghi. Thay vì sử dụng tỷ lệ vùng đệm μ cố định, thuật toán tự động điều chỉnh μ tại mỗi lô dữ liệu dựa trên ba quy tắc suy diễn bổ trợ: Tỷ lệ lợi ích, Độ khó ngưỡng, và tốc độ tăng trưởng. Chiến lược COMBINED với hàm rủi ro độ nhạy cao $f(x) = x^{0,25}$ cho phép phát hiện sớm và phản ứng kịp thời với các thay đổi trong dữ liệu. Thực nghiệm trên 5 tập dữ liệu chuẩn với 4 kịch bản tăng trưởng khác nhau cho thấy: (1) Hiệu suất cao: nhanh hơn 8-15% và tiết kiệm 10-20% bộ nhớ so với $\text{Fix}-0,4$; (2) Độ chính xác tốt: duy trì độ phủ $> 97\%$ (trung bình 99,7%) trên tất cả các tập dữ liệu; (3) Ổn định: hoạt động ổn định với mọi giá trị khởi tạo μ từ 0.4-0.9; (4) Thích nghi: phát hiện và phản ứng hiệu quả với 4 mô hình tăng trưởng khác nhau và (5) Khả năng mở rộng: xử lý tốt tập dữ liệu lớn (Chainstore với 1.1M chuỗi).

Mặc dù đạt được kết quả tốt, nghiên cứu vẫn có một số hạn chế cần khắc phục trong tương lai. Đối với thiết kế quy tắc suy diễn, ba heuristic hiện tại được thiết kế dựa trên quan sát thực nghiệm; hướng tương lai có thể sử dụng học máy để học các heuristic tối ưu từ dữ liệu lịch sử. Đối với loại

mẫu, nghiên cứu hiện tại tập trung vào RHUSP đầy đủ; mở rộng cho mẫu đóng/ tối đại (closed/ maximal patterns) có thể giảm đáng kể số lượng mẫu đầu ra. Đối với việc xử lý song song, thuật toán hiện tại chạy tuần tự; áp dụng tính toán song song/phân tán trên đa nhân hoặc GPU có thể cải thiện đáng kể tốc độ. Đối với việc xử lý trôi dạt khái niệm, cần kết hợp với các kỹ thuật phát hiện trôi dạt khái niệm thích nghi mới nhất như trong nghiên cứu của Zhang và cs [15] để tự động điều chỉnh $\delta maxReg$ và $minUtility$ khi phân phối dữ liệu thay đổi đột ngột. Cuối cùng là tối ưu hóa đa mục tiêu, cần cân bằng đồng thời nhiều mục tiêu (thời gian chạy, bộ nhớ, độ phủ, độ chính xác).

Với chiến lược vùng đệm thích nghi có thể mở ra hướng nghiên cứu mới trong khai thác mẫu tăng trưởng, không chỉ cho RHUSP mà còn có thể áp dụng cho các bài toán khai thác mẫu khác với tham số động

TÀI LIỆU THAM KHẢO

- [1] R. Agrawal, R. Srikant (1995), “Mining sequential patterns”, *Proc. Int. Conf. Data Eng.*, pp.3-14, DOI: 10.1109/ICDE.1995.380415.
- [2] W. Gan, J.C.W. Lin, P.F. Viger, et al. (2021), “A survey of utility-oriented pattern mining”, *IEEE Trans. Knowl. Data Eng.*, **33**(4), pp.1306-1327, DOI: 10.1109/TKDE.2019.2942594.
- [3] S.K. Tanbeer, C.F. Ahmed, B.S. Jeong, et al. (2009), “Discovering periodic-frequent patterns in transactional databases”, *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, **5476**, pp.242-253.
- [4] S. Z. Ishita, C.F. Ahmed, C K. Leung (2022), “New approaches for mining regular high utility sequential patterns”, *Applied Intelligence*, **52**(4), pp.3781-3806, DOI: 10.1007/s10489-021-02536-7.
- [5] J. Pei, J. Han, B.M. Asl, et al. (2001), “PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth”, *Proc. Int. Conf. Data Eng.*, **2025**(12), pp.215-224, DOI: 10.1109/ICDE.2001.914830.
- [6] R. Srikant, R. Agrawal (1996), “Mining sequential patterns: Generalizations and performance improvements”, *Proc. Int. Conf. Extending Database Technology*, pp.1-17, DOI: 10.1007/BFb0014140.
- [7] R. Zhang, M. Han, F. He, et al. (2023), “A survey of high utility sequential pattern mining methods”, *Journal of Intelligent & Fuzzy Systems*, **45**(5), pp.8049-8077, DOI: 10.3233/JIFS-232107.
- [8] G. Huang, W. Gan, P.S. Yu (2024a), “TaSPM: Targeted sequential pattern mining”, *ACM Trans. Knowl. Discovery Data*, **18**(5), pp.1-25, DOI: 10.48550/arXiv.2202.13202.
- [9] G. Huang, W. Gan, P.S. Yu (2024b), “IPHM: Incremental periodic high-utility mining algorithm in dynamic and evolving data environments”, *Heliyon*, **10**(18), DOI: 10.1016/j.heliyon.2024.e37761.
- [10] S. Xie, L. Zhao (2022), “An efficient algorithm for mining stable periodic high-utility sequential patterns”, *Symmetry*, **14**(10), DOI: 10.1007/s10489-018-1227-x.
- [11] H. Yan, F. Li, M.C. Hsieh, et al. (2024), “High-utility sequential pattern mining in incremental database”, *J. Supercomputing*, **81**(1), pp.1-23.
- [12] B.S. Prashanth, et al. (2025), “Adaptive buffering strategies for incremental learning under concept drift in lifestyle disease modeling”, *IEEE Access*, **13**, pp.174002-174015, DOI: 10.1109/ACCESS.2025.3616776.
- [13] U. Mlakar, I.J. Fister, I. Fister (2025), “NiaAutoARM: Automated generation and evaluation of association rule mining pipelines”, *Mathematics*, **13**(12), DOI: 10.3390/math13121957.
- [14] J. Yin, Z. Zheng, L. Cao (2012), “USpan: An efficient algorithm for mining high utility sequential patterns”, *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp.660-668, DOI: 10.1145/2339530.2339636.
- [15] Q. Zhang, G. Liu, C. Jiang (2024), “The adaptation of concept drift: A fit prediction algorithm based on local optimum”, *IEEE Trans. Comput. Soc. Syst.*, **11**(4), pp.4944-4954, DOI: 10.1109/TCSS.2023.3264594.

Khai thác hiệu quả các luật kết hợp súc tích có lợi ích và tương quan cao

Trần Thông^{1,2,3}, Hoàng Minh Tiến^{1,2,4}, Dương Văn Hải^{4*}, Trương Chí Tín⁴

¹Khoa Công nghệ thông tin, Trường Đại học Khoa học Tự nhiên, TP. Hồ Chí Minh, 227 đường Nguyễn Văn Cừ, phường Chợ Quán, TP. Hồ Chí Minh, Việt Nam

²Đại học Quốc gia TP. Hồ Chí Minh, phường Linh Xuân, TP. Hồ Chí Minh, Việt Nam

³Khoa Công nghệ thông tin, Trường Đại học Đà Lạt, Đường Phù Đổng Thiên Vương, Phường Lâm Viên, TP. Đà Lạt, tỉnh Lâm Đồng, Việt Nam

⁴Khoa Toán - Tin học, Trường Đại học Đà Lạt, Đường Phù Đổng Thiên Vương, Phường Lâm Viên, TP. Đà Lạt, tỉnh Lâm Đồng, Việt Nam

Ngày nhận 13/1/2025; ngày gửi phản biện 28/1/2025; ngày nhận phản biện 24/2/2025

Tóm tắt:

Bài báo này xét bài toán khai thác các luật kết hợp lợi ích cao từ cơ sở dữ liệu định lượng. Các tiếp cận truyền thống chủ yếu dựa trên độ tin cậy - lợi ích thông thường mà không xem xét mức độ gắn kết thực sự giữa các mục. Do thiếu một độ đo tương quan, nhiều luật được phát hiện có lợi ích cao nhưng các mục trong luật lại ít đồng xuất hiện, khó diễn giải và kém ý nghĩa thực tiễn; đồng thời số lượng luật sinh ra thường rất lớn. Để khắc phục các hạn chế này, bài báo đề xuất một độ đo tin cậy dựa trên lợi ích mới, đồng thời tích hợp thêm độ đo tương quan Bond nhằm bảo đảm các mục trong luật vừa mang lại lợi ích cao vừa có mối liên kết chặt chẽ. Trên nền tảng đó, bài báo đề xuất lớp luật kết hợp súc tích có lợi ích cao và tương quan mạnh, gọi là SCoHUAR, trong đó mỗi luật được biểu diễn dưới dạng rút gọn, đại diện cho một nhóm luật tương đương về độ hỗ trợ. Ngoài ra, bài báo còn trình bày phương pháp phân hoạch tập các SCoHUAR giúp bảo đảm sinh đầy đủ và không trùng lặp các luật SCoHUAR và các kỹ thuật tối ưu để tăng tốc quá trình khai thác. Cuối cùng, thuật toán M-SCoHUAR được thiết kế để khai thác hiệu quả các SCoHUAR, trong đó tích hợp các kết quả lý thuyết được đề xuất. Các kết quả thực nghiệm chỉ ra rằng thuật toán được đề xuất có nhiều ưu điểm nổi bật hơn so với các phương pháp trước đây về thời gian thực thi, mức bộ nhớ sử dụng và chất lượng của các mẫu được khám phá.

Từ khóa: độ đo tương quan Bond; hàm tin cậy - lợi ích; luật kết hợp lợi ích cao; luật kết hợp súc tích; luật kết hợp tương quan cao; mẫu đồng; mẫu sinh.

Efficient mining of succinct correlated high utility association rules

Thong Tran^{1,2,3}, Tien Hoang Minh^{1,2,4}, Hai Duong Van^{4*}, Tin Truong Chi⁴

¹Faculty of Information Technology, VNUHCM-University of Science, Ho Chi Minh City, Vietnam

²Vietnam National University Ho Chi Minh City, Linh Xuan Ward, Hochiminh City, Vietnam

³Faculty of Information Technology, Da Lat University, Phu Dong Thien Vuong street, Lam Vien Ward, Da Lat City, Lam Dong province, Vietnam

⁴Faculty of Mathematics and Computer Science, Da Lat University, Phu Dong Thien Vuong Street, Lam Vien Ward, Da Lat City, Lam Dong province, Vietnam

Received 13 January 2025; revised 20 February 2025; accepted 24 February 2025

Abstract:

This paper investigates the problem of mining high utility association rules from quantitative databases. Traditional approaches mainly rely on conventional utility-confidence measures without considering the actual strength of association among items. Due to the absence of a correlation measure, many discovered rules exhibit high utility while the items within the rules co-occur infrequently, making them difficult to interpret and of limited practical significance; moreover, the number of generated rules is often very large. To address these limitations, this paper proposes a novel utility-confidence measure and integrates the Bond correlation measure to ensure that the items in a rule not only yield high utility but also exhibit strong correlations. On this basis, the paper introduces a class of succinct correlated high utility association rules, called SCoHUARs, in which each rule is represented in a compact form that serves as a representative of an equivalence class of rules sharing the same support. In addition, a partitioning method for the set of SCoHUARs is presented to guarantee complete and non-redundant generation of all SCoHUARs, together with several optimization techniques to accelerate the mining process. Finally, the M-SCoHUAR algorithm is designed to efficiently mine SCoHUARs by integrating the proposed theoretical results. Experimental results demonstrate that the proposed algorithm significantly outperforms existing methods in terms of execution time, memory consumption, and the quality of the discovered patterns.

Keywords: Bond correlation; closed pattern; correlated association rule; generator pattern; high utility association rule; succinct association rule; utility-confidence measure.

*Tác giả liên hệ: Email: haidv@dlu.edu.vn

1. Giới thiệu

Khai thác luật kết hợp là một trong những hướng nghiên cứu nền tảng của khai phá dữ liệu và đã thu hút sự quan tâm của nhiều nhà nghiên cứu trong những năm qua [1]. Các phương pháp truyền thống dựa chủ yếu trên độ hỗ trợ và độ tin cậy. Cách tiếp cận này xem mọi mục là như nhau và chỉ quan tâm đến tần suất xuất hiện, nên không phản ánh được tầm quan trọng hay lợi ích của từng mục trong giao dịch. Điều này dẫn đến khả năng thu được các luật có độ tin cậy cao nhưng đóng góp lợi ích thấp, kém giá trị trong các bài toán ra quyết định. Để khắc phục hạn chế đó, các nghiên cứu về khai thác tập mục lợi ích cao (high-utility itemset (HUI) mining - HUIM) đã được đề xuất [2], trong đó mỗi mục được gán một trọng số lợi ích và một tập mục được xem là lợi ích cao nếu tổng lợi ích của nó vượt một ngưỡng do người dùng chỉ định [2]. Bài toán này có nhiều ứng dụng trong thực tế, chẳng hạn như khám phá các sản phẩm có lợi ích cao khi được mua đồng thời, phân tích hành vi người dùng trong các cửa hàng trực tuyến, nghiên cứu dữ liệu y-sinh, và tiếp thị chéo [2-4]. Các thuật toán như Two-Phase [5], HUI-Miner [6], FHM [7] và EFIM [8] đã được phát triển để xử lý những khó khăn do độ đo lợi ích không thỏa tính đơn điệu, thông qua việc thiết kế các chặn trên (UB) và cấu trúc dữ liệu đặc thù nhằm giảm số lượng ứng viên và chi phí tính toán.

