Các công trình nghiên cứu, phát triển và ứng dụng

Công nghệ Thông tin và Truyền thông

https://www.ictmag.vn/cntt-tt

Tập 2023, Số 2, Tháng 12

Số đặc biệt VNICT 2023

MỤC LỤC

Một phương pháp phân cụm bán giám sát mờ đồng huấn luyện trên dữ liệu đa khung nhìn

Hoàng Thị Cành[1,2], Phùng Thế Huân[1], Vũ Thuỳ Trang[3], Phạm Huy Thông[4], Nguyễn Như Sơn[5], Lê Trường Giang[6]
[1] Trường Đại học Công nghệ Thông tin và Truyền thông, Đại học Thái Nguyên, Thái Nguyên, Việt Nam
[2] Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Hà Nội, Việt Nam
[3] Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội, Hà Nội, Việt Nam
[4] Viện Công nghệ thông tin, Đại học Quốc gia Hà Nội, Hà Nội, Việt Nam
[5] Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Hà Nội, Việt Nam
[6] Trường Đại học Công nghiệp Hà Nội, Hà Nội, Việt Nam
Tác giả liên hệ: Phùng Thế Huân, pthuan@ictu.edu.vn
Ngày nhận bài: 30/06/2023, ngày sửa chữa: 10/09/2023, ngày duyệt đăng: 26/09/2023
Định danh DOI: 10.32913/mic-ict-research-vn.v2023.n2.1230

Tóm tắt: Trong thực tế hiện nay, dữ liệu đa khung nhìn ngày càng phổ biến. Dữ liệu đa khung nhìn (Multi-view data) đề cập đến loại dữ liệu bao gồm nhiều quan điểm hoặc góc nhìn về một đối tượng. Dữ liệu trong mỗi khung nhìn riêng lẻ có thuộc tính cụ thể thực hiện nhiệm vụ khám phá tri thức riêng và cung cấp các thông tin về cùng một vấn đề với độ chính xác và độ tin cậy khác nhau. Việc kết hợp nhiều loại thông tin từ các khung nhìn, có thể thu được biểu diễn đầy đủ và chính xác hơn về các đối tượng, dẫn đến việc phân tích dữ liệu và ra quyết định được cải thiện. Phân cụm đa khung nhìn là hướng nghiên cứu đã thu hút được sự quan tâm của các nhà khoa học trong nhiều năm gần đây. Tuy nhiên, chưa có nghiên cứu nào tập trung vào phân cụm bán giám sát mờ kết hợp thuật toán đồng huấn luyện để đánh giá độ chính xác và chất lượng phân cụm trên tập dữ liệu đa khung nhìn. Bài báo này đề xuất một phương pháp mới trong phân cụm bán giám sát mờ, sử dụng thuật toán đồng huấn luyện trên dữ liệu đa khung nhìn thu thập từ một nguồn dữ liệu. Đồng thời, bài báo cũng cung cấp các kết quả thực nghiệm để đánh giá tính hiệu quả và độ chính xác của thuật toán đề xuất.

Từ khóa: *Dữ liệu đa khung nhìn, phân cụm đa khung nhìn, phân cụm bán giám sát mờ, thuật toán đồng huấn luyện*

Title: **A Semi-Supervised Fuzzy Clustering Co-Training Approach on Multi-View Data**

Abstract: In today's practical reality, multi-view data is increasingly prevalent. Multi-view data refers to a type of data that encompasses multiple perspectives or viewpoints of an object. Data within each individual view possesses specific attributes dedicated to knowledge discovery and provides information on the same subject with varying degrees of accuracy and reliability. Combining various types of information from different views can yield a more comprehensive and accurate representation of objects, thereby improving data analysis and decision-making. Multi-view clustering has emerged as a research direction that has garnered the interest of scientists in recent years. However, there has been no research focusing on semi-supervised fuzzy clustering combined with co-training algorithms to assess the accuracy and quality of clustering on multi-view datasets. This paper proposes a novel method in semi-supervised clustering, utilizing co-training algorithms on multi-view data collected from a data source. Additionally, the paper provides experimental results to evaluate the effectiveness and accuracy of the proposed algorithm.

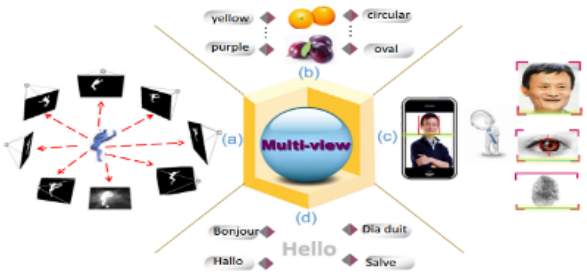
Keywords: *Multi-view data, multi-view clustering, semi-supervised fuzzy clustering, co-training algorithm.*

I. MỞ ĐẦU

Quá trình phân cụm dữ liệu đóng vai trò quan trọng trong khai phá dữ liệu nhằm tìm kiếm, phát hiện các nhóm dữ liệu theo đặc trưng để cung cấp tri thức cho quá trình ra quyết định. Phân cụm dữ liệu là quá trình phân chia các phần tử dữ liệu về các cụm dựa trên mức độ tương đồng giữa chúng, sao cho mức độ tương đồng giữa các phần tử trong cùng một cụm được tối đa hóa và mức độ tương đồng giữa các phần tử khác cụm được cực tiểu hóa [1]. Hầu hết các thuật toán phân cụm hiện tại được thiết kế cho dữ liệu một khung nhìn, trong khi các bài toán thực tế hiện nay nhu cầu sử dụng dữ liệu đa khung nhìn rất phổ biến.

Dữ liệu đa khung nhìn (Multi-view data) đề cập đến loại dữ liệu bao gồm nhiều quan điểm hoặc góc nhìn về một đối

tượng. Trong đó, dữ liệu được biểu diễn dưới nhiều khung nhìn khác nhau, mỗi khung nhìn cung cấp các thông tin và thuộc tính khác nhau về dữ liệu. Dữ liệu trong các khung nhìn được thu thập từ các phương thức, nguồn, các dạng khác nhau hoặc được quan sát từ các góc nhìn khác nhau [2]. Dữ liệu đa khung nhìn được áp dụng trong nhiều bài toán thực tế như: học máy, xử lý ảnh, kinh doanh, y tế, hệ thống tư vấn, khoa học dữ liệu [3], v.v. Điều này thúc đẩy sự cần thiết phát triển các phương pháp phân cụm tiên tiến, nhằm khám phá tri thức từ các bộ dữ liệu đa khung nhìn.

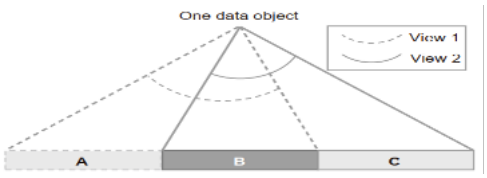


Hình 1. Ví dụ về dữ liệu đa khung nhìn [4]

Phân cụm đa khung nhìn (Multi-view Clustering - MvC) là một phương pháp phân cụm dữ liệu dựa trên việc sử dụng nhiều khung nhìn độc lập, nhằm tìm ra các nhóm dữ liệu tương đồng với nhau trong các khung nhìn. Rõ ràng, việc tích hợp thông tin từ nhiều khung nhìn và phát hiện ra tri thức tiềm ẩn chung được chia sẻ bởi nhiều khung nhìn mang lại lợi ích lớn cho việc phân cụm dữ liệu [5]. So sánh với các phương pháp phân cụm trên một khung nhìn thì phân cụm đa khung nhìn có một số ưu điểm vượt trội đó là nâng cao chất lượng phân cụm, giảm thiểu sự phụ thuộc vào khung nhìn và xử lý dữ liệu phức tạp [6]. Tuy nhiên, phân cụm đa khung nhìn đang đối diện với nhiều thách thức như đòi hỏi các khung nhìn độc lập, tốn nhiều chi phí cho việc thu thập, xử lý, lưu trữ dữ liệu từ nhiều khung nhìn và khó khăn trong việc kết hợp các phương pháp phân cụm khác nhau[6].

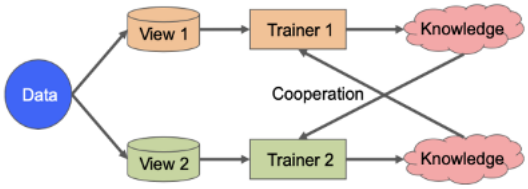
Trên thế giới, nhiều nhà nghiên cứu đang nỗ lực tìm cách giải quyết những khó khăn đã nêu ở trên. Vào năm 2004, Bickel và Scheffer [7] tiên phong trong nghiên cứu phương pháp phân cụm đa khung nhìn. Họ đã mở rộng phương pháp phân cụm K-means và Expectation Maximization (EM) để phù hợp với môi trường đa khung nhìn và xử lý dữ liệu văn bản với hai khung nhìn độc lập có điều kiện. Nhiều phương pháp phân cụm đa khung nhìn đã được đề xuất dựa trên nghiên cứu quan trọng này [8,9,10]. Phương pháp phân cụm bán giám sát sâu đa khung nhìn (DMSC - Deep Multi-view Semi-supervised Clustering) [11] do Rui Chen và cộng sự đề xuất, có thể tăng hiệu suất của phân cụm đa khung nhìn một cách hiệu quả bằng cách lấy thông tin được giám sát yếu có trong các ràng buộc theo cặp mẫu và bảo vệ các thuộc tính của dữ liệu đa khung nhìn. Trong một nghiên cứu khác, Li B và các cộng sự [12] đã đưa ra một phương pháp phân cụm đa khung nhìn mới dựa trên phân tích ma trận phi tuyến, với mục tiêu tối ưu hóa sự giống nhau giữa các khung nhìn. Phương pháp này sử dụng một mô hình ma trận phi tuyến hóa không âm (NMF - Non-negative matrix factorization) để tách các yếu tố ẩn chung từ các khung nhìn khác nhau của dữ liệu.

Hai nguyên tắc quan trọng đảm bảo sự thành công của thuật toán MvC chính là nguyên tắc bổ sung và nguyên tắc đồng thuận [6]. Nguyên tắc bổ sung khẳng định rằng nên sử dụng nhiều khung nhìn khác nhau để mô tả các đối tượng dữ liệu một cách toàn diện và chính xác hơn. Mặc dù mỗi khung nhìn riêng lẻ đã cung cấp đủ thông tin cho một nhiệm vụ khám phá tri thức cụ thể tuy nhiên, các khung nhìn khác nhau thường chứa các thông tin bổ sung cần được khai thác. Nguyên tắc đồng thuận nhằm mục đích tối đa hóa tính nhất quán giữa nhiều khung nhìn. Hình 2 minh hoạ nguyên tắc bổ sung và nguyên tắc đồng thuận [6]. Giả sử một đối tượng dữ liệu có hai khung nhìn, được ánh xạ vào một không gian dữ liệu ẩn: (i) thành phần A và thành phần C tồn tại trong khung nhìn riêng, phần A trong khung nhìn 1 và phần C trong khung nhìn 2, thể hiện tính bổ sung của hai khung nhìn; (ii) thành phần B của đối tượng được chia sẻ bởi cả hai khung nhìn, thể hiện sự đồng thuận giữa hai khung nhìn.



Hình 2. Minh hoạ nguyên tắc bổ sung và nguyên tắc đồng thuận

Thuật toán đồng huấn luyện, được giới thiệu bởi Blum và Mitchell vào năm 1998 [13], đã trở thành một công nghệ mang tính tiên phong và được coi là một trong những phương pháp phổ biến nhất trong lĩnh vực học đa khung nhìn. Mục tiêu của phương pháp này là tối đa hoá sự đồng thuận qua các khung nhìn để đạt được sự đồng thuận rộng nhất và cải thiện hiệu suất phân cụm.



Hình 3. Quy trình chung của thuật toán đồng huấn luyện [6]

Mặc dù đã có nhiều nghiên cứu trước đây tìm cách giải quyết các thách thức của phân cụm đa khung nhìn, nhưng các nghiên cứu lại chủ yếu tập trung sử dụng phương pháp phân cụm rõ. Mà đối với các loại dữ liệu phức tạp, khi các điểm dữ liệu có thể cùng lúc thuộc vào nhiều cụm với trọng số khác nhau thì việc áp dụng các thuật toán phân cụm rõ là hoàn toàn không hiệu quả. Trong khi đó, phân cụm bán giám sát mờ cho phép phân tích dữ liệu và phát hiện các cụm dữ liệu khác nhau. Điều này hỗ trợ trong việc đưa ra quyết định về phân tích dữ liệu, phát hiện dữ liệu nhiễu và tìm kiếm những đặc trưng quan trọng trong dữ liệu. Vì vậy, bài báo này nhằm lấp đầy khoảng trống nghiên cứu bằng cách đề xuất một phương pháp phân cụm bán giám sát mờ sử dụng thuật toán đồng huấn luyện trên dữ liệu đa khung nhìn có tên gọi SSFCC (Semi Supervised Fuzzy Multi-View Co-training Clustering).

Phần còn lại của bài báo được tổ chức như sau: Phần II trình bày một số phương pháp tiếp cận về phân cụm đa khung nhìn, phân cụm đồng huấn luyện. Phần III trình bày phương pháp đề xuất. Phần IV trình bày kết quả thực nghiệm và phân tích, đối sánh. Một số kết luận và định hướng nghiên cứu trong bài báo tiếp theo được đưa ra trong phần kết luận.

II. MỘT SỐ PHƯƠNG PHÁP TIẾP CẬN

1. Phân cụm K-means đa khung nhìn

Thuật toán K-means là một trong những thuật toán phân cụm dữ liệu phổ biến nhất và có khả năng xử lý tốt các tập dữ liệu quy mô lớn. K-means đã được áp dụng rộng rãi trong nhiều lĩnh vực như: phân tích mạng xã hội, kinh doanh, y tế v.v. Mặc dù nó đã được nghiên cứu kỹ trong nhiều thập kỷ qua, nhưng nhiều biến thể của K-means liên tục được đưa ra.

Đối với các bài toán phân cụm đa khung nhìn, thuật toán K-means được mở rộng như sau [14]:

Giả sử $\mathcal{X} = \{X^{(1)}, X^{(2)}, ..., X^{(V)}\} \in \mathbb{R}^V$ là tập dữ liệu tổng quát của tất cả V khung nhìn. Hàm mục tiêu của K-means đa khung nhìn có dạng như sau:

$$J = \sum_{v=1}^{V} \mu_v^{\gamma} J_v \tag{1}$$

Với các ràng buộc:

$$\begin{cases} \mu_v \geq 0 \\ \sum_{v=1}^{V} \mu_v = 1 \\ \gamma > 1 \end{cases}$$

Trong đó, μ_v là trọng số của view thứ v, γ là số mũ điều chỉnh μ_v , J_V tướng ứng với hàm mục tiêu K-means của view thứ v:

$$J_v = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ik} ||x_i^{(v)} - v_k^{(v)}||^2 \tag{2}$$

Phương pháp phân cụm K-means đa khung nhìn sử dụng thông tin từ nhiều khung nhìn để cải thiện quá trình phân cụm. Bằng cách kết hợp các khung nhìn, thuật toán sẽ gán các điểm dữ liệu vào cùng một cụm, đảm bảo tính nhất quán trong quá trình phân cụm giữa các khung nhìn và tăng cường hiệu suất phân cụm.

2. Phân cụm quang phổ đa khung nhìn đồng huấn luyện

Phân cụm quang phổ đa khung nhìn cho phép khám phá các cấu trúc cụm tiềm ẩn bằng cách kết hợp thông tin từ nhiều đồ thị. Kumar và đồng nghiệp đã đưa ra một phương pháp phân cụm quang phổ đa khung nhìn sử dụng ý tưởng đồng huấn luyện [15]. Phương pháp này tuân theo tính nhất quán của việc học đa khung nhìn, cho phép kết hợp thông tin từ nhiều khung nhìn để cải thiện quá trình phân cụm. Trong đó, mỗi khung nhìn cung cấp các nhãn giống nhau cho tất cả các mẫu dữ liệu. Nhờ đó, phương pháp này có thể sử dụng vector riêng của một khung nhìn để "gán nhãn" cho một khung nhìn khác và ngược lại.

Thuật toán quang phổ đa khung nhìn đồng huấn luyện được triển khai thông qua các bước như sau [15]:

Bước 1: Sử dụng phân cụm quang phổ trên từng khung nhìn, thu được các vector riêng gọi là U_1 và U_2.

Bước 2: Phân cụm các điểm sử dụng U_1 và sử dụng phân cụm này để điều chỉnh cấu trúc đồ thị trong khung nhìn 2.

Bước 3: Phân cụm các điểm sử dụng U_2 và sử dụng phân cụm này để điều chỉnh cấu trúc đồ thị trong khung nhìn 1.

Bước 4: Quay lại **Bước 1** và lặp lại một số vòng lặp.

III. PHƯƠNG PHÁP ĐỀ XUẤT

Bài báo đề xuất một giải pháp phân cụm bán giám sát mờ đa khung nhìn sử dụng thuật toán đồng huấn luyện. Nó được áp dụng cho dữ liệu đa khung nhìn được thu thập từ một nguồn có những đặc điểm sau: số lượng bản ghi trên mỗi khung nhìn là bằng nhau, số lượng thuộc tính trên mỗi khung nhìn có thể khác nhau và quan hệ giữa hai khung nhìn là ánh xạ một – một.

1. Ý tưởng thuật toán

Thuật toán SSFCC được đề xuất bao gồm 2 bước cụ thể như sau:

Bước 1: Phân cụm mờ cho dữ liệu chưa được gán nhãn trên mỗi khung nhìn. Trong bước này, thuật toán FCM (Fuzzy C-Means) được sử dụng độc lập cho từng khung nhìn (viewA và viewB) để chia các điểm dữ liệu vào các cụm. Kết quả

của bước này gồm các tâm cụm và độ thuộc tương ứng trên từng khung nhìn. Độ thuộc được ký hiệu là: $\bar{u}^A$ (độ thuộc của điểm dữ liệu trên viewA so với các tâm cụm) và $\bar{u}^B$ (độ thuộc của điểm dữ liệu trên viewB so với các tâm cụm).

Bước 2: Phân cụm bán giám sát mờ đồng huấn luyện đa khung nhìn. Trong bước này, kết quả của viewA được sử dụng làm dữ liệu huấn luyện cho viewB và ngược lại. Mục tiêu là tối đa hóa sự đồng thuận chéo qua tất cả các khung nhìn và đạt được sự đồng thuận rộng nhất. Đầu vào của bước này là các thành phần bán giám sát $\bar{u}^A$ và $\bar{u}^B$ thu được từ **Bước 1**. Đầu ra của bước này là các giá trị tâm cụm cuối cùng và giá trị độ thuộc tương ứng trên mỗi khung nhìn.

2. Chi tiết thuật toán SSFCC

Dựa trên ý tưởng đã trình bày ở phần trước, phần này sẽ trình bày mô hình hoá của phương pháp đề xuất. Hàm mục tiêu của phương pháp được biểu diễn bởi ba thành phần, như sau:

$$
\begin{aligned}
J(u^A, v^A, u^B, v^B) &= \sum_{k=1}^{N}\sum_{j=1}^{c} {u_{kj}^A}^2 ||x_k^A - v_j^A||^2 \\
&+ \sum_{k=1}^{N}\sum_{j=1}^{c} {u_{kj}^B}^2 ||x_k^B - v_j^B||^2 \\
&+ \sum_{k=1}^{N}\sum_{j=1}^{c} (u_{kj}^A - u_{kj}^B)^2 (||x_k^A - v_j^A||^2 + ||x_k^B - v_j^B||^2) \\
&+ \sum_{k=1}^{N}\sum_{j=1}^{c} (u_{kj}^A - \bar{u}_{kj}^A)^2 ||x_k^A - v_j^A||^2 \\
&+ \sum_{k=1}^{N}\sum_{j=1}^{c} (u_{kj}^B - \bar{u}_{kj}^B)^2 ||x_k^B - v_j^B||^2 \rightarrow Min
\end{aligned} \tag{3}
$$

Với các ràng buộc:

$$
\begin{cases}
\sum_{j=1}^{c} u_{kj}^A = \sum_{j=1}^{c} u_{kj}^B = 1 \\
u_{kj} \in [0,1]
\end{cases}
$$

Chú thích:

- x_k^A, x_k^B : đại diện cho điểm dữ liệu thứ k trên viewA và viewB tương ứng
- v_j^A, v_j^B : đại diện cho tâm cụm thứ j trên viewA và viewB tương ứng
- u_{kj}^A, u_{kj}^B : đại diện cho độ thuộc của điểm dữ liệu thứ k ở cụm thứ j trên viewA và viewB tương ứng
- $\bar{u}_{kj}^A, \bar{u}_{kj}^B$: đại diện cho độ thuộc của điểm dữ liệu thứ k ở cụm thứ j sau khi sử dụng thuật toán FCM trên viewA và viewB tương ứng
- $||x_k^A - v_j^A||^2, ||x_k^B - v_j^B||^2$: đại diện cho khoảng cách từ điểm dữ liệu thứ k đến tâm cụm thứ j trên viewA và viewB tương ứng

Hàm mục tiêu (3), thể hiện rõ thành phần của phân cụm mờ, phân cụm đồng huấn luyện và phân cụm bán giám sát, cụ thể:

- Thành phần thể hiện phân cụm mờ:

$$\sum_{k=1}^{N}\sum_{j=1}^{c} {u_{kj}^A}^2 ||x_k^A - v_j^A||^2$$

và

$$\sum_{k=1}^{N}\sum_{j=1}^{c} {u_{kj}^B}^2 ||x_k^B - v_j^B||^2$$

- Thành phần thể hiện phân cụm đồng huấn luyện:

$$\sum_{k=1}^{N}\sum_{j=1}^{c} (u_{kj}^A - u_{kj}^B)^2 (||x_k^A - v_j^A||^2 + ||x_k^B - v_j^B||^2)$$

- Thành phần thể hiện phân cụm bán giám sát:

$$\sum_{k=1}^{N}\sum_{j=1}^{c} (u_{kj}^A - \bar{u}_{kj}^A)^2 ||x_k^A - v_j^A||^2$$

và

$$\sum_{k=1}^{N}\sum_{j=1}^{c} (u_{kj}^B - \bar{u}_{kj}^B)^2 ||x_k^B - v_j^B||^2$$

Các tâm cụm và độ thuộc của bài toán tối ưu (3) được tính toán như sau:

Tâm cụm v_j^A và v_j^B có công thức như sau:

$$
v_j^A = \frac{\sum_{k=1}^{N} \left[{u_{kj}^A}^2 + (u_{kj}^A - u_{kj}^B)^2 + (u_{kj}^A - \bar{u}_{kj}^A)^2 \right] . x_k^A}{\sum_{k=1}^{N} \left[{u_{kj}^A}^2 + (u_{kj}^A - u_{kj}^B)^2 + (u_{kj}^A - \bar{u}_{kj}^A)^2 \right]} \tag{4}
$$

$$
v_j^B = \frac{\sum_{k=1}^{N} \left[{u_{kj}^B}^2 + (u_{kj}^A - u_{kj}^B)^2 + (u_{kj}^B - \bar{u}_{kj}^B)^2 \right] . x_k^B}{\sum_{k=1}^{N} \left[{u_{kj}^B}^2 + (u_{kj}^A - u_{kj}^B)^2 + (u_{kj}^B - \bar{u}_{kj}^B)^2 \right]} \tag{5}
$$

Phương pháp nhân tử Lagrange được sử dụng để xác định độ thuộc u_{ij}^A và u_{ij}^B , được xác định bằng công thức sau:

$$
u_{kj}^A = \frac{\frac{\lambda_A}{2} + \Delta_{kj}^A}{3{d_{kj}^A}^2 + {d_{kj}^B}^2} \tag{6}
$$

Trong đó:

$$\Delta_{kj}^A = u_{kj}^B ({d_{kj}^A}^2 + {d_{kj}^B}^2) + \bar{u}_{kj}^A {d_{kj}^A}^2$$

$$
\frac{\lambda_A}{2} = \frac{1 - \sum_{i=1}^{c} \frac{\Delta_{ki}^A}{3{d_{ki}^A}^2 + {d_{ki}^B}^2}}{\sum_{i=1}^{c} \frac{1}{3{d_{ki}^A}^2 + {d_{ki}^B}^2}}
$$

$$u_{kj}^{B} = \frac{\frac{\lambda_B}{2} + \Delta_{kj}^{B}}{3{d_{kj}^{B}}^2 + {d_{kj}^{A}}^2} \quad (7)$$

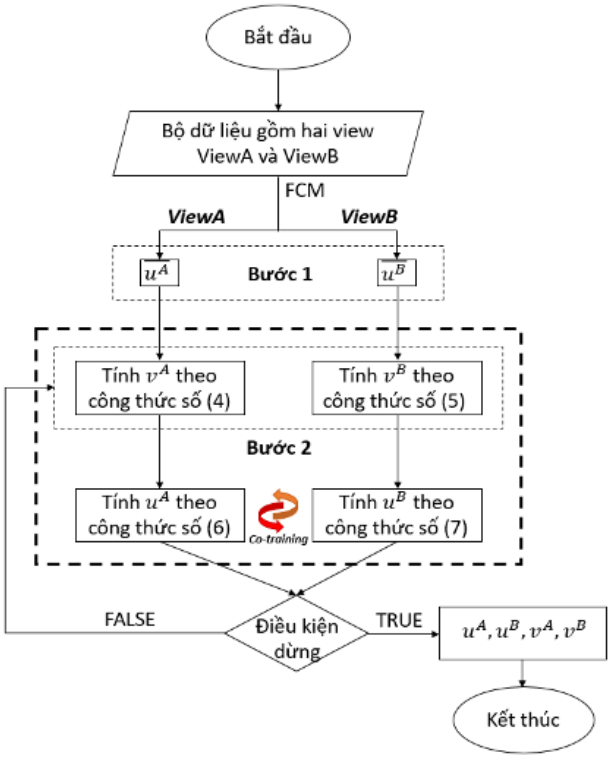
Trong đó:

$$\Delta_{kj}^{B} = u_{kj}^{A}({d_{kj}^{A}}^2 + {d_{kj}^{B}}^2) + \bar{u}_{kj}^{B}{d_{kj}^{B}}^2$$

$$\frac{\lambda_B}{2} = \frac{1 - \sum_{i=1}^{c} \frac{\Delta_{ki}^{B}}{3{d_{ki}^{B}}^2 + {d_{ki}^{A}}^2}}{\sum_{i=1}^{c} \frac{1}{3{d_{ki}^{B}}^2 + {d_{ki}^{A}}^2}}$$

3. Sơ đồ thuật toán SSFCC

Hình 4 dưới đây mô tả sơ đồ của thuật toán SSFCC.



Hình 4. Sơ đồ thuật toán

4. Thuật toán SSFCC

Thuật toán 1 SSFCC

Input:
- Tập dữ liệu X gồm hai view: viewA: $X^A = \{x_1^A, x_2^A, x_3^A, \ldots, x_n^A\}$, viewB: $X^B = \{x_1^B, x_2^B, x_3^B, \ldots, x_n^B\}$.
- Số lượng bản ghi trên viewA và viewB là bằng nhau.
- Số thuộc tính trên viewA và viewB có thể khác nhau.
- Số cụm c
- Số lần lặp Maxstep
- Sai số cho phép ϵ

Output: u^A, u^B, v^A, v^B

BEGIN

Bước 1: Phân cụm mờ cho dữ liệu chưa được gán nhãn trên mỗi khung nhìn

1.1. Áp dụng thuật toán FCM lên viewA thu được $\bar{u}^A$ và $\bar{v}^A$

1.2. Áp dụng thuật toán FCM lên viewB thu được $\bar{u}^B$ và $\bar{v}^B$

Bước 2: Phân cụm bán giám sát mờ đồng huấn luyện đa khung nhìn

2.1. Khởi tạo t = 0

Repeat

2.2. $t = t + 1$

2.3. Cập nhật v^A theo công thức số (4)

2.4. Cập nhật v^B theo công thức số (5)

2.5. Cập nhật u^A theo công thức số (6)

2.6. Cập nhật u^B theo công thức số (7)

Until

$Max\{||u^{A^{(t+1)}} - u^{A^{(t)}}||, ||u^{B^{(t+1)}} - u^{B^{(t)}}||\} \leq \epsilon$ hoặc $t \geq$ Maxstep

END.

IV. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

1. Các điều kiện thực nghiệm

Nhằm kiểm chứng hiệu suất của phương pháp đề xuất, nhóm nghiên cứu đã tiến hành cài đặt mô phỏng trên 9 bộ dữ liệu lấy từ kho dữ liệu học máy UCI [16]. Các bộ dữ liệu bao gồm Australian, Balance-scale, Heart, Iris, Spambase, Tae, Waweform, Wdbc, Wine.

Đối với mỗi bộ dữ liệu, chúng tôi đã chia thành hai view (viewA và viewB) và các thuộc tính trên mỗi view được lựa chọn ngẫu nhiên. Thông tin chi tiết về các bộ dữ liệu được trình bày trong Bảng I.

Việc cài đặt thực nghiệm được thực hiện trên máy tính thương hiệu Apple MacBook Air M1 2020 với cấu hình 8GB/256GB/7-core GPU, ngôn ngữ lập trình Python phiên bản 3.10.

Bảng I
THÔNG TIN CHI TIẾT CÁC BỘ DỮ LIỆU DÙNG CHO THỰC NGHIỆM

ST	Bộ dữ liệu	viewA		viewB		Số cụm
		Số lượng mẫu	Số thuộc tính	Số lượng mẫu	Số thuộc tính	
1	Australian	690	7	690	7	2
2	Balance-scale	625	2	625	2	3
3	Heart	270	6	270	7	2
4	Iris	150	2	150	2	3
5	Spambase	4601	29	4601	28	2
6	Tae	151	2	151	2	3
7	Waweform	5000	20	5000	20	3
8	Wdbc	569	15	569	15	2
9	Wine	178	6	178	7	3

Bảng II
BẢNG GIÁ TRỊ SO SÁNH HIỆU NĂNG VỀ ĐỘ CHÍNH XÁC PHÂN CỤM

Bộ dữ liệu	SSFCC		MKC		CMSC	
	Trung bình	Phương sai	Trung bình	Phương sai	Trung bình	Phương sai
Australian	**0.7109**	0.0037	0.6227	0.0162	0.4644	0.0009
Balance-scale	**0.5511**	0.013	0.549	0.0305	0.4032	0.0038
Heart	0.4517	0.2376	**0.5414**	0.0194	0.1533	0.0516
Iris	**0.8217**	0.0373	0.6411	0.0175	0.1573	0.0339
Spambase	0.3645	0.0563	**0.5144**	0.0143	0.3972	0.0001
Tae	**0.7785**	0.0009	0.458	0.0181	0.3278	0.0001
Waweform	0.3619	0.0214	0.5711	0.0253	**0.5885**	0.0087
Wdbc	**0.8493**	0.0625	0.5683	0.0153	0.2425	0.0097
Wine	0.3401	0.0477	**0.5889**	0.0189	0.2303	0.0126

Bảng III
BẢNG GIÁ TRỊ SO SÁNH HIỆU NĂNG VỀ CHẤT LƯỢNG PHÂN CỤM

Bộ dữ liệu	SSFCC		MKC		CMSC	
	Trung bình	Phương sai	Trung bình	Phương sai	Trung bình	Phương sai
Australian	**1.9306**	1.6895	2.2527	0.0377	3.5724	0.9613
Balance-scale	**1.0102**	0.0001	3.9196	0.1511	5.0247	0.6785
Heart	**2.3391**	0.0004	2.5519	0.9348	2.3409	0.3888
Iris	**0.7246**	2.8865	3.0464	0.0497	6.4878	0.9114
Spambase	**1.5221**	0.9296	2.7543	0.1186	2.2847	0.4525
Tae	**2.8793**	10.0895	2.8909	0.1649	6.3583	1.6738
Waweform	28.6892	0.0504	22.9106	1.937	**2.0502**	0.005
Wdbc	0.6525	0.0014	2.0328	0.0299	**0.5745**	0.0222
Wine	3.2525	2.4586	**2.8765**	0.0191	4.6377	1.8099

Để đánh giá hiệu suất, chúng tôi sử dụng các tiêu chí sau: i) Độ chính xác phân cụm: chúng tôi sử dụng độ chính xác phân cụm CA (Clustering Accuracy) [17]. Độ chính xác phân cụm đo lường mức độ chính xác của việc gán nhãn cho các điểm dữ liệu trong từng cụm. Giá trị CA càng cao cho thấy hiệu suất phân cụm càng tốt. ii). Chất lượng phân cụm: Chúng tôi sử dụng độ đo DB (Davies–Bouldin index) [18]. Chất lượng phân cụm đo lường độ tách biệt và đồng nhất giữa các cụm. Giá trị DB càng nhỏ cho thấy chất lượng phân cụm càng tốt.