Dựa trên các HUI, bài toán khai thác các luật kết hợp lợi ích cao (high-utility association rules - HUAR) tiếp tục được quan tâm với mục tiêu cung cấp thông tin hỗ trợ ra quyết định trong các bối cảnh như thương mại điện tử, tài chính hay Internet vạn vật (IoT). Sahoo và cs [9] đã đề xuất khung đo lường độ tin cậy - lợi ích (utility-confidence) dựa trên hàm tin cậy - lợi ích, định nghĩa lại luật kết hợp trong không gian lợi ích, và đưa ra các khái niệm tập mục đóng lợi ích cao, tập mục sinh lợi ích cao cùng thuật toán HUCI-Miner để khai thác đồng thời hai lớp tập mục này. Từ đó, nhóm tác giả xây dựng được một cơ sở luật lợi ích cao không dư thừa, có tiền đề (về trái của luật) tối thiểu và hậu đề (về phải của luật) tối đa. Các thuật toán sau đó sử dụng cấu trúc dàn của tập mục lợi ích cao để sinh toàn bộ hoặc một tập con không dư thừa các HUAR với hiệu quả tốt hơn [10-12].

Tuy đạt được nhiều kết quả quan trọng, các hướng tiếp cận trên chủ yếu đánh giá luật dựa trên lợi ích và độ tin cậy theo lợi ích, chưa tích hợp trực tiếp các độ đo tương quan để kiểm soát mức độ gắn kết giữa các mục trong luật. Trong thực tế, một luật có lợi ích cao nhưng tương quan yếu (chẳng hạn khi về phải của luật xuất hiện rất phổ biến, ít phụ thuộc vào về trái) có thể ít mang lại giá trị giải thích và hỗ trợ ra quyết định. Đồng thời, các cơ sở luật không dư thừa hiện có mới chủ yếu khai thác khái niệm đóng/sinh trong không gian lợi ích, chưa kết hợp rõ ràng cả hai yếu tố: lợi ích cao và tương quan mạnh.

Xuất phát từ những nhận định đó, bài báo này nghiên cứu bài toán khai thác các luật kết hợp súc tích có lợi ích và tương quan cao (Succinct Correlated High-Utility Association Rules - SCoHUAR) trên cơ sở dữ liệu định lượng. Trước hết, bài báo sử dụng khái niệm độ tin cậy - lợi ích trong [9] nhưng đề xuất một hàm độ tin cậy - lợi ích mới gọi là ρ , nhấn mạnh phần lợi ích mà về phải thực sự đóng góp khi về trái xảy ra. Đồng thời, bài báo tích hợp thêm độ đo tương quan bond [13], vốn phản ánh tỉ lệ giữa độ hỗ trợ đồng xuất hiện và độ hỗ trợ rời rạc của tập mục, để bảo đảm các luật thu được không chỉ có lợi ích cao mà còn thể hiện mối liên hệ chặt chẽ giữa các mục. Trong khai thác các tập mục phổ biến, nhiều độ đo tương quan đã được đề xuất nhằm đánh giá mức độ gắn kết giữa các mục, tiêu biểu như bond [13], lift [14], all-confidence [15], kulec [16], v.v. Mỗi độ đo phản ánh mối quan hệ giữa các mục dưới những góc nhìn khác nhau. Cụ thể, lift đánh giá mức độ phụ thuộc thống kê bằng cách so sánh tần suất đồng xuất hiện quan sát được với tần suất kỳ vọng dưới giả định độc lập, qua đó cho phép phát hiện tương quan dương hoặc âm giữa các thuộc tính. Trong khi đó, kulec tập trung đo lường mức độ gắn kết hai chiều, thông qua trung bình xác suất có điều kiện theo cả hai hướng giữa các tập mục. So với các độ đo trên, bond, được định nghĩa là tỉ lệ giữa support của giao và support của hợp của các tập mục, sở hữu những ưu điểm nổi bật. Cụ thể, bond là một độ đo có tính chất đơn điệu giảm nên đặc biệt phù hợp để tích hợp vào các chiến lược cắt tỉa hiệu quả trong quá trình khai thác. Hơn nữa, như đã được chỉ ra trong phần thực nghiệm của [13], việc sử dụng bond mang lại chất lượng tập kết quả tốt hơn và thời gian thực thi nhanh hơn so với all-confidence trong bài toán khai thác các tập mục có lợi ích và tương quan cao. Nhờ những đặc tính này, bond cho phép loại bỏ sớm các mẫu không tương quan, đồng thời nâng cao khả năng mở rộng của thuật toán. Vì các lý do nêu trên, bài báo này lựa chọn bond làm độ đo tương quan để khai thác các SCoHUAR.

Trên cơ sở đó, bài báo đồng thời giới thiệu lớp luật SCoHUAR, trong đó về trái là tập mục sinh lợi ích cao, hợp của hai về của luật là tập mục đóng lợi ích cao [17], và cả hai đồng thời thỏa các ràng buộc về lợi ích, độ tin cậy - lợi ích và tương quan mạnh.

Để thấy được ý nghĩa của hàm tin cậy - lợi ích mới và độ đo tương quan bond, bài báo trình bày một ví dụ về cơ sở dữ liệu thực tế minh họa việc bán lẻ phụ kiện công nghệ trong một cửa hàng online, trong đó mỗi giá trị là lợi ích thu được từ mặt hàng trong một giao dịch. Cơ sở dữ liệu được trình bày trong bảng 1, gồm các thuộc tính: *Smartphone*, *ốp lưng*, *kính cường lực*, *bảo hiểm rơi vỡ*, *gói data* và *tai nghe*. Giá sử ngưỡng độ tin cậy-lợi ích tối thiểu và ngưỡng tương quan tối thiểu, xét hàm tin cậy - lợi ích được giới thiệu trong [13] (xem định nghĩa 16), hàm tin cậy - lợi ích mới được trình bày trong định nghĩa 17, và độ đo tương quan bond được trình bày trong định nghĩa 8.

Xét luật R1: Smartphone $\rightarrow$ Ôp lung, ta có $uconf(R1) = 0,616 > muconf$, $cuconf(R1) = 0,03 < muconf$ và $bond(\{Smartphone, \text{ôp lung}\}) = 0,375 < mb$. Luật này minh họa trường hợp $uconf$ đủ lớn (về trái “giữ lợi ích” khi đi kèm về phải), nhưng $cuconf$ rất nhỏ, tức Ôp lung chỉ đóng góp lợi ích không đáng kể. Do đó, nếu chỉ dùng $uconf$, ta có thể giữ lại các luật mà về phải mang ít giá trị kinh tế. Tuy nhiên, nếu sử dụng thêm $cuconf$, ta có thể nhận diện và hạ ưu tiên các luật loại này. Ngoài ra, ta cũng có thể loại bỏ luật này do mức tương quan không đạt ngưỡng. Đáng chú ý hơn, luật R2: Ôp lung $\rightarrow$ Tai nghe cho kết quả $uconf(R2) = 0.4 < muconf$, $cuconf(R2) = 1,467 > muconf$ và $bond(\{\text{Ôp lung}, \text{Tai nghe}\}) = 0.5 > mb$, cho thấy Tai nghe đóng góp lợi ích vượt trội trong các giao dịch kết hợp Ôp lung, mặc dù $uconf$ không đạt ngưỡng. Ví dụ này minh họa rõ ràng $cuconf$ có khả năng phát hiện các luật bán kèm sinh lợi cao mà $uconf$ có thể loại bỏ, đồng thời việc xét thêm $bond$ giúp đảm bảo các luật thu được phản ánh những mối liên kết đồng xuất hiện có ý nghĩa. Ngoài ra, $cuconf$ có thể hỗ trợ $uconf$ trong việc loại bỏ các luật mặc dù có độ tin cậy - lợi ích cao (theo $uconf$) nhưng về phải mang ít giá trị kinh tế.

Bảng 1. Cơ sở dữ liệu định lượng minh họa bán lẻ phụ kiện công nghệ.

ID	Các thuộc tính (với lợi ích) trong giao tác
	Smartphone (900), Ôp lung (40), Kính cường lực (25), Bảo hiểm rơi vỡ (200)
	Smartphone (850), Bảo hiểm rơi vỡ (200), Gói data (150)
	Smartphone (820), Ôp lung (45), Kính cường lực (30)
	Ôp lung (35), Kính cường lực (20), Tai nghe (120)
	Smartphone (880), Bảo hiểm rơi vỡ (200)
	Ôp lung (30), Kính cường lực (20), Tai nghe (110)
	Smartphone (950), Ôp lung (50), Bảo hiểm rơi vỡ (200)
	Ôp lung (25), Kính cường lực (15), Tai nghe (100)

Ví dụ trên cho thấy rằng việc khai thác các luật kết hợp có lợi ích và tương quan cao, dựa trên độ tin cậy - lợi ích mới $cuconf$ kết hợp với độ đo tương quan $bond$, mang lại giá trị rõ rệt và có ý nghĩa thực tiễn trong các ứng dụng thực tế. Trên cơ sở đó, các đóng góp chính của bài báo được tóm tắt như sau:

- (1) Một hàm tin cậy - lợi ích mới được đề xuất, thay thế hoặc hỗ trợ (tùy vào ứng dụng thực tế) cho hàm tin cậy - lợi ích đã được sử dụng bởi những công trình trước đây.
- (2) Đề xuất bài toán khai thác các luật kết hợp sức tích có lợi ích và tương quan cao (SCoHUAR) sử dụng hàm tin cậy - lợi ích mới, độ đo tương quan $bond$ và các khái niệm tập mục đóng và sinh lợi ích cao và tương quan mạnh.

- (3) Giới thiệu một phương pháp phân hoạch mới để sinh ra đầy đủ và không trùng lặp tất cả các SCoHUAR.

- (4) Thiết kế một thuật toán hiệu quả M-SCoHUAR nhằm khai thác nhanh các SCoHUAR, trong đó tích hợp các kết quả lý thuyết đã đề xuất, kết hợp với các kỹ thuật tối ưu nhằm giảm thiểu tối đa số phép kiểm tra quan hệ bao hàm giữa các tập mục, qua đó tăng tốc quá trình khai thác.

- (5) Cung cấp các kết quả thực nghiệm thuyết phục nhằm xác nhận tính đúng đắn và ý nghĩa của các kết quả lý thuyết được đề xuất, đồng thời chứng minh hiệu quả của thuật toán M-SCoHUAR trong việc khai thác các luật SCoHUAR.

Phần còn lại của bài báo được tổ chức như sau: (i) Mục 2 trình bày tổng quan các công trình nghiên cứu liên quan đến khai thác các tập mục lợi ích cao và các dạng biểu diễn sức tích của chúng, cũng như các công trình liên quan đến khai thác luật kết hợp lợi ích cao; (ii) Mục 3 giới thiệu các khái niệm và ký hiệu cần thiết cho việc khám phá các luật kết hợp sức tích có lợi ích và tương quan cao; (iii) Mục 4 trình bày quy trình tổng quát để khai thác các luật kết hợp hữu ích, cùng với phương pháp và thuật toán nhằm sinh nhanh các luật này; (iv) Kết quả thực nghiệm được thảo luận trong (v) mục 5; Cuối cùng, những kết luận và công việc tương lai được trình bày trong mục 6.

2. Những công việc liên quan

1. Khai thác các tập mục lợi ích cao

Khai thác các tập mục lợi ích cao (HUIM) đã là một chủ đề nghiên cứu phổ biến trong khai thác dữ liệu, với nhiều thuật toán được phát triển để khám phá hiệu quả những tập mục như vậy [18, 19, 20-23]. Các thuật toán này thường dựa vào các chặn trên (UBs) để giảm không gian tìm kiếm. Việc thiết kế các UBs chặt có thể dẫn đến cải thiện trực tiếp hiệu quả của HUIM. Do đó, một số UBs đã được thiết kế. Chặn trên đầu tiên là TWU (transaction weight utility), được sử dụng trong thuật toán Two-Phase [5]. Thuật toán này trước tiên xác định các tập mục có giá trị TWU cao, và sau đó lọc bỏ các tập mục có lợi ích thấp (LUIs). Cách tiếp cận này cũng được áp dụng trong các thuật toán khác như GPA và UP-Growth+ [4, 24]. UB thứ hai, gọi là, được trình bày trong [2], và chặt hơn TWU. Để tính toán hiệu quả, các tác giả của [2] đã giới thiệu cấu trúc utility-list để lưu trữ thông tin cần thiết của các tập mục. Cách tiếp cận này cũng được sử dụng trong các thuật toán FHM [7] và HMiner, với các chiến lược bổ sung như U-Prune, EUCP, và LA-Prune. Bên cạnh đó, các phương pháp khác cũng được phát triển để cải thiện hiệu quả của HUIM, dựa trên các cấu trúc dữ liệu mới [25, 26] [27, 28]. Gần đây, một chặn trên yếu (WUB) gọi là được

đề xuất trong [29], chặt hơn và được sử dụng trong các thuật toán C-HUIM và MaxC-HUIM để loại bỏ sớm hơn nhiều tập mục có lợi ích thấp.

2. Khai thác các tập mục đóng và sinh có lợi ích cao

Để khắc phục vấn đề sinh quá nhiều kết quả trong các thuật toán HUIM, nhiều dạng biểu diễn súc tích của tập mục lợi ích cao đã được đề xuất. Tuy nhiên, việc thiết kế các thuật toán hiệu quả cho các biểu diễn này không đơn giản, vì đòi hỏi phải loại bỏ sớm các tập mục lợi ích thấp (LUI), cắt tía sớm các tập mục lợi ích cao không là đóng và/hoặc không là sinh, đồng thời thường xuyên kiểm tra quan hệ bao hàm giữa các tập mục. Một số thuật toán tiêu biểu theo hướng này gồm CHUD [30], CHUI-Miner [31], EFIM-Closed [32], HMiner-Closed [33], HUG-Miner [34], HUCI-Miner [9,35], C-HUIM [29] và MaxC-HUIM [29]. Các thuật toán này nhìn chung sử dụng các chặn trên TWU và để loại bỏ sớm các LUI; CHUD và CHUI-Miner sử dụng hai tập PrevSet và PostSet để cắt tía nhanh các HUI nhưng không là đóng; EFIM-Closed dùng cấu trúc utility-bin, kỹ thuật closure jumping và gộp giao dịch; HMiner-Closed dùng cấu trúc MCUL mở rộng từ CUL cùng khái niệm lợi ích đóng và không đóng để giảm số lần quét dữ liệu. Gần đây, trong [17], các tác giả đã đề xuất thuật toán FCGHUI-Miner để khai thác đồng thời các tập mục đóng và sinh có lợi ích cao, sử dụng chặn trên yếu của hàm lợi ích, chiến lược cắt tía địa phương theo hai mức liên tiếp của cây tiền tố, tránh kiểm tra quan hệ bao hàm, qua đó giảm mạnh ứng viên dư thừa và cải thiện đáng kể hiệu quả của quá trình khai thác.

3. Khai thác các luật kết hợp lợi ích cao

J. Sahoo và cs (2015) [9] là một trong những tác giả đầu tiên đề xuất bài toán khai thác luật kết hợp lợi ích cao. Họ đã đề xuất độ đo tin cậy dựa trên lợi ích (utility-confidence) để đánh giá luật trong không gian lợi ích, đồng thời giới thiệu các khái niệm tập mục đóng và tập mục sinh lợi ích cao, cùng thuật toán HUCI-Miner để khai thác đồng thời các loại tập mục này từ cơ sở dữ liệu định lượng. Từ các tập mục đóng và sinh có lợi ích cao, nhóm tác giả xây dựng một cơ sở luật kết hợp lợi ích cao không dư thừa, trong đó về trái của luật là tập sinh và hợp hai vế của luật là tập đóng. Dựa trên hướng tiếp cận này, T. Mai và cs [12] đã đề xuất thuật toán LNR-HAR, sử dụng dàn của các tập mục lợi ích cao để sinh trực tiếp các luật kết hợp lợi ích cao không dư thừa, trong khi L.T.T. Nguyen và cs (2020) [11] giới thiệu các thuật toán HGB-HAR và LARM, kết hợp giữa tập đóng và tập sinh lợi ích cao, và cấu trúc dàn để cải thiện hiệu quả về thời gian và bộ nhớ của quá trình khai thác. Điểm chung của các công trình trên là tập trung vào việc giảm số lượng luật được khai thác và loại bỏ dư thừa sử dụng hàm tin cậy - lợi ích, nhưng chưa tích hợp một độ đo tương quan để bảo đảm mức độ gắn kết giữa các mục. Điều này dẫn đến khả năng

sinh ra các luật có lợi ích cao nhưng sự tương quan giữa các thuộc tính trong luật còn yếu.

Khoảng trống nghiên cứu trên là động lực để bài báo này đề xuất bài toán khai thác các luật kết hợp súc tích có lợi ích và tương quan cao (SCoHUAR). Khác với các hướng tiếp cận trước đây, bài báo đề xuất một hàm lợi ích - tin cậy mới gọi là *cuconf* và tích hợp độ đo tương quan bond vào quy trình khai thác, đồng thời sử dụng các tập mục đóng và sinh lợi ích cao để bảo đảm tính súc tích và không dư thừa của các luật được khai thác.

3. Các khái niệm và định nghĩa cơ sở

Phần này trình bày bài toán khai thác các luật kết hợp súc tích có lợi ích và tương quan cao trên các cơ sở dữ liệu giao dịch định lượng. Định dạng dữ liệu đầu vào sử dụng trong bài toán này được xác định như sau.

Định nghĩa 1 (Cơ sở dữ liệu định lượng). Cho $A \stackrel{\text{def}}{=} \{a_j, j \in J \stackrel{\text{def}}{=} \{1, 2, \dots, M\}\}$ là tập hữu hạn các thuộc tính riêng biệt và $A \subseteq A$ là tập con của A . Nếu A chứa k thuộc tính, nó được gọi là một k -itemset, nghĩa là, $k=|A|$. Không mất tính tổng quát, giả sử rằng các mục trong mỗi tập mục được sắp xếp theo một thứ tự toàn phần $<$ (ví dụ thứ tự từ điển). Vector lợi ích $P \stackrel{\text{def}}{=} (p(a_j), j \in J \text{ và } p(a_j) > 0)$ biểu diễn lợi ích ngoài của các mục trong A , trong đó $p(a_j)$ thể hiện sự quan trọng tương đối của mục a_j , chẳng hạn như lợi ích đơn vị, giá, trọng số hay các thừa số khác. Một q -item là một cặp (a, q) , trong đó a là một mục trong A và q là một số không âm thể hiện số lượng mua hay lợi ích trong của a . Một cơ sở dữ liệu định lượng (QDB) D là tập tất cả các giao tác $D \stackrel{\text{def}}{=} \{T_i, i \in I \stackrel{\text{def}}{=} \{1, 2, \dots, N\}\}$, trong đó mỗi giao tác $T_i, T_i \stackrel{\text{def}}{=} \{(a_j, q_j), j \in J_i\}$ gồm tập các q -items, được xác định duy nhất bởi TID (ví dụ, TID= i) và J_i là tập con của $J, J_i \subseteq J$. Lợi ích của q -item (a, q) , ký hiệu là $u((a, q))$, được định nghĩa là $u((a, q)) \stackrel{\text{def}}{=} q \times p(a)$. Bảng 2 chỉ ra ví dụ về QDB và vector lợi ích tương ứng (bảng 3).

Bảng 2. Cơ sở dữ liệu định lượng.

TID \ Item						
T_1	5	0	3	0	7	3
T_2	0	0	0	3	5	6
T_3	0	5	0	2	6	3
T_4	0	0	0	2	4	7
T_5	5	0	2	0	2	0
T_6	0	2	0	0	0	3
T_7	0	0	0	3	2	5
T_8	0	0	0	8	4	0

Bảng 3. Vector lợi ích.

a_j	a	b	c	d	e	f
$p(a_j)$	4	6	4	3	2	3

Định nghĩa 2 (Cơ sở dữ liệu định lượng tích hợp). Để tránh tính toán lặp lại lợi ích u của các q -items (a_j, q_{ij}) trong các giao tác T_i trong $\mathcal{D}$, các giá trị này được tính toán 1 lần và được lưu trữ trong QDB tương ứng gọi là $\mathcal{D}'$, $\mathcal{D}' \stackrel{\text{def}}{=} \{T'_i \stackrel{\text{def}}{=} \{(a_j, q'_{ij}), j \in J_i, J_i \subseteq J, i \in I\}$, trong đó $q'_{ij} = q_{ij} * p(a_j)$ nếu $(a_j, q_{ij}) \in T_i$ (ngược lại, $q'_{ij} = 0$), $\forall i \in I, j \in J$. Khi đó, $\mathcal{D}'$ được gọi là một QDB tích hợp của $\mathcal{D}$.

Ví dụ minh họa. Bảng 2 và bảng 3 lần lượt biểu diễn QDB $\mathcal{D}$ và vector lợi nhuận $\mathcal{P}$, trong khi Bảng 4 minh họa QDB tích hợp $\mathcal{D}'$ thu được bằng cách kết hợp $\mathcal{D}$ và $\mathcal{P}$. QDB $\mathcal{D}'$ sẽ được sử dụng làm ví dụ minh họa xuyên suốt bài báo.

Bảng 4. Cơ sở dữ liệu định lượng tích hợp $\mathcal{D}'$.

TID \ Item						
T ₁	20	0	12	0	14	9
T ₂	0	0	0	9	10	18
T ₃	0	30	0	6	12	9
T ₄	0	0	0	6	8	21
T ₅	20	0	8	0	4	0
T ₆	0	12	0	0	0	9
T ₇	0	0	0	9	4	15
T ₈	0	0	0	24	8	0

Định nghĩa 3 (Độ hỗ trợ của tập mục). Cho bất kỳ tập mục $A \subseteq \mathcal{A}$, $A \neq \emptyset$, đặt $\rho(A) \stackrel{\text{def}}{=} \{T_i \in \mathcal{D}' \mid q'_{ij} > 0, \forall a_j \in A\}$ là tập các giao dịch trong $\mathcal{D}'$ chứa A , và theo quy ước $\rho(\emptyset) \stackrel{\text{def}}{=} \mathcal{D}'$. Độ hỗ trợ của A , ký hiệu $\text{supp}(A)$, được xác định là số lượng giao dịch trong QDB $\mathcal{D}'$ chứa A , nghĩa là $\text{supp}(A)$ chính là lực lượng (cardinality) của tập $\rho(A)$. Lưu ý rằng ta chỉ xét các tập mục khác rỗng A và xuất hiện ít nhất một lần trong $\mathcal{D}'$, nghĩa là $A \in \mathcal{IS} \stackrel{\text{def}}{=} \{A \subseteq \mathcal{A} \mid A \neq \emptyset \wedge \rho(A) \neq \emptyset\}$.

Ví dụ 1. Cho tập mục $A = ac$, vì $A \subseteq T_1$ và $A \subseteq T_5$, ta có $\rho(A) = \{T_1, T_5\}$ và $\text{supp}(A) = 2$.

Định nghĩa 4 (Lợi ích của tập mục trong một giao tác). Lợi ích của một mục a_j ($j \in J$) trong giao tác $T_i \in \mathcal{D}'$ được ký hiệu và định nghĩa là $u(a_j, T_i) \stackrel{\text{def}}{=} q'_{ij}$. Lợi ích của tập mục $A \in \mathcal{IS}$ trong T_i chứa A được định nghĩa như sau:

$$u(A, T_i) \stackrel{\text{def}}{=} \sum_{a_j \in A} u(a_j, T_i). \quad (1)$$

Định nghĩa 5 (Lợi ích của tập mục trong cơ sở dữ liệu). Lợi ích của tập mục trong được định nghĩa là

$$u(A) \stackrel{\text{def}}{=} \sum_{T_i \in \rho(A)} u(A, T_i), \quad (2)$$

trong đó $A \in \mathcal{IS}$ và $u(A) \in \mathbb{R}_+ \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid x > 0\}$.

Định nghĩa 6 (Lợi ích giao tác và lợi ích cơ sở dữ liệu). Lợi ích của giao tác T_i được xác định là $tu(T_i) \stackrel{\text{def}}{=} \sum_{q'_{ij} > 0} q'_{ij}$. Khi đó, lợi ích tổng cộng TU của $\mathcal{D}'$ là tổng lợi ích của tất cả các giao tác trong $\mathcal{D}'$, nghĩa là, $TU \stackrel{\text{def}}{=} \sum_{T_i \in \mathcal{D}} tu(T_i)$.

Ví dụ 2. Cho tập mục $A = ac$, khi đó $\rho(A) = \{T_1, T_5\}$, $u(A, T_1) = 20 + 12 = 32$ và $u(A, T_5) = 20 + 8 = 28$. Khi đó, $u(A) = u(A, T_1) + u(A, T_5) = 32 + 28 = 60$. Ta có $tu(T_1) = 20 + 12 + 14 + 9 = 55$ và $TU = 297$.

Định nghĩa 7 (Tập mục lợi ích cao). Với một ngưỡng lợi ích tối thiểu do người dùng chỉ định, ký hiệu là mu , một tập mục A được gọi là tập mục lợi ích cao (HUI) nếu lợi ích của nó trong QDB $\mathcal{D}'$ không nhỏ hơn ngưỡng này, nghĩa là $u(A) \geq mu$. Ngược lại, nếu $u(A) < mu$, thì A được xem là tập mục lợi ích thấp (LUI). Mục tiêu của bài toán khai thác tập mục lợi ích cao (HUIM) là khám phá tập $\mathcal{HUI}$ gồm tất cả các HUI, được xác định cụ thể như sau: $\mathcal{HUI} \stackrel{\text{def}}{=} \{A \in \mathcal{IS} \mid u(A) \geq mu\}$.

Định nghĩa 8 (Độ đo tương quan bond [36]). Độ đo bond đánh giá mức độ gắn kết giữa các mục trong một tập mục dựa trên tần suất đồng xuất hiện của chúng trong cơ sở dữ liệu. Với một tập mục A , giá trị bond của A được định nghĩa như sau:

$$\text{bond}(A) = \frac{\text{supp}(A)}{\text{dissup}(A)}, \quad (3)$$

trong đó $\text{dissup}(A) = |\{T \in \mathcal{D} \mid A \cap T \neq \emptyset\}|$ chỉ ra số giao tác trong $\mathcal{D}$ chứa ít nhất một mục của A .

Độ đo bond của tập mục A thỏa điều kiện: $0 < \text{bond}(A) \leq 1$. Giá trị bond càng lớn cho thấy mức độ tương quan giữa các mục trong A càng mạnh, trong khi giá trị nhỏ phản ánh sự gắn kết yếu.

Định nghĩa 9 (Tập mục lợi ích và tương quan cao). Ký hiệu mu là ngưỡng lợi ích tối thiểu do người dùng đặt ra và mb là ngưỡng bond tối thiểu. Một tập mục $A \subseteq \mathcal{A}$ được gọi là tập mục lợi ích và tương quan cao (CoHUI) nếu nó đồng thời thỏa mãn hai ràng buộc về lợi ích và tương quan, nghĩa là $u(A) \geq mu$ và $\text{bond}(A) \geq mb$. Tập tất cả các CoHUI trong $\mathcal{D}'$ được xác định như sau:

$\text{CoHUI} = \{A \in \mathcal{IS} \mid u(A) \geq mu \wedge \text{bond}(A) \geq mb\}$. (4)
 trong đó, $\mathcal{IS}$ là danh sách tất cả các tập mục có thể trong không gian tìm kiếm. Định nghĩa này bảo đảm rằng các mẫu được phát hiện không chỉ có lợi ích cao mà còn thể hiện sự tương quan mạnh giữa các mục, từ đó nâng cao khả năng diễn giải và giá trị ứng dụng trong thực tiễn.

Ví dụ 3. Cho tập mục $A = ce$, ta có $\text{supp}(A) = 2$, và $\text{dissup}(A) = 7$. Do đó, $\text{bond}(A) = \frac{\text{supp}(A)}{\text{dissup}(A)} = \frac{2}{7} = 0.2857$. Ngoài ra, $u(A) = 38$. Nếu $mu = 30$ và $mb = 0.2$, thì $u(A) > mu$ and $\text{bond}(A) > mb$. Vì vậy, A là một CoHUI.