2. Kết quả thực nghiệm

Phương pháp SSFCC được so sánh với hai phương pháp khác là Multi-view K-means Clustering (MKC) [14] và Co-trained Multi-view Spectral Clustering (CMSC) [15].

a) Kết quả thực nghiệm đánh giá theo độ chính xác phân cụm

Kết quả đánh giá độ chính xác phân cụm của phương pháp đề xuất (SSFCC) đã được so sánh với hai phương pháp MKC và CMSC trên 9 bộ dữ liệu được trình bày trong bảng II.

Trong Bảng II, phương pháp đề xuất SSFCC đã đạt giá trị tốt nhất trong 5/9 trường hợp (Australian, Balance-scale, Iris, Tae, Wdbc), trong khi phương pháp MKC chỉ có 3/9 trường hợp nhận giá trị tốt nhất (Heart, Spambase, Wine) và phương pháp CMSC chỉ có 1/9 trường hợp nhận giá trị tốt nhất (Waweform). Do vậy, phương pháp SSFCC cho kết quả tốt hơn phương pháp MKC và CMSC trong việc đạt được độ chính xác phân cụm.

b) Kết quả thực nghiệm đánh giá chất lượng phân cụm

Kết quả đánh giá chất lượng phân cụm của phương pháp đề xuất (SSFCC) đã được so sánh với hai phương pháp MKC và CMSC trên 9 bộ dữ liệu được trình bày trong bảng III.

Trong Bảng III, phương pháp SSFCC đã đạt giá trị tốt nhất trong 6/9 trường hợp (Autralian, Balance-scale, Heart, Iris, Spambase, Tae), trong khi phương pháp MKC chỉ có 1/9 trường hợp nhận giá trị tốt nhất (Wine) và phương pháp CMSC chỉ có 2/9 trường hợp nhận giá trị tốt nhất (Waveform, Wdbc). Do vậy, chất lượng phân cụm theo phương pháp SSFCC tốt hơn phương pháp MKC và phương pháp CMSC.

V. KẾT LUẬN

Bài báo này đã đề xuất một mô hình mới có tên gọi SSFCC, đó là một phương pháp phân cụm bán giám sát mờ đa khung nhìn đồng huấn luyện. Mô hình này áp dụng cho dữ liệu đa khung nhìn thu thập từ một nguồn dữ liệu và mang đặc trưng của dữ liệu trên cùng một nguồn. Kết quả thực nghiệm cho thấy phương pháp SSFCC vượt trội hơn so với các phương pháp MKC và CMSC khi đánh giá độ chính xác và chất lượng của việc phân cụm. Điều này khuyến khích và thúc đẩy nhiều nghiên cứu tiếp theo trong lĩnh vực phân cụm đa khung nhìn.

Mặc dù phương pháp SSFCC hiệu quả trong trường hợp dữ liệu trên hai khung nhìn được thu thập từ một nguồn dữ liệu, nhưng nó vẫn còn một số hạn chế. Mô hình có nhiều tham số, quá trình đồng huấn luyện lặp lại nhiều lần dẫn đến thời gian tính toán cao và chưa hiệu quả trong trường hợp dữ liệu được thu thập từ nhiều nguồn dữ liệu với các đặc điểm: số lượng bản ghi trên mỗi khung nhìn khác nhau, số lượng thuộc tính trên mỗi khung nhìn có thể khác nhau và quan hệ giữa hai khung nhìn là ánh xạ nhiều – nhiều.

Để giải quyết các vấn đề liên quan đến dữ liệu đa khung nhìn, đặc biệt là dữ liệu thu thập từ nhiều nguồn khác nhau, cần tiếp tục phát triển các thuật toán mới trong các nghiên cứu tiếp theo.

TÀI LIỆU THAM KHẢO

[1] Al-Amri, Salem Saleh, and Namdeo V. Kalyankar. "Image segmentation by using threshold techniques." arXiv preprint arXiv:1005.4020 (2010).

[2] Li, Xin, et al. "A multi-view model for visual tracking via correlation filters." Knowledge-Based Systems 113 (2016): 88-99.

[3] Elkahky, Ali Mamdouh, Yang Song, and Xiaodong He. "A multi-view deep learning approach for cross domain user modeling in recommendation systems." Proceedings of the 24th international conference on world wide web. (2015).

[4] Xu, Chang, Dacheng Tao, and Chao Xu. "A survey on multi-view learning." arXiv preprint arXiv:1304.5634 (2013).

[5] Zhu, Zhenfeng, et al. "Shared Subspace Learning for Latent Representation of Multi-View Data." J. Inf. Hiding Multim. Signal Process. 5.3 (2014): 546-554.

[6] Yang, Yan, and Hao Wang. "Multi-view clustering: A survey." Big Data Mining and Analytics 1.2 (2018): 83-107.

[7] Bickel, Steffen, and Tobias Scheffer. "Multi-view clustering." ICDM. Vol. 4. No. 2004. (2004).

[8] Ye, Fanghua, et al. "New approaches in multi-view clustering." Recent applications in data clustering 195 (2018).

[9] Xu, Chang, Dacheng Tao, and Chao Xu. "A survey on multi-view learning." arXiv preprint arXiv:1304.5634 (2013).

[10] Sun, Shiliang. "A survey of multi-view machine learning." Neural computing and applications 23 (2013): 2031-2038.

[11] Chen, Rui, et al. "Deep multi-view semi-supervised clustering with sample pairwise constraints." Neurocomputing 500 (2022): 832-845.

[12] Li, B., Li, X., Zhang, L., Yu, Z., & Wu, X. (2021), "Multiview clustering via robust probabilistic non-negative matrix factorization", IEEE Transactions on Neural Networks and Learning Systems, 32(5), 1975-1986

[13] Blum, Avrim, and Tom Mitchell. "Combining labeled and unlabeled data with co-training." Proceedings of the eleventh annual conference on Computational learning theory. (1998).

[14] Ye, F., Chen, Z., Qian, H., Li, R., Chen, C., & Zheng, Z. New approaches in multi-view clustering. Recent applications in data clustering, 195.(2018).

[15] Kumar, Abhishek, and Hal Daumé. "A co-training approach for multi-view spectral clustering." Proceedings of the 28th international conference on machine learning (ICML-11). (2011).

[16] D. Dua and C. Graff, "UCI Machine Learning Repository," 2019.[Online]. Available. http://archive.ics.uci.edu/ml. [Accessed Jan. 10, 2022].

[17] Wang, Jiada, Yijun Liu, and Wujian Ye. "FMvC: Fast Multi-View Clustering." IEEE Access 11 (2023): 12808-12820.

[18] Bernard, Desgraupes. "Clustering Indices." University Paris Ouest Lab Modal'X November (2017).

Hoàng Thị Cành tốt nghiệp Đại học Thái Nguyên (ICTU). Nhận bằng Kỹ sư CNTT tại ICTU năm 2009. Nhận bằng Thạc sĩ KHMT tại ICTU năm 2012. Hiện đang NCS tại Viện Hàn lâm Khoa học và Công nghệ Việt Nam và là giảng viên Trường Đại học Công nghệ Thông tin và Truyền thông. Các lĩnh vực nghiên cứu bao gồm Thị giác máy tính, Nhận dạng mẫu, Khai phá dữ liệu, Tính toán mềm.
Email: htcanh@ictu.edu.vn

Phùng Thế Huân tốt nghiệp Đại học Thái Nguyên (ICTU). Nhận bằng Kỹ sư CNTT tại ICTU năm 2009. Nhận bằng Thạc sĩ KHMT tại ICTU năm 2012. Nhận bằng Tiến sĩ KHMT tại ICTU năm 2023. Hiện là giảng viên Trường Đại học Công nghệ Thông tin và Truyền thông.
Các lĩnh vực nghiên cứu bao gồm Thị giác máy tính, Nhận dạng mẫu, Khai phá dữ liệu, Tính toán mềm.
Email: pthuan@ictu.edu.vn

Vũ Thuỳ Trang hiện là sinh viên Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội.
Các lĩnh vực nghiên cứu bao gồm Thị giác máy tính, Nhận dạng mẫu, Khai phá dữ liệu, Tính toán mềm, Logic mờ.
Email: vuthuytrangt_65@hus.edu.vn

Nguyễn Như Sơn nhận bằng Tiến sĩ Khoa học máy tính tại Đại học Queensland - Australia năm 2007. Hiện là giảng viên, nghiên cứu viên tại Viện Công nghệ Thông tin - Viện Hàn lâm Khoa học và Công nghệ Việt Nam.
Các lĩnh vực nghiên cứu bao gồm Thị giác máy tính, Học máy, Trí tuệ nhân tạo, Khai phá dữ liệu, Tính toán mềm, Logic mờ.
Email: nnson@ioit.ac.vn

Lê Trường Giang nhận bằng Kỹ sư Khoa học máy tính năm 2011 tại Trường Đại học Công nghiệp Hà Nội và Thạc sĩ Khoa học máy tính tại Trường Đại học Công nghệ Thông tin và Truyền thông, Đại học Thái Nguyên (ICTU) năm 2014. Nhận bằng Tiến sĩ Hệ thống thông tin tại Học viện Khoa học Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam năm 2023. Hiện là cán bộ tại Trung tâm Đảm bảo Chất lượng, Trường Đại học Công nghiệp Hà Nội.
Các lĩnh vực nghiên cứu bao gồm Tối ưu hóa, Học máy, Khai phá dữ liệu, Trí tuệ nhân tạo, Tính toán mềm, Logic mờ.
Email: letruonggiang@haui.edu.vn

Phạm Huy Thông nhận bằng Tiến sĩ Toán Tin tại Đại học Quốc Gia Hà Nội. Hiện là giảng viên, nghiên cứu viên tại Viện Công nghệ Thông tin – Đại học Quốc Gia Hà Nội. Năm 2020.
Các lĩnh vực nghiên cứu bao gồm Tối ưu hóa, Khai phá dữ liệu, Tính toán mềm, Logic mờ.
Email: thongph@vnu.edu.vn

Về một thuật toán gia tăng tìm tập rút gọn trên bảng quyết định khi loại bỏ tập đối tượng

Phạm Việt Anh[1,4], Nguyễn Long Giang[2], Nguyễn Ngọc Thủy[3], Nguyễn Thế Thủy[1,5], Phạm Đình Khánh[6]
[1] Học Viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ, Việt Nam
[2] Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ, Việt Nam
[3] Trường Đại học Khoa học, Đại học Huế, Việt Nam
[4] Viện Công nghệ HaUI, Đại học Công nghiệp Hà Nội, Việt Nam
[5] Trung tâm CNTT và truyền thông Bắc Ninh, Việt Nam
[6] Công ty Cổ phần Công nghệ Neurond, Đà Nẵng, Việt Nam

Tác giả liên hệ: Phạm Việt Anh, anhpv@haui.edu.vn
Ngày nhận bài: 10/07/2022, ngày sửa chữa: 05/09/2023, ngày duyệt đăng: 26/09/2023
Định danh DOI: 10.32913/mic-ict-research-vn.v2023.n2.1229

Tóm tắt: Rút gọn hay lựa chọn thuộc tính (RGTT) trên các hệ thông tin quyết định từ lâu đã được coi là bài toán then chốt và không thể thiếu của các lĩnh vực về khai phá cũng như phân tích dữ liệu. Một số tiếp cận theo hướng lý thuyết tập thô cùng các mở rộng đã mang tới nhiều phương pháp RGTT đạt hiệu quả ấn tượng. Tuy nhiên, cho tới nay, một số phương pháp theo lý thuyết tập mờ trực cảm chưa thực sự được biết tới. Các phương pháp theo tiếp cận này có một điểm mạnh là khả năng nâng cao hiệu năng phân lớp trên các bảng quyết định (DT) có tính nhiễu và không nhất quán. Bài báo này xuất phát từ một độ đo khoảng cách giữa hai phân hoạch mờ trực cảm và từ đó đề xuất một thuật toán RGTT hiệu quả. Cụ thể như sau, đầu tiên, chúng tôi thiết kế một thuật toán RGTT trên DT chưa có sự biến động. Tiếp theo, chúng tôi thiết kế một thuật toán gia tăng để xử lý trên DT trong trường hợp xóa bỏ tập đối tượng. Một số kết quả trong quá trình thử nghiệm đã chứng mình rằng, các phương pháp RGTT được đề xuất của chúng tôi có hiệu năng vượt trội khi so sánh với các phương pháp dựa trên cách tiếp cận tập thô, tập mờ về kích thước tập rút gọn (TRG) và hiệu quả phân lớp.

Từ khóa: *Rút gọn thuộc tính, tập thô, tập mờ trực cảm, tập mờ, quan hệ tương đương mờ trực cảm*

Title: **A Novel Incremental Algorithm for Finding the Reduct on the Decision Table when Deleting the Object Set**

Abstract: Feature selection or attribute reduction for decision information systems has long been considered a key and indispensable problem in data mining and analysis. Some approaches based on rough set theory and extensions have brought many attribute reduction methods with impressive efficiency. However, up to now, some attribute reduction methods according to intuitionistic fuzzy sets have not received much interest. The advance of this approach is the ability to improve classification performance on noisy and inconsistent decision tables. This paper starts from a distance measure between two intuitionistic fuzzy partitions and then proposes an effective attribute reduction algorithm. Specifically, we first design an attribute reduction algorithm on the decision table without change. Next, we construct an incremental algorithm to process the decision table when deleting an object set. Some experimental results have shown that our proposed methods have superior performance to methods based on the rough set and fuzzy set in terms of the size of the reduct and the classification efficiency.

Keywords: *Attribute reduction, rough set, intuitionistic fuzzy set, fuzzy set, intuitionistic fuzzy equivalence relation*

I. GIỚI THIỆU

Một vấn đề khá cần thiết ở giai đoạn tiền xử lý của dữ liệu đó là việc rút gọn hay lựa chọn thuộc tính (RGTT). Mục tiêu chính của RGTT nhằm giữ lại các thuộc tính thiết yếu từ DT và loại đi các thuộc tính chứa thông tin dư thừa nhằm cải thiện hiệu năng phân lớp cho các mô hình dự đoán. Khái niệm tập thô mờ (FRS) được đề xuất bởi Duboi đã mang tới sự thành công cho các phương pháp RGTT khi có thể xử lý trên các DT chứa các thuộc tính mang miền giá trị số và liên tục. Cũng theo lý thuyết này, có nhiều công trình đã được công bố dựa trên không gian xấp xỉ.

Một số thuật toán điển hình bao gồm hàm thuộc mờ [1], miền khẳng định mờ [2]-[4], entropy mờ [5]-[8] và khoảng cách mờ [9]. Trước xu thế của dữ liệu lớn ngày nay, các DT có số chiều vô cùng lớn và liên tục được cập nhật. Điều này đã mang tới những khó khăn và thách thức cho các phương pháp RGTT theo cách tiếp cận truyền thống. Một trong số đó có thể nói tới những khó khăn về tốc độ tính toán và không gian lưu trữ. Ngoài ra, khi DT được cập nhật, các phương pháp trên phải tìm kiếm lại TRG trên toàn bộ DT. Điều này càng làm tăng thời gian xử lý của thuật toán. Để giải quyết vấn đề này, các kỹ thuật về tính toán gia tăng để tìm TRG trên DT biến động đã được phát triển bởi nhiều nhà nghiên cứu. Các kỹ thuật này đã làm cho các thuật toán RGTT mang ưu điểm nổi trội về thời gian tính toán do chỉ cập nhật các tập con thuộc tính trên phần thay đổi mà không cần thực thi lại trên toàn bộ DT gốc. Hơn nữa, với các bảng có số chiều lớn, DT sẽ được chia thành nhiều phần và áp dụng thuật toán gia tăng (TTGT). TRG sau đó sẽ được cập nhật khi DT được lược bỏ hay bổ sung từng phần. Theo cách tiếp cận này, nhiều TTGT đã được đề xuất trong các trường hợp loại bỏ hoặc bổ sung tập đối tượng [10]-[14] và loại bỏ hoặc bổ sung tập thuộc tính [15]. Thêm vào đó, một số nghiên cứu đã mở rộng các TTGT khi thực thi trên các DT biến động và thiếu dữ liệu. Cụ thể, Giang trong [16] đã xây dựng tập thô mờ dung sai để thiết kế thuật toán lai ghép cho DT không đầy đủ dữ liệu khi lược bỏ và bổ sung tập đối tượng. Cũng theo cách tiếp cận này, Thắng trong [17] đã xây dựng một số công thức trong TTGT cho hai trường hợp loại bỏ và bổ sung tập thuộc tính. Tuy nhiên, Hung và Yang [18] đã chỉ ra rằng, các phương pháp theo hướng tiếp cận FRS có hiệu quả chưa tốt khi được xử lý trên các DT có hiệu quả phân lớp ban đầu thấp và không nhất quán.

Trong thời gian trở lại gần đây, một mô hình mới sử dụng tập mờ trực cảm (IFS) nhằm giải quyết bài toán RGTT đã được đề xuất. Mô hình này hạn chế được các thông tin không chắc chắn trên DT do sự bổ sung của hàm không thành viên (hàm không thuộc) có khả năng điều chỉnh rất tốt các đối tượng có tính nhiễu về gần đúng phân lớp [19]. Đối với các DT chứa thông tin không chắc chắn và có hiệu năng phân lớp ban đầu không cao, hiệu quả của các phương pháp RGTT dựa trên IFS được báo cáo là kỳ vọng hơn so với các phương pháp dựa trên FRS. Từ cách tiếp cận này, Thắng trong [20] đã xây dựng khoảng cách trên hai phân hoạch mờ trực cảm và thiết kế một phương pháp lai ghép để tìm kiếm tập con thuộc tính trên DT. Các kết quả trong thực nghiệm đã minh họa rõ rệt khả năng cải thiện hiệu quả phân lớp của TRG thu được từ thuật toán là tốt hơn so với các thuật toán theo cách tiếp cận FRS trên các bộ dữ liệu chứa thông tin nhiễu, đặc biệt có hiệu quả phân lớp ban đầu thấp. Do đó, cách tiếp cận IFS mở ra một hướng mới và tiềm năng khi dữ liệu ngày nay càng đa dạng và không chắc chắn.

Đối với nội dung của nghiên cứu này, chúng tôi xây dựng một TTGT trên DT khi loại bỏ tập đối tượng. Nghiên cứu này đóng góp hai phần chính như sau:

(1) Mở rộng công thức gia tăng để tính toán khoảng cách phân hoạch mờ trực cảm trên DT khi loại bỏ một tập đối tượng.

(2) Đề xuất một TTGT để tính toán TRG trong trường hợp loại bỏ tập các đối tượng nhằm giảm thiểu thời gian xử lý.

Phần tiếp theo của bài báo này sẽ khái quát một số lý thuyết cơ sở về IFS và các tính chất liên quan. Trong phần III, dựa trên độ đo khoảng cách của hai phân hoạch, chúng tôi định nghĩa TRG và xây dựng một thuật toán lọc để lựa chọn một TRG trên DT chưa biến động. Sau đó, chúng tôi tiếp tục mở rộng độ đo khoảng cách và thiết kế một TTGT trong trường hợp DT loại bỏ tập các đối tượng. Phần IV và V sẽ minh họa các thực nghiệm, đưa ra các kết luận, đánh giá và hướng nghiên cứu sẽ triển khai trong tương lai.

II. MỘT SỐ KHÁI NIỆM LIÊN QUAN

Đầu tiên, bảng quyết định được biểu diễn là một đôi $DT = (O, C \cup D)$ với O đại diện cho một tập khác rỗng các đối tượng, C và D là tập các thuộc tính sao cho với mỗi $a \in C \cup D$ sẽ xác định một ánh xạ $a : O \to V_a$, trong đó V_a là ký hiệu biểu diễn giá trị của thuộc tính a. Khi đó, với $o \in O$ và $a \in C \cup D$, giá trị của thuộc tính a với đối tượng o được viết là $a(o)$. Chúng tôi sẽ gọi C là tập các thuộc tính điều kiện và D là tập các thuộc tính quyết định.

Cho một bảng quyết định $DT = (O, C \cup D)$. Một IFS P trên O có dạng $P = \{\langle o, \eta_P(o), \gamma_P(o)\rangle \,|o \in O\}$ với $\eta_P(o) : O \to [0,1]$ và $\gamma_P(o) : O \to [0,1]$ tương ứng là độ thuộc (độ thành viên) và độ không thuộc (độ không thành viên) của đối tượng o trong P sao cho $0 \le \eta_P(o) + \gamma_P(o) \le 1, \forall o \in O$ [21]. Độ do dự (độ lưỡng lự) của o trong P được tính bởi $\pi_P(o) = 1 - \eta_P(o) - \gamma_P(o)$. Khi đó, $\pi_P(o) = 0 \forall o \in O$, IFS trở thành một tập mờ truyền thống. Lực lượng của IFS P được viết là $|P|$ và được tính toán theo công thức sau [22]:

$$|P| = \sum_{o \in O} \frac{1 + \eta_P(o) - \gamma_P(o)}{2} \tag{1}$$

Xét hai IFS P và Q trên O, chúng tôi sẽ định nghĩa một vài phép toán như sau:

(1) $P \subseteq Q$ nếu $\eta_P(o) \le \eta_P(o)$ và $\gamma_P(o) \ge \gamma_P(o) \forall o \in O$.

(2) $P = Q$ nếu $P \subseteq Q$ và $Q \subseteq P$.

(3) $P \cap Q = \{(o, min(\eta_P(o), \eta_Q(o)), max(\gamma_P(o), \gamma_Q(o)))\}$

(4) $P \cup Q = \{(o, max(\eta_P(o), \eta_Q(o)), min(\gamma_P(o), \gamma_Q(o)))\}$

Để đơn giản, một IFS P có thể được biểu diễn dưới dạng $P = \{\eta_P(o), \gamma_P(o) \,|o \in O\}$.

Cho O là một tập hữu hạn khác rỗng bao gồm các đối tượng, định nghĩa về một quan hệ nhị phân mờ trực cảm R trên O^2 được trình bày như sau:

$$R = \left\{ (o,q), \eta_R(o,q), \gamma_R(o,q) \,|\, (o,q) \in O^2 \right\} \quad (2)$$

với $\gamma_R(o,q) \in [0,1]$ và $\eta_R(o,q) \in [0,1]$ lần lượt là độ khác biệt và độ tương tự của hai đối tượng o và q, cặp $(\eta_R(o,q), \gamma_R(o,q))$ được gọi là một số mờ trực cảm giữa hai đối tượng o và q thỏa mãn $0 \le \eta_R(o,q) + \gamma_R(o,q) \le 1$. Khi đó, gọi R là một quan hệ tương đương mờ trực cảm (IFER) nếu như R thỏa mãn:

(1) Tính phản xạ: $\eta_R(o,o) = 1$ và $\gamma_R(o,o) = 0 \quad \forall o \in O$.

(2) Tính đối xứng: $\eta_R(o,q) = \eta_R(q,o)$ và $\gamma_R(o,q) = \gamma_R(q,o) \quad \forall o,q \in O$.

(3) Tính bắc cầu min-max:

$\eta_R(o,q) \ge \max\limits_{t \in O} \{min(\eta_R(o,t), \eta_R(t,q))\}$;

$\gamma_R(o,q) \le \min\limits_{t \in O} \{max(\gamma_R(o,t), \gamma_R(t,q))\} \forall o,q \in O$.

Cho một IFER R trên O, một tập con thuộc tính $A \subseteq C$ và một đối tượng $o \in O$. Khi đó, lớp tương đương mờ trực cảm (IFEC) của o theo R trên A được ký hiệu là $R_A[o]$ và được biểu diễn như sau:

$$R_A[o] = \left\{ \left(o, \eta_{R_A[o]}(q), \gamma_{R_A[o]}(q)\right) | q \in O \right\} \quad (3)$$

Cho $DT = (O, C \cup D)$, mỗi tập con các thuộc tính $A \subseteq C$ xác định một IFER, ký hiệu là R_A. Một IFER R_A sẽ tạo ra một phân hoạch mờ trực cảm (IFP) trên O, ký hiệu là $\mho_A$ với $\mho_A = \{R_A[o] \,|\, o \in O\}$, trong đó $R_A[o]$ là một IFEC của đối tượng o theo quan hệ R_A. Chúng ta có thể thấy rằng mỗi IFEC $R_A[o]$ là một IFS trên O. Để đơn giản, với mỗi đối tượng q trong $R_A[o]$, ký hiệu: $R_A[o](q) = (\eta_A[o](q), \gamma_A[o](q))$.

Cho $A, B \subseteq C$, chúng ta có $R_A[o] = \cap_{a \in A} R_a[o]$ và $R_{A \cup B}[o] = R_A[o] \cap R_B[o]$. Điều này nhấn mạnh rằng $\mho_{A \cup B} = \mho_A \cap \mho_B$. Với $A \subseteq C$ và $q \in O$, chúng ta xét hai trường hợp đặc biệt dưới đây:

(1) Nếu $R_A[o](q) = (0,1), \; o \ne q$ và $R_A[o](q) = (1,0)$ thì $|R_A[o]| = 1$, IFP $\mho_A$ được xem là trường hợp mịn nhất và được ký pháp là $\mho_\omega$.

(2) Nếu $R_A[o](q) = (1,0)$ thì $|R_A[o]| = n$, IFP $\mho_A$ được xem là trường hợp thô nhất và có ký pháp là $\mho_\delta$.

III. THUẬT TOÁN RÚT GỌN THUỘC TÍNH

1. Rút gọn thuộc tính dựa trên IFPD

Xét $DT = (O, C \cup D)$ với $O = \{o_1, o_2, \ldots, o_m\}$ và các IFP $\mho_A, \mho_B$ được sinh bởi các IFEC $R_A[o_i]$, $R_B[o_i]$ $(o_i \in O)$ trên các tập con $A, B \subseteq C$, khi đó:

$$\mathcal{D}(\mho_A, \mho_B) = \sum_{i=1}^{m} \frac{(|R_A[o_i] \cup R_B[o_i]| - |R_A[o_i] \cap R_B[o_i]|)}{m^2} \quad (4)$$

được gọi là khoảng cách phân hoạch mờ trực cảm (IFPD) giữa $\mho_A$ và $\mho_B$ [20]. Rõ ràng, giá trị cực tiểu của $\mathcal{D}(\mho_A, \mho_B) = 0$ khi $\mho_A = \mho_B$ và giá trị cực đại của $\mathcal{D}(\mho_A, \mho_B) = (m-1)/m$ nếu như $\mho_A = \mho_\delta$ và $\mho_B = \mho_\omega$ (hoặc khi $\mho_A = \mho_\omega$ và $\mho_B = \mho_\delta$). Do đó, chúng ta luôn có $0 \le \mathcal{D}(\mho_A, \mho_B) \le (m-1)/m$. Dựa trên (4), IFPD được tạo bởi C và $C \cup D$ trên O được tính theo công thức:

$$\mathcal{D}(\mho_C, \mho_{C \cup D}) = \sum_{i=1}^{m} \frac{(|R_C[o_i]| - |R_C[o_i] \cap R_D[o_i]|)}{m^2} \quad (5)$$

Nếu $B \subseteq C$ thì $\mathcal{D}(\mho_B, \mho_{B \cup D}) \ge \mathcal{D}(\mho_C, \mho_{C \cup D})$. Rõ ràng, công thức (5) thỏa mãn tính chất phản đơn điệu với kích thước của tập thuộc tính điều kiện. Khi đó, giá trị $\mathcal{D}(\mho_B, \mho_{B \cup D})$ càng nhỏ thì kích thước của B càng lớn và ngược lại.

Định nghĩa 1. Cho $DT = (O, C \cup D)$, một tập con thuộc tính $B \subseteq C$ được gọi là một tập rút gọn của C nếu:

(1) $\mathcal{D}(\mho_B, \mho_{B \cup D}) = \mathcal{D}(\mho_C, \mho_{C \cup D})$.

(2) $\forall B' \subset B, \mathcal{D}(\mho_{B'}, \mho_{B' \cup D}) > \mathcal{D}(\mho_B, \mho_{B \cup D})$.

Định nghĩa 1 chỉ ra rằng, với mọi thuộc tính e thuộc B, nếu $\mathcal{D}(\mho_{B \setminus \{e\}}, \mho_{B \setminus \{e\} \cup D}) \ne \mathcal{D}(\mho_C, \mho_{C \cup D})$ thì e sẽ được coi là một thuộc tính quan trọng trong B. Với trường hợp ngược lại, thì e sẽ được xem là một thuộc tính dư thừa trong B.

Định nghĩa 2. Cho $DT = (O, C \cup D)$, một tập con thuộc tính $B \subseteq C$ và một thuộc tính bất kỳ $e \in C/B$. Khi đó, độ quan trọng (độ ý nghĩa) của thuộc tính e với B ký hiệu là $SIG_B(e)$ được xác định theo công thức:

$$SIG_B(e) = \mathcal{D}(\mho_{B \cup \{e\}}, \mho_{B \cup \{e\} \cup D}) - \mathcal{D}(\mho_B, \mho_{B \cup D}) \quad (6)$$

Dựa trên định nghĩa này, chúng tôi thiết kế một phương pháp RGTT để lựa chọn một tập con thuộc tính tối ưu từ DT khi chưa có sự biến động. Phương pháp này có tên gọi là ARIFPD và được trình bày như sau.

Algorithm 1: Attribute Reduction based on IFPD

Input: $DT = (O, C \cup D)$
Output: Approximation reduct B
1 *Initialize:* $B := \varnothing, \mathcal{D}(\mho_B, \mho_{B \cup D}) = (m-1)/m$
2 *Calculate:* $\mathcal{D}(\mho_C, \mho_{C \cup D})$ using (5).
3 **While** $\mathcal{D}(\mho_B, \mho_{B \cup D}) > \mathcal{D}(\mho_C, \mho_{C \cup D})$ **do**
4 **For** b in $C \setminus B$ **do**
5 Compute $\mathcal{D}(\mho_{B \cup \{b\}}, \mho_{B \cup \{b\} \cup D})$
6 Select b_0 with: $SIG_B(b_0) = \max\limits_{b \in C \setminus B} \{SIG_B(b)\}$.
7 $B := B \cup \{b_0\}$
8 **End while**
9 Return B

Giả sử rằng $|O|, |C|$ được ký hiệu lần lượt cho tổng số đối tượng và tổng số thuộc tính điều kiện trên DT. Một IFP $\mho_C$ được xác định với độ phức tạp thời gian là $\mathcal{O}\left(|C| * |O|^2\right)$. Trong trường hợp này, độ phức tạp thuật toán ở dòng số 2 khi tính $\mathcal{D}(\mho_C, \mho_{C \cup D})$ là $\mathcal{O}\left(|C| * |O|^2\right)$. Ngoài ra, độ phức tạp tính toán khi tính $\mathcal{D}(\mho_{B \cup \{b\}}, \mho_{B \cup \{b\} \cup D})$ và $SIG_B(b)$ trong dòng 6 là

$\mathcal{O}\left(|O|^2\right)$. Do đó độ phức tạp trong vòng For và While lần lượt là $\mathcal{O}\left(|C|*|O|^2\right)$ và $\mathcal{O}\left(|C|^2*|O|^2\right)$. Như vậy độ phức tạp của ARIFPD là $\mathcal{O}\left(|C|^2*|O|^2\right)$.