Định nghĩa 10 (Lớp tương đương trên các tập mục). Dựa trên định nghĩa của toán tử ρ , quan hệ tương đương trên các tập mục trong $\mathcal{IS}$ được ký hiệu là $\sim$ và được định nghĩa như sau: $\forall A, B \in \mathcal{IS}$, $A \sim B \Leftrightarrow \rho(A) = \rho(B)$. Khi đó, lớp tương đương của A là $[A] \stackrel{\text{def}}{=} \{B \in \mathcal{IS} \mid \rho(B) =$

Định nghĩa 11 (Tập mục lợi ích cao đóng). Một HUI A được gọi là *tập mục lợi ích cao đóng* (CHUI) nếu không tồn tại một HUI nào khác là tập cha thật sự của A và có cùng độ hỗ trợ với A . Tập đầy đủ tất cả các CHUI được ký hiệu và định nghĩa hình thức như sau:

$$\mathcal{CHUI} \stackrel{\text{def}}{=} \{A \in \mathcal{HUI} \mid \nexists B \in \mathcal{HUI}: B \supset A \wedge \text{supp}(B) = \text{supp}(A)\}. \quad (5)$$

Định nghĩa 12 (Bao đóng của tập mục). Bao đóng của một tập mục A , ký hiệu $h(A)$, được định nghĩa là một tập mục C sao cho C là tập cha lớn nhất của A và có độ hỗ trợ bằng độ hỗ trợ của A , nghĩa là $\text{supp}(A) = \text{supp}(C)$. Nếu A là CHUI, thì $A = h(A)$.

Định nghĩa 13 (Tập mục sinh lợi ích cao). Một HUI A được gọi là *tập sinh lợi ích cao* (GHUI) nếu không tồn tại một HUI nào khác là tập con thật sự của A và có cùng độ hỗ trợ với A . Tập tất cả các GHUI được ký hiệu và được định nghĩa như sau:

$$\mathcal{GHUI} \stackrel{\text{def}}{=} \{A \in \mathcal{HUI} \mid \nexists B \in \mathcal{HUI}: B \subset A \wedge \text{supp}(B) = \text{supp}(A)\}. \quad (6)$$

Cho bất kỳ $A \in \mathcal{CHUI}$ và $B \in \mathcal{GHUI}$, B được gọi là tập mục sinh của A nếu $B \subseteq A$ và $\text{supp}(B) = \text{supp}(A)$. Ký hiệu $\mathcal{G}(A)$ là tập tất cả các GHUI của A .

Dựa trên định nghĩa của 2 tập $\mathcal{GHUI}$, $\mathcal{CHUI}$ và độ đo *bond*, ta đặt được hai tập $\mathcal{CoGHUI}$ và $\mathcal{CoCHUI}$ được định nghĩa hình thức như sau:

$$\begin{aligned} \mathcal{CoGHUI} &\stackrel{\text{def}}{=} \{A \in \mathcal{GHUI} \mid \text{bond}(A) \geq mb\}, \text{ and} \\ \mathcal{CoCHUI} &\stackrel{\text{def}}{=} \{A \in \mathcal{CHUI} \mid \text{bond}(A) \geq mb\}. \end{aligned} \quad (7)$$

Định nghĩa 14 (Lợi ích cục bộ của một mục [9]). Cho bất kỳ mục a_j và tập mục A , lợi ích cục bộ của a_j trong A , $\text{luv}(a_j, A)$, là tổng lợi ích của a_j trong các giao tác chứa A , cụ thể:

$$\text{luv}(a_j, A) \stackrel{\text{def}}{=} \sum_{T_i \in \rho(A)} u(a_j, T_i). \quad (8)$$

Ví dụ 4. Xét tập mục $A = ce$. Ta có $\rho(A) = \{T_1, T_5\}$, $u(e, T_1) = 14$ và $u(e, T_5) = 4$. Vì vậy, $\text{luv}(e, A) = 14 + 4 = 18$.

Định nghĩa 15 (Lợi ích cục bộ của một tập mục [9]). Cho bất kỳ 2 tập mục A và B sao cho $A \subseteq B$, lợi ích cục bộ của A trong B được định nghĩa như sau:

$$\text{luv}(A, B) \stackrel{\text{def}}{=} \sum_{a_j \in A} \text{luv}(a_j, B). \quad (9)$$

Ví dụ 5. Xét hai tập mục $A = de$ và $B = def$. Khi đó, $\rho(B) = \{T_2, T_3, T_4, T_7\}$ và $\text{luv}(A, B) = \text{luv}(d, B) + \text{luv}(e, B) = 30 + 34 = 64$.

Định nghĩa 16 (Độ tin cậy lợi ích của một luật [9]). Với hai tập mục khác rỗng A và B , trong đó $A, B \in \mathcal{IS}$ và

$A \cap B = \emptyset$, một luật kết hợp R có dạng $R: A \rightarrow B$, trong đó A và B lần lượt được gọi là vế trái (antecedent) và vế phải (consequent) của luật. Độ tin cậy - lợi ích của R được ký hiệu và định nghĩa như sau:

$$\text{uconf}(R) \stackrel{\text{def}}{=} \frac{\text{luv}(A, A \cup B)}{u(A)}. \quad (10)$$

Độ tin cậy-lợi ích $\text{uconf}(R)$ phản ánh mức độ đóng góp lợi ích của các mục trong A khi chúng xuất hiện cùng với B .

Ví dụ 6. Xét luật $R: e \rightarrow f$. Ta có, $\text{luv}(e, ef) = 48$ và $u(e) = 60$. Do đó, $\text{uconf}(R) = \frac{48}{60} = 0.8$.

Lưu ý rằng, mặc dù hàm tin cậy - lợi ích uconf truyền thống phản ánh mức độ lợi ích của A trong các giao tác mà A cùng xuất hiện với B , nhưng nó chỉ cung cấp một góc nhìn một chiều: uconf chỉ xem xét đóng góp lợi ích của A mà bỏ qua phần lợi ích thực sự do B tạo ra trong các giao tác đó. Trong nhiều tình huống thực tế, như đánh giá gói sản phẩm hoặc chiến lược bán kèm, việc biết mức lợi nhuận mà B đóng góp khi xuất hiện cùng với A là thông tin quan trọng không kém, thậm chí quan trọng hơn. Hạn chế này dẫn đến nhu cầu xây dựng một hàm tin cậy - lợi ích mới gọi là cuconf , so sánh lợi ích của B trong các giao tác chứa $A \cup B$ với tổng lợi ích của A . Nhờ đó, cuconf cung cấp cái nhìn rõ ràng hơn về hiệu quả kết hợp giữa hai tập mục A và B .

Định nghĩa 17 (Độ tin cậy - lợi ích chéo của luật kết hợp). Cho bất kỳ luật $R: A \rightarrow B$, độ tin cậy-lợi ích chéo của luật kết hợp R được ký hiệu và định nghĩa như sau:

$$\text{cuconf}(R) \stackrel{\text{def}}{=} \frac{\text{luv}(B, A \cup B)}{u(A)}. \quad (11)$$

Độ tin cậy-lợi ích chéo $\text{cuconf}(R)$ đo lường mức đóng góp lợi ích của tập mục B trong các giao tác chứa cả A và B , và so sánh phần lợi ích này với tổng lợi ích của A . Nếu $\text{cuconf}(R)$ cao, điều đó cho thấy rằng khi A xuất hiện, việc bổ sung B mang lại phần lợi ích đáng kể, phản ánh sự gắn kết mạnh của cặp tập mục này.

Khác với uconf , vốn chỉ xem xét đóng góp lợi ích của A , độ đo cuconf trực tiếp đánh giá giá trị mà B tạo ra trong ngữ cảnh của luật, từ đó cung cấp góc nhìn đầy đủ hơn về tính hiệu quả của cặp tập mục A và B .

Xét bối cảnh trong siêu thị với A là mì gói và B là xúc xích. Khi phân tích dữ liệu giao dịch, giả sử rằng trong những giao tác chứa cả mì gói và xúc xích, xúc xích tạo ra phần lớn doanh thu, trong khi mì gói chỉ đóng góp mức lợi nhuận nhỏ. Khi đó, $\text{cuconf}(A \rightarrow B)$ sẽ cao và cho nhà quản lý biết rằng: “*Khi khách mua mì gói, họ thường mua xúc xích và phần lợi nhuận đáng kể đến từ xúc xích*”. Điều này giúp doanh nghiệp đưa ra quyết định về chiến lược bán kèm, đặt kệ hàng, hoặc gói combo để tối đa hóa lợi nhuận.

Ví dụ 7. Xét luật $R: e \rightarrow f$ trên cơ sở dữ liệu minh họa. Ta có $u(e) = 60$. Trong các giao tác chứa đồng thời e và

f , tổng lợi ích của e là $luv(e, ef) = 48$ và tổng lợi ích của f là $luv(f, ef) = 72$. Do đó, $uconf(R) = 48/60 = 0,8$, $cuconf(R) = 72/60 = 1.2$. Giá trị $uconf(R)$ cho thấy mức lợi ích của e được duy trì trong các giao tác có chứa cả e và f . Trong khi đó, $cuconf(R) > 1$ phản ánh rằng f đóng góp lợi ích lớn hơn tổng lợi ích của e , cho thấy việc mở rộng từ e sang f mang lại hiệu quả mạnh hơn nhiều. Điều này minh họa rõ sự khác biệt trong góc nhìn của hai độ đo và vai trò bổ sung của $cuconf$ khi đánh giá các luật kết hợp theo lợi ích.

Định nghĩa 18 (Luật kết hợp lợi ích và tương quan cao). Luật $R: A \rightarrow B$ được gọi là một luật kết hợp lợi ích và tương quan cao (CoHUAR) nếu R thỏa những điều kiện sau: $u(A) \geq mu$, $u(A \cup B) \geq mu$, $bond(A \cup B) \geq mb$ và $cuconf(R) \geq muconf$, trong đó $muconf$ là ngưỡng tin cậy - lợi ích tối thiểu được định nghĩa bởi người dùng. Tập $CoHUAR$ gồm tất cả các CoHUARs được định nghĩa hình thức như sau:

$$CoHUAR = \{R: A \rightarrow B \mid u(A) \geq mu, u(A \cup B) \geq mu, bond(A \cup B) \geq mb \wedge cuconf(R) \geq muconf\}. \quad (12)$$

Ví dụ 8. Xét cơ sở dữ liệu minh họa trong Bảng 3 với các ngưỡng $mu = 30$, $mb = 0.2$ và $muconf = 0.6$. Với luật $R: e \rightarrow f$, ta có $u(e) = 60$, $u(ef) = 120$, và $bond(ef) = \frac{5}{8} = 0.625$. Ngoài ra, độ tin cậy-lợi ích chéo của luật là $cuconf(R) = 1.2$. Do đó, luật thỏa đồng thời bốn điều kiện: $u(e) \geq mu$, $u(ef) \geq mu$, $bond(ef) \geq mb$ và $cuconf(R) \geq muconf$. Vì vậy, R là một CoHUAR.

Dựa trên các định nghĩa về CHUI, GHUI và độ đo bond, bài báo này giới thiệu khái niệm luật kết hợp sức tích có lợi ích và tương quan cao (SCoHUAR), trong đó về trái của mỗi luật là một GHUI và hợp của về trái với về phải tạo thành một CHUI. Định nghĩa hình thức của SCoHUAR được trình bày như sau.

Định nghĩa 19 (Luật kết hợp sức tích có lợi ích và tương quan cao). Cho trước một CoHUAR $R: A \rightarrow B$, R được gọi là luật kết hợp sức tích có lợi ích và tương quan cao (SCoHUAR) nếu $A \in CoGHUI$ và $A \cup B \in CoCHUI$. Tập $SCoHUAR$ gồm tất cả các SCoHUAR trong $\mathcal{D}'$ được định nghĩa hình thức như sau:

$$SCoHUAR = \{R: A \rightarrow B \mid R \in CoHUAR \wedge A \in CoGHUI \wedge A \cup B \in CoCHUI\}. \quad (13)$$

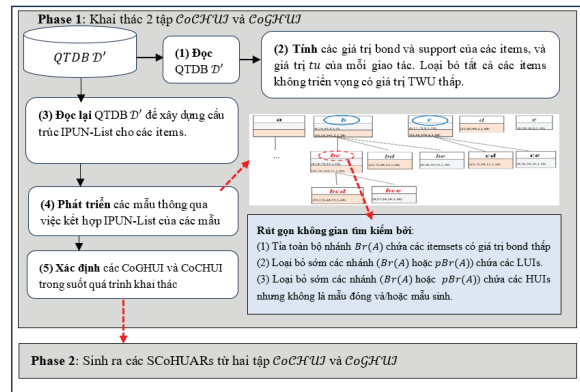
Phát biểu bài toán. Cho SDB $\mathcal{D}'$ và ba ngưỡng dương do người dùng chỉ định, gồm lợi ích tối thiểu (mu), độ tin cậy-lợi ích tối thiểu ($muconf$), và mức tương quan bond tối thiểu (mb), mục tiêu của bài toán đặt ra trong bài báo này là khám phá tập $SCoHUAR$ gồm tất cả các SCoHUAR có trong $\mathcal{D}'$.

4. Khai thác các luật kết hợp sức tích có lợi ích và tương quan cao

4.1. Quá trình khai thác tổng thể các SCoHUAR

Mục này trình bày chi tiết quy trình khai thác của phương pháp được đề xuất, được minh họa trong hình 1. Quy trình tổng thể gồm hai giai đoạn. Giai đoạn thứ nhất dùng để khám phá hai tập $CoCHUI$ và $CoGHUI$ từ SDB $\mathcal{D}'$, trong khi giai đoạn thứ hai dùng để sinh tất cả các SCoHUAR từ hai tập này. Giai đoạn thứ nhất bắt đầu bằng việc đọc QDB $\mathcal{D}'$ để tính các giá trị bond và support của các mục, cũng như tổng lợi ích của các giao dịch (tu). Các mục có giá trị TWU thấp bị loại bỏ khỏi $\mathcal{D}'$, và các mục còn lại được sắp xếp theo thứ tự tăng dần của support. QDB $\mathcal{D}'$ sau đó được quét lại để mỗi giao dịch được sắp xếp theo thứ tự support. Cấu trúc dữ liệu IPUN-list, được cải tiến từ MPUN-list trong [29], sau đó được xây dựng cho các mục còn lại. Sau khi quét xong tất cả các giao dịch, các IPUN-list của các 1-itemset được kết hợp để hình thành các 2-itemset và IPUN-list tương ứng của chúng. Tiếp theo, các 3-itemset được sinh ra bằng cách sử dụng IPUN-list của các 2-itemset. Quy trình tiếp tục được lặp lại cho đến khi hai tập $CoCHUI$ và $CoGHUI$ được xác định.

Lưu ý rằng, để khai thác đồng thời hai tập $GHUI$ và $CHUI$ trong giai đoạn 1, nghiên cứu [17] đã đề xuất thuật toán FCGHUI-Miner. Bài báo này kế thừa và áp dụng trực tiếp thuật toán đó để khám phá hai tập $GHUI$ và $CHUI$. Vì vậy, các kết quả lý thuyết liên quan đến việc thiết kế các chiến lược loại bỏ các tập mục có lợi ích thấp hoặc các tập mục có lợi ích cao nhưng không phải là mẫu đóng và/hoặc mẫu sinh sẽ không được trình bày lại trong. Tuy nhiên, do bài báo còn tích hợp thêm độ đo tương quan $bond$ nhằm khai thác các mẫu vừa có lợi ích cao vừa có mức độ tương quan mạnh, chúng tôi mở rộng FCGHUI-Miner bằng cách tích hợp tính chất trong mệnh đề 1 (liên quan đến độ đo



Hình 1. Quá trình khai thác tổng thể các SCoHUAR.

bond) vào quá trình khai thác để tìm 2 tập $CoCHUI$ và $CoGHUI$. Thuật toán sau khi sửa đổi được gọi là CoFCGHUI-Miner. Mệnh đề 1 được phát biểu như sau:

Mệnh đề 1. (*Chiến lược tia dựa trên bond*). Xét tập mục A bất kỳ, nếu giá trị *bond* của A nhỏ hơn ngưỡng tối thiểu mb , nghĩa là $bond(A) < mb$, thì A và tất cả tập cha của A được gọi là tương quan yếu, và có thể được loại bỏ.

Một thách thức quan trọng khi tích hợp độ đo *bond* vào thuật toán CoFCGHUI-Miner là chi phí tính toán *bond* cho các tập mục trong suốt quá trình khai thác thường rất lớn. Để giải quyết hạn chế này, bài báo tích hợp *bit vector* (*bv*) vào cấu trúc dữ liệu IPUN-list (được sử dụng trong thuật toán CoFCGHUI-Miner) để giảm thời gian tính toán giá trị *bond* của các tập mục. Với sự mở rộng này, bit vector trong cấu trúc IPUN-list của mỗi tập mục được tính toán như sau.

Cách xác định *bv* trong IPUN-List. Việc tính toán giá trị *bond* của tập mục A phụ thuộc vào hai giá trị $supp(A)$ và $dissup(A)$. Giá trị $supp(A)$ được tính toán trực tiếp từ các thông tin có sẵn trong IPUN-List. Tuy nhiên, $dissup(A)$ thường được tính bằng cách quét lại toàn bộ QDB, một thao tác tốn kém, đặc biệt đối với các QDB lớn. Để khắc phục hạn chế này, IPUN-List bổ sung thêm trường *bv* (bit vector). Mỗi bit trong *bv* tương ứng với một giao tác trong cơ sở dữ liệu: bit có giá trị 1 nếu giao tác chứa ít nhất một mục thuộc A , và 0 nếu ngược lại. Đối với các 1-itemset (tập mục chỉ có 1 thuộc tính), *bv* được khởi tạo trực tiếp trong lần quét QDB đầu tiên. Đối với các k-itemsets ($k > 1$), *bv* được cập nhật hiệu quả trong quá trình thực hiện phép mở rộng A bằng cách thực hiện phép toán OR trên các bit vector của hai tập cha: $bv(Pxy) = bv(Px) \text{ OR } bv(Py)$. Ví dụ, giả sử $bv(a) = 10011$ và $bv(b) = 11101$, ta có $bv(ab) = 11111$. Khi đó, $dissup(A)$ của tập mục A được xác định như sau: $dissup(X) = |bv(A)|$, tức là số lượng bit 1 trong bit vector.

4.2. Sinh ra nhanh các SCoHUAR

Nhắc lại rằng việc sinh ra các SCoHUAR dạng $R: A \rightarrow B$, trong đó $A \in CoGHUI$ và $A \cup B \in CoCHUI$ giúp đảm bảo rằng các luật SCoHUAR được biểu diễn ở dạng ngắn gọn nhất và không dư thừa. Các luật này không chỉ giảm số lượng luật được sinh ra mà còn tăng khả năng diễn giải, bởi mỗi luật đều được xem như đại diện cho một lớp tương đương các luật có cùng độ hỗ trợ. Điều này giúp các nhà phân tích dễ hiểu và ứng dụng các luật vào các bài toán ra quyết định dựa trên dữ liệu.

Để bảo đảm việc sinh ra đầy đủ và không trùng lặp tất cả các luật trong tập $SCoHUAR$, bài báo đề xuất một cách tiếp cận dựa trên phân hoạch (partitioning) của không gian

luật. Phân hoạch này cho phép chia tập $SCoHUAR$ thành các tập con rời nhau, mỗi tập con được đại diện bởi một cặp tập đóng lồng nhau. Ý tưởng cốt lõi là: đối với mỗi tập mục $A \in CoGHUI$, ta chỉ cần xem xét các mở rộng $A \cup B \in CoCHUI$, và từ đó sinh ra tất cả các luật $R: A \rightarrow B$ thuộc lớp tương ứng. Với cách sinh luật này, toàn bộ không gian của tập $SCoHUAR$ được chia thành các phân hoạch độc lập dựa trên các CoGHUI và CoCHUI. Cách tiếp cận này không chỉ đảm bảo rằng mỗi luật được sinh ra đúng một lần mà còn giúp giảm đáng kể chi phí khai thác bằng cách hạn chế vùng tìm kiếm cho mỗi phân hoạch. Phương pháp phân hoạch được trình bày trong định lý sau.

Định lý 1 (*Phân hoạch tập SCoHUAR*).

$$SCoHUAR = \sum_{(L,S) \in NCHUI} SCoHUAR(L,S), \quad (14)$$

trong đó $NCHUI = \{(L,S) \mid \emptyset \neq L \subseteq S \wedge L, S \in CoCHUI\}$ và $SCoHUAR(L,S)$ được xác định như sau:

- $SCoHUAR(L,S) = \{R: L_i \rightarrow S \setminus L_i \mid L_i \in \mathcal{G}(L) \wedge cuconf(R) \geq muconf\}, \text{ if } L \subset S.$
- $SCoHUAR(L,L) = \{R: L_i \rightarrow L \setminus L_i \mid L_i \in \mathcal{G}(L) \wedge cuconf(R) \geq muconf\}, \text{ if } L \notin \mathcal{G}(L).$

Tất cả các luật trong tập $SCoHUAR$ được sinh ra đầy đủ và không trùng lặp.

Chứng minh.

1) Các luật trong tập $SCoHUAR$ được sinh ra đầy đủ.

+ Ta sẽ Chứng minh $SCoHUAR \subseteq \sum_{(L,S) \in NCHUI} SCoHUAR(L,S)$. Lấy tùy ý $R: A \rightarrow B \in SCoHUAR$. Khi đó $A \in CoGHUI$, $A \cup B \in CoCHUI$ và $cuconf(R) \geq muconf$. Đặt $L = h(A)$, $S = h(A \cup B)$, trong đó $h(A)$ là bao đóng của A . Vì $A \cup B \in CoCHUI$ nên $S = A \cup B$. Do $A \in CoGHUI$ nên $A \in \mathcal{G}(L)$. Suy ra, $\emptyset \neq L \subseteq S$, $S \in CoCHUI$, nên $(L,S) \in NCHUI$. Ta lại có $S \setminus A = (A \cup B) \setminus A = B$.

Nếu $L \subset S$ thì $R: A \rightarrow B$ có dạng $L_i \rightarrow S \setminus L_i$ với $L_i = A \in \mathcal{G}(L)$, nên $R \in SCoHUAR(L,S)$ theo (a).

Nếu $L = S$ và $L \notin \mathcal{G}(L)$ thì $R: A \rightarrow B$ có dạng $L_i \rightarrow L \setminus L_i$ với $L_i = A \in \mathcal{G}(L)$, nên $R \in SCoHUAR(L,L)$ theo (b). Vậy $SCoHUAR \subseteq \sum_{(L,S) \in NCHUI} SCoHUAR(L,S)$.

+ Chiều ngược lại là hiển nhiên từ định nghĩa của $SCoHUAR(L,S)$.

2) Các luật trong $SCoHUAR$ được sinh không trùng lặp.

Giả sử tồn tại $R: A \rightarrow B$ sao cho $R \in SCoHUAR(L_1, S_1) \cap SCoHUAR(L_2, S_2)$, $(L_1, S_1) \neq (L_2, S_2)$. Theo định nghĩa, vế trái của R là một GHUI của mỗi L_j , nên $A \in \mathcal{G}(L_1)$, $A \in \mathcal{G}(L_2) \Rightarrow h(A) = L_1 = L_2$. Mặt khác, vế phải của R lần lượt là $S_1 \setminus A$ và $S_2 \setminus A$, nên $S_1 \setminus A =$

$B = S_2 \setminus A \Rightarrow S_1 = A \cup B = S_2$. Suy ra $(L_1, S_1) = (L_2, S_2)$, mâu thuẫn với giả thiết. Do đó, nếu $(L_1, S_1) \neq (L_2, S_2)$ thì $\mathcal{SCoHUAR}(L_1, S_1) \cap \mathcal{SCoHUAR}(L_2, S_2) = \emptyset$.

Từ (1) và (2), ta có các luật SCoHUAR được sinh ra đầy đủ và không trùng lặp.

4.3. Kỹ thuật tối ưu hóa

(a) Giảm số lần kiểm tra quan hệ bao hàm giữa các tập mục

Để tăng tốc việc kiểm tra quan hệ bao hàm giữa các tập mục khi áp dụng Định lý 1 để sinh ra các SCoHUAR trong tập $\mathcal{SCoHUAR}$, bài báo áp dụng kỹ thuật băm dựa trên giá trị TWU . Mỗi tập mục A được gán một khóa băm $TWU(A) = \sum_{T_i \in \rho(A)} tu(T_i)$ (xem [17]). Nếu hai tập mục xuất hiện trong cùng tập giao tác của QDB thì chúng có cùng TWU . So với việc dùng $supp(A)$ làm khóa băm, số tập mục có cùng TWU thường ít hơn đáng kể, nên số lần trùng khóa băm và số lần kiểm tra quan hệ bao hàm cũng được giảm đáng kể, góp phần cải thiện hiệu quả tính toán của thuật toán M-SCoHUAR được đề xuất trong Mục 4.3

Ví dụ 9. Xét QDB tích hợp trong Bảng 4 và tập mục $A = \{b\}$ với $supp(A) = 2$ và $TWU(A) = 78$. Kết quả ch thấy có 8 tập mục có cùng độ hỗ trợ bằng 2, trong khi ch có 2 tập mục có cùng giá trị TWU bằng 78. Sự chênh lệch này cho thấy việc sử dụng TWU làm khóa băm giúp giảm đáng kể số lượng tập mục trùng khóa so với sử dụng độ hỗ trợ.

(b) Xác định nhanh tập $\mathcal{G}(L)$ và $cuconf$

Dựa trên ý tưởng khóa băm theo TWU , sau khi khai thác 2 tập $\mathcal{CoCHUI}$ và $\mathcal{CoGHUI}$ sử dụng thuật toán CoCGHUI-Miner, với mỗi tập mục $L \in \mathcal{CoCHUI}$, ta xác định bản ghi $\langle L, supp(L), u(L), TWU(L), \{luv(x, L)\}_{x \in L}, \mathcal{G}(L) \rangle$ và chèn bản ghi này vào bảng băm HT_{closed} với khóa là $TWU(L)$. Trường $luv(x, L)$ lưu sẵn giá trị $luv(x, L) = \sum_{T_i \in \rho(L)} u(x, T_i)$ cho từng mục $x \in L$; trường $\mathcal{G}(L)$ là danh sách các CoGHUI có bao đóng là L . Với mỗi CoGHUI $X \in \mathcal{CoGHUI}$, ta truy cập danh sách $HT_{closed}[TWU(X)]$, xác định tập đóng L sao cho $X \subset L$ và $supp(X) = supp(L)$, và đưa X vào $\mathcal{G}(L)$. Chú ý rằng, các thông tin $supp$, u và TWU của L và X được tính sẵn trong thuật toán CoFCGHUI-Miner.

Nhờ cách tổ chức này, khi sinh một luật $R: A \rightarrow B$ với $A \in \mathcal{G}(L)$ và $A \cup B = L$, giá trị $luv(B, A \cup B) = luv(B, L) = \sum_{b \in B} luv(b, L)$ được tính rất nhanh bằng cách cộng các giá trị $luv(b, L)$ đã lưu sẵn trong bản ghi của L . Do đó, $cuconf(R) = luv(B, A \cup B)/u(A)$ được tính trong thời gian tuyến tính theo $|B|$, giúp toàn bộ quá trình sinh các SCoHUAR theo Định lý 1 diễn ra hiệu quả.

4.4. Thuật toán đề xuất M-SCoHUAR

Dựa vào phương pháp sinh các SCoHUAR trong Định lý 1 và các kỹ thuật tối ưu hóa trong Nhận xét 1, bài báo đề xuất thuật toán M-SCoHUAR để sinh ra nhanh tất cả các luật trong tập $\mathcal{SCoHUAR}$. Mã giả của M-SCoHUAR được trình bày trong Hình 2.