2. Xây dựng thuật toán gia tăng

Tính chất 1. Cho bảng quyết định $DT = (O, C \cup D)$, $O = \{o_1, o_2, \ldots, o_m\}$ và một IFER R được định nghĩa trên giá trị của tập thuộc tính và một tập đối tượng bị lược bỏ $\Delta O = \{o_m, o_{m-1}, \ldots, o_{m-s+1}\}$, $s < m$, IFPD được xác định như sau:

$$\mathcal{D}_{O\backslash\Delta O}\left(\mho_C, \mho_{C\cup D}\right) = \frac{m^2}{(m-s)^2}\mathcal{D}_O\left(\mho_C, \mho_{C\cup D}\right) - \frac{2}{(m-s)^2} * \sum_{i=0}^{s-1}\left(\left|R_C\left[o_{m-i}\right]\right| - \left|R_C\left[o_{m-i}\right] \cap R_D\left[o_{m-i}\right]\right| - b_j\right) \quad (7)$$

trong đó: $b_j = \frac{1}{2}\sum_{j=0}^{s}\left(\eta_C\left[o_{m+i}\right]\left(o_{m+j}\right) - \min\left\{\eta_C\left[o_{m+i}\right]\left(o_{m+j}\right), \eta_D\left[o_{m+i}\right]\left(o_{m+j}\right)\right\} - \gamma_C\left[o_{m+i}\right]\left(o_{m+j}\right) + \max\left\{\gamma_C\left[o_{m+i}\right]\left(o_{m+j}\right), \gamma_D\left[u_{m+i}\right]\left(u_{m+j}\right)\right\}\right)$

Tính chất 2. Cho một bảng quyết định $DT = (O, C \cup D)$ với $O = \{o_1, o_2, \ldots, o_m\}$. Giả sử rằng $B \subseteq C$ là một TRG dựa trên IFPD của tập đối tượng ban đầu O và $\Delta O = \{o_m, o_{m-1}, \ldots, o_{m-s+1}\}$, $s < m$ là tập đối tượng bị lược bỏ. Khi đó:

(1) Nếu toàn bộ các đối tượng thuộc ΔO có thuộc tính quyết định mang giá trị giống nhau thì $\mathcal{D}_{O\backslash\Delta O}\left(\mho_C, \mho_{C\cup D}\right) = \frac{m^2}{(m-s)^2}\mathcal{D}_O\left(\mho_C, \mho_{C\cup D}\right) - \frac{2}{(m-s)^2} * \sum_{i=0}^{s-1}\left(\left|R_C\left[o_{m-i}\right]\right| - \left|R_C\left[o_{m-i}\right] \cap R_D\left[o_{m-i}\right]\right|\right)$.

(2) Nếu $R_B\left[o_{m-i}\right] \subseteq R_D\left[o_{m-i}\right]$ với $i = 0, 1, ..., s-1$ thì $\mathcal{D}_{O\backslash\Delta O}\left(\mho_B, \mho_{B\cup D}\right) = \mathcal{D}_{O\backslash\Delta O}\left(\mho_C, \mho_{C\cup D}\right)$.

Từ kết quả Tính chất 1 và 2, chúng tôi thiết kế một TTGT để tìm TRG trong trường hợp DT xóa bỏ tập đối tượng. Chúng tôi gọi thuật toán là ARIFPD-DO. Bốn bước chính của thuật toán ARIFPD-DO bao gồm việc khởi tạo các IFP trên các thuộc tính, kiểm tra tập đối tượng bị xóa bỏ, tìm kiếm TRG theo phương pháp lọc và xóa bỏ các thuộc tính dư thừa trong TRG tìm được.

Chúng tôi ký hiệu $|\Delta O|$ là tổng số đối tượng bị loại bỏ. Độ phức tạp khi tính các IFPs trong dòng 2 là $\mathcal{O}\left(\left|B_{O\backslash\Delta O}\right| * \Delta O\right)$. Với trường hợp đơn giản, TTGT sẽ dừng lại ở dòng 6 và có thời gian tính toán là $\mathcal{O}\left(\left|B_{O\backslash\Delta O}\right| * \Delta O\right)$. Đối với các trường hợp còn lại, độ phức tạp tính toán của dòng 9 là $\mathcal{O}\left(|C| * |\Delta O| * (|O| - |\Delta O|)\right)$ và trên tính toán gia tăng $\mathcal{D}_{O\backslash\Delta O}\left(\mho_{B_{O\backslash\Delta O}\cup\{b\}}, \mho_{B_{O\backslash\Delta O}\cup\{b\}\cup D}\right)$ có độ phức tạp theo thời gian là $\mathcal{O}\left(|\Delta O| * (|O| - |\Delta O|)\right)$. Thời gian tính toán trong vòng lặp While từ dòng 10 tới 17 là $\mathcal{O}\left(\left(|C| - \left|B_{O\backslash\Delta O}\right|\right)^2 * |\Delta O| * (|O| - |\Delta O|)\right)$. Từ dòng 18 tới dòng 21, độ phức tạp trong vòng lặp for là $\mathcal{O}\left(|C|^2 * |\Delta O| * (|O| + |\Delta O|)\right)$. Như vậy, độ phức tạp của toàn bộ thuật toán ARIFPD-DO là giá trị lớn nhất của $\mathcal{O}\left(\left|B_{O\backslash\Delta O}\right| * |\Delta O| * (|O| - |\Delta O|)\right)$ hoặc $\mathcal{O}\left(|C|^2 * |\Delta O| * (|O| - |\Delta O|)\right)$. Rõ ràng, có thể thấy thời gian xử lý của ARIFPD là lớn hơn khá nhiều so với thuật toán ARIFPD-DO.

Algorithm 2: Incremental Algorithm ARIFPD-DO

Input: $DT = (O, C \cup D)$, $O = \{o_1, o_2, \ldots, o_m\}$, R, $B \subseteq C$, $\mho_B, \mho_C$ and the deleted set of objects $\Delta O = \{o_m, o_{m-1}, \ldots, o_{m-s+1}\}$.
Output: Reduct $B_{O\backslash\Delta O}$ of $DT' = (O\backslash\Delta O, C \cup D)$.

/* **Initialization** */
1 $B_{O\backslash\Delta O} := B$
2 Compute IFPs on $O\backslash\Delta O$: $\mho_B, \mho_D$
/* **Check the removed set of objects** */
3 Set $S := \Delta O$
4 **For** $i = 0$ to $s-1$ **do**:
5 **If** $R_B\left[o_{m-i}\right] \subseteq R_D\left[o_{m-i}\right]$ **then** $S := S\backslash\{o_{m-i}\}$
6 **If** $S = \varnothing$ **then** return $B_{O\backslash\Delta O}$
7 Set $\Delta O := S, s := |\Delta O|$
/* **Find the reduct by method Filter** */
8 Compute original IFPDs: $\mathcal{D}\left(\mho_{B_{O\backslash\Delta O}}, \mho_{B_{O\backslash\Delta O}\cup D}\right)$ and $\mathcal{D}\left(\mho_C, \mho_{C\cup D}\right)$
9 Compute IFPDs $\mathcal{D}_{O\backslash\Delta O}\left(\mho_{B_{O\backslash\Delta O}}, \mho_{B_{O\backslash\Delta O}\cup D}\right)$ and $\mathcal{D}_{O\backslash\Delta O}\left(\mho_C, \mho_{C\cup D}\right)$
10 **While** $\mathcal{D}_{O\backslash\Delta O}\left(\mho_{B_{O\backslash\Delta O}}, \mho_{B_{O\backslash\Delta O}\cup D}\right) > \mathcal{D}_{O\backslash\Delta O}\left(\mho_C, \mho_{C\cup D}\right)$ **do**
11 **For** each $b \in C\backslash B_{O\backslash\Delta O}$ **do**:
12 Calculate $\mathcal{D}_{O\backslash\Delta O}\left(\mho_{B_{O\backslash\Delta O}\cup\{b\}}, \mho_{B_{O\backslash\Delta O}\cup\{b\}\cup D}\right)$ by (7)
13 Calculate $SIG_{B_{O\backslash\Delta O}}(b) = \mathcal{D}_{O\backslash\Delta O}\left(\mho_{B_{O\backslash\Delta O}}, \mho_{B_{O\backslash\Delta O}\cup D}\right) - \mathcal{D}_{O\cup\Delta O}\left(\mho_{B_{O\backslash\Delta O}\cup\{b\}}, \mho_{B_{O\backslash\Delta O}\cup\{b\}\cup D}\right)$
14 **End for**
15 Select b_0 which satisfies: $SIG_{B_{O\backslash\Delta O}}(b_0) = \max_{b\in C\backslash B_{O\backslash\Delta O}}\left\{SIG_{B_{O\backslash\Delta O}}(b)\right\}$
16 $B_{O\backslash\Delta O} := B_{O\backslash\Delta O} \cup \{b_0\}$
17 **End while**
/* **Remove redundant attributes** */
18 **For** each $b \in B_{O\backslash\Delta O}$ **do**:
19 Calculate $\mathcal{D}_{O\backslash\Delta O}\left(\mho_{B\backslash\{b\}}, \mho_{B_{O\cup\Delta O}\backslash\{b\}\cup D}\right)$ by (7).
20 **If** $\mathcal{D}_{O\backslash\Delta O}\left(\mho_{B_{O\backslash\Delta O}\backslash\{b\}}, \mho_{B_{O\cup\Delta O}\backslash\{b\}\cup D}\right) = \mathcal{D}_{O\backslash\Delta O}\left(\mho_{B_{O\backslash\Delta O}}, \mho_{B_{O\backslash\Delta O}\cup D}\right)$ **then** $B_{O\backslash\Delta O} := B_{O\backslash\Delta O}\backslash\{b\}$
21 **End**
22 Return $B_{O\backslash\Delta O}$

IV. THỰC NGHIỆM

Để minh chứng cho hiệu năng của phương pháp đề xuất, trong phần này, chúng tôi sẽ minh họa một số thực nghiệm cụ thể. Bài báo sẽ đánh giá hiệu quả của ba phương pháp RGTT là ARIFPD-DO, IFSD [23] và FDAR-DO [10] thông qua ba tiêu chuẩn: Hiệu quả của mô hình phân lớp KNN

(K=10) theo kiểm định chéo 10-folds, kích thước TRG và thời gian tính toán của từng phương pháp. Trong đó, các thuật toán IFSD và FDAR-DO là các TTGT tương ứng theo hướng tiếp cận của tập thô và tập mờ. Mục đích so sánh thuật toán đề xuất với hai thuật toán này để làm rõ hiệu quả của cách tiếp cận IFS trong vấn đề tìm TRG trên các bảng dữ liệu nhiễu hay mang các thông tin không chắc chắn. Các dữ liệu mẫu sử dụng trong thực nghiệm được chúng tôi lấy tại kho lưu trữ dữ liệu UCI [24].

1. Thiết lập quá trình thực nghiệm

Dữ liệu trong thực nghiệm được phân chia theo hai phần có kích thước bằng nhau. Phần thứ nhất ký hiệu là O_{ori} và phần thứ hai là O_{inc}. Tập O_{inc} được phân chia tiếp thành 6 phần bằng nhau (biểu diễn từ O_1 đến O_6) được sử dụng để làm các tập đối tượng bị loại bỏ. Trong Bảng I, các cột $|O_{ori}|$, $|O_{inc}|$, $|C|$, $|D|$ được ký hiệu lần lượt cho tổng số đối tượng hay bản ghi trong O_{ori}, O_{inc}, số lượng thuộc tính điều kiện và số lượng lớp quyết định.

Bảng I
CÁC BỘ DỮ LIỆU THỰC NGHIỆM

ID	Data sets	$\|O\|$	$\|O_{ori}\|$	$\|O_{inc}\|$	$\|C\|$	$\|D\|$
1	Robot	164	82	82	90	5
2	Iono	351	175	176	34	2
3	Pc4	1458	729	729	37	2
4	Ozone	2534	1267	1267	72	2
5	Mfeat	2000	1000	1000	76	10
6	Wall	5456	2728	2728	24	4

Đầu tiên, bài báo sử dụng các thuật toán ARIFPD, GFS trong [23] và FDAR trong [10] để tìm TRG trên toàn bộ O. Tiếp theo dựa trên TRG thu được từ ba thuật toán, chúng tôi đánh giá ba TTGT là ARIFPD-DO, IFSD và FDAR-DO sau khi bảng quyết định loại bỏ lần lượt các tập đối tượng từ O_1 đến O_6 của O_{inc}. Để phục vụ cho bước tính toán các IFP, chúng tôi xây dựng IFER bao gồm độ khác biệt và độ tương tự của đối tượng o và q theo thuộc tính a.
(1) Nếu miền giá trị của a là liên tục khi đó mức tương tự và khác biệt của p và q được tính lần lượt như sau:

$$\eta_{\{a\}}[o](q) = 1 - \frac{a(o) - a(q)}{\max(a) - \min(a)} \tag{8}$$

$$\gamma_{\{a\}}[o](q) = \frac{1 - \eta_{\{a\}}[o](q)}{1 + \partial_a * \eta_{\{a\}}[o](q)} \; with \; \partial_a > 0 \tag{9}$$

Trong công thức (9), giá trị ∂_a được xác định:

$$\partial_a = \begin{cases} 1 \;\; if \;\; \sigma_a = 0 \\ \beta_a/\sigma_a \;\; if \;\; \sigma_a > 0 \end{cases}, \tag{10}$$

Trong đó, $\sigma_a = \sqrt{\frac{1}{|O|-1}\sum_{o\in O}(a(o) - \bar{a})^2}$ là giá trị độ lệch chuẩn của miền giá trị từ thuộc tính a, $\beta_a = \left|\mho^F_{\{a\}\cup D}\right| / \left|\mho^F_D\right|$ là độ nhất quán của a trong DT, $\mho^F_{\{a\}\cup D}$ là phân hoạch mờ của $\{a\} \cup D$.
(2) Nếu thuộc tính a là kiểu phân loại, khi đó:

$$\eta_{\{a\}}[o](q) = \begin{cases} 1, & a(o) = a(q) \\ 0, & a(o) \neq a(q) \end{cases} \tag{11}$$

2. Kết quả thực nghiệm

Như đã đề cập, chúng tôi ban đầu so sánh hiệu quả của ba phương pháp FDAR, GFS và ARIFPD. Kích thước TRG cũng như hiệu quả phân lớp của ba phương pháp được trình bày trong Bảng II. Có thể thấy, kích thước TRG thu được từ ARIFPD là nhỏ nhất, trong khi kích thước TRG của FDAR và GFS vẫn còn rất lớn trên một số tập dữ liệu. Tiếp theo, chúng tôi phân tích về thời gian thực thi của ba phương pháp. Thời gian thực thi của các phương pháp được tính sau bước tiền xử lý dữ liệu đến khi TRG được xác định. Từ kết quả thu được trong Bảng II, thời gian thực thi của GFS là nhanh nhất trên tất cả các bộ dữ liệu. Điều này giải thích rằng, các thuật toán dựa trên FRS và IFR phải tính toán trên các ma trận quan hệ với số lượng phần tử lớn. Hơn nữa, các ma trận quan hệ theo FDAR chỉ sử dụng độ tương tự, trong khi đó, các ma trận quan hệ của ARIFPD được đặc trưng bởi độ khác biệt và độ tương tự. Vì vậy, tốc độ xử lý của ARIFPD là chậm nhất.

Bảng II cho thấy, TTGT đề xuất xác định một tập con thuộc tính rất hiệu quả trên nhiều tập dữ liệu khác nhau. Cụ thể, khi so sánh với dữ liệu gốc, độ chính xác trong mô hình phân lớp KNN của thuật toán đề xuất thể hiện khả năng vượt trội trong 5 trường hợp. Chỉ có duy nhất một trường hợp mà TTGT đề xuất có độ chính xác thấp hơn dữ liệu gốc. Hơn nữa, độ chính xác trung bình của mô hình phân lớp theo ARIFPD là 81.6% cao hơn so với độ chính xác phân lớp trung bình của dữ liệu gốc là 78.9% và cao nhất trong ba thuật toán mặc dù số lượng thuộc tính hay kích thước TRG thu được là nhỏ nhất.

Tiếp theo, chúng tôi tiếp tục so sánh hiệu quả xử lý trên các TTGT. Rõ ràng, có thể thấy, thời gian của FDAR, GFS và ARIFPD là lớn hơn rất nhiều so với FDAR-DO, IFSD và ARIFPD-DO. Điều này nhấn mạnh rằng, các TTGT chỉ tính toán chủ yếu trên các phần thay đổi của dữ liệu nên giảm thiểu rất nhiều về thời gian xử lý. Bảng III cho thấy, trong đa số các trường hợp, thời gian thực thi của FDAR-DO và IFSD nhanh hơn so với ARIFPD-DO. Điều này được giải thích tương tự khi so sánh thời gian chạy của các thuật toán FDAR, GFS và ARIFPD. Tuy nhiên, trên một số tập dữ liệu như Pc4, Ozone và Wall, thuật toán đề xuất lại xử lý nhanh hơn do kích thước của TRG thu được nhỏ hơn. Trong mỗi giai đoạn gia tăng, phần lớn kích thước TRG của ARIFPD-DO cũng nhỏ hơn đáng kể so với FDAR-DO và IFSD, đặc biệt trên các tập dữ liệu có số thuộc tính lớn như Robot, Ozone và Mfeat.

Bảng II
KẾT QUẢ XỬ LÝ CỦA FDAR, GFS VÀ ARIFPD

ID	RAW	FDAR			GFS			ARIFPD		
	Acc	$\|B\|$	Acc	Time	$\|B\|$	Acc	Time	$\|B\|$	Acc	Time
1	0.505	25	0.578	2.450	18	0.475	5.346	**9**	**0.597**	2.147
2	0.843	19	0.852	3.670	14	0.835	2.375	**7**	**0.877**	3.008
3	0.885	7	0.871	38.07	11	0.874	5.368	**2**	**0.878**	45.33
4	0.933	8	0.935	289.9	12	0.936	132.8	**2**	**0.937**	296.1
5	0.809	56	**0.826**	173.3	**17**	0.767	119.6	28	0.819	399.1
6	0.759	18	**0.817**	648.1	23	0.758	193.1	**8**	0.789	647.1
AVG	0.759	22.2	0.813	192.6	15.8	0.774	76.43	**9.33**	**0.816**	236.6

Bảng III
KẾT QUẢ XỬ LÝ CỦA FDAR-DO, GFS VÀ ARIFPD-DO

ID	Adding data sets	RAW	FDAR-DO			IFSD			ARIFPD-DO		
		Acc	$\|B\|$	Acc	Time	$\|B\|$	Acc	Time	$\|B\|$	Acc	Time
1	$\|O_1\| = 151$	0.509	24	**0.582**	0.899	18	0.477	0.005	**9**	**0.582**	0.001
	$\|O_2\| = 138$	0.471	24	0.559	0.001	17	0.493	0.006	**5**	**0.609**	0.017
	$\|O_3\| = 125$	0.511	24	0.559	0.001	16	0.535	0.004	**2**	**0.626**	0.546
	$\|O_4\| = 112$	0.553	24	0.590	0.001	15	0.582	0.004	**2**	**0.643**	0.561
	$\|O_5\| = 99$	0.607	24	0.599	0.001	15	0.668	0.003	**2**	**0.658**	0.001
	$\|O_6\| = 86$	0.581	24	0.614	0.001	13	0.672	0.003	**2**	**0.686**	0.001
2	$\|O_1\| = 322$	0.835	18	0.820	1.802	14	0.820	0.013	**7**	**0.870**	0.001
	$\|O_2\| = 293$	0.819	17	0.816	1.161	13	0.816	0.191	**7**	**0.867**	0.001
	$\|O_3\| = 264$	0.815	16	0.784	0.962	12	0.815	0.009	**4**	**0.872**	0.014
	$\|O_4\| = 235$	0.783	15	0.749	0.781	11	0.796	0.008	**3**	**0.890**	0.084
	$\|O_5\| = 206$	0.766	15	0.732	0.724	10	**0.791**	0.007	**3**	0.787	0.135
	$\|O_6\| = 177$	0.767	15	0.739	0.703	10	**0.779**	0.006	**3**	0.750	0.108
3	$\|O_1\| = 1377$	0.884	6	**0.879**	1.910	11	0.874	0.106	**2**	**0.879**	0.031
	$\|O_2\| = 1216$	0.888	6	**0.884**	0.005	11	0.878	0.081	**2**	0.883	0.026
	$\|O_3\| = 1095$	0.879	6	0.874	0.004	10	0.874	0.068	**2**	**0.881**	0.025
	$\|O_4\| = 974$	0.885	6	0.880	0.002	10	0.868	0.080	**2**	**0.883**	0.018
	$\|O_5\| = 853$	0.882	6	0.882	0.002	10	0.870	0.053	**2**	**0.883**	0.012
	$\|O_6\| = 732$	0.884	6	0.881	0.002	10	**0.889**	0.044	**2**	0.888	0.009
4	$\|O_1\| = 2323$	0.931	8	0.934	0.222	12	**0.935**	0.354	**2**	**0.935**	0.121
	$\|O_2\| = 2112$	0.928	7	0.929	0.019	11	**0.933**	0.345	**2**	0.929	0.087
	$\|O_3\| = 1901$	0.927	6	0.930	0.017	11	0.930	0.235	**2**	**0.931**	0.071
	$\|O_4\| = 1690$	0.923	5	**0.925**	0.014	11	0.924	0.254	**2**	**0.925**	0.058
	$\|O_5\| = 1479$	0.918	4	0.914	0.011	11	0.916	0.183	**2**	**0.918**	0.049
	$\|O_6\| = 1268$	0.909	3	0.908	0.009	11	**0.912**	0.143	**2**	**0.912**	0.035
5	$\|O_1\| = 1834$	0.871	56	0.892	0.001	**17**	0.839	0.268	28	**0.893**	0.074
	$\|O_2\| = 1668$	0.881	56	0.903	0.001	**17**	0.847	0.287	28	**0.905**	0.059
	$\|O_3\| = 1502$	0.874	56	0.895	0.001	**16**	0.846	0.193	28	**0.902**	0.067
	$\|O_4\| = 1336$	0.896	56	**0.920**	0.001	**15**	0.852	0.187	28	0.918	0.048
	$\|O_5\| = 1170$	0.902	56	**0.921**	0.001	**15**	0.872	0.104	28	0.920	0.029
	$\|O_6\| = 1004$	0.902	56	0.921	0.001	**14**	0.877	0.083	28	**0.928**	0.018
6	$\|O_1\| = 5002$	0.757	18	0.756	0.076	23	0.754	1.756	**8**	**0.779**	0.637
	$\|O_2\| = 4548$	0.752	18	0.740	0.051	23	0.747	1.625	**8**	**0.755**	0.546
	$\|O_3\| = 4094$	0.731	18	0.708	0.050	23	0.731	1.289	**8**	**0.747**	0.410
	$\|O_4\| = 3640$	0.701	18	0.670	0.041	22	0.691	1.081	**8**	**0.711**	0.308
	$\|O_5\| = 3186$	0.679	18	0.661	0.030	19	0.665	0.939	**8**	**0.696**	0.197
	$\|O_6\| = 2732$	0.635	18	0.607	0.017	19	0.616	0.668	**8**	**0.654**	0.178

Cuối cùng, chúng tôi phân tích độ chính xác từ mô hình phân lớp của các TTGT. Chúng tôi xét 36 trường hợp khi DT loại bỏ đi các tập đối tượng. Có 7 trường hợp TRG từ phương pháp có hiệu năng phân lớp thấp hơn các TTGT được so sánh. Trong 29 trường hợp còn lại, TTGT đề xuất đã thể hiện hiệu quả vượt trội tương đối rõ ràng, đặc biệt trên các tập dữ liệu không nhất quán và có hiệu quả phân lớp ban đầu thấp. Đây có thể coi là điểm mạnh lớn nhất của thuật toán khi sự đóng góp của thành phần độ khác biệt giúp điều chỉnh nhiễu một cách đáng kể. Trong khi, các TTGT theo cách tiếp cận từ tập thô cũng như các mở rộng của nó chưa thể giải quyết.

V. KẾT LUẬN

Nghiên cứu này đã xây dựng một công thức gia tăng dựa trên cơ sở từ độ đo khoảng cách của hai FPDs để áp dụng cho DT khi loại bỏ tập đối tượng. Thêm vào đó, bài báo này cũng thiết kế hai thuật toán theo cách tiếp cận IFS. Thuật toán thứ nhất được sử dụng để tìm một TRG của

bảng quyết định chưa có sự biến động. Thuật toán thứ hai là TTGT để tìm kiếm một rút gọn xấp xỉ khi DT loại bỏ tập đối tượng. Quá trình so sánh, đánh giá với các phương pháp dựa trên cách tiếp cận tập thô và tập mờ đã chỉ ra những ưu điểm nổi trội của hai thuật toán đề xuất trong việc nâng cao độ chính xác trên các bộ dữ liệu không nhất quán và có độ chính xác phân lớp ban đầu thấp. Tuy nhiên, hạn chế của hai phương pháp đề xuất là thời gian thực thi chậm do sự bổ sung của thành phần độ khác biệt trong các lớp tương đương mờ trực cảm.

Trong tương lai, chúng tôi sẽ tiếp tục tập trung phát triển tiếp các TTGT để giảm thiểu thời gian tính toán và cải thiện độ chính xác phân lớp. Cụ thể, chúng tôi cũng sẽ thiết kế các thuật toán gia tăng theo hướng IFS sử dụng cấu trúc hạt và các độ đo khác nhau cho các trường hợp loại bỏ và bổ sung tập thuộc tính. Ngoài ra, chúng tôi cũng sẽ nghiên cứu một số thuật toán RGTT phân phối cho các dữ liệu có số chiều vô cùng lớn và được phân bố ở các kho dữ liệu khác nhau.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi dự án nghiên cứu số 27-2022-RD/HĐ-ĐHCN, Đại học Công nghiệp Hà Nội. Các tác giả xin chân thành cảm ơn sự hỗ trợ và hợp tác nhiệt tình của phòng mô phỏng và tính toán hiệu năng cao (SHPC) thuộc Viện Công nghệ HaUI, Đại học Công nghiệp Hà Nội trong việc cung cấp một số cơ sở vật chất cần thiết để thực hiện nghiên cứu. Chúng tôi cũng xin dành lời cảm ơn đặc biệt tới các chuyên gia nghiên cứu tại Viện Công nghệ Thông tin, Viện Hàn Lâm Khoa học và Công nghệ Việt Nam vì sự cộng tác trong suốt quá trình xây dựng hướng nghiên cứu.

TÀI LIỆU THAM KHẢO

[1] X. Zhang, C. L. Mei, D. G. Chen and Y. Y. Yang, "A fuzzy rough set-based feature selection method using representative instances", *Knowledge-Based Systems, vol. 151*, (2018): 216-229.

[2] T. K. Sheeja and A. S. Kuriakose, "A novel feature selection method using fuzzy rough sets", *Computers in Industry, vol. 97*, (2018): 111-116.

[3] X. Che, D. Chen and J. Mi, "Label correlation in multi-label classification using local attribute reductions with fuzzy rough sets", *Fuzzy Sets and Systems, vol. 426*, (2022): 121-144.

[4] Z. Huang and J. Li, "A fitting model for attribute reduction with fuzzy - covering", *Fuzzy Sets and Systems, vol. 413*, (2021): 114-137.

[5] Q. H. Hu, D. R. Yu and Z. X. Xie, "Information-preserving hybrid data reduction based on fuzzy-rough techniques", *Pattern Recognition Letters, vol. 27, no. 5*, (2016): 414-423.

[6] A. Meriello and R. Battiti, "Feature Selection Based on the Neighborhood Entropy", *IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 12*, (2018): 6313-6322.

[7] B. Sang, H. Chen, L. Yang, T. Li and W. Xu, "Incremental Feature Selection Using a Conditional Entropy Based on Fuzzy Dominance Neighborhood Rough Sets", *IEEE Transactions on Fuzzy Systems, vol. 30, no. 6*, (2021): 1683-1697.

[8] Z. Yuan, H. Chen and T. Li, "Exploring interactive attribute reduction via fuzzy complementary entropy for unlabeled mixed data", *Pattern Recognition, vol. 127*, (2022).

[9] C. Z. Wang, Y. Huang, M. W. Shao and X. D. Fan, "Fuzzy rough set based attribute reduction using distance measures", *Knowledge-Based Systems, vol. 164*, (2019): 205-212.

[10] N. L. Giang, L. H. Son, T. T. Ngan, T. M. Tuan, H. T. Phuong et al., "Novel Incremental Algorithms for Attribute Reduction from Dynamic Decision systems using Hybrid Filter-Wrapper with Fuzzy Partition Distance", *IEEE Transactions on Fuzzy Systems, vol. 28, no. 5*, (2020): 858-873.

[11] Y. M. Liu, S. Y. Zhao, H. Chen, C. P. Li and Y. M. Lu, "Fuzzy Rough Incremental Attribute Reduction Applying Dependency Measures", *APWeb-WAIM 2017: Web and Big Data, vol. 10366*, (2017): 484-492.

[12] Y. Y. Yang, D. G. Chen, H. Wang, E. C. C. Tsang and D. L. Zhang, "Fuzzy rough set based incremental attribute reduction from dynamic data with sample arriving", *Fuzzy Sets and Systems, vol. 312*, (2017): 66-86.

[13] Y. Y. Yang, D. G. Chen, H. Wang and X. H. Wang, "Incremental perspective for feature selection based on fuzzy rough sets", *IEEE Transactions on Fuzzy Systems, vol. 26, no. 3*, (2017): 1257-1273.

[14] X. Zhang, C. L. Mei, D. G. Chen, Y. Y. Yang and J. H. Li, "Active Incremental Feature Selection Using a Fuzzy-Rough-Set-Based Information Entropy", *IEEE Transactions on Fuzzy Systems, vol. 28, no. 5*, (2020): 901- 915.

[15] P. Ni, S. Y. Zhao, X. H. Wang, H. Chen, C. P. Li and E. C. C. Tsang, "Incremental Feature Selection Based on Fuzzy Rough Sets", *Information Sciences, vol. 536*, (2020): 185-204.

[16] N. L. Giang, L. H. Son, N. A. Tuan, T. T. Ngan, N. N. Son et al., "Filter-Wrapper Incremental Algorithms for Finding Reduct in Incomplete Decision Systems When Adding and Deleting an Attribute Set", *International Journal of Data Warehousing and Mining, vol. 17, no. 2*, (2021): 39-62.

[17] N. T. Thang, N. L. Giang, H. V. Long, N. A. Tuan, T. M Tuan et al., "Efficient Algorithms for Dynamic Incomplete Decision Systems", *International Journal of Data Warehousing and Mining, vol. 17, no. 3*, (2021): 44-67.

[18] W. L. Hung and M. S. Yang, "Similarity measures of intuitionistic fuzzy sets based on Lp metric", *International Journal of Approximate Reasoning: Official Publication of the North American Fuzzy Information Processing Society, vol. 46, no. 1*, (2007): 120–136.

[19] B. Huang, C. X. Guo, Y. L. Zhuang, H. X. Li and X. Z. Zhou, "Intuitionistic fuzzy multigranulation rough sets", *Information Sciences, vol. 277*, (2014): 299-320.

[20] N. T. Thang, N.L. Giang, T. T. Dai, T. T. Tuan, N. Q. Huy et al., "A Novel Filter-Wrapper Algorithm on Intuitionistic Fuzzy Set for Attribute Reduction from Decision Tables", *International Journal of Data Warehousing and Mining, vol 17, no. 4*, (2021): 67-100.

[21] K. T. Atanassov, "Intuitionistic fuzzy sets. Fuzzy Sets and Systems", *An International Journal in Information Science and Engineering, vol. 20, no. 1*, (1986): 87–96.

[22] I. Iancu, "Intuitionistic fuzzy similarity measures based on Frank t-norms family", *Pattern Recognition Letters, vol. 42*, (2014): 128–136.

[23] W. Shu, W. Qian and Y. Xie, "Incremental approaches for feature selection from dynamic data with the variation of multiple objects", *Knowledge-Based Systems, vol. 163*, (2019): 320-331.

[24] UCI Machine learning repository, “Data,” 2021. [Online]. Available:https://archive.ics.uci.edu/ml/index.