Thuật toán M-SCoHUAR hoạt động như sau. Trước hết, thuật toán xây dựng bảng băm các tập mục đóng có lợi ích và tương quan cao theo khóa $TWU(L)$ (dòng 1–6): với mỗi $L \in \mathcal{CoCHUI}$, bản ghi $rec(L) = \langle L, supp(L), u(L), TWU(L), \{luv(x, L)\}_{x \in L}, \mathcal{G}(L) := \emptyset \rangle$ được tạo ra và chèn vào trong $HT_{closed}[TWU(L)]$. Tiếp theo, đối với mỗi tập sinh $A \in \mathcal{CoGHUI}$, thuật toán tính $TWU(A)$, duyệt các bản ghi trong $HT_{closed}[TWU(A)]$ để tìm tập đóng L thỏa $h(A) = L$ (hay $A \subseteq L$ và $supp(A) = supp(L)$); khi điều kiện này được thỏa, A được thêm vào danh sách $\mathcal{G}(L)$ (dòng 7–15). Dựa trên cấu trúc này, thuật toán hình thành tập $NCHUI = \{(L, S) \mid \emptyset \neq L \subseteq S \wedge L, S \in \mathcal{CoCHUI}\}$ theo Định lý 1 (dòng 16), trong đó mỗi cặp (L, S) xác định một phân hoạch con của tập kết quả $\mathcal{SCoHUAR}$. Cuối cùng, với mỗi $(L, S) \in NCHUI$, thuật toán duyệt từng $L_i \in \mathcal{G}(L)$, xác định vế phải của luật là $B = S \setminus L_i$, sau đó tính nhanh $luv(B, S) = \sum_{b \in B} luv(b, S)$ nhờ các giá trị $luv(b, S)$ đã lưu sẵn trong $rec(S)$, và tính $cuconf(R) = luv(B, S)/u(L_i)$; nếu $cuconf(R) \geq muconf$ thì luật $R: L_i \rightarrow B$ được đưa vào tập $\mathcal{SCoHUAR}$ (dòng 17–28). Nhờ việc kết hợp băm theo TWU , ánh xạ trực tiếp từ mỗi tập đóng L sang danh sách các tập mục sinh trong $\mathcal{G}(L)$ và phân hoạch $NCHUI$, thuật toán M-SCoHUAR sinh đầy đủ và không trùng lặp tất cả các SCoHUAR trong tập $\mathcal{SCoHUAR}$ một cách hiệu quả.

Ví dụ 10. Với QDB cho trong Bảng 4, $mu = 30$, $mb = 0.2$ và $muconf = 0.6$, ta có $\mathcal{CoCHUI} = \{e, f, ef, ace, def, de, bf\}$ và $\mathcal{CoGHUI} = \{e, f, ef, a, ce, df, d, b\}$. Khi đó, ta có tập kết quả $\mathcal{SCoHUAR}$ gồm 9 luật với thông tin chi tiết cho trong bảng 5.

Bảng 5. Danh sách các luật kết hợp súc tích có lợi ích và tương quan cao.

TT	Luật $R: A \rightarrow B$	$u(A)$	$u(A \cup B)$	Bond ($A \cup B$)	Cuconf (R)
1	$e \rightarrow f$	60	120	0.625	1.2
2	$e \rightarrow ac$	60	78	0.286	1.0
3	$e \rightarrow df$	60	127	0.5	1.55
4	$e \rightarrow d$	60	96	0.714	0.9
5	$a \rightarrow ce$	40	78	0.286	0.95
6	$ce \rightarrow a$	38	78	0.286	1.05
7	$f \rightarrow de$	81	127	0.5	0.79
8	$d \rightarrow ef$	54	127	0.5	1.796
9	$d \rightarrow e$	54	96	0.71	0.78

Độ phức tạp của thuật toán. Trong Mục IV.1, bài báo chỉ ra rằng quá trình tổng thể sinh các SCoHUIR bao gồm hai giai đoạn. Giai đoạn thứ nhất sử dụng thuật toán CoFCGHUI-Miner, được cải tiến từ FCGHUI-Miner [17] thông qua việc tích hợp độ đo bond, để khai thác hai tập $CoCHUI$ và $CoGHUI$. Giai đoạn thứ hai tiến hành sinh toàn bộ các SCoHUIR từ hai tập kết quả này. Dựa trên các quan sát thực nghiệm được thực hiện trên những cơ sở dữ liệu lớn với ngưỡng mu , ms và mb nhỏ, giai đoạn thứ nhất thường tiêu tốn thời gian xử lý nhiều hơn đáng kể so với giai đoạn thứ hai. Do đó, trong bài báo này, độ phức tạp thời gian trong trường hợp xấu nhất của thuật toán CoFCGHUI-Miner được phân tích như một ví dụ minh họa. Cần lưu ý rằng độ phức tạp tính toán của thuật toán chủ yếu phụ thuộc vào số lượng các phép mở rộng có thể xảy ra. Gọi M là số lượng các mục trong tập $\mathcal{A}$, và $L = \max\{|t| : t \in \mathcal{D}\}$ biểu thị độ dài giao dịch lớn nhất trong cơ sở dữ liệu $\mathcal{D}$. Xét tập mục A bất kỳ với chiều dài $K = |A|$, và đặt $a_k = lastItem(A)$ với $K \leq k \leq M$. Thủ tục đệ quy Find-FCloGenHUI(A , mu , ms , mb , $CoCHUI$, $CoGHUI$) (hay ngắn gọn Find-FCloGenHUI(A)) có thể sinh ra nhiều nhất $E(k) \stackrel{\text{def}}{=} M - k$ các tập mở rộng (xem dòng 15 của Hình 4 trong [17]). Tại mức K ($1 \leq K \leq L$), gọi $C(K, k)$ và $\mathcal{C}(K)$ là độ phức tạp thời gian của Find-FCloGenHUI(A) để xử lý tương ứng một tập mục A và tất cả các tập mục A với $|A| = K$. Gọi $HUS(K)$ là tập tất cả các tập mục lợi ích cao có chiều dài K , và quy ước $|HUS(0)| = 1$. Khi đó, số lượng tập mở rộng tối đa của tất cả các tập mục có độ dài K là $ME(K) \stackrel{\text{def}}{=} \sum_{k=K..M} E(k) = \frac{(M-K)(M-K+1)}{2} = O(M^2)$. Do đó, ta có $C(K, k) = E(k) + \sum_{k'=k+1..M} C(K+1, k')$ với $K = 1, \dots, L-1$, $\mathcal{C}(K) = |HUS(K-1)| \times \sum_{k=K..M} C(K, k)$, và các điều kiện biên $C(L, k) \stackrel{\text{def}}{=} 0$, $\mathcal{C}(L) \stackrel{\text{def}}{=} 0$. Vì vậy, ta đạt được kết quả $C(L-1, k) = E(k) = M - k = O(M - k)$ và $\mathcal{C}(L-1) = |HUS(L-2)| \times ME(L-1) = |HUS(L-2)| \times \frac{(M-L+1)(M-L+2)}{2} = |HUS(L-2)| \times O((M-L+2)^2)$. Tương tự, $C(L-2, k) = \frac{(M-k)(M-k+1)}{2} = O((M-k)^2)$ và $\mathcal{C}(L-2) = |HUS(L-3)| \times \left[\frac{(M-L+2)(M-L+3)(M-L+4)}{6} \right] = |HUS(L-3)| \times O((M-L+3)^3)$. Bằng phương pháp quy nạp, ta thu được dạng tổng quát sau: $C(K, k) = O((M-k)^{L-K})$ và $\mathcal{C}(K) = |HUS(K-1)| \times O((M-L+2)^{L-K+1})$. Cụ thể, độ phức tạp thời gian trong trường hợp xấu nhất của thủ tục Find-FCloGenHUI là $\mathcal{C}(1) = O((M-L+2)^L)$. Khi $L = M$, $\mathcal{C}(1) = 2^M - 1 = O(2^M)$, tương ứng với số lượng tất cả các tập con khác rỗng của tập $\mathcal{A}$ gồm M thuộc tính. Tuy nhiên, khi các điều kiện cắt tĩa sớm được kích hoạt (xem các dòng 1 và 12 trong Hình 4, các dòng 1–5 trong Hình 5, và các dòng 1–6 trong Hình 6 trong [17]), toàn bộ các mở rộng

Bảng 6. Đặc trưng của các cơ sở dữ liệu.

Cơ sở dữ liệu	Số giao tác (D)	Số thuộc tính (N)	Chiều dài trung bình của giao tác (T)	Mật độ (T*100/N %)
Chess	3,196	75	37	49.33
Foodmart	4,141	1,559	4.42	0.28
Mushroom	8,124	119	23	19.33
Retail	88,162	16,470	10,30	0.06
C20D10K1	10,000	192	20	10.42

tiến thật sự của A có thể được loại bỏ. Ngoài ra, khi tích hợp độ đo tương quan bond vào thuật toán, nhờ tính chất đơn điệu giảm của bond như đã trình bày trong Mệnh đề 1, nếu điều kiện $bond(A) < mb$ được thỏa mãn, thì tập mục A và tất cả các mở rộng tiến của nó đều có thể bị cắt tĩa. Khi đó, số lượng tối đa các tập mục bị loại bỏ là $\mathcal{C}(K, k) = O((M-k)^{L-K})$.

Phân tích trên cho thấy rằng, trong thủ tục Find-FCloGenHUI (và thuật toán CoFCGHUI-Miner), các điều kiện cắt tĩa được thỏa mãn càng sớm và càng thường xuyên thì số lượng các tập ứng viên bị loại bỏ càng lớn. Nói cách khác, các điều kiện cắt tĩa giúp thu hẹp đáng kể không gian tìm kiếm, từ đó giảm độ phức tạp thời gian thực thi và nâng cao hiệu quả của thuật toán CoFCGHUI-Miner. Trên cơ sở đó, hiệu quả của toàn bộ quá trình sinh các SCoHUIR (bao gồm cả thuật toán M-SCoHUIR) cũng được cải thiện đáng kể.

5. Đánh giá thực nghiệm

Mục này đánh giá hiệu quả của phương pháp đề xuất so với các thuật toán trước đây để khai thác các luật kết hợp súc tích có lợi ích cao. Các thực nghiệm được thực hiện trên máy tính chạy hệ điều hành Windows 10, với bộ vi xử lý Intel(R) Core(TM) i5-2400 và bộ nhớ RAM 16 GB. Các thuật toán được xem xét để so sánh gồm HGB [9] và LNR-HAR [12], và được cài đặt bằng ngôn ngữ Java 8 SE. Mã nguồn của thuật toán HGB được lấy từ thư viện SPMF [37], trong khi chúng tôi cài đặt thuật toán LNR-HAR dựa trên phương pháp được mô tả trong [12]. Các bộ dữ liệu thực được lấy từ [37] và các bộ dữ liệu tổng hợp được sinh bằng công cụ tạo dữ liệu IBM Quest (lấy từ [37]), cho phép kiểm soát nhiều tham số như độ dài giao tác trung bình (T), kích thước trung bình của tập mục phổ biến tối đại (I), số lượng mục khác nhau (N), và số lượng giao tác (D) trong QDB. Đặc trưng của các QDB được trình bày trong Bảng 6.

M-SCoHUAR($CoCHUI, CoGHUI, mu, mb, muconf$)
 Đầu vào: $CoCHUI, CoGHUI$ được khám phá bởi thuật toán $CoFCGHUI$ -Miner, các ngưỡng $mu, mb, muconf$.
 Đầu ra: tập $SCoHUAR$ gồm tất cả các SCoHUAR.
 //Xây dựng bảng băm các tập đóng theo TWU

1. $SCoHUAR := \emptyset$;
2. Khởi tạo bảng băm rỗng HT_{closed} ; // khóa: TWU(L)
3. **for each** tập đóng $L \in CoCHUI$ **do** {
4. Tạo bản ghi $rec(L) = \langle L, supp(L), u(L), TWU(L), \{luv(x, L)\}_{x \in L}, \mathcal{G}(L) := \emptyset \rangle$;
5. Chèn $rec(L)$ vào $HT_{closed}[TWU(L)]$;
6. }
- // Gán mỗi CoGHUI A với tập đóng $L = h(A)$ thông qua TWU hashing
7. **for each** tập sinh $A \in CoGHUI$ **do** {
8. $t := TWU(A)$;
9. **for each** $rec(L)$ trong $HT_{closed}[t]$ **do** {
10. **if** $h(A) = L$ **then** {
11. Thêm A vào $\mathcal{G}(L)$;
12. **break**;
13. }
14. }
15. }
16. Xác định tập $NCHUI = \{(L, S) \mid \emptyset \neq L \subseteq S \wedge L, S \in CoCHUI\}$ dựa vào HT_{closed} ;
17. **for each** $(L, S) \in NCHUI$ **do** {
18. **for each** $L_i \in \mathcal{G}(L)$ **do** { // L_i là CoGHUI của L
19. $B := S \setminus L_i$; // về phải duy nhất theo phân hoạch
- // Tính nhanh $cuconf(R)$ nhờ $luv(b, S)$
20. $luvB := 0$;
21. **for each** $b \in B$ **do** $luvB := luvB + luv(b, S)$;
22. $cuconf := luvB / u(L_i)$;
23. **if** $cuconf \geq muconf$ **then** {
24. $R := (L_i \rightarrow B)$;
25. Thêm R vào $SCoHUAR$;
26. }
27. }
28. }
29. **return** $SCoHUAR$;

Hình 2. Thuật toán M-SCoHUAR.

5.1. So sánh kích thước của hai tập $CoCHUI$ và $CoGHUI$ với hai tập $CHUI$ và $GHUI$

Để đánh giá tác động của việc áp dụng độ đo tương quan bond đến số lượng các tập mục đóng và sinh có lợi ích cao, trong thực nghiệm này, chúng tôi tiến hành so sánh kích thước của hai tập $CHUI$ và $GHUI$ với hai tập $CoCHUI$ và $CoGHUI$ dưới các ngưỡng lợi ích tối thiểu khác nhau.

Nhắc lại rằng, việc khai thác hai tập $CHUI$ và $GHUI$ được thực hiện bằng thuật toán $FCGHUI$ -Miner trong [17], trong khi hai tập $CoCHUI$ và $CoGHUI$ được khai thác bằng thuật toán $CoFCGHUI$ -Miner, là phiên bản được điều chỉnh từ $FCGHUI$ -Miner thông qua việc tích hợp độ đo bond. Kết quả thực nghiệm được minh họa trong Hình 3.

Dựa trên kết quả thực nghiệm trong hình 3, ta có thể nhận thấy rằng khi ngưỡng lợi ích tối thiểu giảm, kích thước của các tập $CHUI$, $GHUI$, $CoCHUI$ và $CoGHUI$ đều tăng một cách nhất quán, phản ánh đặc điểm phổ biến của các thuật toán khai thác mẫu lợi ích cao.

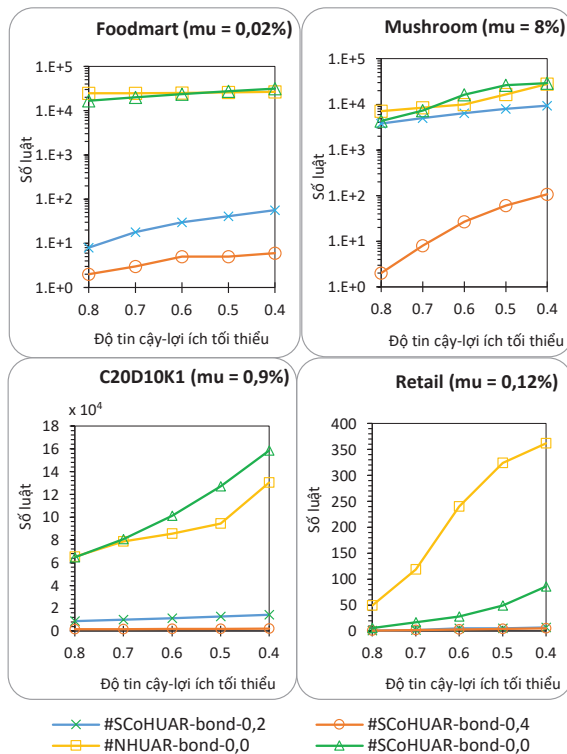
Tuy nhiên, tại mọi mức ngưỡng lợi ích, kích thước của hai tập $CoCHUI$ và $CoGHUI$ luôn nhỏ hơn đáng kể so với các tập $CHUI$ và $GHUI$ tương ứng, qua đó cho thấy tác động rõ rệt của việc tích hợp độ đo tương quan bond vào quá trình khai thác. Cụ thể, độ đo bond đã loại bỏ hiệu quả các tập mục mặc dù có lợi ích cao nhưng có mức độ tương quan yếu, dẫn đến sự giảm đáng kể số lượng mẫu được sinh ra. Những kết quả này khẳng định rằng độ đo bond không chỉ đóng vai trò như một ràng buộc tương quan bổ sung, mà còn hoạt động như một cơ chế lọc mạnh, giúp thu gọn không gian kết quả, nâng cao tính cô đọng và cải thiện khả năng diễn giải của các mẫu lợi ích cao được khai thác.

5.2. So sánh số luật được khai thác khi sử dụng $uconf$ và $cuconf$

Trong thực nghiệm này, bài báo so sánh số lượng luật được sinh ra trên các QDB khi sử dụng các hàm tin cậy - lợi ích $uconf$ và $cuconf$ kết hợp với các giá trị ngưỡng bond khác nhau. Cụ thể, #NHUAR-bond-0,0 biểu thị số luật được khai thác khi sử dụng hàm tin cậy - lợi ích $uconf$ được trình bày trong Định nghĩa 16 và đề xuất trong [9], với giá trị $mb = 0$. Trong khi đó, #SCoHUAR-bond-0,0, #SCoHUAR-bond-0,2 và #SCoHUAR-bond-0,4 lần lượt là số luật được sinh ra khi sử dụng hàm tin cậy - lợi ích mới $cuconf$ được trình bày trong Định nghĩa 17, tương ứng với các giá trị ngưỡng $mb = 0$, $mb = 0,2$ và $mb = 0,4$.

Dựa trên các kết quả thực nghiệm được trình bày trong Hình 4, ta có thể quan sát thấy rằng khi $mb = 0$, số lượng luật #SCoHUAR-bond-0.0 thu được khi sử dụng hàm tin cậy - lợi ích $cuconf$ khác biệt rõ ràng so với #NHUAR-bond-0.0 sử dụng $uconf$. Cụ thể, tùy thuộc vào đặc trưng của từng cơ sở dữ liệu, $cuconf$ có thể sinh ra nhiều luật hơn hoặc ít luật hơn so với $uconf$. Hiện tượng này phản ánh sự khác biệt về bản chất trong cách hai độ đo đánh giá mức độ tin cậy - lợi ích của luật, trong đó $cuconf$ nhấn mạnh hơn vào đóng góp lợi ích tương đối của vế phải, từ đó có thể làm nổi bật các luật mà $uconf$

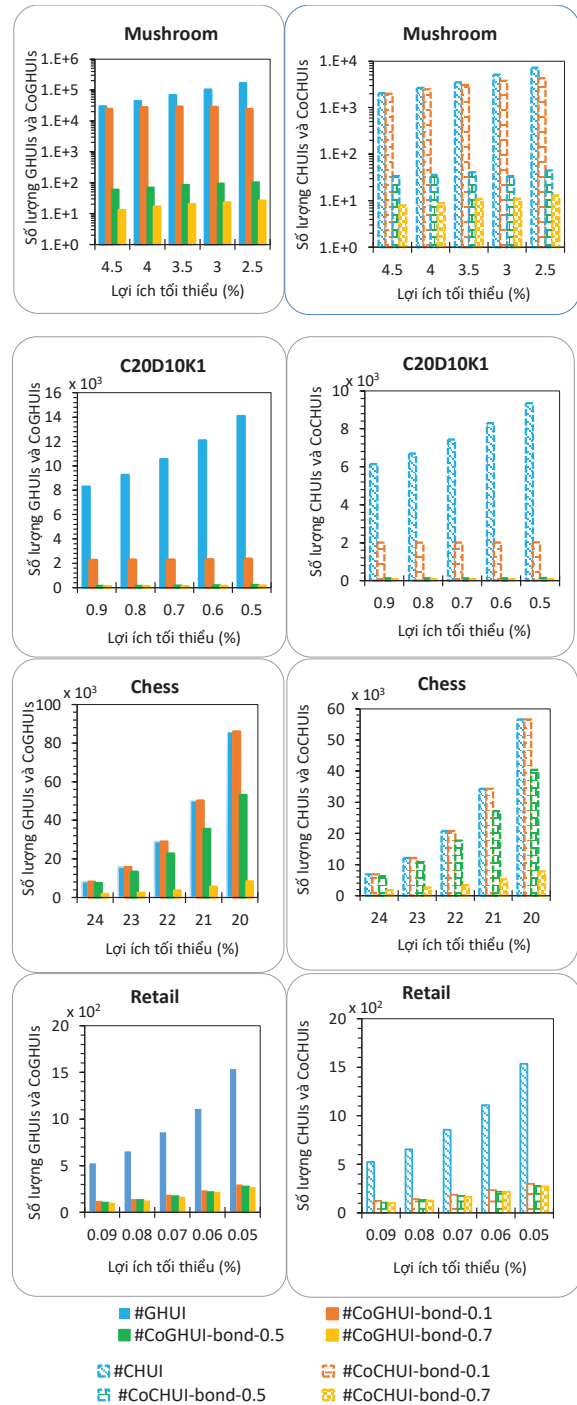
chưa phản ánh đầy đủ. Khi ngưỡng bond tăng lên $mb = 0,2$ và $mb = 0,4$, số lượng luật tương ứng (#SCoHUAR-bond-0,2 và #SCoHUAR-bond-0,4) giảm mạnh so với cả #SCoHUAR-bond-0,0 và #NHUAR-bond-0,0. Xu hướng này cho thấy độ đo bond đóng vai trò như một ràng buộc tương quan hiệu quả, giúp loại bỏ các luật tuy có lợi ích cao nhưng mức độ liên kết giữa các thuộc tính trong luật còn yếu. Đặc biệt, với $mb = 0,4$, tập luật thu được là gọn nhất, chỉ giữ lại các luật vừa thỏa mãn tiêu chí lợi ích và tin cậy cao, vừa có tương quan chặt chẽ, qua đó nâng cao đáng kể tính cô đọng và khả năng diễn giải của tập luật được khai thác.



Hình 4. So sánh số luật được sinh ra khi sử dụng các hàm tin cậy - lợi ích *uconf* và *cuconf*.

5.3. Đánh giá hiệu quả của thuật toán đề xuất

Trong mục này, bài báo tiến hành so sánh hiệu quả của thuật toán đề xuất M-SCoHUAR với các thuật toán trước đây dựa trên hai tiêu chí chính là thời gian chạy và mức sử dụng bộ nhớ cực đại. Theo hiểu biết của chúng tôi, cho đến nay hai thuật toán tiêu biểu được sử dụng để khai thác các luật kết hợp lợi ích cao không dư thừa là HGB [9] và LNR-HAR [12], đều dựa trên hàm độ tin cậy - lợi ích *uconf* được định nghĩa trong định nghĩa 16. Tuy nhiên, như đã chỉ ra trong các thực nghiệm của [12], thuật toán LNR-HAR cho thấy hiệu quả vượt trội hơn HGB cả về thời gian thực thi lẫn mức tiêu thụ bộ nhớ. Do đó, trong bài báo này, chúng tôi chỉ tiến hành so sánh thuật toán đề



Hình 3. So sánh kích thước của *CHUI* và *GHUI* với *CoCHUI* và *CoGHUI*.

xuất M-SCoHUAR với LNR-HAR. Để bảo đảm tính công bằng, hai thuật toán được so sánh trong cùng điều kiện với ngưỡng $mb=0$. Sau đó, nhằm đánh giá tác động của tham số mb đến hiệu quả của thuật toán đề xuất, M-SCoHUAR tiếp tục được thực thi với ngưỡng $mb=0,2$ trên các cơ sở dữ liệu đã sử dụng trong thực nghiệm.

Dựa trên các kết quả thực nghiệm được trình bày trong hình 5 và hình 6, ta có thể đưa ra các nhận xét và thảo luận chi tiết hơn về hiệu quả của thuật toán đề xuất M-SCoHUAR so với LNR-HAR, cũng như tác động của tham số *bond* đến thời gian chạy và mức sử dụng bộ nhớ.

Trước hết, về thời gian chạy, các đồ thị cho thấy M-SCoHUAR-bond-0,0 luôn đạt thời gian thực thi thấp hơn so với LNR-HAR-bond-0,0 trên tất cả các QDB được khảo sát. Sự khác biệt này xuất phát chủ yếu từ cách tiếp cận của các thuật toán. Cụ thể, LNR-HAR được xây dựng dựa trên phương pháp xây dựng và duyệt dần (lattice) các mẫu lợi ích cao, trong đó các mối quan hệ bao hàm giữa các tập mục phải được kiểm tra thường xuyên khi xây dựng dần. Cách tiếp cận dựa trên dần này làm phát sinh chi phí tính toán lớn, đặc biệt khi số lượng các mẫu lợi ích cao tăng nhanh, dẫn đến thời gian thực thi kéo dài. Ngược lại, M-SCoHUAR áp dụng chiến lược phân hoạch để chia không gian kết quả thành các lớp rời nhau, nhờ đó tránh được việc sinh trùng lặp cũng như giảm đáng kể chi phí kiểm tra trùng lặp giữa các tập CoCHUI và CoGHUI. Bên cạnh đó, thuật toán còn khai thác các cấu trúc dữ liệu hiệu quả và các kỹ thuật tối ưu đã nêu trong Nhận xét 1, trong đó các mẫu được lưu trữ trong bảng băm với khóa băm là giá trị TWU, giúp hạn chế mạnh việc kiểm tra quan hệ bao hàm giữa các mẫu. Việc sử dụng cấu trúc bit vector để tính toán nhanh giá trị *bond* cũng góp phần làm giảm đáng kể chi phí xử lý.

Tiếp theo, khi so sánh M-SCoHUAR-bond-0.2 với M-SCoHUAR-bond-0,0, ta có thể thấy rằng, thời gian chạy của thuật toán với *mb*=0,2 luôn nhỏ hơn so với trường hợp *mb*=0,0. Điều này cho thấy tác động tích cực của ngưỡng *bond* đến hiệu quả của thuật toán: khi giá trị *bond* tăng, các ràng buộc tương quan trở nên chặt chẽ hơn, cho phép loại bỏ sớm nhiều mẫu và luật có mức độ liên kết yếu. Hệ quả là không gian tìm kiếm được thu hẹp đáng kể, số lượng mẫu trung gian cần xử lý giảm, dẫn đến thời gian thực thi ngắn hơn. Ngược lại, với *mb*=0,0, việc không áp đặt ràng buộc tương quan khiến thuật toán phải xử lý nhiều mẫu hơn, làm gia tăng chi phí tính toán.

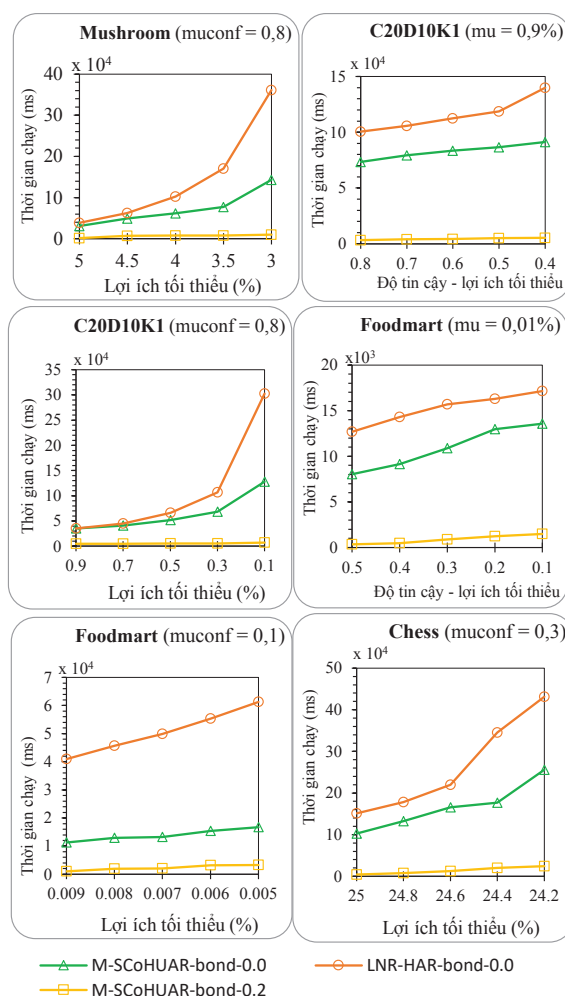
Cuối cùng, về mức sử dụng bộ nhớ, các kết quả trong Hình 6 cho thấy LNR-HAR thường tiêu thụ bộ nhớ lớn hơn so với M-SCoHUAR. Nguyên nhân chính là do phương pháp dựa trên dần của LNR-HAR đòi hỏi phải

lưu trữ một số lượng lớn các mẫu lợi ích cao. Trong khi đó, M-SCoHUAR nhờ cơ chế phân hoạch, tổ chức mẫu bằng bảng băm theo TWU và các cấu trúc dữ liệu gọn nhẹ, đã kiểm soát tốt bộ nhớ sử dụng ngay cả khi tích hợp thêm độ đo *bond*.