Phạm Việt Anh tốt nghiệp Đại học chuyên ngành Hệ thống Thông tin Quản lý tại Đại học Kinh tế Quốc dân năm 2016 và nhận bằng thạc sĩ chuyên ngành Khoa học máy tính tại Đại học Công nghệ Thông tin và Truyền thông, Thái Nguyên năm 2019. Hiện đang là nghiên cứu sinh tại Học viện khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam và là giảng viên thuộc Viện Công nghệ HaUI, Đại học Công nghiệp Hà Nội. Các lĩnh vực nghiên cứu chính bao gồm Trí tuệ nhân tạo, tính toán tối ưu, học máy, khai phá dữ liệu và tính toán mềm.
Email: anhpv@haui.edu.vn

Nguyễn Long Giang nhận bằng Tiến sĩ Toán học năm 2012, Phó Giáo sư tại Viện Công nghệ thông tin - Viện Hàn lâm khoa học Việt Nam năm 2017. Tổng biên tập tạp chí Tin học và điều khiển (JCC). Lĩnh vực nghiên cứu: Trí tuệ nhân tạo (AI), tính toán mềm, tính toán mờ, khai phá dữ liệu.
Email: nlgiang@ioit.ac.vn

Nguyễn Ngọc Thủy tốt nghiệp Đại học và Cao học ngành Khoa học máy tính tại Đại học Huế vào các năm 2012 và 2015. Nhận bằng tiến sĩ ngành Khoa học máy tính tại Đại học Khon Kaen, Thái Lan năm 2020. Hiện công tác tại: Khoa CNTT, Trường Đại học Khoa học, Đại học Huế. Hướng nghiên cứu: Lý thuyết tập thô, Khai phá dữ liệu và học máy.
Email: nnthuy.cs@hueuni.edu.vn

Nguyễn Thế Thủy hiện công tác tại Trung tâm Công nghệ thông tin và Truyền thông - Sở Thông tin và Truyền thông tỉnh Bắc Ninh. Tốt nghiệp Thạc sĩ Công nghệ thông tin tại Đại học Mở Hà Nội năm 2017. Các lĩnh vực nghiên cứu gồm Trí tuệ nhân tạo, Khai thác dữ liệu, Tập thô, Tập thô mờ, Chuyển đổi số, Trung tâm dữ liệu.
Email: thethuy@gmail.com

Phạm Đình Khánh tốt nghiệp cử nhân ngành Toán Ứng Dụng đại học Kinh tế Quốc dân, Hà Nội năm 2015. Hiện là chuyên viên tư vấn cao cấp các dự án AI tại Công ty Cổ phần Công nghệ Neurond, Đà Nẵng, Việt Nam. Lĩnh vực nghiên cứu chính bao gồm Trí tuệ Nhân tạo, khai phá dữ liệu, giảm chiều Dữ liệu.
Email: phamdinhkhanh.tkt53.neu@gmail.com

Thuật toán song song khai thác itemset lợi nhuận phổ biến Skyline

Nguyễn Mạnh Hùng[1], Nguyễn Thị Thùy Trâm[2]
[1] Viện CNTT & TT, HVKTQS, Hà Nội, Việt Nam
[2] Trung tâm Phát triển Công nghệ Thông Tin, ĐH Công nghệ Thông Tin - ĐH Quốc gia Tp.HCM, Việt Nam
Tác giả liên hệ: Nguyễn Mạnh Hùng, Manhhungk12@mta.edu.vn
Ngày nhận bài: 28/07/2023, ngày sửa chữa: 09/09/2023, ngày duyệt đăng: 26/09/2023
Định danh DOI: 10.32913/mic-ict-research-vn.v2023.n2.1233

Tóm tắt: Các itemset lợi nhuận phổ biến Skyline (SFUI) có thể cung cấp nhiều thông tin hữu ích hơn cho việc ra quyết định với việc xem xét cả hai yếu tố là tần suất xuất hiện và lợi nhuận của chúng. Kể từ khi bài toán khai thác itemset lợi nhuận phổ biến Skyline được Goyal V. và các cộng sự đề xuất vào năm 2015, đến nay đã có nhiều thuật toán tuần tự được đề xuất nhằm cải thiện hiệu suất khai thác, tuy nhiên hầu hết các thuật toán đều có hiệu suất kém khi khai thác các tập dữ liệu lớn phổ biến hiện nay. Trong bài báo này, chúng tôi đề xuất một thuật toán song song có tên là ParaSFUI-UF dựa trên thuật toán tuần tự SFUI-UF là một thuật toán khai thác itemset lợi nhuận phổ biến Skyline hiệu quả nhất hiện nay. Các kết quả thực nghiệm cho thấy thuật toán ParaSFUI-UF vượt trội so với thuật toán SFUI-UF.

Từ khóa: *Data mining, frequent itemset, high utility itemset, Skyline Frequent-Utility Itemset, multi-core, parallel processing*

Title: **Parallel Algorithm Exploits Skyline Common Interest Element Set**

Abstract: Skyline common-utility element sets (SFUIs) can provide more useful information for decision-making by considering both their frequency and their benefits. Since the Skyline utility-common element set mining problem was proposed by Goyal V. and colleagues in 2015, up to now, many sequential algorithms have been proposed to improve mining performance. However, most algorithms have poor performance when exploiting today's popular large data sets. In this paper, we propose a parallel algorithm called ParaSFUI-UF based on the sequential algorithm SFUI-UF, which is the most effective algorithm for exploiting the Skyline common-benefit element set today. Experimental results show that the ParaSFUI-UF algorithm outperforms the SFUI-UF algorithm.

Keywords: *Data mining, frequent itemset, high utility itemset, Skyline Frequent-Utility Itemset, multi-core, parallel processing*

I. GIỚI THIỆU

Khai thác tập lợi nhuận phổ biến Skyline (SFUIM) nhằm khắc phục hạn chế của khai thác tập phổ biến và khai thác tập lợi nhuận cao. Cụ thể, khác với khai thác tập phổ biến và khai thác tập lợi nhuận cao, để khai thác tập lợi nhuận phổ biến Skyline người sử dụng không cần phải thiết lập các tham số đầu vào là ngưỡng độ hỗ trợ tối thiểu minSup hay là ngưỡng lợi nhuận tối thiểu minUtil. Từ khi bài toán khai thác tập lợi nhuận phổ biến Skyline được Goyal V. và các cộng sự đề xuất [1] vào năm 2015, đến nay đã có một số thuật toán được công bố nhằm nâng cao hiệu suất khai thác. Tuy nhiên, do không gian các itemset ứng viên của bài toán có độ phức tạp hàm mũ cho nên bài toán khai thác tập lợi nhuận phổ biến Skyline vẫn là một thách thức, đặc biệt khi xử lý dữ liệu lớn. Các thuật toán tuần tự [2-5] hoạt động tốt trên các tập dữ liệu nhỏ, tuy nhiên hầu hết chúng không thể hoàn thành nhiệm vụ khai thác trên các bộ dữ liệu quy mô lớn trong thời gian chấp nhận được do giới hạn về khả năng tính toán.

Mặt khác, tính toán song song [6,7], được phát triển để xử lý các tập dữ liệu lớn, cung cấp một giải pháp tiềm năng cho bài toán này. Đối với các bài toán khai thác tập phổ biến và bài toán khai thác tập lợi nhuận cao thì đã có khá nhiều thuật toán song song được công bố [11-14]. Tuy nhiên, tính đến hiện nay để giải bài toán SFUIM chỉ có các thuật toán tuần tự được công bố, trong đó thuật toán tuần tự SFUI-UF được cho là hiệu quả nhất.

Trong bài báo này chúng tôi đề xuất thuật toán song song ParaSFUI-UF sử dụng ưu điểm của tính toán song song để nâng cao hiệu suất khai thác. Nội dung tiếp theo của bài báo được tổ chức như sau: mục 2 trình bày các nghiên cứu liên quan, mục 3 thuật toán song song ParaSFUI-UF và thử

nghiệm đánh giá thuật toán, mục 4 kết luận.

II. CÁC NGHIÊN CỨU LIÊN QUAN

1. Khai thác itemset lợi nhuận - phổ biến Skyline

Cho I là một tập hữu hạn các item $I = i_1, i_2, \ldots, i_m$ và $X \subseteq I$ là một itemset, nếu X chứa k item thì X được gọi là *k-itemset*. Một cơ sở dữ liệu giao dịch (CSDL) D chứa n giao dịch: $D = T_1, T_2, \ldots, T_n$. Mỗi giao dịch $T_i \in D(1 \leq i \leq n)$ chứa một tập con các item thuộc tập I. Tần suất xuất hiện của tập X, ký hiệu là $f(X)$, là số lượng giao dịch trong D có chứa tập X.

Lợi nhuận trong của một item i_p trong giao dịch T_i, ký hiệu là $q(i_p, T_i)$, là số lượng của item i_p xuất hiện trong giao dịch T_i. Lợi nhuận ngoài của item i_p, kí hiệu là $prof(i_p)$, là đơn vị lợi nhuận của item i_p (là giá bán hoặc lợi nhuận khi bán được một item i_p). Lợi nhuận của item ip trong giao dịch T_i, kí hiệu là $u(i_p, T_i) = q(i_p, T_i) * prof(i_p)$.

Lợi nhuận của itemset X trong giao dịch T_i chứa tập X, ký hiệu là $u(X, T_i) = \sum_{i_p \in X} u(i_p, T_i)$. Lợi nhuận của itemset X trong CSDL D, ký hiệu là $u(X) = \sum_{X \subseteq T_i} u(X, T_i)$. Lợi nhuận giao dịch của giao dịch T_i, ký hiệu là $TU(T_i) = u(T_i, T_i)$.

Vì lợi nhuận của itemset không có tính chất đóng, nên Liu và cộng sự [8] đã đề xuất một khái niệm gọi là lợi nhuận giao dịch có trọng số, ký hiệu là $TWU(X) = \sum_{X \subseteq T_i} TU(T_i)$. TWU được sử dụng để cắt tỉa không gian các itemset ứng viên trong các thuật toán khai thác itemset lợi nhuận cao. Ví dụ CSDL D trong Bảng I và lợi nhuận ngoài của từng item trong Bảng II.

Bảng I
CSDL GIAO DỊCH

TID	Transaction	TU
T1	(A2, 2), (A4, 1)	9
T2	(A1, 2), (A2, 1), (A3, 2), (A4, 1), (A5, 1)	25
T3	(A2, 2), (A4, 2),(A5, 1)	18
T4	(A4, 2), (A5, 1)	14
T5	(A1, 4), (A2, 2), (A4, 1), (A5, 2)	41
T6	(A2, 4), (A5, 2)	16
T7	(A3, 5)	5

Bảng II
BẢNG LỢI NHUẬN CỦA CÁC ITEM

item	A1	A2	A3	A4	A5
Lợi nhuận	6	2	1	5	4

Xem xét một itemset dựa trên 2 yếu tố là tần suất xuất hiện và lợi nhuận: Một tập X là trội hơn tập Y nếu $f(X) \geq f(Y)$ và $u(X) \geq u(Y)$, **HOẶC** $f(X) \geq f(Y)$ và $u(X) \geq u(Y)$, ký hiệu là: $X \succ Y$.

Một itemset X trong CSDL D là *itemset lợi nhuận phổ biến Skyline* (SFUI) nếu X không bị trội hơn bởi bất kỳ một itemset nào khác.

Bài toán khai thác itemset lợi nhuận phổ biến Skyline (SFUIM): Cho một CSDL D với bảng lợi nhuận ngoài tương ứng của từng item, khai thác itemset lợi nhuận phổ biến Skyline(SFUIM) chính là việc đi tìm tất cả các tập SFUIs trong CSDL D.

Điểm khác biệt của bài toán khai thác itemset lợi nhuận phổ biến Skyline so với các bài toán khai thác itemset phổ biến và bài toán khai thác itemset lợi nhuận cao đó là khi thực hiện khai thác các itemset lợi nhuận phổ biến Skyline, người sử dụng không cần phải xác định các giá trị ngưỡng về độ hỗ trợ tối thiểu (minSup) và lợi nhuận tối thiểu (minUtil).

Như đã giới thiệu ở trên, Goyal V. và các cộng sự đã đề xuất bài toán SFUIM và thuật toán SKYMINE [1] sử dụng cấu trúc UP-Tree và kỹ thuật tăng trưởng mẫu để khai thác các tập SFUIs.

Sau đề xuất của Goyal V., đã có một số thuật toán được công bố nhằm nâng cao hiệu năng khai thác SFUIs: Trong [2, 10] các tác giả đã đề xuất cấu trúc mảng umax và sử dụng cấu trúc danh sách lợi nhuận UL [9] để xây dựng thuật toán SFU-Miner khai thác các tập SFUIs. So với thuật toán SKYMINE, thuật toán SFU-Miner ưu điểm hơn nhờ sử dụng cấu trúc umax để cắt tỉa không gian các itemset ứng viên và cấu trúc UL có thể xác định trực tiếp SFUIs mà không cần tạo các tập ứng viên.

Lin và công sự [3] đã đề xuất cấu trúc utility-max và sử dụng cấu trúc danh sách lợi nhuận UL để xây dựng 02 thuật toán duyệt theo chiều sâu SKYFUP-D và thuật toán duyệt theo chiều rộng SKYFUP-B khai thác các tập SFUIs. SKYFUP hiệu quả hơn SKYMINE và SFU-Miner.

Nguyen và công sự [4] đã đề xuất cấu trúc danh sách lợi nhuận mở rộng EUL, so sánh với UL, EUL bổ sung thêm 02 trường: lợi nhuận của itemset và lợi nhuận của item cuối cùng trong itemset. Trên cơ sở cấu trúc EUL, Nguyen và công sự đề xuất thuật toán FSKYMINE khai thác SFUIs. Thuật toán FSKYMINE hiệu quả hơn so với thuật toán được đề xuất trong [2, 10].

Wei Song và cộng sự [5] đã đề xuất một số cấu trúc và chiến lược để nâng cao hiệu quả khai thác các tập SFUIs:

- Thứ nhất, đề xuất cấu trúc mảng lợi nhuận tối đa, ký hiệu là MUA và sử dụng danh sách lợi nhuận UL kết hợp với MUA để cắt tỉa các itemset và các mở rộng không tiềm năng.

- Thứ 2, sử dụng TWU để lọc các item không phải là SFUIs. Thứ 3, đề xuất khái niệm lợi nhuận tối thiểu của tập mở rộng, ký hiệu là MUE, để cắt tỉa không gian các itemset ứng viên. Trên cơ sở các kỹ thuật được đề xuất ở

trên, Wei Song và cộng sự đã đề xuất thuật toán SFUI-UF. SFUI-UF là thuật toán khai thác các tập SFUIs hiệu quả nhất hiện nay.

2. Một số khái niệm cơ sở

Định nghĩa 2.1 (Itemset lợi nhuận phổ biến tiềm năng Skyline): [2]. Một itemset X là itemset lợi nhuận phổ biến tiềm năng (PSFUI) nếu không tồn tại tập Y bất kỳ sao cho $f(Y) = f(X)$ và $u(Y) \geq u(X)$.

Trong [2], Pan và các cộng sự đã chứng minh tất cả các SFUIs đều là PSFUIs, vì vậy nhiều thuật toán khai thác các tập SFUIs có thể thực hiện theo phương pháp 2 bước: bước 1, tìm ra tất cả các tập PSFUIs, và sau đó tiến hành lọc PSFUIs để tìm ra các tập SFUIs.

Định nghĩa 2.2 (mảng MUA): [5]. Gọi fmax là tần suất tối đa của tất cả các itemset trong CSDL D và MUA là một mảng có kích thước bằng fmax. Giá trị của item thứ $f(1 \leq f \leq fmax)$ trong mảng MUA được định nghĩa như sau: $MUA(f) = max\{u(X)|f(X) \geq f\}$

Trong đó X là itemset bất kỳ trong CSDL D. Mảng MUA tương tự như mảng utilmax[3] nhưng khác nhau về kích thước. Cụ thể, theo [3], utilmax có kích thước bằng số lượng giao dịch của CSDL D, trong khi đó MUA có kích thước bằng $fmax$. Trong [5] Wei Song và cộng sự đã chứng minh $|MUA| \leq |utilmax|$, trong đó $|MUA|$ và $|ultimax|$ tương ứng là kích thước của mảng MUA và mảng utilmax.

Định lý 2.1: Gọi X là một itemset. Nếu tổng của tất cả các giá trị iutil và rutil trong X. UL nhỏ hơn $MUA(f(X))$, thì X và tất cả các phần mở rộng của X không phải là SFUIs [5].

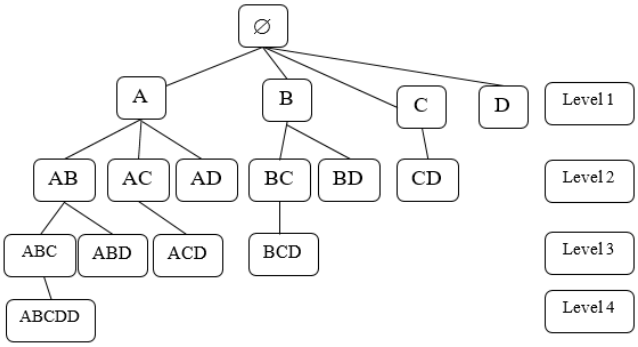
Định nghĩa 2.3 (Lợi nhuận tối thiểu của SFUIs):

[5]. Xét CSDL D, lợi nhuận tối thiểu của SFUIs, ký hiệu là MUS, trong D là lợi nhuận tối đa của các item đơn lẻ có tần suất cao nhất trong D. Cụ thể MUS được định nghĩa như sau: $MUS = max\{u(i_p)|f(i_p) = fmax\}$,

Trong đó i_p là một item trong D và fmax là tần suất lớn nhất của tất cả các tập chứa một item trong D.

Định lý 2.2: Gọi i_p là tập một item. Nếu $TWU(i_p) \leq MUS$ thì i_p và tất cả các itemset có chứa i_p không phải là $SFUIs$. Như vậy, khi itemset chứa một item có TWU thấp hơn MUS thì itemset này và tất cả các tập chứa nó có thể được cắt tỉa một cách an toàn. Trong thuật toán SFUI-UF, MUS được đặt là các giá trị ban đầu của mỗi item trong MUA [5].

Định nghĩa 2.4 (Lợi nhuận tối thiểu của phần mở rộng): Cho X là một itemset, i_p là một item, và tập $X \cup i_p$ là một tập mở rộng của itemset X theo item i_p. Khi đó, lợi nhuận tối thiểu của tập mở rộng $X \cup i_p$, ký hiệu là $MUE(X \cup i_p)$, được định nghĩa bằng $u(X)$, cụ thể như sau: $MUE(X \cup i_p) = u(X)$.



Hình 1. Không gian tìm kiếm của thuật toán SFUI-UF.

Ví dụ, xét CSDL trong Bảng I và Bảng II, ta có: $MUE(A1A4) = u(A1) = 36; MUE(A2A4) = u(A2) = 22$

Định lý 2.3: Cho X là một itemset và itemset Y là một mở rộng của X. Nếu $u(Y) \leq MUE(Y)$, Y không phải là SFUI [5].

Thuật toán SFUI-UF[5] dựa trên các định lý 1, 2, 3 để cắt tỉa không gian các itemset ứng viên. Nội dung tiếp theo, bài báo đề xuất thuật toán song song khai thác SFUIs.

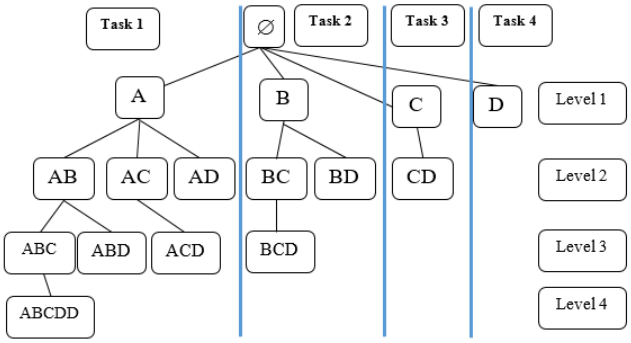
III. THUẬT TOÁN SONG SONG KHAI THÁC ITEMSET LỢI NHUẬN - PHỔ BIẾN SKYLINE

Ngày nay ngay cả các máy tính cá nhân cũng đã có bộ xử lý đa nhân (multi-core) và việc sử dụng lập trình song song nhằm phát huy tối đa sức mạnh của bộ xử lý đa nhân cũng khá phổ biến trong các ứng dụng. Lập trình song song mang lại nhiều lợi ích như: nâng cao hiệu năng xử lý và giúp giải quyết được các bài toán có kích thước lớn. Tuy nhiên, việc phát triển các thuật toán song song hoặc cải tiến các thuật toán tuần tự hiện có để có thể thực hiện song song sẽ gặp phải một số thách thức như: vấn đề về đồng bộ, vấn đề về truyền thông và vấn đề về cân bằng tải.

Nhằm phát huy tốt nhất các ưu điểm của các bộ xử lý đa nhân trên các máy tính ngày nay, trong bài báo này chúng tôi đề xuất một thuật toán song song khai tác tập lợi nhuận phổ biến Skyline có tên là ParaSFUI-UF. Thuật toán ParaSFUI-UF là mở rộng của thuật toán SFUI-UF theo hướng có thể thực hiện song song việc khám phá không gian các itemset ứng viên.

Thuật toán SFUI-UF khám phá các tập SFUIs theo phương pháp tìm kiếm theo chiều sâu trên cây liệt kê các itemset bắt đầu từ nút gốc rỗng. Hình 1 minh hoạ một cây liệt kê với 04 item A, B, C, D.

Phương pháp ParaSFUI-UF thực hiện song song công việc khám phá không gian các itemset ứng viên theo phương pháp chia để trị: chia không gian các itemset ứng



Hình 2. Phân hoạch không gian các itemset ứng viên trong ParaSFUI-UF.

viên thành các không gian con, mỗi không gian con sẽ gồm MP item ở mức Level1 trên cây liệt kê, MP là tham số được tính toán theo số lượng nhân của bộ xử lý đa nhân; Việc khám phá trên mỗi không gian con được gọi là một tác vụ (task); Các tác vụ sẽ được xử lý song song trên các nhân của bộ xử lý đa nhân (multi-core). Hình 2 minh họa việc chia không gian các itemset ứng viên thành các không gian con, mỗi không gian con gồm 1 item ở Level1 và các mở rộng của nó ở mức tiếp theo.

Phương pháp ParaSFUI-UF gồm 2 giai đoạn:

Giai đoạn 1, từ dòng 1 đến dòng 13 trong Thuật toán 1 nhằm xác định danh sách lợi nhuận UL của các tập 1 item, mảng MUA.

Giai đoạn 2, từ dòng 14 đến dòng 19 trong Thuật toán 1. Chi tiết giai đoạn 2 được thực hiện trong Thuật toán 2 và Thuật toán 3. Hàm tail(X) trong tham số đầu vào của Thuật toán 2 được định nghĩa là tất cả các item đứng sau tất cả các item trong tập X theo một thứ tự Δ nào đó. Điểm thay đổi chính của thuật toán ParaSFUI-UF so với thuật toán SFUI-UF nằm ở vòng lặp for từ dòng 1 đến dòng 13 của Thuật toán 2 và vòng lặp for từ dòng 1 đến dòng 9 của Thuật toán 3. Cụ thể, trong Thuật toán 2, mỗi MP item trong tail(X) sẽ được khám phá song song theo phương pháp tìm kiếm theo chiều sâu trong một tác vụ trên một nhân của bộ xử lý. Hình 2 cho thấy các không gian con là độc lập với nhau và hợp của chúng sẽ bằng không gian tổng thể của bài toán. Mỗi tác vụ sẽ giữ riêng cho mình X, tail(X) và mảng MUA.

Trong thuật toán 3, thực hiện song song hóa việc tìm ra các tập không phải là itemset lợi nhuận phổ biến Skyline từ các tập tiềm năng S-PSFUIs. Cụ thể, mỗi MP item trong S-PSFUIs sẽ được xử lý song song trong các tác vụ.

1. Vấn đề đồng bộ kết quả khai thác

Trong thuật toán 2, các tác vụ được thực thi song song trên các nhân của bộ xử lý và chúng phải đồng bộ hóa thao tác ghi kết quả mỗi khi tìm ra được một tập tiềm năng PSFUI. Cụ thể trong dòng lệnh 6 và dòng lệnh 7, khi phát hiện ra tập X' là itemset lợi nhuận phổ biến tiềm năng Skyline, thao tác ghi tập X' ra tập kết quả và thao tác xóa các tập Y không phải là tập tiềm năng ra khỏi tập kết quả phải được đồng bộ giữa các tác vụ.

Algorithm 1 ParaSFUI-UF

Input CSDL D cùng với bảng lợi nhuận ngoài của từng item

Output Tất cả các tập SFUIs

1: #Giai đoạn 1
2: Scan D once to calculate $fmax$ and MUS;
3: Delete item with TWU values lower than MUS;
4: Calculate the TWU values of the remaining items;
5: **for all** $T_d \in D$ **do**
6: Sort items in T_d using TWU - ascending oder;
7: **for all** $i \in T_d \in D$ **do**
8: Update i.UL;
9: **end for**
10: **end for**
11: **for** $j = 1$ to $fmax$ **do**
12: $MUA(j) = MUS$;
13: **end for**
14: #Giai đoạn 2
15: Shared S-PSFUI=∅;
16: Shared S-SFUI=∅;
17: ParaP-Miner(∅,tail(∅),MUA);
18: ParaS-Miner(S-PSFUI);
19: **return** $all\ SFUIs$;

Algorithm 2 ParaP-Miner

Input X, $tail(X)$, MUA

Output Tất cả các tập PSFUIs là mở rộng của tập X

1: **for all** Parallel MP item $i \in tail(X)$ **do**
2: $X' = X \cup i$;
3: **if** $u(X') \geq MUE(X')$ **then**
4: **if** $u(X') \geq MUA(f(X'))$ **then**
5: **Update** MUA according to $f(X')$ and $u(X')$;
6: **Synchronize**: $S - PSFUI = S - PSFUI \cup X'$;
7: **Synchronize**: Remove Y from $S - PSFUI$ if $(f(Y) == f(X'))$;
8: **end if**
9: **if** $\sum d \in X.tids(iutil(X', T_d) + rutil(X', T_d)) \geq MUA(f(X'))$ **then**
10: $P - Mine(X', tail(X'), MUA)$;
11: **end if**
12: **end if**
13: **end for**

Algorithm 3 ParaS-Miner

Input S-PSFUIs
Output Tất cả các tập SFUIs

1: **for all** Parallel MP itemset X in S-PSFUI **do**
2: **for all** itemset Y in S-PSFUI **do**
3: **if** $f(X) \geq f(Y)$ and $u(X) \geq u(Y)$ or $f(X) \geq f(Y)$ and $u(X) \geq u(Y)$ **then**
4: **Synchronize**: Remove Y from $S - PSFUI$;
5: **end if**
6: **end for**
7: **end for**
8: $S - SFUI = S - PSFUI$;
9: **return** $S - SFUI$;

2. Vấn đề cân bằng tải

Trong các thuật toán song song, việc xử lý tốt vấn đề cân bằng tải sẽ cải thiện được hiệu năng tổng thể của chương trình. Việc cân bằng tải bao gồm hai yếu tố: thứ nhất, giảm thiểu dữ liệu phải truyền thông giữa các tác vụ; thứ 2, tối thiểu tổng thời gian nhàn rỗi của các bộ xử lý (các nhân trong bộ xử lý đa nhân). Bộ xử lý đa nhân được thiết kế theo mô hình xử lý song song sử dụng bộ nhớ chia sẻ do đó người lập trình sẽ không phải quan tâm đến vấn đề truyền thông dữ liệu giữa các tác vụ. Hơn nữa, các nhân của một bộ xử lý đa nhân thường có khả năng tính toán như nhau, vì vậy để cân bằng tải, thuật toán phải giao khối lượng công việc như nhau cho mỗi nhân xử lý nhằm đảm bảo các nhân luôn ở trong trạng thái xử lý công việc trước khi thuật toán kết thúc. Các tác vụ trong các thuật toán song song sẽ được quản lý trong một task-pool. Sau đó, các tác vụ này sẽ được xử lý song song theo chiến lược tác vụ đến trước sẽ được xử lý trước trong các nhân của bộ xử lý. Như vậy, trên bộ xử lý đa nhân, nếu thời gian xử lý đồng bộ giữa các tác vụ nhỏ, để thực hiện cân bằng tải chúng ta nên chia tác vụ có khối lượng tính toán nhỏ. Tuy nhiên, mỗi tác vụ trong task-pool sẽ được cấp phát bộ nhớ để lưu trữ các biến riêng của chúng, do đó nếu kích thước của task-pool quá lớn sẽ gây ra hiện tượng tràn bộ nhớ. Để tối thiểu hóa kích thước của task-pool, chúng ta có thể chọn số tác vụ bằng đúng số nhân của bộ xử lý. Tuy nhiên, điều này lại có thể ảnh hưởng đến vấn đề cân bằng tải. Cụ thể, trên Hình 2 cho thấy các tác vụ ở phía bên trái (ứng với các item ở đầu của tail(X)) sẽ có không gian các itemset ứng viên lớn hơn các tác vụ phía bên phải, do đó những nhân xử lý các tác vụ ở phía bên trái sẽ có thể hoàn thành công việc chậm hơn so với những nhân thực hiện các tác vụ ở phía bên phải.

3. Thử nghiệm

Các tác giả trong [5], khi thực nghiệm thuật toán SFUI-UF, đã sắp xếp các item theo thứ tự giảm dần của giá trị TWU và duyệt các item trên cây liệt kê theo thứ tự từ trái sang phải để đánh giá thuật toán SFUI-UF. Tuy nhiên, trong thuật toán song song, thứ tự tìm kiếm của các item sẽ bị thay đổi và không nhất quán giữa các lần chạy khác nhau. Vì vậy, để đánh giá ảnh hưởng của thứ tự tìm kiếm đến kích thước của không gian các itemset ứng viên và tốc độ thực hiện của thuật toán, trong phần thực nghiệm này chúng tôi sẽ thực hiện 02 nội dung: Nội dung 1, các item ở Level 1 của cây liệt kê được tổ chức thành một hàng đợi xoay vòng theo thứ tự giảm dần của giá trị TWU, đánh giá thuật toán SFUI-UF bằng cách thay đổi vị trí xuất phát: TH1: Xuất phát từ đầu hàng đợi (từ item bên trái cùng trên cây liệt kê); TH2: Xuất phát từ vị trí cách đầu hàng đợi một khoảng bằng 1/3 chiều dài hàng đợi; TH3: Xuất phát từ giữa hàng đợi; TH4: Xuất phát từ vị trí cách đầu hàng đợi một khoảng bằng 2/3 chiều dài hàng đợi; TH5: Xuất phát từ cuối hàng đợi. Nội dung 2, so sánh phương pháp ParaSFUI-UF với thuật toán SFUI-UF ở 2 trường hợp điển hình là TH1 và TH5, các trường hợp khác hoàn toàn tượng tự.

Mã nguồn thuật toán SFUI-UF được lấy từ công bố của nhóm tác giả [5] trong bộ thư viện nguồn mở SPMF[15].

Phương pháp ParaSFUI-UF được cài đặt và đánh giá trên môi trường Java 17 sử dụng fork-join framework để chia không gian các itemset ứng viên thành các không gian con theo phương pháp đệ quy. Số lượng tác vụ con được thực hiện song song được tính toán tự động theo số nhân của bộ xử lý trên máy tính. Trong quá trình tiến hành thực nghiệm, chúng tôi thấy rằng để cân bằng tải, và không bị tràn bộ nhớ do kích thước hàng đợi lớn thì số lượng tác vụ được định nghĩa bằng 2 lần số nhân của bộ xử lý. Các thực nghiệm chúng tôi đã thực hiện trên máy tính có cấu hình 8-core 3.00GHz, 8GB RAM chạy hệ điều hành Windows 10 phiên bản 64 bit.