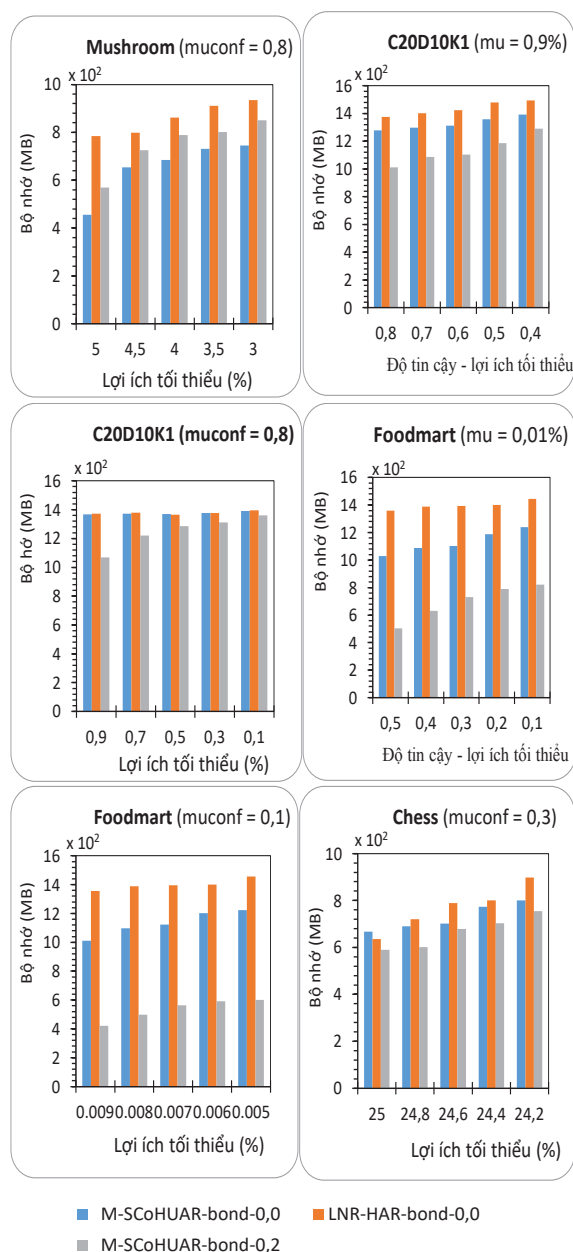
Tóm lại, các kết quả thực nghiệm cho thấy M-SCoHUAR không chỉ vượt trội hơn LNR-HAR về thời gian chạy và mức sử dụng bộ nhớ, mà còn có ưu điểm rõ rệt từ việc tích hợp tham số *bond*, qua đó nâng cao khả năng mở rộng, hiệu quả tính toán và tính thực tiễn trong khai thác các luật kết hợp súc tích có lợi ích và tương quan cao.

6. Kết luận và công việc tương lai

Bài báo này nghiên cứu bài toán khai thác các luật kết hợp súc tích, có lợi ích cao và tương quan mạnh, nhằm khắc phục tình trạng dư thừa cũng như hạn chế về khả



Hình 5. So sánh thời gian chạy của các thuật toán.



Hình 6. So sánh mức sử dụng bộ nhớ của các thuật toán.

năng diễn giải của các luật lợi ích cao truyền thống, đồng thời cho phép đánh giá toàn diện hơn cả khía cạnh lợi ích và mức độ liên kết giữa các thành phần trong luật. Trên cơ sở đó, thuật toán M-SCoHUAR được đề xuất để khai thác hiệu quả tất cả các SCoHUAR sử dụng hàm tin cậy - lợi ích mới kết hợp với độ đo tương quan bond. Ngoài ra, thuật toán M-SCoHUAR áp dụng phương pháp phân hoạch không gian của tập các SCoHUAR và cấu trúc dữ liệu hiệu quả dựa trên bảng băm theo TWU, cũng như các

kỹ thuật tối ưu nhằm giảm chi phí kiểm tra trùng lặp và quan hệ bao hàm giữa các tập mục. Kết quả thực nghiệm trên các QDB cho thấy việc tích hợp độ đo bond có tác động rõ rệt trong việc kiểm soát kích thước tập mẫu và tập luật, qua đó loại bỏ hiệu quả các luật có mức độ tương quan yếu. So với thuật toán tiêu biểu trước đây LNR-HAR, thuật toán đề xuất M-SCoHUAR đạt hiệu quả vượt trội về thời gian chạy và mức sử dụng bộ nhớ. Những kết quả này khẳng định tính hiệu quả và khả năng mở rộng của phương pháp đề xuất trong khai thác các SCoHUAR.

Trong tương lai, hướng nghiên cứu có thể được mở rộng sang các độ đo tương quan khác, các dạng cơ sở dữ liệu phức tạp hơn như dữ liệu chuỗi hoặc dữ liệu luồng, cũng như tích hợp các cơ chế giải thích (XAI) nhằm nâng cao hơn nữa khả năng ứng dụng thực tiễn của các luật được khai thác.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Trường Đại học Đà Lạt, Việt Nam theo Quyết định số 198/QĐ-ĐHDL ngày 06/03/2025.

TÀI LIỆU THAM KHẢO

- [1] R. Agrawal, R. Srikant (1994), "Fast algorithms for mining association rules in large databases", *Proc. of the 20th International Conference on Very Large Data Bases*, pp.487-499.
- [2] M. Liu, J. Qu (2012), "Mining high utility itemsets without candidate generation", *Proceedings of ACM International Conference on Information and Knowledge Management*, pp.55-64.
- [3] B.E. Shie, P.S. Yu, V.S. Tseng (2012), "Mining interesting user behavior patterns in mobile commerce environments", *Applied Intelligence*, **38**, pp.418-435.
- [4] B.E. Shie, C.W. Wu, V.S. Tseng, et al. (2013), "Efficient algorithms for mining high utility itemsets from transactional databases", *IEEE Transactions on Knowledge and Data Engineering*, **25**(8), pp.1772-1786, DOI: 10.1109/TKDE.2012.59.
- [5] Y. Liu, L. Wang, L. Feng (2020), "Mining high utility itemsets based on pattern growth without candidate generation", *Mathmetics*, pp.1-22, DOI:10.3390/math9010035.

- [6] M. Liu, J. Qu (2012), “Mining high utility itemsets without candidate generation”, *Proceedings of The 21st ACM International Conference on Information and Knowledge Management*, pp.55-64, DOI: 10.1145/2396761.2396773
- [7] P.F. Viger, C.W. Wu, S. Zida, et al. (2014), “FHM: Faster high-utility itemset mining using estimated utility cooccurrence pruning”, *Lecture Notes in Computer Science*, **8502**, pp.83-92.
- [8] S. Zida, P.F. Viger, J.C.W. Lin, et al. (2015), “A highly efficient algorithm for high-utility itemset mining”, *Proceedings of Mexican International Conference on Artificial Intelligence*, pp.530-546.
- [9] J. Sahoo, A. Kumar, A. Goswami (2015), “An efficient approach for mining association rules from high utility itemsets”, *Expert Syst. Appl.* **42(13)**, pp.5754-5778, DOI: 10.1016/j.eswa.2015.02.051.
- [10] T. Mai, B. Vo, L.T.T. Nguyen (2017), “A lattice-based approach for mining high utility association rules”, *Inf. Sci.*, **399**, pp.81-97, DOI: 10.1016/j.ins.2017.02.058.
- [11] L.T.T. Nguyen, T. Mai, B. Vo (2019), *High Utility Association Rule Mining, High-Utility Pattern Mining*, Springer, pp.161-174.
- [12] T. Mai, L.T.T. Nguyen, B. Vo, et al. (2020), “Efficient algorithm for mining non-redundant high-utility association rules”, *Sensors (Switzerland)*, **20(4)**, pp.1-17, DOI: 10.3390/s20041078..
- [13] P.F. Viger, J.C.W. Lin, T. Dinh, et al. (2016), *Mining Correlated High-Utility Itemsets Using The Bond Measure*, Springer, pp.53-65.
- [14] C.C. Aggarwal, P.S. Yu (1998), “New framework for itemset generation, Proc”, *ACM Symp. Principles of Database Systems*, pp.18-24, DOI: 10.1145/275487.275490.
- [15] E.R. Omiecinski (2023), “Alternative interest measures for mining associations in databases”, *IEEE Trans. Knowl. Data Eng.*, pp.57-69.
- [16] T. Wu, Y. Chen, J. Han (2010), “Re-examination of interestingness measures in pattern mining: A unified framework”, *Data Min. Knowl. Discov.*, pp.371-397.
- [17] T. Tran, H. Duong, T. Truong, et al. (2023), “Efficient mining of concise and informative representations of frequent high utility itemsets”, *Eng. Appl. Artif. Intell.*, **126**, DOI: 10.1016/j.engappai.2023.107111.
- [18] L.T.T. Nguyen, P. Nguyen, T.D.D. Nguyen, et al. (2019), “Mining high-utility itemsets in dynamic profit databases”, *Knowl. Based. Syst.*, **175**, pp.130-144.
- [19] T. Truong, H. Duong, B. Le, et al. (2019), “Efficient vertical mining of high average-utility itemsets based on novel upper-bounds”, *IEEE Trans. Knowl. Data Eng.*, **31**, pp.301-314.
- [20] J.F. Qu, P.F. Viger, M. Liu, et al. (2023), “Mining high utility itemsets using prefix trees and utility vectors”, *IEEE Trans. Knowl. Data Eng.*, **35**, pp.10224-10236.
- [21] Z. Cheng, W. Fang, W. Shen, et al. (2023), “An efficient utility-list based high-utility itemset mining algorithm”, *Applied Intelligence*, **53**, pp.6992-7006.
- [22] R.U. Kiran, P. Veena, P. Ravikumar, et al. (2023), “HDSHUI-miner: a novel algorithm for discovering spatial high-utility itemsets in high-dimensional spatiotemporal databases”, *Applied Intelligence*, **53**, pp.8536-8561.
- [23] J.M. Luna, R.U. Kiran, P.F. Viger (2023), “Efficient mining of top-k high utility itemsets through genetic algorithms”, *Inf. Sci.*, **624**, pp.529-553.
- [24] G.C. Lan, T.P. Hong, V.S. Tseng (2014), “An efficient projection-based indexing approach for mining high utility itemsets”, *Knowl. Inf. Syst.*, **38**, pp.85-107.
- [25] S. Dawar, V. Goyal, D. Bera (2017), “A hybrid framework for mining high-utility itemsets in a sparse transaction database”, *Applied Intelligence*, **47**, pp.809-827.
- [26] D.Q. Huy, P.F. Viger, H. Ramampiaro, et al. (2018), “Efficient high utility itemset mining using buffered utility-lists”, *Applied Intelligence*, **48**, pp.1859-1877.
- [27] Z. Deng (2018), “An efficient structure for fast mining high utility itemsets”, *Applied Intelligence*, **48**, pp.3161-3177.
- [28] T. Truong, H. Duong, B. Le, et al. (2019), “Efficient high average-utility itemset mining using novel vertical weak upper-bounds”, *Knowl. Based. Syst.*, **183**, DOI: 10.1016/j.knosys.2019.07.018.
- [29] H. Duong, T. Hoang, T. Tran, et al. (2022), “Efficient algorithms for mining closed and maximal high utility itemsets”, *Knowl. Based. Syst.*, **257**, DOI: 10.1016/j.knosys.2022.109921.

- [30] V.S. Tseng, C. Wu, P.F. Viger (2015), “Efficient algorithms for mining the concise and lossless representation of high utility itemsets”, *IEEE Trans. Knowl. Data Eng.*, **27**, pp.726-739.
- [31] C.W. Wu, P.F. Viger, J.Y. Gu, et al. (2015), “Mining closed + high utility itemsets without candidate generation”, *Conference on Technologies and Applications of Artificial Intelligence*, pp.187-194.
- [32] P.F. Viger, S. Zida, J.C.W. Lin, et al. (2016), “EFIM-Closed: Fast and memory efficient discovery of closed high-utility itemsets”, *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp.199-213.
- [33] L.T.T. Nguyen, V.V. Vu, M.T.H. Lam, et al. (2019), “An efficient method for mining high utility closed itemsets”, *Inf. Sci.*, **495**, pp.78-99.
- [34] P.F. Viger, C.W. Wu, V.S. Tseng (2014), *Novel Concise Representations of High Utility Itemsets Using Generator Patterns*, International Conference on Advanced Data Mining and Applications, pp.30-43.
- [35] J. Sahoo, A.K. Das, A. Goswami (2014), “An algorithm for mining high utility closed itemsets and generators”, *Proceedings of The 2014 IEEE International Conference on Data Science and Advanced Analytics*, pp.1-9.
- [36] S. Bouasker, S. Yahia (2015), “Key correlation mining by simultaneous monotone and anti-monotone constraints checking”, *Proceedings of The ACM Symposium on Applied Computing*, pp.851-856.
- [37] P.F. Viger, A. Gomariz, A. Soltani, et al. (2014), “SPMF: A java open-source pattern mining library”, *J. Mach. Learn. Res.*, **15**, pp.3569-3573.