Các cơ sở dữ liệu sử dụng gồm: mushroom_utility (viết tắt là M_Util), chess_utility (viết tắt là C_Util) và connect_utility (viết tắt là Con_Util) được lấy từ [15].

Kết quả thực nghiệm nội dung 1 được trình bày trong Bảng III và Bảng IV:

Thời gian chạy 05 trường hợp trong Bảng IV là kết quả trung bình của 20 lần chạy. Từ kết quả thực nghiệm thuật toán SFUI-UF trên 03 CSDL ở Bảng III và Bảng IV, chúng ta thấy kích thước của không gian các itemset ứng viên và thời gian thực hiện của thuật toán có xu hướng giảm khá nhanh khi vị trí bắt đầu duyệt dần về cuối hàng đợi. Kết quả thực nghiệm nội dung 2 được trình bày trong 02 bảng: Bảng V và Bảng VI như sau:

Các kết quả đánh giá thời gian thực hiện và không gian các itemset ứng viên của thuật toán ParaSFUI-UF trong các Bảng V và Bảng VI là các giá trị trung bình của 20 lần chạy chương trình. Kết quả thực nghiệm trong Bảng V và

Bảng III
SO SÁNH KHÔNG GIAN CÁC ITEMSET ỨNG VIÊN CỦA SFUI-UF

Dataset	#SFUIs	TH1	TH2	TH3	TH4	TH5
M_Util	17	22,710	12,820	9,236	4,506	4,743
C_Util	35	915,101	269,561	117,050	86,364	81,301
Con_Util	46	4,349,427	4,952,510	1,968,260	685,520	654,885

Bảng IV
SO SÁNH KẾT QUẢ THỜI GIAN THỰC HIỆN CỦA SFUI-UF (MS)

Dataset	TH1	TH2	TH3	TH4	TH5
M_Util	1,588	1,500	1,322	998	1,012
C_Util	28,273	17,732	12,708	12,524	11,925
Con_Util	4,141,415	4,267,812	2,821,565	1,810,984	1,762,345

Bảng V
SO SÁNH KHÔNG GIAN CÁC ITEMSET ỨNG VIÊN

Dataset	SFUI-UF (TH1)	SFUI-UF (TH5)	ParaSFUI-UF	
	#PSFUIs	#PSFUIs	#SFUIs	#PSFUIs
M_Util	22,710	4,743	17	11,650
C_Util	915,101	81,301	35	17,981
Con_Util	4,349,427	654,885	46	54,081

Bảng VI
SO SÁNH THỜI GIAN THỰC HIỆN(MS)

Dataset	SFUI-UF (TH1)	SFUI-UF (TH5)	ParaSFUI-UF
M_Util	1,588	1,012	637
C_Util	28,273	11,925	788
Con_Util	4,141,415	1,762,345	43,439

Bảng VI cho thấy thuật toán ParaSFUI-UF vượt trội so với thuật toán SFUI-UF nguyên bản [5] và thuật toán SFUI-UF thực hiện theo TH5.

IV. KẾT LUẬN

Trong bài báo này, chúng tôi đã đề xuất thuật toán song song ParaSFUI-UF trên cơ sở mở rộng thuật toán hiệu quả nhất hiện nay SFUI-UF. Thuật toán ParaSFUI-UF đã thực hiện chia không gian các itemset ứng viên SFUIs thành các không gian con và thực hiện khám phá các không gian con một cách song song trên các nhân của bộ xử lý đa nhân. Phần thực nghiệm đã tiến hành 2 nội dung đánh giá: Thứ nhất là đánh giá sự ảnh hưởng của thứ tự tìm kiếm đến kích thước của không gian các itemset ứng viên và tốc độ thực hiện của thuật toán SFUI-UF; Thứ hai là đánh giá, so sánh phương pháp ParaSFUI-UF so với thuật toán SFUI-UF: kết quả cho thấy, phương pháp ParaSFUI-UF có hiệu quả vượt trội so với SFUI-UF.

TÀI LIỆU THAM KHẢO

[1] Vikram Goyal, Ashish Sureka, Dhaval Patel, "Efficient skyline itemsets mining", *C3S2E '15: Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering* (2015), 119–124.

[2] Pan Jeng-Shyanga, Lin Jerry Chun-Weib, Yang Lub, Fournier-Viger Philippec, Hong Tzung-Peid, "Efficiently mining of skyline frequent-utility patterns", *Intelligent Data Analysis*, vol. 21, no. 6 (2017), 1407-1423.

[3] Jerry Chun-Wei Lin, Lu Yang, Philippe Fournier-Viger, Tzung-Pei Hong, "Mining of skyline patterns by considering both frequent and utility constraints", *Engineering Applications of Artificial Intelligence*, Volume 77 (2019), 229-238.

[4] Hung Manh Nguyen; Anh Viet Phan; Lai Van Pham: FSKYMINE, "A Faster Algorithm For Mining Skyline Frequent Utility Itemsets", *Proceedings of the 6th NAFOSTED Conference on Information and Computer Science*, (2019), 251–255.

[5] Wei Song, Chuanlong Zheng, Philippe Fournier-Viger, "Mining Skyline Frequent-Utility Itemsets with Utility Filtering", *18th Pacific Rim International Conference on Artificial Intelligence*, PRICAI (2021), Part I ,411–424.

[6] Bertil Schmidt, Jorge Gonzalez-Martinez, Christian Hundt, Moritz Schlarb, Parallel Programming: Concepts and Practice, Morgan Kaufmann (2017).

[7] Julian Shun: Shared-Memory Parallelism Can Be Simple, Fast, and Scalable, Association for Computing Machinery and Morgan & Claypool (2017).

[8] Ying Liu, Wei-keng Liao, Alok Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets", *Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference*, PAKDD (2005), 689–695.

[9] Mengchi Liu, Junfeng Qu: "Mining high utility itemsets without candidate generation", *21st ACM International Conference on Information and Knowledge Management*, (2012), 55–64.

[10] Jerry Chun-Wei Lin, Lu Yang, Philippe Fournier-Viger, Siddharth Dawar, Vikram Goyal, Ashish Sureka, and Bay Vo, "A more efficient algorithm to mine skyline frequent-utility patterns", *In International Conference on Genetic and Evolutionary Computing*, 2017, 127–135.

[11] Bay Vo, Loan T. T. Nguyen, Trinh D. D. Nguyen, Philippe Fournier-Viger, Unil Yun, "A Multi-Core Approach to Efficiently Mining High-Utility Itemsets in Dynamic Profit Databases", *IEEE Access* (2020), 85890 – 85899.

[12] Mohammed J. Zaki, "Parallel Sequence Mining on Shared-

Memory Machines", *Journal of Parallel and Distributed Computing*, Volume 61, Issue 3 (2001), 401-426.
[13] Krishan Kumar Sethi, Dharavath Ramesh , Damodar Reddy Edla: P-FHM+, "Parallel high utility itemset mining algorithm for big data processing", *Procedia Computer Science*, Volume 132 (2018), 918-927.
[14] O. Jamsheela, G. Raju, "Parallelization of Frequent Itemset Mining Methods with FP-tree: An Experiment with PrePost+ Algorithm", *The International Arab Journal of Information Technology*, Vol. 18, No. 2(2021), 208-231.
[15] An Open-Source Data Mining Library: http://www.philippe-fournier-viger.com/spmf/

Nguyễn Mạnh Hùng nhận bằng Tiến sĩ Bảo đảm toán học cho Máy tính và Hệ thống tính toán năm 2004 tại Trung tâm Tính toán, Viện Hàn lâm Khoa học, Liên bang Nga. Hiện đang công tác tại Viện CNTT&TT, Học viện KTQS.
Lĩnh vực nghiên cứu: Khai phá dữ liệu; Tính toán song song; Khoa học dữ liệu.
Email: ManhHungk12@mta.edu.vn

Nguyễn Thị Thùy Trâm tốt nghiệp Đại học chuyên ngành Công nghệ Thông tin, Trường ĐH Công nghệ Thông tin – ĐH Quốc Gia - Thành phố Hồ Chí Minh. Tốt nghiệp Thạc sĩ Khoa học máy tính năm 2020 của Học viện Kỹ thuật Quân sự. Hiện nay công tác tại Trung tâm Phát triển Công nghệ Thông tin – Trường ĐH Công nghệ thông tin – ĐH Quốc gia TP.HCM.
Lĩnh vực nghiên cứu: Khai phá dữ liệu; Tính toán song song; Rút gọn thuộc tính.
Email: tramntt@uit.edu.vn

Nghiên cứu phương pháp phát hiện va chạm của cánh tay robot cộng tác 6 bậc tự do

Nghiêm Văn Hưng[1,2], Đặng Văn Đức[3], Nguyễn Văn Căn[2], Trịnh Hiền Anh[3]

[1] Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Hà Nội, Việt Nam

[2] Trường Đại học Kỹ thuật - Hậu cần Công an nhân dân, Bộ Công an, Bắc Ninh, Việt Nam

[3] Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Hà Nội, Việt Nam

Tác giả liên hệ: Nghiêm Văn Hưng, nghiemvanhung1985@gmail.com

Ngày nhận bài: 08/04/2023, ngày sửa chữa: 29/07/2023, ngày duyệt đăng: 01/10/2023

Định danh DOI: 10.32913/mic-ict-research-vn.v2023.n2.1236

Tóm tắt: Robot cộng tác (Collaborative robot - Cobot) là những robot có thể tiếp xúc với con người một cách trực tiếp, phát huy đồng thời được các lợi thế của cả con người và robot để tăng hiệu suất công việc. Cobot hoạt động thân thiện và tương tác với con người vì được lập trình để phát hiện va chạm an toàn. Do đó yêu cầu phát hiện chính xác và nhanh chóng các va chạm của cánh tay cobot là một chủ đề thu hút được sự quan tâm của nhiều nhà nghiên cứu. Nghiên cứu của chúng tôi đề xuất hướng tiếp cận áp dụng hai kỹ thuật học máy có giám sát là SVMR (Support Vector Machine Regression) và 1D CNN (1D Convolutional Neural Network) cho kết quả phát hiện va chạm hiệu quả với cánh tay robot CURA6 trên bộ dữ liệu của Intema.

Từ khóa: *Phát hiện va chạm, cánh tay robot cộng tác, 6-DoF.*

Title: **Collision Detection for 6-DoF Collaborative Robot Arm**

Abstract: Cobots are robots that can directly contact humans, simultaneously promoting the advantages of both humans and robots to increase work efficiency. Cobots operate friendly and interactive with humans because they are programmed to detect collisions safely. Therefore, the need to accurately and quickly detect collisions of cobot arms is a topic that attracts the attention of many researchers. The our proposed technique using the SVMR model and the 1D CNN model is tested and gives good collision detection results with CURA6 cobot arm on the Intema's dataset.

Keywords: *Collision detection, cobot arm, 6-DoF.*

I. GIỚI THIỆU

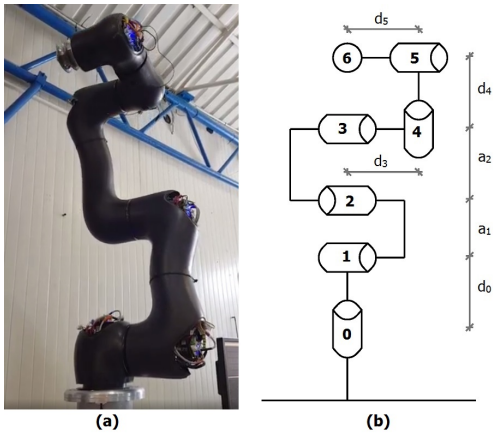
Cobot là những robot có thể làm việc đồng thời với người vận hành. Theo ISO 10218 [7], cobot là robot có thể tiếp xúc với con người một cách trực tiếp trong một không gian làm việc cộng tác xác định. Khác với robot công nghiệp, cobot được thiết kế để bảo đảm tương tác an toàn với con người cũng như môi trường làm việc cộng tác. Cobot được chế tạo bằng vật liệu nhẹ và thiết kế với hình dạng mô phỏng tương tự cánh tay người. Cobot hoạt động phối kết hợp với con người để tăng hiệu năng quá trình sản xuất hoặc khắc phục các hạn chế kỹ thuật mà không cần đến các thiết bị vật lý bảo hộ như tấm chắn, lưới, hàng rào bảo vệ như robot công nghiệp [18].

Đã có nhiều công bố về bài toán phát hiện va chạm với đối tượng nghiên cứu là cánh tay cobot [2, 8, 13, 20]. Điển hình là cánh tay cobot sáu bậc tự do 6-DoF (Six Degrees of Freedom) được thiết kế nhằm thực hiện các tác vụ chính xác và an toàn với con người. Hình 1 mô tả cánh tay robot CURA6 (Cooperative Universal Robotic Assistant 6) của Intema [2].

Các phần tiếp theo của bài báo gồm: Mục II giới thiệu các công trình nghiên cứu có liên quan; mục III khái quát về động lực học robot; mục IV mô tả phương pháp đề xuất của chúng tôi. Các kết quả thử nghiệm trên bộ dữ liệu Intema được trình bày trong mục V. Cuối cùng là Mục VI đưa ra kết luận và định hướng cho các nghiên cứu phát triển trong thời gian tới.

II. CÁC NGHIÊN CỨU LIÊN QUAN

S. Haddadin và các cộng sự [5] công bố một khảo cứu khá toàn diện về bài toán xử lý va chạm của robot. Bài toán xử lý va chạm được chia thành những tác vụ chính là: phát hiện (detection) va chạm, nhận dạng (identification) va chạm, phân loại (classification) va chạm và phản ứng



Hình 1. (a) Hình chụp robot CURA6 tại phòng thí nghiệm của Intema; (b) Mô hình robot CURA6

(reaction) va chạm. Trọng tâm nghiên cứu của chúng tôi là bài toán phát hiện va chạm, đây chính là bước đầu tiên, có tính chất định hướng các tác vụ khác trong toàn bộ hệ xử lý va chạm. Chúng tôi phân chia các phương pháp phát hiện va chạm trên cánh tay cobot thành hai nhóm: (i) nhóm phương pháp sử dụng học máy và (ii) nhóm phương pháp không sử dụng học máy. Nhóm phương pháp (ii) thường dựa vào các cảm biến bề mặt xúc giác (sensor skins) bao phủ bên ngoài robot [1, 17]. Tuy nhiên, các cảm biến bề mặt xúc giác thường có chi phí cao, không bền và có thể bị hỏng khi bị va chạm mạnh hay va chạm nhiều lần. Các phương pháp dựa trên học máy [2, 5, 13] thường ước tính các mômen xoắn khớp bên ngoài do va chạm, sau đó so sánh với một tập giá trị ngưỡng cho trước để kiểm tra xem va chạm có xảy ra hay không. Đo mômen xoắn trực tiếp sẽ là lý tưởng, nhưng cảm biến mômen xoắn có thể tốn kém, giải pháp thay thế là ước tính mômen xoắn khớp từ các phép đo dòng điện tại các bộ truyền động khớp.

Hiện nay, việc áp dụng các phương pháp học máy để phát hiện va chạm đã trở nên phổ biến. Với điều kiện thu thập đủ lượng dữ liệu đa dạng, chất lượng tốt, các phương pháp học máy mở ra khả năng có thể khắc phục được sự không chắc chắn (uncertainty) của các tham số trong mô hình động lực học cũng như các hiệu ứng không được mô hình hóa như ma sát và nhiễu đo lường.

A. N. Sharkawy và cộng sự [14] đã sử dụng M-FNN (Multilayer Feedforward Neural Network) với sai số vị trí khớp và vận tốc làm đầu vào (input), đồng thời ước tính mômen xoắn khớp bên ngoài mà không cần sử dụng động lực học robot (nhưng cũng yêu cầu điều chỉnh ngưỡng va chạm thủ công cho từng khớp), trong khi [15] xem xét các mômen khớp đo được và vectơ lực hấp dẫn của tay máy. Tuy nhiên, các phương pháp này chỉ được thực nghiệm với các chuyển động hai và ba khớp của một tay máy robot 7-DoF. Tác giả Z. Zhang và các cộng sự [19] huấn luyện một FNN với các tín hiệu đầu vào 36 chiều được trích xuất từ cả các đặc điểm miền thời gian và tần số của các phép đo cảm biến mômen xoắn khớp. Đối với các robot có hình dạng đầy đủ giống như con người, nhóm nghiên cứu của K. Narukawa [11] đề xuất cải tiến kỹ thuật máy vectơ hỗ trợ một lớp (OC-SVM), trong đó đầu vào được coi là vectơ điểm mômen của robot hình người và các phép đo lực và mômen xoắn cho từng chi.

Gần đây, Y. J. Heo và các cộng sự [6] đề xuất phương pháp phát hiện nhanh va chạm của cánh tay robot 6-DoF sử dụng mạng neural tích chập một chiều (1D CNN) với kích thước là $W \times 66$ (với W biểu thị kích thước cửa sổ thời gian) trong đó có các mômen xoắn khớp bên ngoài ước tính làm đầu vào; đầu ra là kết quả phát hiện va chạm dạng nhị phân $(True, False)$. K. M. Park và các cộng sự [13] tập trung nghiên cứu giải quyết vấn đề phát hiện va chạm của cánh tay robot Doosan M0609. Trong khi M. Czubenko và nhóm nghiên cứu [2] đề xuất kết hợp một số mạng neural khác nhau như AR, RNN, CNN-LSTM và MC-LSTM nhằm phát hiện chính xác va chạm của cánh tay robot CURA6. Tuy nhiên, hạn chế chung của các phương pháp do K. M. Park và M. Czubenko đề xuất là sự phụ thuộc vào ma sát của động cơ, đặc biệt là ma sát Coulomb.

Trong bài báo này, nhóm nghiên cứu của chúng tôi đề xuất áp dụng hai phương pháp học có giám sát là SVMR (Support Vector Machine Regression) và 1D CNN (1D Convolutional Neural Network) để phát hiện va chạm của cánh tay robot CURA6 trên cơ sở các phép đo dòng điện cùng với mô hình động lực học của robot. Hướng nghiên cứu của chúng tôi có ưu điểm là không yêu cầu bất kỳ cảm biến bên ngoài bổ sung nào khác, so với [19] yêu cầu cảm biến mômen xoắn khớp tại các khớp. Hai phương pháp của chúng tôi không yêu cầu bù ma sát. Chi phí tính toán trong phương pháp đề xuất ít hơn đáng kể cho cả huấn luyện và suy luận (đầu vào 4 chiều cho SVMR và 6 chiều cho 1D CNN) trong khi phương pháp của Y. J. Heo [6] có đầu vào 66 chiều.

III. ĐỘNG LỰC HỌC ROBOT

1. Cấu tạo của cánh tay robot

Các bộ phận chủ yếu của một robot CURA6 bao gồm: đế, các khớp (Joint) quay hoặc tịnh tiến, các liên kết (Link) giữa các khớp và bàn tay robot còn được gọi khác là khâu chấp hành cuối (End of Effect). Cánh tay robot CURA6 của Intema có đặc trưng nhỏ gọn với trọng tải 5000 g, bán kính hoạt động rộng 1200 mm. Robot CURA6 có thể thực hiện được các nhiệm vụ phức tạp với yêu cầu tính chính xác cao như thao tác xử lý, tiếp xúc đối với những vật có đặc tính dễ biến dạng hoặc dễ rách, vỡ trong những điều kiện vận hành với yêu cầu kỹ thuật rất khắt khe.

Bảng I thể hiện bảng tham số động học D-H (Denavit

Bảng I
BẢNG THAM SỐ D-H CỦA ROBOT CURA6

i	Khoảng cách a_i	Góc quay α_i	Khoảng cách d_i	Mômen τ_i	Góc quay θ_i
0	0	$\frac{\pi}{2}$	0.105	τ_0	θ_0
1	0.4	0	-	τ_1	θ_1
2	0.4	0	-	τ_2	θ_2
3	0	$\frac{\pi}{2}$	0.220	τ_3	θ_3
4	0	$-\frac{\pi}{2}$	0.200	τ_4	θ_4
5	0	0	0.140	τ_5	θ_5

- Hartenberg) của robot CURA6 [2]. Ký hiệu trong bảng I như sau:

a_i là tham số khoảng cách tính từ trục Z_{i-1} tới trục Z_i đo dọc theo trục X_i.

α_i là tham số góc quay trục Z_{i-1} xung quanh trục X_i tới song song hoặc trùng với trục Z_i.

d_i là tham số khoảng cách giữa 2 trục X_{i-1} và trục X_i đo dọc theo trục Z_{i-1}.

θ_i là tham số góc quay của trục X_{i-1} xung quanh trục Z_{i-1} tới song song với trục X_i.

2. Phương trình động lực học

Phương trình động lực học được định nghĩa cho robot CURA6 là:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + \tau_F = \tau_m \tag{1}$$

với $q(t) = [q_0(t), q_1(t), ..., q_{n-1}(t)] \in \mathbb{R}^n$ bao gồm vị trí của mỗi khớp (với $n = 6$), $\dot{q}(t) \in \mathbb{R}^n$ và $\ddot{q}(t) \in \mathbb{R}^n$ lần lượt biểu thị vectơ vận tốc và gia tốc. $M(q) \in \mathbb{R}^{n\times n}$ là ma trận quán tính robot, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ là ma trận Coriolis, $g(q) \in \mathbb{R}^n$ là lực hấp dẫn, $\tau_F \in \mathbb{R}^n$ biểu thị lực ma sát và $\tau_m \in \mathbb{R}^n$ là vectơ mômen khớp. Lưu ý: $M(q)$ là ma trận đối xứng; $\tau_m = K_i i_m$ với $K_i \in \mathbb{R}^{n\times n}$ là ma trận khuếch đại và $i_m \in \mathbb{R}^n$ là dòng điện động cơ. Khi va chạm xảy ra, phương trình (1) trở thành:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + \tau_F = \tau_m + \tau_{ext} \tag{2}$$

với $\tau_{ext} \in \mathbb{R}^n$ là mômen xoắn khớp ngoài do va chạm hoặc lực bên ngoài khác tác động lên robot.

Ma trận $\dot{M}(q) - 2C(q,\dot{q})$ là đối xứng xiên, với $\dot{M}(q)$ là ký hiệu đạo hàm của $M(q)$ theo thời gian (chi tiết trong [10]).

$$\dot{M}(q) = C(q,\dot{q}) + C^T(q,\dot{q}) \tag{3}$$

Động lượng tổng quát của robot xác định như sau:

$$\rho = M(q)\dot{q} \in \mathbb{R}^n \tag{4}$$

Đạo hàm của ρ theo thời gian được biểu diễn là:

$$\dot{\rho} = \dot{M}(q)\dot{q} + M(q)\ddot{q} \tag{5}$$

hay

$$\dot{\rho} = \tau_m + \tau_{ext} - \tau_F + C^T(q,\dot{q})\dot{q} - g(q) \tag{6}$$

Ký hiệu $\beta(q,\dot{q}) = g(q) - C^T(q,\dot{q})\dot{q}$ và ma trận $K_O = diag(k_{O,i}) \in \mathbb{R}^{n\times n}$, động lực học của bộ quan sát động lượng được định nghĩa như sau:

$$\dot{\hat{\rho}} = \tau_m - \tau_F - \hat{\beta}(q,\dot{q}) + r_m \tag{7}$$

và

$$\dot{r}_m = K_O\left(\dot{\rho} - \dot{\hat{\rho}}\right) \tag{8}$$

Đầu ra $r_m(t) \in \mathbb{R}^n$ của bộ quan sát động lượng được mô tả như sau [13]:

$$r_m = K_O\left(\rho(t) - \int_0^t \left(\tau_m - \tau_F - \hat{\beta} + r_m\right) ds - \rho(0)\right) \tag{9}$$

với $\rho = \hat{M}(q)\dot{q}$ và $\hat{\beta} = \hat{g}(q) - \hat{C}^T(q,\dot{q})\dot{q}$, trong đó $\hat{M}$, $\hat{C}$ và $\hat{g}$ biểu thị các tham số mô hình. Trong các điều kiện lý tưởng khi $\hat{M} = M$ và $\hat{\beta} = \beta$, mối quan hệ động lực học giữa $r_m(t)$ và mômen xoắn khớp ngoài τ_{ext} là:

$$\dot{r}_m = K_O(\tau_{ext} - r_m) \tag{10}$$

sao cho $r_m(t)$ đóng vai trò ước tính cho τ_{ext}.

Mômen xoắn khớp ngoài ước tính được so sánh theo từng thành phần với ngưỡng cố định do người dùng xác định $\epsilon_{cd} \in \mathbb{R}^n$, sao cho hàm xác định va chạm $cd : r_m(t) \to \{True, False\}$ được định nghĩa là:

$$cd(r_m(t)) = \begin{cases} True & if\ |r_m(t)| > \epsilon_{cd} \\ False & if\ |r_m(t)| \le \epsilon_{cd} \end{cases} \tag{11}$$

Hàm trả về kết quả dạng nhị phân cho biết có xảy ra va chạm hay không [3, 5].

IV. PHƯƠNG PHÁP ĐỀ XUẤT

1. Mô tả bộ dữ liệu

Có 15 lát (slices) chuyển động ngẫu nhiên của robot đã được chuẩn bị để làm dữ liệu huấn luyện, mỗi lát chứa 10000 mẫu (07 phút chuyển động). Tốc độ của mỗi khớp được bộ giải thiết kế an toàn nằm trong khoảng 25% tốc độ tối đa của động cơ.

Có 03 lát được thu thập mà robot không có tải trọng và 12 lát còn lại có tải (đơn vị tính bằng gam) từ danh sách sau {782, 1016, 1282, 2298, 2757, 4039}, với 02 lát cho mỗi tải (phạm vi tải xấp xỉ là từ 500 g đến 4000 g). Trong đó, dữ liệu học (90%) và dữ liệu xác thực (10%). Bộ dữ liệu này được tải về từ trang Gitlab (https://gitlab.com/intema-gdansk/cura6-dataset/-/tree/main).

Các tín hiệu cần được chuẩn hóa trong phạm vi [0,1] trước khi được xử lý làm đầu vào. Đầu ra không xét đến ma sát được mô tả bằng phương trình:

$$r = K_O\left(\rho\left(t\right) - \int_0^t \left(\tau_m - \hat{\beta} + r\right) ds - \rho\left(0\right)\right) \tag{12}$$

với

$$\dot{\hat{\rho}} = \tau_m - \hat{\beta}\left(q, \dot{q}\right) + r \tag{13}$$

và

$$\dot{r} = K_O\left(\dot{\rho} - \dot{\hat{\rho}}\right) \tag{14}$$

Trong các điều kiện lý tưởng, mối quan hệ động giữa $r\left(t\right)$ và mômen xoắn khớp ngoài τ_{ext} là:

$$\dot{r} = K_O\left(\tau_{ext} - \tau_F - r\right) \tag{15}$$

2. Phương pháp phát hiện va chạm

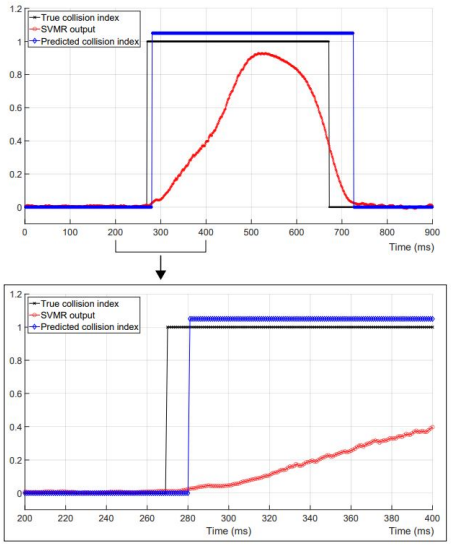
Trên cơ sở phương pháp SVMR [16] và 1D CNN [9, 12] là nền tảng, chúng tôi nghiên cứu phát triển mô hình phát hiện va chạm cho cánh tay robot CURA6. Quy ước: Một tín hiệu f-chiều thay đổi theo thời gian $s\left(t\right) \in \mathbb{R}^f$ được lấy mẫu tại khoảng thời gian lấy mẫu t_I trên một cửa sổ thời gian có kích thước t_W; số lượng mẫu là $N+1 = \frac{t_W}{t_I} + 1 \in \mathbb{N}$.

a) Phát hiện va chạm với SVMR

SVMR lấy giá trị tuyệt đối của đầu ra (12) làm đầu vào.

- Thiết kế vectơ đặc trưng: Thiết kế vectơ đặc trưng bao gồm chuyển tín hiệu thành dạng vectơ và gán nhãn cho các đặc trưng. Để lấy mẫu theo thời gian, chúng tôi đặt t_W = 80 ms và t_I = 8 ms. Sau khi lấy mẫu đầu ra của bộ quan sát động lượng và biến đổi thành ma trận $R \in \mathbb{R}^{n\times(N+1)}$, giá trị các hàng 1, 2, 3 và 4 của R tạo thành một vectơ duy nhất, dẫn đến kết quả vectơ đặc trưng $x\left(t\right)$ như sau:

$$x = \begin{bmatrix} r_{J_1} & r_{J_2} & r_{J_3} & r_{J_4} \end{bmatrix}^T \in \mathbb{R}^{4(N+1)} \tag{16}$$



Hình 2. Đầu ra của SVMR và chỉ số va chạm tương ứng

trong đó $r_{J_i} \in \mathbb{R}^{1(N+1)}$ biểu thị giá trị của hàng có thứ tự i của R. Lưu ý rằng hàng thứ i của R biểu thị đầu ra quan sát động lượng được lấy mẫu của khớp i. So với việc sử dụng tín hiệu của tất cả các khớp, kích thước của vectơ đặc trưng x này nhỏ hơn, giảm lượng tính toán cho SVMR.

- Huấn luyện SVMR: Vì hàm mũ yêu cầu tính toán nhiều hơn các hàm đa thức, nên chúng tôi sử dụng hàm SRQ [4] thay vì hàm RBF:

$$K_{SRQ}\left(x, z\right) = \frac{1}{\left(1 + \frac{\|x-z\|^2}{\sigma^2}\right)^2} \tag{17}$$

với các tham số $C = 1$, $\varepsilon = 0.02$ (dựa theo [16] và $\sigma = 3$.

- Điều chỉnh độ nhạy phát hiện va chạm: Đầu ra của SVMR được huấn luyện là một giá trị vô hướng được chuẩn hóa thành một đơn vị được mô tả như trong Hình 2 sau đó được đặt ngưỡng và chuyển thành chỉ số va chạm nhị phân tương ứng theo (11).

Giá trị đầu ra gần bằng 0 khi không có lực bên ngoài tác dụng và tăng khi mômen xoắn bên ngoài ước tính của khớp 1–4 (16) tăng và giảm khi mômen xoắn bên ngoài giảm. Mômen xoắn bên ngoài ước tính càng lớn và số lượng khớp bị tác động của ngoại lực càng nhiều thì giá trị đầu ra SVMR càng lớn.

b) Phát hiện va chạm với 1D CNN

- Thiết kế đầu vào mạng: Tín hiệu được lấy mẫu trong cửa sổ thời gian t_W=50 ms ở khoảng thời gian lấy mẫu t_I=10 ms và được chuyển đổi thành ma trận $R \in \mathbb{R}^{n\times(N+1)}$. Ma trận đặc trưng đầu vào mạng $Z\left(t\right)$ được định nghĩa là:

$$Z = \begin{bmatrix} | & | & & | \\ \left|r_{J_1}^T\right| & \left|r_{J_2}^T\right| & \cdots & \left|r_{J_6}^T\right| \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times(N+1)} \quad (18)$$

trong đó $\left|r_{J_i}\right| \in \mathbb{R}^{1\times(N+1)}$ biểu thị giá trị của hàng thứ i của ma trận R.

- Thiết kế kiến trúc: Kiến trúc CNN được mô tả trong Hình 3 chứa bốn lớp tích chập 1-D, một lớp làm phẳng, một lớp fully connected và một bộ lọc đầu ra. Hai trong số các lớp tích chập 1-D sử dụng cùng một phần đệm, trong khi hai lớp còn lại sử dụng phần đệm hợp lệ. Sau khi đi qua các lớp làm phẳng và được kết nối đầy đủ và bộ lọc đầu ra, giá trị chỉ số va chạm nhị phân được xuất ra, loại bỏ quá trình điều chỉnh ngưỡng phát hiện. Tất cả các lớp đều sử dụng hàm kích hoạt *relu*.

c) Lọc đầu ra

Chúng tôi thực hiện lọc đầu ra để giảm dương tính giả (cảnh báo sai). Va chạm chỉ được cảnh báo nếu đầu ra của phương pháp phát hiện va chạm liên tục là *True* trong khoảng thời gian t_c ms. Bộ lọc đầu ra loại bỏ các lỗi (cảnh báo sai với thời gian kéo dài trong phạm vi $1 - t_c$ ms) giúp phát hiện va chạm hiệu quả hơn. Chúng tôi đặt t_c=3 ms cho phương pháp SVMR và t_c=4 ms cho CNN.

V. THỬ NGHIỆM VÀ KẾT QUẢ

Thử nghiệm của chúng tôi dùng *PRAUC* (Area Under the Precision-Recall Curve) làm tiêu chí để đánh giá hiệu suất phát hiện va chạm. *Precision* và *Recall* được định nghĩa như sau:

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

và

$$Recall = \frac{TP}{TP + FN} \quad (20)$$

với *TP* biểu thị số lượng dương tính thật, *FP* biểu thị số lượng dương tính giả và *FN* biểu thị số lượng âm tính giả (va chạm được coi là dương tính). *Precision* cao thể hiện tỉ lệ dương tính giả thấp trong khi *Recall* cao thể hiện tỉ lệ âm tính giả thấp. *PRAUC* cao thể hiện cả *Precision* và *Recall* cao.

Trước tiên, chúng tôi đánh giá *PRAUC* của SVMR với các ngưỡng phát hiện khác nhau (α); đánh giá *PRAUC* của CNN với các tham số bộ lọc liên tục đầu ra khác nhau (t_c). Sau đó, đối với mỗi phương pháp, với ngưỡng cố định và tham số bộ lọc đầu ra cố định, chúng tôi đo độ trễ phát hiện trung bình và số lần phát hiện không thành công đối với các va chạm. Đối với SVMR, trong khi tăng dần ngưỡng phát hiện α từ α =-0.08 (tất cả các nhãn được dự đoán là 1) lên α =0.98 (tất cả các nhãn được dự đoán là 0), chúng tôi tính toán cặp *Precision*, *Recall* cho từng giá trị ngưỡng và vẽ đường cong như trong Hình 4.

Tiếp theo, đánh giá PRAUC của CNN với tham số bộ lọc đầu ra t_c thay đổi từ t_c = 1 ms (không áp dụng lọc) đến t_c = 619 ms (tất cả các nhãn được dự đoán là 1 được lọc thành 0). Chúng tôi lưu ý rằng do quá trình lọc liên tục chỉ có thể biến đổi các giá trị 1 được dự đoán thành 0 (điều ngược lại là không thể; không thể loại bỏ các phủ định sai hiện có), nên *Recall* tối đa không phải là 1. Như minh họa trong Hình 4, SVMR có PRAUC là 80.95% trong khi CNN có PRAUC là 82.12%. Như vậy, cả hai phương pháp học máy có giám sát đều phát hiện hiệu quả các va chạm và không va chạm đối với nhiều ngưỡng phát hiện và tham số bộ lọc đầu ra.

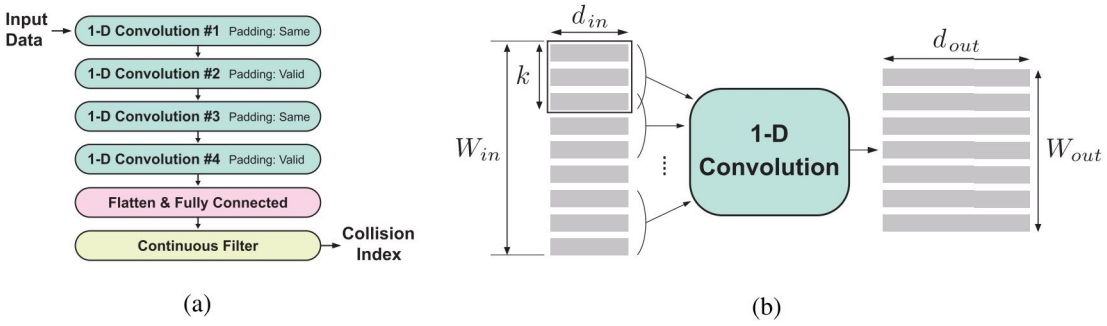
Chúng tôi lập trình thử nghiệm bằng ngôn ngữ Python. Các phương pháp đề xuất đã khắc phục được những trở ngại về sự không chắc chắn của các tham số trong mô hình động lực học và các hiệu ứng không được mô hình hóa (ma sát, nhiễu đo cảm biến).

VI. KẾT LUẬN VÀ ĐỊNH HƯỚNG NGHIÊN CỨU PHÁT TRIỂN

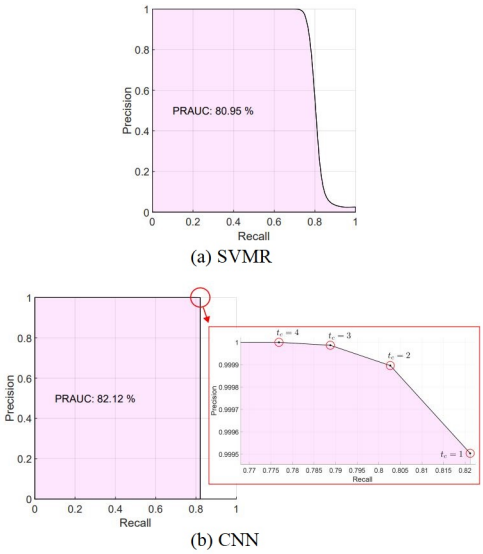
Phương pháp phát hiện va chạm dựa trên SVMR và 1D CNN do chúng tôi phát triển đã phát hiện va chạm của cánh tay robot 6-DoF CURA6 khá hiệu quả. Ưu thế nổi bật trong phương pháp đề xuất là chỉ yêu cầu phép đo các cảm biến dòng điện của động cơ cùng với mô hình động lực học của robot; không cần mô hình hóa hoặc xác định các mômen ma sát trong các khớp.

Qua thử nghiệm trên bộ dữ liệu của robot CURA6, phương pháp dựa trên SVMR chỉ yêu cầu điều chỉnh một tham số không đổi, trong khi phương pháp dựa trên 1D CNN không yêu cầu bất kỳ điều chỉnh nào. Chúng tôi so sánh và nhận thấy rằng phương pháp dựa trên 1D CNN phát hiện va chạm cho kết quả nhanh phương pháp dựa trên SVMR hơn khoảng 9 ms; trong khi phương pháp dựa trên SVMR yêu cầu về dữ liệu va chạm ít hơn để huấn luyện so với phương pháp dựa trên 1D CNN. Phương pháp dựa trên 1D CNN thích hợp với ngữ cảnh dữ liệu huấn luyện phong phú, còn phương pháp dựa trên SVMR phù hợp với ngữ cảnh dữ liệu hạn chế.

So với kết quả nghiên cứu trong [2] và [6], phương pháp của chúng tôi chiếm ưu thế trong ngữ cảnh xử lý tác động



Hình 3. Kiến trúc mạng CNN



Hình 4. (a) PRAUC của SVMR; (b) PRAUC của CNN

của các tải trọng khác nhau. Tuy nhiên, việc xác thực cho các tải trọng ngẫu nhiên không xác định vẫn là một bài toán nan giải. Hơn nữa, bài toán phát hiện va chạm của hai hay nhiều cánh tay robot rất phức tạp và cần được nghiên cứu chuyên sâu hơn. Đối với các robot cộng tác được sản xuất hàng loạt, một vấn đề thực tế khác là các quy trình cần thiết để phát hiện va chạm hiệu quả phải được nhân rộng trên quy mô lớn. Đây sẽ là những chủ đề mở cho nghiên cứu trong tương lai của chúng tôi.

VII. LỜI CẢM ƠN

Để hoàn thành bài báo này, chúng tôi xin chân thành cảm ơn sự hỗ trợ của nhóm nghiên cứu Intema, các tác giả M. Czubenko, K. M. Park và cộng sự.

TÀI LIỆU THAM KHẢO

[1] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical human–robot interaction", *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 41-48, Jan. 2016.

[2] M. Czubenko and Z. Kowalczuk, "A simple neural network for collision detection of collaborative robots", *Sensors*, vol. 21, no. 12, pp. 4235-4254, 2021.

[3] C. Gaz, E. Magrini, and A. De Luca, "A model-based residual approach for human-robot collaboration during manual polishing operations", *Mechatronics*, vol. 55, pp. 234-247, 2018.

[4] M. G. Genton, "Classes of kernels for machine learning: A statistics perspective", *J. Mach. Learn. Res.*, vol. 2, no. Dec, pp. 299–312, 2001.

[5] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification", *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, Dec. 2017.

[6] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, "Collision detection for industrial collaborative robots: A deep learning approach", *IEEE Robot. and Autom. Lett.*, vol. 4, no. 2, pp. 740–746, Apr. 2019.

[7] ISO 10218-1/2:2011, Robots and robotic devices - Safety requirements for industrial robots, *International Organization for Standardization*, Geneva, Switzerland, 2011.

[8] Y. Jiang, Y. Wu, X. Wu and B. Zhao, "Research on Collision Detection of Collaborative Robot using improved Momentum-based Observer", *IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, London, United Kingdom, pp. 1-6, May. 2023.

[9] Y. LeCun et al. "Gradient-based learning applied to document recognition", *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[10] K. M. Lynch and F. C. Park, Modern Robot, *Cambridge University Press*, 2017.

[11] K. Narukawa, T. Yoshiike, K. Tanaka, and M. Kuroda, "Real-time collision detection based on one class SVM for safe movement of humanoid robot", *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot*, pp. 791–796, Nov. 2017.

[12] D. Palaz, R. Collobert, and M. M. Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks", *Proc. Interspeech*, pp. 1766–1770, 2013.

[13] K. M. Park, J. Kim, J. Park, and F. C. Park, "Learning-Based Real-Time Detection of Robot Collisions Without Joint Torque Sensors", *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 103–110, Jan. 2021.

[14] A.-N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "Neural network design for manipulator collision detection based only on the joint position sensors", *Robotica*, vol. 38, no. 10, pp. 1737–1755, 2020.

[15] A.-N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, “Human–robot collisions detection for safe human–robot interaction using one multi-input–output neural network”, *Soft Comput.*, vol. 24, no. 9, pp. 6687–6719, 2020.
[16] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression”, *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.
[17] M. Strohmayr, H. Wörn, and G. Hirzinger, “The DLR artificial skin Step I: Uniting sensitivity and collision tolerance”, *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1012–1018, 2013.
[18] V. Villani, F. Pini, F. Leali, and C. Secchi, “Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications”, *Mechatronics*, vol. 55, pp. 248–266, 2018.
[19] Z. Zhang, K. Qian, B. W. Schuller, and D. Wollherr, “An online robot collision detection and identification scheme by supervised learning and bayesian decision theory”, *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1144-1156, 2021.
[20] D. Zurlo, T. Heitmann, M. Morlock and A. De Luca, "Collision Detection and Contact Point Estimation Using Virtual Joint Torque Sensing Applied to a Cobot", *IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, pp. 7533-7539, 2023.

SƠ LƯỢC VỀ TÁC GIẢ

Nghiêm Văn Hưng tốt nghiệp Cử nhân ngành Công nghệ thông tin tại Trường Đại học Sư phạm Đà Nẵng năm 2008, Thạc sĩ ngành Hệ thống thông tin tại Học viện Kỹ thuật quân sự năm 2012. Nghiên cứu sinh ngành Hệ thống thông tin tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Hiện công tác tại Khoa Công nghệ thông tin, Trường Đại học Kỹ thuật – Hậu cần Công an nhân dân.
Lĩnh vực nghiên cứu: Đa phương tiện, Mô phỏng, Thực tại ảo, Hệ thống thông tin.
Điện thoại: 0946.555.189
E-mail: nghiemvanhung1985@gmail.com

Đặng Văn Đức nhận học vị Tiến sĩ năm 1996, nhận chức danh Phó Giáo sư năm 2002. Hiện công tác tại Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.
Lĩnh vực nghiên cứu: Đa phương tiện, GIS, Công nghệ phần mềm, Phân tích và thiết kế hệ thống thông tin.
Điện thoại: 0912.223.163
E-mail: dvduc@ioit.ac.vn

Nguyễn Văn Căn nhận học vị Tiến sĩ chuyên ngành Cơ sở toán học cho tin học năm 2016. Hiện công tác tại Trường Đại học Kỹ thuật – Hậu cần Công an nhân dân.
Lĩnh vực nghiên cứu: Công nghệ nhận dạng, Thị giác máy tính, Xử lý ảnh, Đa phương tiện, Học máy, An ninh mạng, An toàn thông tin.
Điện thoại: 0986.919.333
E-mail: nguyenvancan@gmail.com

Trịnh Hiền Anh tốt nghiệp Thạc sĩ ngành Công nghệ phần mềm tại Đại học Công nghệ-ĐHQG Hà Nội năm 2011. Nghiên cứu sinh tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Hiện công tác tại Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.
Lĩnh vực nghiên cứu quan: Đa phương tiện, Công nghệ mô phỏng, Thực tại ảo, Hệ thống thông tin.
Điện thoại: 0989.197.288
E-mail: hienanh@ioit.ac.vn

Thuật toán song song khai thác nhanh các mẫu trọng số hữu ích phổ biến từ cơ sở dữ liệu định lượng động

Lê Hoàng Bình Nguyên[1], Nguyễn Duy Hàm[2], Nguyễn Thị Hồng Minh[3]
[1] Trường Cao đẳng An ninh mạng iSPACE, TP. Hồ Chí Minh, Việt Nam
[2] Trường Đại học Sài Gòn, TP. Hồ Chí Minh, Việt Nam
[3] Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội, Việt Nam

Tác giả liên hệ: Nguyễn Duy Hàm, ndham@sgu.edu.vn
Ngày nhận bài: 29/07/2023, ngày sửa chữa: 10/09/2023, ngày duyệt đăng: 26/09/2023
Định danh DOI: 10.32913/mic-ict-research-vn.v2023.n2.1238

Tóm tắt: Trong những năm gần đây, bài toán khai thác các mẫu trọng số hữu ích phổ biến (FWUP) đang được quan tâm nghiên cứu. Đây là một là một biến thể của bài toán khai thác mẫu. Một số phương pháp giải quyết khá hiệu quả bài toán này đã được đề xuất. Tuy nhiên, trong thực tế, khi dữ liệu có kích thước lớn và trọng số các mục thường xuyên thay đổi dẫn đến các thuật toán xử lý tốn nhiều thời gian trong quá trình khai thác mẫu. Nếu tận dụng khả năng tính toán song song của các hệ thống tính toán, hoặc của các bộ vi xử lý đa lõi (multi-core) thì có thể cải thiện được thời gian tính toán của các thuật toán. Trong bài báo này chúng tôi giới thiệu một giải pháp song song hoạt động trên bộ xử lý đa lõi, đặt tên là pdFWUNL, để khai thác các mẫu trọng số hữu ích phổ biến từ cơ sở dữ liệu định lượng động có trọng số các mục thay đổi. Kết quả thực nghiệm cho thấy thời gian khai thác mẫu của thuật toán song song pdFWUNL của chúng tôi hiệu quả hơn phương pháp tuần tự tốt nhất hiện có.

Từ khóa: *mẫu trọng số hữu ích phổ biến, cơ sở dữ liệu định lượng động, dữ liệu lớn, tính toán song song*

Title: **A Parallel Algorithm for Fast Mining Frequent Weighted Ultility Patterns from Dynamic Quantitative Databases**

Abstract: In recent years, the problem of mining frequent weighted utility patterns has been receiving research attention. This is a variation of the pattern mining problem. Many effective methods have been proposed to solve this problem. However, when the data is extensive and the weights of items change frequently, the algorithms take much time during the mining process. If we take advantage of the parallel computing capabilities of computing systems or multi-core processors, we can improve the mining time of algorithms. This paper presents a parallel solution operating on multi-core processors, named pdFWUNL, to exploit frequent weighted utility patterns from dynamic quantitative databases with variable item weight. The experimental results show that the mining time of our parallel algorithm, pdFWUNL, is more efficient than the best available sequential method.

Keywords: *frequent weighted utility patterns, dynamic quantitative database, big data, parallel computation*

I. GIỚI THIỆU

Công nghệ đang ngày càng phát triển và tác động đến tất cả các lĩnh vực của đời sống xã hội, mang lại những hiệu quả tích cực. Cùng với đó là lượng lớn dữ liệu đang liên tục được tạo ra trong mỗi phút giây. Những dữ liệu này có thể được tạo ra bởi con người hoặc máy móc, và được thu thập bằng nhiều phương thức, công nghệ khác nhau. Có thể kể đến như dữ liệu bán hàng, mạng xã hội, nhật ký hệ thống, dữ liệu đo đạc hiện tượng thiên nhiên... Nếu khai thác được những thông tin hữu ích từ dữ liệu này sẽ mang lại rất nhiều lợi thế, đặc biệt là trong kinh doanh, bán hàng, trong các hệ thống hỗ trợ ra quyết định, các hệ thống thông minh...

Trong lĩnh vực khai thác dữ liệu, khai thác mẫu là một trong những bài toán quan trọng [1]. Nhiệm vụ của khai thác mẫu là khám phá tất cả các mẫu phổ biến xuất hiện trong cơ sở dữ liệu (CSDL), ví dụ như tập sản phẩm được mua nhiều nhất bởi khách hàng. Khai thác mẫu trọng số hữu ích phổ biến (Frequent Weighted Utility Pattern - FWUP) [2, 3] là một biến thể của khai thác mẫu với mục tiêu tìm

ra các mẫu vừa xuất hiện thường xuyên, vừa có lợi ích và và tầm quan trọng (trọng số) cao. Đã có nhiều thuật toán được đề xuất để giải quyết hiệu quả bài toán này như MBiS [4], WUN-Miner [5], SWUNL-FWUP [6] trên CSDL định lượng tĩnh và dFWUT [7], dFWUNL [7] trên CSDL định lượng động. CSDL định lượng động (dQDB) là CSDL mà trọng số của các mục có thể thay đổi trong quá trình giao dịch chứ không cố định. Tuy nhiên, các thuật toán khai thác FWUP nêu trên đều chỉ chạy trên các bộ xử lý đơn lõi. Điều này có thể làm hạn chế hiệu suất, không thể mở rộng trên bộ xử lý đa lõi, cũng như gặp khó khăn trong việc khai thác trên các CSDL lớn, dữ liệu được tích luỹ nhanh chóng theo thời gian.

CSDL định lượng động xuất hiện nhiều trong thực tế, trọng số của mỗi mục dữ liệu có thể là lợi nhuận, giá cả hay lợi ích… của mặt hàng, các giá trị này có thể thay đổi theo từng giao dịch. Ví dụ giá của mặt hàng có thể thay đổi theo số lượng mua (khi mua với số lượng nhiều có thể được giá thấp hơn), có thể thay đổi theo thời gian (giá mùa hè có thể khác giá mùa đông với những mặt hàng điện lạnh)… Điều này dẫn đến bài toán có tính thực tiễn cao là khai thác mẫu phổ biến từ CSDL định lượng động với trọng số thay đổi.

Trong bài báo này, chúng tôi thực hiện một số nội dung sau: (1) Giới thiệu bài toán khai thác FWUP từ CSDL định lượng động; (2) Đề xuất một phương pháp song song, đặt tên là pdFWUNL để khai thác FWUP từ CSDL định lượng động, với khả năng chia tính toán thành nhiều tác nhiệm có thể thực hiện đồng thời, nhằm giảm chi phí xử lý; (3) Đề xuất cơ chế cân bằng tải động để phân phối công việc giữa các tiến trình khi có bộ xử lý nhàn rỗi trong quá trình tính toán; và (4) Tiến hành thực nghiệm, đánh giá và so sánh thuật toán đề xuất pdFWUNL với thuật toán gốc dFWUNL. Kết quả thử nghiệm cho thấy thuật toán pdFWUNL tốt hơn phiên bản gốc về thời gian thực hiện đối với tất cả các bộ dữ liệu thử nghiệm.

Nội dung bài viết được tổ chức như sau: Phần II giới thiệu các nghiên cứu liên quan. Phần III trình bày một số khái niệm và kiến thức cơ bản. Phần IV mô tả thuật toán dFWUNL và trình bày chi tiết về thuật toán song song được đề xuất. Phần V mô tả kết quả thực nghiệm. Cuối cùng, Phần VI rút ra kết luận và hướng nghiên cứu tiếp theo.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Cơ sở dữ liệu định lượng thường xuất hiện trong dữ liệu bán hàng. Điểm đặc trưng của loại dữ liệu này là mỗi giao dịch lưu giữ số lượng các sản phẩm được mua và mỗi sản phẩm có một trọng số khác nhau. Trọng số này có thể là giá cả, lợi nhuận, hoặc tầm quan trọng của sản phẩm đó trong CSDL. Đã có nhiều hướng tiếp cận để khai thác thông tin có giá trị từ CSDL định lượng. Hướng nghiên cứu đầu tiên đó là khai thác các mẫu lợi ích cao (High Utility Pattern- HUP) do Yao và các đồng sự đề xuất [8]. Khai thác HUP tập trung vào việc tìm kiếm các mẫu có giá trị hữu ích không nhỏ hơn một ngưỡng hữu ích cho trước. Sau đó, nhiều thuật toán giải quyết bài toán này đã được đề xuất như Two-Phase [9], UP-Growth+ [10], HUI-Miner [11], EFIM [12], và các biến thể của HUP như khai thác HUP đóng [13], top-k HUP [14], HUP trung bình [15],...

Khai thác HUP chỉ quan tâm đến lợi ích của các mẫu, trong khi tần số xuất hiện của các mẫu cũng đóng vai trò quan trọng. Do đó, Khan và đồng sự đã đề xuất một hướng nghiên cứu mới đó là khai thác luật kết hợp trọng số hữu ích [2]. Tác giả đã đưa ra độ đo mới là trọng số hữu ích giao dịch, kết hợp hai yếu tố là tần số và tầm quan trọng để trích xuất được các FWUP, đồng thời phát triển một thuật toán dựa trên phương pháp Apriori. Tiếp đó, Vo và các đồng sự [3] đề xuất cấu trúc MWIT-tree với chỉ một lần quét CSDL để khai thác FWUP. Nguyen và các đồng sự [4] đã đề xuất cấu trúc MBiS để tối ưu việc lưu trữ tidset nhằm tăng hiệu quả khai thác. Tiếp theo, Bui và các đồng sự [5] đề xuất cấu trúc WUN-set cùng với thuật toán WUN-Miner, và sau đó, Nguyen và các đồng sự [6] phát triển thuật toán FWUP-SWUNL sử dụng cấu trúc SWUN-list và đề xuất các định lý tăng tốc quá trình khai thác. Kết quả thực nghiệm cho thấy FWUP-SWUNL là thuật toán khai thác FWUP tốt nhất hiện nay.

Các nghiên cứu khai thác FWUP ở trên đều tập trung vào CSDL định lượng tĩnh, trọng số các mục là cố định. Tuy nhiên trên thực tế, tầm quan trọng của mỗi sản phẩm chịu ảnh hưởng của nhiều yếu tố khác nhau như chiến lược giảm giá, thanh lý mặt hàng, thay đổi hành vi người dùng,... Điều này dẫn đến trọng số của từng sản phẩm cần được điều chỉnh cho phù hợp. Để giải quyết vấn đề này, Nguyen và các đồng sự [7] đã phát biểu bài toán và đề xuất phương pháp (thuật toán dFWUNL) khai thác FWUP từ CSDL định lượng động, trong đó trọng số của các mục có thể thay đổi theo thời gian. Thuật toán dFWUNL mặc dù giải quyết khá hiệu quả bài toán nhưng vẫn còn hạn chế trên các CSDL lớn, đặc biệt về mặt thời gian.

Để giải quyết thách thức về thời gian tính toán khi dữ liệu lớn, ý tưởng song song hoá các thuật toán khai thác dữ liệu là một hướng tiếp cận. Một trong những mô hình phổ biến để áp dụng tính toán song song là sử dụng bộ xử lý đa lõi. Thiết kế đa lõi, công nghệ khá phổ biến hiện nay trong sản xuất chip xử lý, cho phép thực hiện nhiều tác nhiệm đồng thời để cải thiện hiệu suất của các quá trình tính toán. Tuy nhiên cần có thuật toán có khả năng phân rã các tiến trình và điều phối phù hợp. Đối với bài toán khai thác mẫu, đã có khá nhiều nghiên cứu sử dụng kiến trúc đa lõi và đạt được hiệu quả đáng kể [16–18].

III. CÁC KHÁI NIỆM CƠ BẢN

Một CSDL định lượng động là một bộ bao gồm ba thành phần (T, I, DW). Trong đó $T = \{t_1, t_2, ..., t_m\}$ là tập các giao dịch định lượng với m là số lượng giao dịch, $I = \{i_1, i_2, ..., i_n\}$ là tập mục với n là số lượng mục, và $DW = \{dw_1, dw_2, ..., dw_p\}$ là các tập trọng số với p là số lượng lô (batch). Mỗi giao dịch t_k có dạng $t_k = \{x_{k1}, x_{k2}, ..., x_{kn}\}$, $k = 1..m$, trong đó x_{ki} là số lượng của mục I trong giao dịch t_k (có thể hiểu như số lượng mặt hàng thứ i được mua trong giao dịch t_k), và tương ứng với một $dw_x \in DW$. Mỗi lô $dw_x = \{w_1, w_2, ..., w_n\}$, $x = 1..p$, là một tập trọng số của các mục trong I. Điểm khác biệt của CSDL định lượng động đó là trọng số của một mục có thể thay đổi trong các giao dịch khác nhau dựa trên tầm quan trọng của mục đó tại các thời điểm khác nhau. Mỗi lô giao dịch có thể có một hoặc nhiều giao dịch và tương ứng với một tập trọng số.

Bảng I mô tả về một CSDL định lượng động. CSDL có 5 giao dịch $T = \{t_1, t_2, t_3, t_4, t_5\}$ và năm mục $I = \{A, B, C, D, E\}$ với 2 lô $DW = \{dw1, dw2\}$. Trọng số của các mục trong giao dịch t_1, t_2, t_3 là $dw_1 = \{0.3, 0.5, 0.2, 0.4, 0.7\}$, và các giao dịch t_4, t_5 có trọng số là $dw_2 = \{0.4, 0.3, 0.4, 0.2, 0.3\}$.

Bảng I
VÍ DỤ VỀ CSDL ĐỊNH LƯỢNG ĐỘNG (dQDB$_E$)

Bảng giao dịch của CSDL						
Giao dịch	*A*	*B*	*C*	*D*	*E*	**Trọng số**
t_1	1	0	2	3	1	dw_1
t_2	0	2	1	0	2	
t_3	2	0	1	0	1	
t_4	1	0	3	2	0	dw_2
t_5	3	1	1	2	1	

Bảng trọng số của CSDL					
Trọng số	*A*	*B*	*C*	*D*	*E*
dw_1	0.3	0.5	0.2	0.4	0.7
dw_2	0.4	0.3	0.4	0.2	0.3

Định nghĩa 1. [7] Giá trị trọng số hữu ích giao dịch (twu) của một giao dịch t_k với tập trọng số $dw_p \in DW$ được xác định như sau:

$$twu\,(t_k) = \frac{\sum_{i_j \in t_k} w_{pi_j} \times x_{ki_j}}{|t_k|}$$

Trong đó, $w_{pi_j} \in dw_p$ là trọng số của mục i_j trong giao dịch t_k, x_{ki_j} là số lượng của mục i_j trong t_k, và $|t_k|$ là số lượng các mục trong t_k.

Ví dụ 1. Giao dịch t_2 trong dQDB$_E$ có giá trị twu được tính như sau:

$$twu(t_2) = (0.5 \times 2 + 0.4 + 0.7 \times 2)/3 \approx 0.87$$

Tương tự, giá trị twu của tất cả các giao dịch trong dQDB$_E$ được tính như trong Bảng II.

Bảng II
GIÁ TRỊ twu CỦA TẤT CẢ GIAO DỊCH TRONG (dQDB$_E$)

Giao dịch	**Trọng số**	**Công thức tính**	**twu**
t_1	dw_1	$(0.3 \times 1 + 0.2 \times 2 + 0.5 \times 3 + 0.7)/4$	0.65
t_2		$(0.5 \times 2 + 0.4 + 0.7 \times 2)/3$	0.87
t_3		$(0.3 \times 2 + 0.2 + 0.7)/3$	0.50
t_4	dw_2	$(0.4 + 0.4 \times 3 + 0.2 \times 2)/3$	0.67
t_5		$(0.4 \times 3 + 0.3 + 0.4 + 0.2 \times 2 + 0.3)/5$	0.52
Tổng giá trị twu ($sumtwu$)			3.21

Định nghĩa 2. [7] Gọi $sumtwu = \sum_{t_k \in T} twu(t_k)$ là tổng giá trị twu của các giao dịch trong dQDB. Độ hỗ trợ trọng số hữu ích (wus) của một mẫu X được xác định như sau:

$$wus\,(X) = \frac{\sum_{t_k \in t(X)} twu\,(t_k)}{sumtwu}$$

Trong đó, $t(X)$ là tập các giao dịch trong CSDL có chứa mẫu X.

Ví dụ 2. Giá trị wus của mẫu AE được tính như sau:

$$wus\,(AE) = \frac{twu\,(t_1) + twu\,(t_3) + twu\,(t_5)}{sumtwu} = \frac{0.65 + 0.5 + 0.52}{3.21} \approx 0.52$$

Cho CSDL định lượng động và một ngưỡng hỗ trợ trọng số hữu ích tối thiểu (ký hiệu là $minwus$), bài toán khai thác FWUP là tìm tất cả các mẫu X có giá trị $wus(X)$ lớn hơn hoặc bằng $minwus$.

IV. THUẬT TOÁN PDFWUNL KHAI THÁC MẪU PHỔ BIẾN TRÊN CSDL ĐỊNH LƯỢNG ĐỘNG

1. Thuật toán dFWUNL

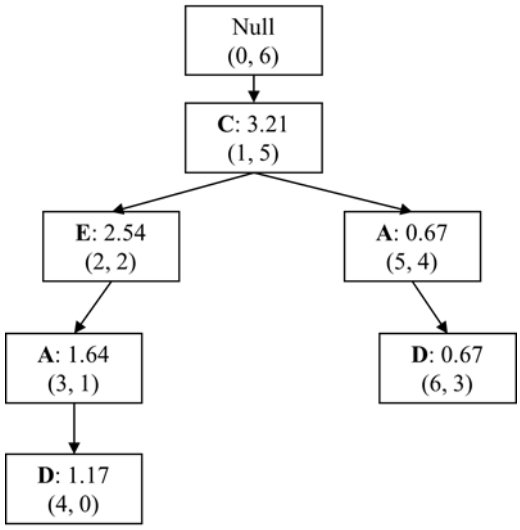
Phần này giới thiệu về thuật toán dFWUNL [7] và các thành phần của nó, đây là cơ sở để đề xuất thuật toán song song hóa của chúng tôi trong phần sau. Thuật toán dFWUNL khai thác các FWUP từ dQDB được đề xuất bởi Nguyen và đồng sự sử dụng cấu trúc WUNList, là mở rộng của cấu trúc N-list [19], được tạo thành từ cấu trúc WUN-tree. Về cơ bản, thuật toán dFWUNL gồm 3 giai đoạn:

1) Giai đoạn đầu tiên quét dQDB để xây dựng WUN-tree, tìm và sắp xếp giảm dần các 1-FWUP theo wus. Các 1-FWUP này được lưu trữ trong I_1. WUN-tree là cấu trúc cây, mỗi nút lưu trữ năm giá trị $name$, pre, $post$, $weight$, và $children$ tương ứng với tên của mục đại diện cho nút, thứ tự của nút khi duyệt pre-order (tiền thứ tự) và post-order (hậu thứ tự), tổng twu của các giao dịch đi qua nút này, và tập các nút con của nút đó.
2) Giai đoạn thứ hai tạo WUNList của các 1-FWUP trong I_1 bằng cách trích xuất thông tin từ WUN-tree. Cấu trúc dữ liệu WUNList của một mục X, ký hiệu là $wl(X)$ là một dãy các WUN-codes

$\{(pre_1, post_1, weight_1), (pre_2, post_2, weight_2), ..., (pre_n, post_n, weight_n)\}$ của các nút đại diện cho mục X trên WUN-tree. WUNList được sắp xếp tăng dần theo giá trị pre $(pre_1 < pre_2 < ... < pre_n)$. Đồng thời, tiến hành tính wus của 2-pattern thông qua WUN-tree và lưu trữ trong ma trận $F2$.

3) Giai đoạn cuối cùng tiến hành tìm kiếm tất cả các FWUP. Thuật toán dFWUNL sử dụng cấu trúc set-enumerations-tree [20] kết hợp với các định lý để tăng tốc quá trình khai thác. Trong quá trình tạo set-enumerations-tree, các nhánh của cây sẽ có cùng tiền tố. Do đó, chúng ta có thể xác định được k-FWUP từ hai $(k-1)$-FWUP và tính nhanh wus thông qua việc giao các WUNList (WUNList intersection). Kết thúc thuật toán, ta thu được đầy đủ tất cả các FWUP thoả mãn ngưỡng $minwus$.

Ví dụ 3. Hình 1 mô tả WUN-tree được tạo thành từ $dQDB_E$ với ngưỡng $minwus = 0.5$. Mục B bị loại bỏ vì $wus(B) = 0.43 < minwus$. Nút đại diện cho mục C có WUN-codes là $(1, 5, 3.21)$. Từ WUN-tree, ta tạo được WUNList của các 1-FWUP. Ví dụ, WUNList của mẫu A là $wl(A) = \{(3, 1, 1.64), (5, 4, 0.67)\}$. WUNList của các 1-FWUP được sử dụng để khai thác các FWUP tiếp theo và cho hiệu quả vượt trội.



Hình 1. WUN-tree được tạo từ $dQDB_E$ với $minwus = 0.5$

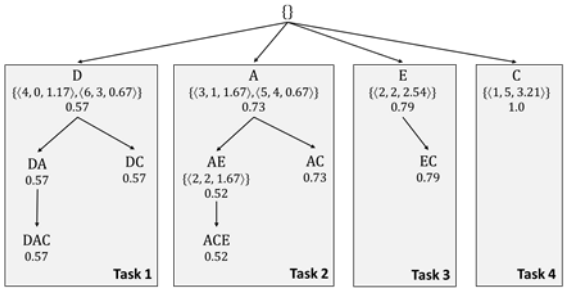
Thuật toán dFWUNL đã triển khai được việc khai thác mẫu trên CSDL định lượng động, tuy nhiên vẫn còn thách thức về thời gian khai thác trên các CSDL lớn và chưa có khả năng tận dụng sức mạnh của các hệ thống tính toán song song.

2. Đề xuất chiến lược song song cho thuật toán dFWUNL

Trong phần này, chúng tôi trình bày thuật toán có tên là pdFWUNL khai thác nhanh các FWUP trên dQDB. Thuật toán pdFWUNL là mở rộng của thuật toán dFWUNL [7], sử dụng kiến trúc bộ xử lý đa lõi để song song hoá quá trình tìm kiếm các mẫu trọng số hữu ích phổ biến.

Giai đoạn 3 của thuật toán dFWUNL tạo thành set-enumerations-tree với các nhánh khác nhau. Các nhánh này bắt đầu từ các 1-FWUP, sau đó được mở rộng dần. Quá trình khai thác trên các nhánh là hoàn toàn độc lập, không có sự phụ thuộc dữ liệu, tính toán lẫn nhau. Do đó, chúng ta có thể áp dụng chiến lược song song vào giai đoạn này. Thuật toán pdFWUNL áp dụng mô hình task-parallelism trên hệ thống bộ nhớ dùng chung. Mô hình này chia chương trình thành các tác vụ (task) riêng lẻ và sau đó tiến hành thực hiện song song các tác vụ độc lập. Thuật toán coi mỗi 1-FWUP trong I_1 là một task riêng biệt và phân phối các task này vào các lõi của bộ xử lý để khai thác độc lập. Mỗi lõi của bộ xử lý sau đó sẽ tiến hành tạo các cây con tương ứng với nút mà nó được chỉ định, sử dụng tìm kiếm theo chiều sâu (DFS).

Ví dụ 4. Hình 2 mô tả quá trình phân chia không gian tìm kiếm thành các công việc độc lập và tiến hành khai thác trên CSDL $dQDB_E$ với $minwus = 0.5$ của thuật toán pdFWUNL. Có bốn 1-FWUP trong I_1 nên có bốn task, và các task này được các lõi của bộ xử lý khai thác song song.



Hình 2. Phân chia không gian tìm kiếm của thuật toán pdFWUNL

Một khía cạnh quan trọng khi áp dụng khai thác song song đó là đảm bảo cân bằng tải. Làm sao phân bổ các công việc vào các bộ hoặc lõi xử lý để thời gian nhàn rỗi (không hoạt động) của các bộ hoặc lõi xử lý là ít nhất có thể. Thuật toán pdFWUNL sử dụng cơ chế cân bằng tải động áp dụng đối với các bộ xử lý đa lõi. Cụ thể như sau: 1-FWUP chưa được khai thác sẽ được đưa vào lõi xử lý khả dụng (đang nhàn rỗi) để tiến hành khai thác; nếu không có lõi xử lý nào, nó sẽ đợi cho đến khi có lõi xử lý thoả mãn để chỉ định 1-FWUP chưa xử lý tiếp theo.

Mã giả của pdFWUNL được hiển thị trong Thuật toán 1. Đầu tiên, thuật toán pdFWUNL tiến hành quét CSDL để tạo I_1, WUN-tree, WUNList của các mục trong I_1 và tính *wus* của 2-pattern lưu trữ trong ma trận F_2 (dòng 1-3). Tiếp theo, mỗi 1-FWUP là một task với nhiệm vụ tìm kiếm các FWUP theo nhánh của mục đó với thủ tục Find_FWUP (dòng 4-6).

Quá trình này được thực thi song song và liên tục được phân phối vào các lõi đang nhàn rỗi. Thủ tục Find_FWUP gồm các tham số $Prefix$, L, S, x và $level$ lần lượt là tiền tố chung, tập mục cuối cùng của các mẫu của lớp hiện tại, tập hợp các mục có thể tính nhanh *wus*, chỉ mục của mục đang xét và độ dài của mẫu. Trong thủ tục Find_FWUP, với độ dài ứng viên là 2 ($level = 2$), ta có thể tính nhanh các *wus* của 2-pattern này thông qua $F2$, nhờ đó nhanh chóng loại bỏ các mẫu không thoả mãn (dòng 10-22). Với các trường hợp khác ($level > 2$), ta tính WUNList của các ứng viên đó thông qua việc giao các WUNList để tìm ra FWUP (dòng 23-43). Thuật toán kết thúc khi không còn mẫu mới được tạo thành và ta thu được tất cả FWUP thoả mãn.

Thuật toán 1: Thuật toán pdFWUNL

```
   Input: CSDL định lượng động dQDB và ngưỡng minwus
   Output: FWUPs, tập tất cả các mẫu trọng số hữu ích phổ biến
1  Quét CSDL để tạo I1 và WUN-Tree
2  FWUPs ← I1
3  Duyệt WUN-tree để tạo WUNList của các mục trong I1, và tính wus
     của 2-pattern lưu trữ trong ma trận F2
4  for x ← |I1| − 1 downto 1 parallel do
5      Prefix2 ← {ix}, L2 ← ∅, S2 ← ∅
6      Find_FWUP(Prefix2, L2, S2, x, 2)
7  return FWUPs
8  Procedure Find_FWUP(Prefixk, Lk, Sk, x, level)
9      Prefixnext ← Prefixk
10     if level == 2 then
11         Prefixnext ← {ix}, Lnext ← ∅, Snext ← ∅
12         for y ← 0 to x − 1 do
13             wus(ixiy) ← F2[ix][iy]/sumtwu
14             if wus(ixiy) == wus(ix) then
15                 Snext ← Snext ∪ iy
16             else if wus(ixiy) ≥ minwus then
17                 wl(ixiy)  ← WUN-
                     List_intersect(wl(ix), wl(iy))
                   Lnext ← Lnext ∪ iy
18                 FWUPs ← FWUPs ∪ ixiy
19         if |Snext| > 0 then
20             Find_FWUP_sameWUS(Prefixnext, Snext)
21         if |Lnext| > 0 then
22             Find_FWUP(Prefixnext, Lnext, Snext, x, level+1)
23     else
24         for i ← |Lk| − 1 downto 1 do
25             Prefixnext ← Prefixnext ∪ Lk[i]
26             Lnext ← ∅, Snext ← Sk
27             for j ← 0 i − 1 do
28                 C ← Lk[j]
29                 wl(C)  ← WUN-
                     List_intersect(wl(Lk[i], wl(Lk[j])
30                 if wl(C) ≠ NULL then
31                     if wus(C) == wus(Lk[i]) then
32                         Snext ← Snext ∪ Lk[j]
33                     else
34                         Lnext ← Lnext ∪ C
35                         FWUPs ← FWUPs ∪ {Prefixk ∪
                             Lk[i] ∪ Lk[j]}
36             if |Snext| > 0 then
37                 Find_FWUP_sameWUS(Prefixnext, Snext)
38             if |Lnext| > 0 then
39                 Find_FWUP(Prefixnext, Lnext, Snext, x, level+
                     1)
40             remove last item of Prefixnext
41         Prefixnext ← Prefixnext ∪ Lk[0]
42         if |Sk| > 0 then
43             Find_FWUP_sameWUS(Prefixnext, Sk)
44 Procedure Find_FWUP_sameWUS(Prefix, S)
45     Sinh ra các tập con của S và lưu trữ vào Child
46     foreach c ∈ Child do
47         FWUPs ← FWUPs ∪ {Prefix ∪ c}
```

V. THỰC NGHIỆM

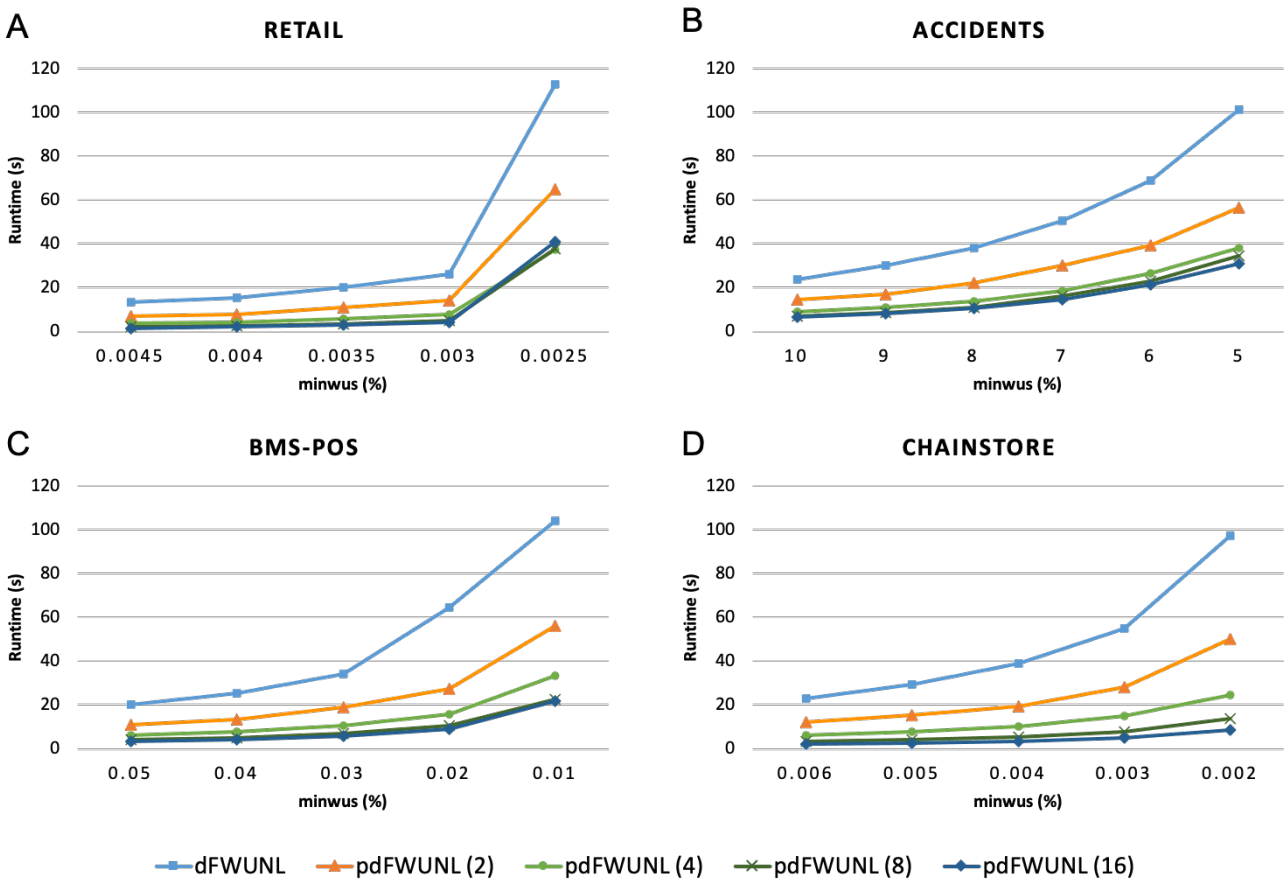
Phần này trình bày kết quả thực nghiệm thuật toán pdFWUNL và so sánh thời gian thực hiện với thuật toán dFWUNL trên một số bộ dữ liệu định lượng động khác nhau. Đồng thời, để kiểm tra khả năng mở rộng và tính hiệu quả của phương pháp song song, chúng tôi cũng thực nghiệm thuật toán pdFWUNL với số lượng lõi xử lý khác nhau, bao gồm: 2, 4, 8 và 16 lõi. Các thực nghiệm đều được tiến hành trên cùng hệ thống Dual CPU Intel Xeon Silver 4114 2.2 GHz (20 lõi), RAM 32GB, chạy hệ điều hành Windows 10. Các thuật toán được phát triển bằng ngôn ngữ C# trên nền tảng .NET Framework 4.7, sử dụng thư viện Task Parallel Library hỗ trợ lập trình song song.

Bảng III mô tả các bộ dữ liệu thực nghiệm bao gồm: Accident, BMS-POS, Retail, và Chainstore được lấy từ thư viện mã nguồn mở SPMF [21]. Để có được CSDL định lượng động và không mất tính tổng quát, chúng tôi đã chia mỗi CSDL thử nghiệm thành nhiều lô. Mỗi lô có số lượng giao dịch bằng nhau, ngoại trừ lô cuối cùng chứa các giao dịch còn lại. Mỗi lô giao dịch tương ứng với một bộ trọng số được tạo ngẫu nhiên trong khoảng [1,10].

Bảng III
CƠ SỞ DỮ LIỆU THỰC NGHIỆM

CSDL	Số lượng mục	Số lượng giao dịch	Số lượng giao dịch mỗi lô
Retail	16,470	88,162	10000
Accidents	468	340,183	5000
BMS-POS	1657	515597	50000
Chainstore	46,086	1,112,949	80000

Hình 3 hiển thị thời gian thực hiện của thuật toán dFWUNL và thuật toán pdFWUNL trên các bộ dữ liệu thực nghiệm với các ngưỡng minwus khác nhau. Kết quả cho thấy thuật toán pdFWUNL vượt trội hoàn toàn so với thuật toán dFWUNL trên tất cả các thực nghiệm với số lượng lõi khác nhau. Điều này là do thuật toán song song có thể tận dụng được sức mạnh của kiến trúc bộ xử lý đa lõi. Khi ngưỡng minwus càng nhỏ, sự chênh lệch về thời gian khai thác càng thể hiện rõ. Ví dụ, CSDL Accidents với



Hình 3. Thời gian khai thác của thuật toán dFWUNL và pdFWUNL

minwus = 6%, thời gian khai thác của thuật toán dFWUNL là 69s, còn pdFWUNL(2) là 39.3s và pdFWUNL (8) là 23s. Bên cạnh đó, thuật toán pdFWUNL cũng cho thấy sự hiệu quả của mình khi số lượng lõi xử lý tăng lên. Ví dụ, CSDL Chainstore với *minwus* = 0.002%, thuật toán pdFWUNL chạy trên hai lõi (pdFWUNL(2)) mất 50s, còn chạy trên 16 lõi (pdFWUNL(16)) chỉ mất 8.6s. Tuy nhiên, sự gia tăng này là không tuyến tính và khi tăng lõi lên một mức độ nhất định, thời gian khai thác không giảm đi, thậm chí còn tăng lên. Điều này là cho thấy, thuật toán pdFWUNL còn phụ thuộc vào sự phân bố dữ liệu. Ví dụ, CSDL Retail với *minwus* = 0.0025%, thuật toán pdFWUNL chạy trên 16 lõi trong thời gian là 40.2s, trong khi chạy trên 8 lõi chỉ mất 38.6s.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo đã giới thiệu một phương pháp sử dụng bộ xử lý đa lõi để song song hoá thuật toán dFWUNL khai thác FWUP từ CSDL định lượng động. Thuật toán song song được đề xuất có tên là pdFWUNL sử dụng cơ chế Task-parallel và cân bằng tải động để tận dụng sức mạnh tính toán của bộ xử lý. Các kết quả thực nghiệm cho thấy thuật toán song song vượt trội hoàn toàn về thời gian khai thác so với thuật toán tuần tự dFWUNL.

Trong thời gian tới, chúng tôi sẽ tiếp tục nghiên cứu các phương pháp để song song hoá hai giai đoạn còn lại của thuật toán pdFWUNL và tối ưu cơ chế cân bằng tải. Cùng với đó, nghiên cứu các phương pháp tính toán phân tán hoặc song song hoá trên các hệ thống khác để áp dụng khai thác FWUP trên những CSDL rất lớn.

TÀI LIỆU THAM KHẢO

[1] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” vol. 22. ACM, 1993, pp. 207–216.

[2] M. S. Khan, M. Muyeba, and F. Coenen, “A weighted utility framework for mining association rules.” IEEE, 2008, pp. 87–92.

[3] B. Vo, B. Le, and J. J. Jung, “A tree-based approach for mining frequent weighted utility itemsets.” Springer, 2012, pp. 114–123.

[4] N. D. Ham, V. D. Bay, N. T. H. Minh, and T.-P. Hong, “Mbis: an efficient method for mining frequent weighted utility itemsets from quantitative databases,” *Journal of Computer Science and Cybernetics*, vol. 31, no. 1, pp. 17–30, 2015.

[5] H. Bui, B. Vo, and H. Nguyen, "Wun-miner: A new method for mining frequent weighted utility itemsets." IEEE, 2016, pp. 001 365–001 370.
[6] H. Nguyen, N. Le, H. Bui, and T. Le, "A new approach for efficiently mining frequent weighted utility patterns," *Applied Intelligence*, vol. 53, no. 1, pp. 121–140, 1 2023.
[7] H. Nguyen, N. Le, H. Bui, and T. Le, "Mining frequent weighted utility patterns with dynamic weighted items from quantitative databases," *Applied Intelligence*, 3 2023, [Online; accessed 2023-06-15]. [Online]. Available: https://link.springer.com/10.1007/s10489-023-04554-z
[8] H. Yao, H. J. Hamilton, and C. J. Butz, "A foundational approach to mining itemset utilities from databases." SIAM, 2004, pp. 482–486.
[9] Y. Liu, W.-k. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets." Springer, 2005, pp. 689–695.
[10] V. S. Tseng, B.-E. Shie, C.-W. Wu, and S. Y. Philip, "Efficient algorithms for mining high utility itemsets from transactional databases," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 8, pp. 1772–1786, 2012.
[11] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," 2012, pp. 55–64.
[12] S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, "Efim: a highly efficient algorithm for high-utility itemset mining." Springer, 2015, pp. 530–546.
[13] L. T. Nguyen, V. V. Vu, M. T. Lam, T. T. Duong, L. T. Manh, T. T. Nguyen, B. Vo, and H. Fujita, "An efficient method for mining high utility closed itemsets," *Information Sciences*, vol. 495, pp. 78–99, 8 2019.
[14] S. Krishnamoorthy, "Mining top-k high utility itemsets with effective threshold raising strategies," *Expert Systems with Applications*, vol. 117, pp. 148–165, 3 2019.
[15] H. Kim, U. Yun, Y. Baek, J. Kim, B. Vo, E. Yoon, and H. Fujita, "Efficient list based mining of high average utility patterns with maximum average pruning strategies," *Information Sciences*, vol. 543, pp. 85–105, 1 2021.
[16] U. Huynh, B. Le, D.-T. Dinh, and H. Fujita, "Multi-core parallel algorithms for hiding high-utility sequential patterns," *Knowledge-Based Systems*, vol. 237, p. 107793, 2 2022.
[17] H. M. Huynh, L. T. Nguyen, B. Vo, Z. K. Oplatková, P. Fournier-Viger, and U. Yun, "An efficient parallel algorithm for mining weighted clickstream patterns," *Information Sciences*, vol. 582, pp. 349–368, 1 2022.
[18] S. Kumar and K. K. Mohbey, "A review on big data based parallel and distributed approaches of pattern mining," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1639–1662, 5 2022.
[19] Z. Deng, Z. Wang, and J. Jiang, "A new algorithm for fast mining frequent itemsets using n-lists," *Science China Information Sciences*, vol. 55, pp. 2008–2030, 2012.
[20] R. Rymon, "Search through systematic set enumeration," in *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 1992, pp. 539–550.
[21] "Spmf: A java open-source data mining library, retrieved from http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php (accessed: 20 august 2020)."

Lê Hoàng Bình Nguyên đang làm Nghiên cứu sinh chuyên ngành Cơ sở toán học cho tin học tại Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội. Hiện công tác tại Khoa Công nghệ thông tin, Trường Cao đẳng An ninh mạng iSPACE, TP. Hồ Chí Minh.
Email: nguyen.le@ispace.edu.vn

Nguyễn Duy Hàm nhận học vị Tiến sĩ ngành Cơ sở toán học cho tin học vào năm 2017 tại Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội. Hiện là giảng viên Khoa Công nghệ thông tin của Trường Đại học Sài Gòn, TP. Hồ Chí Minh.
Email: ndham@sgu.edu.vn

Nguyễn Thị Hồng Minh Nhận bằng Thạc sĩ và Tiến sĩ về Toán - Tin của Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội vào năm 1996 và 2001. Hiện là Phó Giáo sư và công tác tại Khoa Toán - Cơ - Tin học, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội.
Email: minhnth@hus.edu.vn

Social Network Recommendations for Friends with Neo4j Graph Database

Pham Thi Thu Thuy[1], Nguyen Thi Thai Thanh[2], Hwa Soo Kim[3]
[1] Nha Trang University, Nha Trang, Vietnam
[2] Van Ninh Vocational School, Khanh Hoa province, Vietnam
[3] Graduate School of Information and Communication Technology, AJOU University, Korea

Correspondence: Pham Thi Thu Thuy, thuthuy@ntu.edu.vn
Received: 02/28/2023, revised: 11/09/2023, accepted: 04/10/2023
Digital Object Identifier:: 10.32913/mic-ict-research-vn.v2023.n2.1239

Abstract: In recent years, along with the development of the internet, the number of social network users is increasing day by day. Through social networks, users can share, exchange information or make friends with each other. However, with new relationships, users often have a need to assess the credibility of a new friend before making friends on social networks. This paper proposes a friend recommendation system on social networks based on how to calculate the reliability between users. The recommendation system is implemented using the Neo4j graph database. The results of the truth algorithm proposed in this paper are higher than other similar algorithms.

Keywords: *Social Network, Graph Database, Neo4j, Recommendation, Truth algorithm*

Tiêu đề: **Ứng dụng cơ sở dữ liệu đồ thị Neo4j xây dựng hệ thống khuyến nghị kết bạn trên mạng xã hội**

Tóm tắt: Những năm gần đây, cùng với sự phát triển của internet, số lượng người dùng mạng xã hội ngày càng tăng. Thông qua mạng xã hội, người dùng có thể chia sẻ, trao đổi thông tin hoặc kết bạn với nhau. Tuy nhiên, với những mối quan hệ mới, người dùng thường có nhu cầu đánh giá độ tin cậy của một người bạn mới trước khi kết bạn trên mạng xã hội. Bài báo đề xuất hệ thống gợi ý kết bạn trên mạng xã hội dựa trên cách tính toán độ tin cậy giữa những người dùng. Hệ thống đề xuất được triển khai bằng cơ sở dữ liệu đồ thị Neo4j. Kết quả của thuật toán xác thực đề xuất trong bài báo này cao hơn các thuật toán tương tự khác.

Từ khóa: *Mạng xã hội, Cơ sở dữ liệu đồ thị,Neo4j, Khuyến nghị, Thuật toán tính độ tin cậy*

I. INTRODUCTION

In recent years, computers have brought tremendous opportunities for the development of society. We use computers every day; they not only become part of our daily lives, but also help define our personality through our social network profiles, virtual activities, our behavior in discussions, etc. Changes in human behavior inevitably affect the technical and functional requirements of computers from applications. The Internet has become a necessity; this places a huge demand on web applications. Social networking site Facebook has launched Web 3.0, which is not only a modern network but also a network of intelligent, personalized user and multimedia content.

Through social networks, users can interact, share articles, post photos and videos, and help users connect with people currently living in different geographical regions or allow users to make friends and looking for friends. The main components in a social network include nodes and links. A node is an entity or a collection of entities such as individuals, businesses or organizations also called nodes. A link is a relationship between nodes. In a network in its simplest form, a social network is an arrangement of related links between nodes. A social network can be represented by a diagram in which the linked nodes are shown as straight lines.

Therefore, social networks create a social graph of users, including users and their friendship relationships. This social graph is very rich, not just the relationship of one or two friends to each other but the relationship between many friends (for example, friends of a friend). In that vast relationship, sharing and exchanging information or making new friends,... are mostly based on mutual trust. Therefore, just like in life, trust among users in social networks is also concerned and has been mentioned by many researches in

computer science in recent years.

On the other hand, graph databases are used in data with many relationships such as social networks, recommender systems, network management, geospatial, etc. Strengths of graph databases is the ability to store relationships and query on those relationships. A graph database is a type of NoSQL [18] database where data is organized and stored in the form of nodes, edges, and attributes. Unlike the relational database model, the graph database focuses on describing the relationships of the data, which makes it well supported for data types with many complex relationships. complexity and queries related to these relationships.

Currently there are graph databases being used such as: Neo4j [1, 2], RedisGraph [3]; OrientDB [4]. This study chooses Neo4j database to represent social network data. Neo4j is an open source graph database that is a NoSQL database, built in Java and Scala, sponsored by Neo Technology Corporation. The project began to develop in 2003 and became public since 2007. Currently, Neo4j has been used by thousands of companies and organizations in various fields, including network management, software analysis, scientific research, routing, and social networking [17]. Neo4j is well suited for performing association prediction problems, as well as for big data management. Neo4j is mainly designed to capture relationships between items of information, the basic organizing principle of Neo4j is to store information as a network of relationships. The network contains nodes, the nodes are connected through relationships, the nodes are called vertices and the relationships are called edges. Neo4j is designed to handle big data, highly interconnected data, supports multiple platforms, and has the Cypher [5] query language that queries graphs very quickly.

Cypher is a declarative query language similar to SQL, queries are constructed using different clauses. Cypher is designed to be simple, powerful and very convenient when working with Neo4j. Cypher's c structure is designed and built based on the English language, neat, easy to understand, convenient for language manipulators to make reading and writing queries easy and suitable for developers. programmers and database query specialists.

The objective of this study is to learn how to retrieve and represent social network data, specifically Facebook, in the form of a Neo4j database, and then evaluate the trustworthiness of each user account on the Internet. Facebook through their activities. On the basis of the installation results, we install the algorithm that has been previously studied by the previous researchers and propose a new reliability calculation algorithm to improve the reliability calculation results among users on social networks.

The friend recommendation problem is often used in social networks. The basic approach is based on referrals from your friends, the same group, the same events, the same interests… there are many connections to each other, so the use of a graph database to store data exploits many of the strengths of the graph data model.

This study aims to reconstruct the database of users on Facebook as a Neo4j graph database, and install algorithms to determine the trust between users such as Adamic Adar, Common Neighbors and Preferential Attachment, thereby suggesting friend recommendations.

In addition, we propose a new algorithm related to calculating trust among users based on their social relationships. The results of the study have formed an application that recommends making friends based on specific relationships such as: the relationship between friends on each level or the relationship about where you live, or where you work.

II. RELATED WORK

This section presents some commonly used algorithms in social network friend recommendation systems. The most commonly used algorithm is Common Neighbors (CN) [6]. The common neighbor algorithm captures the idea that two strangers who have a mutual friend are more likely to be referred than those who have no mutual friends. The similarity between nodes a and b is calculated as the number of common neighbors between a and b. The greater the number of shared nodes, the higher the similarity between a and b.

$$CN(a,b) = |N(a) \cup N(b)| \tag{1}$$

The similarity between nodes a1 and b1 using CN similarity measurement can be calculated as in equation (2):

$$CN(a1,b1) = |N(a1) \cup N(b1)| = |\{1,2\} \cup \{1,5\}| = 1 \tag{2}$$

Another commonly measure is Adamic Adar (AA) [7] as in the equation 3. It is a measure that is used to calculate the closeness of two nodes based on their common neighbor. The similarity between nodes a and b is calculated as the sum of the reciprocals of the logarithms of each common neighbor z between a and b, the more common nodes at a low level, the higher the similarity between nodes. This is a weighted version of the CN similarity measure.

$$AA(a,b) = \sum_{z \in Na_1 \cap N(b_1)} \frac{1}{log(x_z)} \tag{3}$$

The similarity between nodes a1 and b1 using the AA measure can be calculated as in equation (4):

$$AA(a,b) = \sum_{z \in Na_1 \cap N(b_1)} \frac{1}{log(x_z)} = \frac{1}{log(x_1)} = \frac{1}{log(3)} \quad (4)$$

N(z) is the set of nodes adjacent to z. A value of 0 indicates that the two nodes are not close to each other, while a higher value indicates that the nodes are closer together.

Another algorithm mentioned in the similarity assessment system on social networks is Preferential Attachment (PA) [8] model. In this model, the more connected a node is, the more likely it is to receive new links. A value of 0 indicates that the two nodes are not close to each other, a higher value indicates that the nodes are closer together. The similarity between nodes a and b is calculated as the product of the degrees of nodes a and b. The higher the degree of both nodes, the higher the similarity between a and b, as shown in the equation (5). Where xa is the number of relationship levels of User 1; xb is the number of relationship levels of User 2.

$$PA(a,b) = x_a * x_b \quad (5)$$

In the previous work, the lead author proposed distance-based reliability measurement [9] based on the assumption: the closer a node is to the source node in a trusted network, the more reliable it is. The measurement is given as follows: For power button E; E predicts the reliability of all remaining nodes in the system based on the shortest distance from E to those nodes. Suppose we call d the maximum threshold over which the confidence can propagate. For a node A whose shortest distance from E to A is n($n \leq d$), the reliability of A with E is calculated by the following formula:

$$Trust = \frac{d - n + 1}{d} \quad (6)$$

Value of trust = 0 for nodes with distance greater than d. For example, consider the graph in Figure 1.

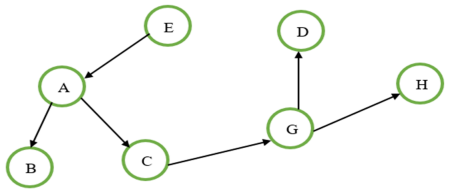


Figure 1. The graph illustrates the example of calculating the distance measure

According to the Figure 1, let's say d=2; Trust (E,A)=1; Trust(E,B)=Trust(E,C)=(2-2+1)/2=1/2;
Trust(E,G)=Trust(E,D)=Trust(E,H)=0

Besides, there are many studies related to link prediction [11–17]. There is an interplay between every pair of different types of relationships. This effect can vary for different couples. For example, students are more likely to form friendships with friends of classmates than with friends of relatives. The "classmate" relationship has a greater impact on the "friend" relationship than the "relative" relationship. In order to predict potential connections in such multi-relational social networks, the correlation and influence between different relations should be considered comprehensively. Most existing relation prediction algorithms are not suitable for multi-relational networks because they only observe one kind of relation and ignore the overall structure and influence among different relations. An intuitive way to predict hidden or future relations in a multi-relational network is to merge multiple relations in the network into a single relation, and perform prediction of potential relations based on this merged relation. However, with this simple approach, we may miss too much topological information to get accurate prediction results. Therefore, we should develop an effective method to utilize multi-relational information in social networks to improve the accuracy of prediction results.

In this paper, we represent the data of users on Facebook through the Neo4j graph database, apply the algorithms to calculate the reliability between users proposed by previous studies, and at the same time recommend the new algorithm to improve the truth measurement. Detail, we install the above 03 algorithms (including CN, AA and PA), and propose a new algorithm to improve and increase accuracy.

III. OUR PROPOSAL: TRUTH CALCULATION ALGORITHM BASED ON RELATIONSHIPS (DT)

In fact, there are different ways to assess trust in social networks based on direct trust relationships between users, assessment based on reputation of network members, and calculation based on trustworthiness. on trust ratings and there are many types of relationships that connect visitors to each other in a social network.

This paper introduces a new method aimed at calculating trust in social networks, which considers user relationships at each level. The calculated confidence values are in the

range 0-1.

Table I
RELIAILITY TABLE BASED ON RELATIONSHIP LEVEL

No	Type of relationships	Trust value
1	Friend	1
2	Your friend	0.8
3	Friend of your friend	0.6
4	Shared workplace	0.4
5	Sharing a place to live	0.2
6	No relationship	0

We construct a direct trust calculation between user v and user v', DT(v; v') as a sorted real value in a unit interval. First, if DT(v; v') = 0, then the confidence v has in v' is minimal, i.e. v is not completely reliable for v'. Conversely, if DT(v; v') = 1, this means that v is completely reliable with v'.

One approach to calculating reliability among users of direct connections is to consider the nature of their interactions based on friendships or co-worker relationships.

Friend: is a type of relationship defined by being a direct relation because v and v' are directly connected through an edge a→a'.

We define the function of a friendship as F:V x R x V → [0,1], this function is used to calculate the confidence ft in the interval [0; 1] between two vertices (two users) depending on their friendship. The formula F(v, r, v') = ft is used as a basis for user v to assign a confidence level to his friends v'.

The input and output parameters of the proposed algorithm are as follows:

Input:

1. U is User
2. d is the level of the algorithm (d > 0)
3. F0 is the set of direct friends of User
4. F1 is the set of direct friends of F.
5. Fd is the set of friends of User according to level d.
6. L is a collection of people living in the same city.
7. W is the set of people who do the workplace.
8. R(m,n) is the resulting 2-dimensional matrix where m is User n is the confidence level.

Output: Truth value, friend recommendation

The pseudo-algorithm code is as in the Figure 2.

Algorithm explanation: The foreach statement line number {2} is executed n times, where n is the number of elements in the set Fd, in each iteration each comparison statement is the lines {3}, {5}, {7}, {9}, {11}, the assignment statements are {4}, {6}, {8},{10},{12}.

```
Begin                                {1}
   foreach f in Fd do:               {2}
     if (f in F0) {                  {3}
         R[f, 1]                     {4}
      } else if (f in F1) {          {5}
         R[f, 0.8]                   {6}
     } else if (f in F2) {           {7}
         R[f, 0.6]                   {8}
     } else if (f in W) {            {9}
         R[f, 0.4]                   {10}
     } else if (f in L) {            {11}
         R[f, 0.2]                   {12}
     }                               {13}
  return R                           {14}
end                                  {15}
```

Figure 2. Pseudocode of the proposed algorithm

IV. EXPERIMENT RESULTS

To conduct the simulation of algorithms, we use a computer with Intel Core i5-2430M CPU 2.40GHZ, Ram: 08GB, Windows 10 Pro-64bit operating system. Neo4j Desktop Graph Database version 1.5.7.

To get data on Facebook, we go to https://developers.facebook.com/apps/ and press the "Create application" button. This is the step to create the Facebook application. Any user account that accepts to link with this application will have access and share the information requested by the application (here is information about friends). We then access the above link as a Consumer to select and add the "user_friends" permission and other permissions if necessary. Then we click "Generate Access Token" to get to the application permission screen. We then use the Graph API [10] engine to collect Facebook social network data as illustrative data for algorithms. Using Graph API, we collected 209 users with the following information: Name, MaUID, Gioitinh, Job, Diachi, City, Work. The data consists of 07 columns, 210 lines and 16.1KB in size, in csv file format.

After having user data on Facebook, we create nodes from users and create friendship from these users. We have created 257 Nodes, 615 Relationships and 3 Labels. After creating the relationship, we look for the relationship based on the edges to suggest friend recommendations, and calculate the reliability on that recommendation to help the user to make a right decision.

In this study, we are interested in factors that increase the probability of Users making friends with each other such as: Potential triangle relationship between Users; Users share a specific community such as: living in the same place; same workplace.

Collected data and additional data installed on Neo4j, shown in Figure 3.

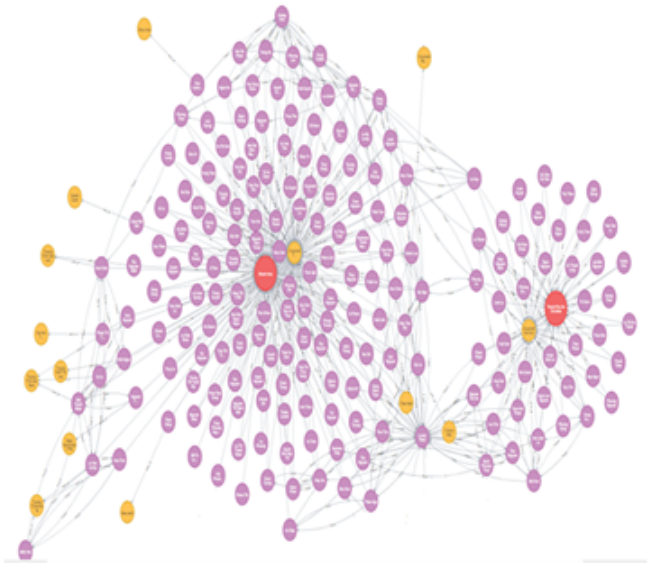


Figure 3. Representation of data collected on Neo4j

In Figure 3, the purple nodes represent the user's nodes on Facebook, the yellow nodes represent the user's workplace, and the red nodes represent the user's workplace.

We install four friend recommendation models, namely: similarity between p1 and p2 nodes by Adamic Adar (AA), common neighbor (CN), priority connection with Preferential Attachment (PA), relationship-based trust proposed by us (DT).

The query to calculate the similarity between node p1 and p2 according to the AA algorithm is as follows:

```
gds.alpha.linkprediction.adamicAdar(p1, p2) as score;";
System.out.println(queryRaw);
Query query = new Query(queryRaw);
Result result = session.run(query);
return result.single().get("score").asDouble();
} catch (NoSuchRecordException e) {
return 0;
```

For the CN algorithm, we replace the first line of the above query with the following statement:

```
gds.alpha.linkprediction.commonNeighbors(p1, p2) AS score;";
```

Similarly, for the PA algorithm, we replace the first line of the above query with the following statement:

```
gds.alpha.linkprediction.preferentialAttachment(p1, p2,) as score;";
```

The software interface of the resulting reliability calculation and friend recommendation for the user is shown in Figure 4. In this figure, when we select any user, the system will list all users in the database that have at least relation with the selected user, for example: Friend, Friend of Friend, Friend of Friend of Friend , where to live, where to work. From there the user can make the decision to make friends if necessary.

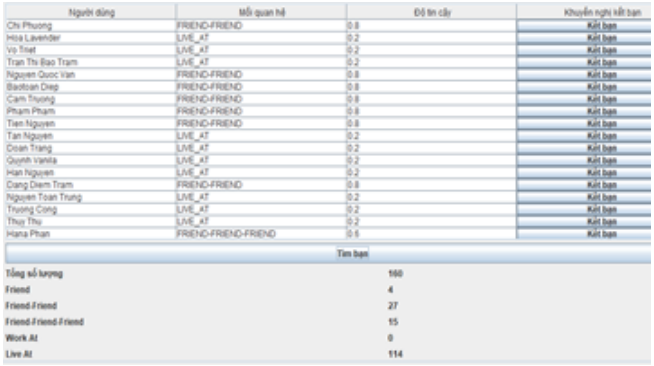


Figure 4. Reliability results and friend recommendations

The matching reliability results on the relationships are shown in Figure 5. Figure 5 shows the matching confidence result of a user named Quang Vinh. Based on the results in Figure 4, we see that our proposed algorithm (DT) always returns the highest matching rate on relationships. The PA algorithm gives the lowest results among the installed algorithms.

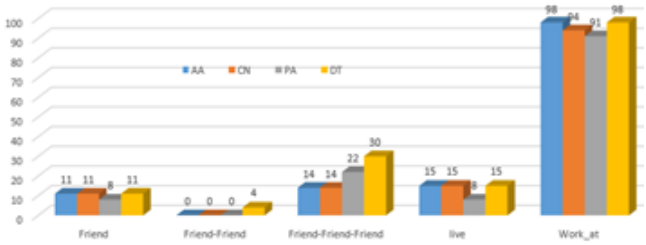


Figure 5. Similarity value matches according to relationships

V. CONCLUSION AND FUTURE WORK

Measuring trust among users on social networks has great importance for decision making friends. Relation prediction in social networks has attracted increased attention from policy makers in many business disciplines. Calculating trust between users is very useful for users

to discover friends around or members of organizations, thereby helping that user and their friends to expand your network. friends, have many opportunities to exchange and cooperate. In actual society, there are many types of relationships between people such as friends, colleagues, relatives, classmates, partners, etc.

To calculate the trust between users in a multi-relationship network, we should take into account the importance of relationships. In this article, we consider four types of relationships and rank them in order of priority from high to low: your friend relationship, your friend's relationship, the same place of work and the same place of residence. Experimental results show that our proposed DT algorithm is superior to other similar algorithms and can achieve more accurate results on multi-relational networks.

In many real-world social networks, users may have attributes such as age, gender, occupation, income, habits, interests, etc., which are valuable for calculating the credibility of Surname. In our current study, we omit such useful information about user properties. Because, the actual collected data on Facebook of users often leave this information blank. However, the data of the users in the social network changes over time. Therefore, our future work is to learn efficient methods to integrate multiple user attributes for inclusion in the reliability algorithm.

REFERENCES

[1] Neo4j, "Neo4j Documentation", https://neo4j.com/docs/, accessed date: March 10th, 2022.

[2] https://neo4j.com/docs/graph-data-science/current/algorithm, accessed date March 10th, 2022.

[3] https://docs.redis.com/latest/stack/deprecated-features/graph/, accessed date March 10th, 2023.

[4] http://orientdb.org/, accessed date March 10th, 2023.

[5] Cypher Query Language, https://neo4j.com/developer/cypher/, accessed date March 10th, 2023.

[6] Iftikhar Ahmad, Muhammad Usman Akhtar, Salma Noor & Ambreen Shahnaz, *Missing Link Prediction using Common Neighbor and Centrality based Parameterized Algorithm*, Sci Rep 10, 364 (2020). https://doi.org/10.1038/s41598-019-57304-y

[7] Sourabh Dadapure, *Recommendation Engine using Adamic Adar Measure*, American Society for Engineering Education, 2022.

[8] Sheridan, P., Onodera, T., *A Preferential Attachment Paradox: How Preferential Attachment Combines with Growth to Produce Networks with Log-normal In-degree Distributions*, Sci Rep 8, 2811 (2018). https://doi.org/10.1038/s41598-018-21133-2.

[9] Pham Thi Thu Thuy, *Algorithm to determine the trustworthiness of friends on social networks based on Ontology*, Science Journal of Da Lat University Vol. 8, Issue 2, 2018 139–150.

[10] Graph API, https://developers.facebook.com/docs/graph-api/, March 10th, 2023.

[11] Ling Chen, Man Gao, Bin Li, Wei Liu, Bolun Chen, *Detect potential relations by link prediction in multi-relational social networks*, Decsup (2018), doi:10.1016/j.dss.2018.09.006.

[12] Kai Zhu, Meng Cao, Heng-yang Lu, *MALP: A More Effective Meta-Paths Based Link Prediction Method in Partially Aligned Heterogeneous Social Networks*, 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI).

[13] Ling Chen, Man Gao, Bin Li, Wei Liu, Bolun Chen, *Detect Potential Relations by Link Prediction in Multi-Relational Social Networks*, Decsup (2018), doi: 10.1016 / j.dss.2018.09.006.

[14] Pinghua Xu, Wenbin Hu, Jia Wu, Bo Du, *Link Prediction with Signed Latent Factors in Signed Social Networks*, Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA. ACM, New York, NY, USA, 9 pages.

[15] Virinchi Srinivas, Pabitra Mitra, *Link Prediction in Social Networks*, Springer, 2016.

[16] Jan Skrasek, *Social Network Recommendation using Graph Databases*, Brno, 2015.

[17] Gorka Sadowksi, Philip Rathle, *Fraud detection: Discovering Connections using Graph Databases*, The World's Leading Graph Database, January 2015.

[18] DB Engines, *Knowledge Base of Relational and NoSQL Database Management Systems*, https://db-engines.com/, accessed date: March 20th, 2022.

Pham Thi Thu Thuy received the B.Eng. degree in Computer Technology from Hanoi University of Science and Technology, Vietnam, in 2001. In 2006, she received the Master degree, and in 2012 received the Ph.D. degree, both on Computer Engineering from Kyung Hee University, South of Korea. She is a main lecturer of the Nha Trang University, Vietnam. Her research interests include ontology matching, semantic measurement, graph database, ontology and Semantic Web.
Email: thuthuy@ntu.edu.vn

Nguyen Thi Thai Thanh received her bachelor's degree on information technology in 2011 from University of Information Technology, Ho Chi Minh City National University. She has got the master degree in Computer Technology at Nha Trang University. Currently she is working at the Training Department of Van Ninh Vocational School in Van Ninh district, Khanh Hoa province, Vietnam.
Email: thaithanhcn@gmail.com

Hwa Soo Kim received the B.Eng. degree in Electronics from the Korea University, Seoul Korea in 1981. In 1984, he received the Master degree in Computer Science from the Naval Postgraduate School, United States and his Ph.D. degree at Case Western Reserve University, United States in 1990. He was a professor of the Korea National Defense University and is a Special Appointment Professor at AJOU University, Korea. His research interests include cyber security, project management, AI technologies.
Email: ajhskim@ajou.ac.kr

Một thuật toán định tuyến cân bằng năng lượng trong mạng cảm biến không dây dựa trên SDN

Lê Đức Huy[1], Lê Hữu Bình[2], Đỗ Thành Công[1], Nguyễn Đỗ Hoàng Giang[3]

[1] Khoa Công nghệ Thông tin, Trường Đại học Kinh doanh và Công nghệ Hà Nội

[2] Khoa Công nghệ Thông tin, Trường Đại học Khoa học, Đại học Huế

[3] Trường Trung học Phổ thông chuyên Khoa học Tự nhiên, Đại học Quốc gia Hà Nội

Tác giả liên hệ: Lê Hữu Bình, lhbinh@hueuni.edu.vn

Ngày nhận bài: 18/07/2023, ngày sửa chữa: 26/09/2023, ngày duyệt đăng: 25/11/2023

Định danh DOI: 10.32913/mic-ict-research-vn.v2023.n2.1240

Tóm tắt: Trong mạng cảm biến không dây (WSN), việc sử dụng năng lượng sao cho hiệu quả để kéo dài thời gian hoạt động của các nút cảm biến là điều cần thiết. Trong bài báo này, chúng tôi đề xuất một thuật toán định tuyến có xét đến mức tiêu thụ năng lượng tại các nút cảm biến. Mục tiêu của thuật toán được đề xuất là cân bằng mức tiêu thụ năng lượng, giảm thiểu số nút phải tiêu thụ nhiều năng lượng nhằm tăng thời gian hoạt động của chúng. Phương pháp của chúng tôi là xây dựng một hàm trọng số cho các kết nối có chứa tham số về năng lượng còn lại tại các nút. Sau đó, sử dụng kỹ thuật định tuyến tập trung dựa trên kiến trúc mạng điều khiển bằng phần mềm (SDN) để tìm lộ trình có trọng số tốt nhất sử dụng cho việc truyền dữ liệu. Kết quả mô phỏng sử dụng OMNeT++ cho thấy rằng, thuật toán được đề xuất cho phép tăng thời gian hoạt động của các nút, tăng thông lượng so với các thuật toán định tuyến phổ biến hiện hành.

Từ khóa: *Mạng cảm biến không dây, định tuyến cân bằng năng lượng, SDN*

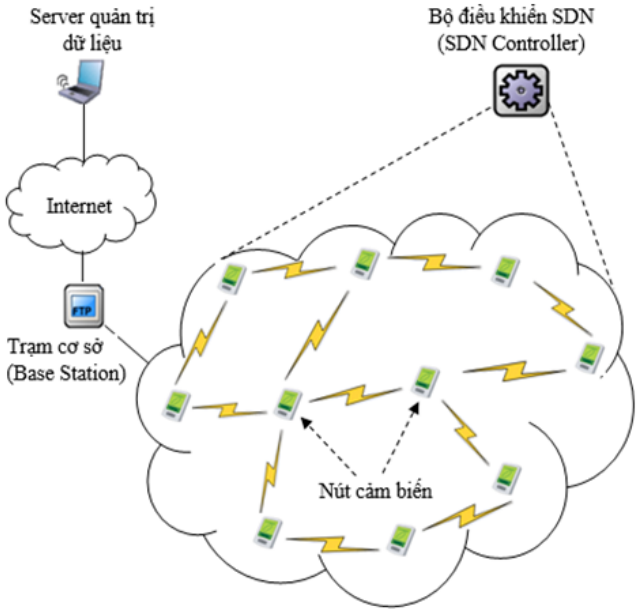
Title: **An Energy-Balanced Routing Algorithm in SDN-based Wireless Sensor Networks**

Abstract: In wireless sensor networks (WSN), it is necessary to use energy efficiently to extend the operating time of sensor nodes. In this study, we propose a routing algorithm that considers the energy consumption between sensor nodes. The goal of the proposed algorithm is to balance energy consumption and minimize the number of nodes that must consume a large amount of energy to increase their uptime. Our method constructs a weight function for wireless links that contains parameters for the energy remaining at the nodes. Then, a centralized routing mechanism based on a software-defined network (SDN) architecture is used to find the best weight route for data transfer. Simulation results on OMNeT++ show that the proposed algorithm increases the uptime of nodes and network throughput compared to the current popular routing algorithms.

Keywords: *Wireless Sensor Networks, energy-balanced routing, software-defined network*

I. GIỚI THIỆU

Thời gian gần đây, mạng cảm biến không dây (Wireless Sensor Networks - WSN) được sử dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là trong giai đoạn bùng nổ các ứng dụng trên nền tảng IoT như hiện nay. Mạng WSN có thể hoạt động theo ba mô hình điều khiển khác nhau, đó là điều khiển theo mô hình tập trung, điều khiển theo mô hình phân cấp và điều khiển theo mô hình phân tán [1]. Với mô hình điều khiển tập trung, có một nút chứa thông tin toàn cục trong hệ thống mạng. Các chức năng điều khiển việc truyền dữ liệu từ các nút mạng về trạm cơ sở (Base Station - BS) được thực hiện tại nút này. Với mô hình điều khiển phân cấp, toàn bộ các nút được chia thành từng cụm, mỗi cụm có một nút đóng vai trò như là "cụm trưởng". Việc truyền dữ liệu từ các nút về BS được thực hiện thông qua nút cụm trưởng. Với mô hình điều khiển phân tán, các nút đóng vai trò ngang hàng nhau. Các giao thức điều khiển được thực hiện độc lập tại mỗi nút mạng. Trong giai đoạn gần đây, kiến trúc điều khiển tập trung dựa trên giải pháp mạng kiểm soát bằng phần mềm (Software Defined Networking - SDN) đã được áp dụng cho mạng WSN [2–10]. Nguyên lý cơ bản của mô hình này như cho thấy ở Hình 1. Các nút cảm biến được kết nối bộ điều khiển SDN một cách trực



Hình 1. Một ví dụ của mạng cảm biến không dây sử dụng SDN

tiếp hoặc gián tiếp (thông qua các nút khác) sử dụng giao thức luồng mở (open flow). Các giao thức điều khiển hoạt động của hệ thống mạng như định tuyến, báo hiệu, điều khiển tô-pô, v.v được thực hiện tập trung tại bộ điều khiển SDN. Mô hình WSN sử dụng SDN đã được triển khai để cải tiến hầu hết các giao thức điều khiển trong mạng WSN, điển hình như các giao thức định tuyến [4, 5, 10–12], điều khiển tô-pô [6, 7, 13], điều khiển phân cụm [8, 9]. Trong đó, định tuyến sử dụng SDN là chủ đề nhận được sự quan tâm của nhiều nhóm nghiên cứu trong nước và trên thế giới thời gian gần đây.

Trong [4], S. Wang cùng các cộng sự đã đề xuất một giao thức truyền thông tích hợp, được đặt tên là MINI-FLOW. Giao thức này tập trung vào ba cơ chế định tuyến dữ liệu, định tuyến đường lên (từ các nút cảm biến đến bộ điều khiển SDN), định tuyến đường xuống (từ bộ điều khiển SDN đến các nút cảm biến) và cung cấp thông tin khi khởi tạo mạng. Cơ chế định tuyến được thiết kế dựa trên một hàm trọng số heuristic bao gồm ba giá trị, tổng số bước truyền đến nút đích, cường độ tín hiệu nhận được và năng lượng còn lại. Bằng phương pháp thực nghiệm, các tác giả đã cho thấy giao thức MINI-FLOW mang lại hiệu quả cao khi xem xét vấn đề kết hợp giữa định tuyến và cân bằng tải. Trong [5], nhóm tác giả đã đề nghị một kiến trúc gọi là phần mềm dựa trên đa luồng cho WSN. Cơ chế này được phát triển dựa trên SDN và phân bổ các kênh riêng biệt cho lớp dữ liệu và lớp điều khiển. Giải pháp đề xuất được sử dụng để duy trì cấu trúc liên kết và quản lý việc sử dụng nguồn năng lượng. Các kết quả thực nghiệm cho thấy rõ cơ chế được đề xuất mang lại hiệu quả cao về mặt thông lượng, độ trễ và mức tiêu thụ năng lượng trong mạng. Trong [10], một cơ chế định tuyến có xét đến năng lượng dựa trên SDN được đề xuất để cải thiện hiệu quả sử dụng năng lượng của WSN với hệ thống IIoT để hỗ trợ công nghiệp 4.0. Bộ điều khiển SDN ước tính mức năng lượng tại các nút quan trọng trong WSN và quyết định lộ trình tốt nhất dựa trên tình trạng sử dụng năng lượng của chúng thay vì tiêu chí đường đi ngắn nhất được sử dụng rộng rãi. Sử dụng phương pháp mô phỏng, nhóm tác giả đã cho thấy rõ phương pháp được đề nghị cho phép giảm thiểu tình trạng các nút của WSN cạn kiệt nguồn năng lượng và làm gián đoạn việc truyền dữ liệu một cách đột ngột.

Thông qua các công trình đã công bố được khảo sát ở trên chúng tôi nhận thấy rằng, việc ứng dụng kiến trúc SDN để điều khiển định tuyến trong mạng WSN là một giải pháp hữu hiệu. Đây cũng là giảp pháp phù hợp với xu hướng phát triển của công nghệ mạng truyền thông không dây thế hệ mới. Trong công trình nghiên cứu này, chúng tôi cũng sử dụng kiến trúc SDN vào việc điều khiển định tuyến trong mạng WSN. Các đóng góp mới của công trình này được liệt kê như sau:

- Đề xuất một hàm để tính toán trọng số cho tất cả các kết nối không dây trong mạng WSN. Hàm trọng số này có chứa các tham số về tỷ lệ giữa năng lượng khởi đầu và năng lượng còn lại tại các nút.
- Đề xuất một thuật toán định tuyến mới với mục tiêu cân bằng việc sử dụng năng lượng cho mạng WSN dựa trên kiến trúc SDN. Thuật toán có tên là SEBR (SDN-based Energy Balanced Routing).

Các phần tiếp theo của bài báo được tổ chức như sau. Phần II tập trung trình bày thuật toán định tuyến được đề xuất. Phần III phân tích các kịch bản mô phỏng và thảo luận về các kết quả. Cuối cùng kết luận các kết quả đã đạt được và nêu ra hướng phát triển tiếp theo của chủ đề nghiên cứu này, được trình bày trong phần IV.

II. THUẬT TOÁN ĐỊNH TUYẾN ĐƯỢC ĐỀ XUẤT

Nội dung chính của phần này trình bày thuật toán định tuyến cân bằng năng lượng tiêu thụ tại các nút, được đề xuất cho mạng WSN dựa trên kiến trúc SDN, được đặt tên SEBR (SDN-based Energy Balanced Routing). Thuật toán SEBR hoạt động theo nguyên lý định tuyến tập trung, sử dụng một hàm trọng số có xét đến tỷ số giữa năng lượng khởi đầu và năng lượng còn lại tại mỗi nút cảm biến. Hàm trọng số này được sử dụng làm tiêu chí để tìm tập lộ trình từ mỗi nút đến trạm cơ sở sử dụng cho việc truyền dữ liệu cảm biến về server quản trị dữ liệu. Để tìm tập lộ trình, thuật toán tìm "đường đi ngắn nhất" được thực thi tại bộ điều khiển SDN. "Đường đi ngắn nhất" được hiểu theo nghĩa tổng trọng số theo hàm được đề xuất là nhỏ nhất.

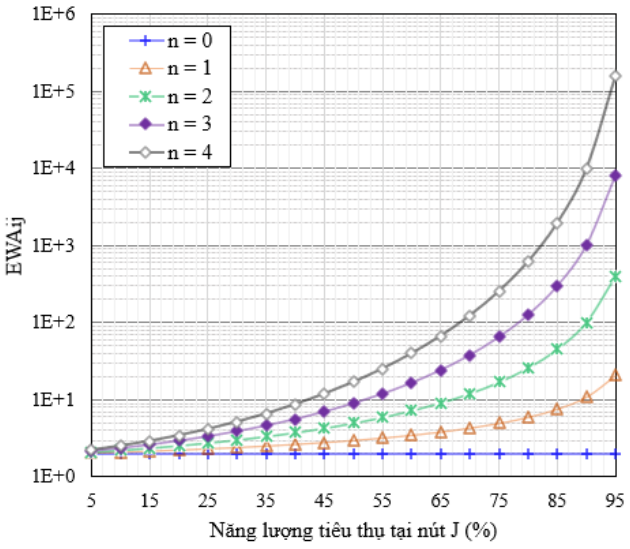
1. Hàm trọng số có xét đến mức tiêu thụ năng lượng tại mỗi nút

Mục tiêu của thuật toán SEBR là cân bằng mức năng lượng tiêu thụ tại các nút, giảm thiểu tình trạng một số nút tiêu thụ năng lượng lớn dẫn đến nhanh chóng bị tắt do hết năng lượng. Trong thuật toán SEBR, việc cân bằng nức tiêu thụ năng lượng được thực hiện thông qua một hàm trọng số có xét đến tỷ số giữa năng lượng khởi đầu và năng lượng còn lại tại mỗi nút, được đặt tên là EAW (Energy Awave Weight), được định nghĩa như sau:

$$EWA_{ij} = \begin{cases} w_{ij} + \left(\dfrac{E_j^{(s)}}{E_j^{(r)}}\right)^n & \text{nếu } E_j^{(r)} \geq 1\% \times E_j^{(s)} \\ \infty & \text{trường hợp ngược lại} \end{cases} \quad (1)$$

trong đó, w_{ij} là trọng số của đường truyền không dây từ nút I đến nút J, $E_j^{(s)}$ là năng lượng tại thời điểm nút J bắt đầu hoạt động và $E_j^{(r)}$ là năng lượng còn lại của nút J tại thời điểm xem xét, n là một số nguyên không âm.

Với hàm trọng số được thiết lập như ở (1), khi năng lượng còn lại của nút đích của mỗi kết nối càng nhỏ thì trọng số của các kết nối đến nút đó càng lớn. Trọng số tăng theo hàm bậc n của tỷ số giữa năng lượng khởi đầu và năng lượng còn lại. Sự phụ thuộc của trọng số EWA_{ij} vào năng lượng tiêu thụ tại nút J và bậc n theo phương trình (1) được thể hiện rõ ở Hình 2. Chúng ta dễ dàng quan sát rằng, khi bậc $n = 0$, trọng số EWA_{ij} là một hằng số vì nó không còn phụ thuộc vào năng lượng tiêu thụ tại nút J. Trường hợp này tương đương với hàm trọng số là hop-count, được sử dụng mặc định trong hầu hết cơ chế thức định tuyến của mạng WSN. Với các trường hợp còn lại, bậc n càng tăng thì trọng số EWA_{ij} càng tăng đột biến khi mức tiêu thụ năng lượng tại nút J lớn, nghĩa là năng lượng còn lại nhỏ. Khi sử dụng hàm trọng số này để tìm tập lộ trình tốt nhất (nghĩa là tập lộ trình có trọng số nhỏ nhất), các lộ trình này sẽ hạn chế đi qua các nút có năng lượng còn lại nhỏ, nhằm kéo dài thời gian hoạt động của các nút này. Trong công trình này, chúng tôi xác định bậc n sao cho việc sử dụng năng lượng tại các nút là cân bằng nhất, nhằm giảm thiểu tình trạng một số nút tiêu thụ năng lượng lớn do tải luuw lượng cao. Việc xác định n được thực hiện thông qua mô phỏng ở phần sau.



Hình 2. Ảnh hưởng của năng lượng tiêu thụ tại nút J và bậc n đến trọng số EWA_{ij} theo hàm (1)

2. Thuật toán SEBR

Thuật toán 1 là mã giả lập của các bước thực thi thuật toán định tuyến được đề xuất, SEBR. Thuật toán này được thực thi tại bộ điều khiển SDN. Việc tính toán tập lộ trình đến BS cho tất cả các nút cảm biến được thực hiện định kỳ với một khoảng thời gian định trước, mà đó là IntervalTime ở bước 2. Cứ mỗi lần cập nhật lại bảng định tuyến, trọng số của tất cả các kết nối trong mạng được tính toán lại dựa trên tỷ số giữa năng lượng khởi đầu và năng lượng còn lại tại các nút cảm biến ở thời điểm xem xét. Trọng số này được cập nhật theo phương trình (1) (bước 5 trong thuật toán). Sau đó, bộ điều khiển SDN thực thi thuật toán *"đường ngắn nhất"*, cụ thể là thuật toán Dijkstra được sử dụng trong mô hình này để tìm tập lộ trình từ tất cả các nút cảm biến đến BS (bước 7). *"Đường ngắn nhất"* trong thuật toán SEBR được hiểu theo nghĩa tổng trọng số *EWA* được xác định theo phương trình (1) là nhỏ nhất. Thông tin các lộ trình tìm được sẽ được cập nhật đến các nút cảm biến sử dụng các gói điều khiển (bước 8).

Thuật toán 1: Tìm tập lộ trình từ các nút cảm biến đến BS trong mạng WSN sử dụng SDN

```
1 while (true) do
2     wait(IntervalTime);
3     Cập nhật năng lượng còn lại tại tất cả các nút
      cảm biến trong mạng thông qua các gói điều
      khiển;
4     for (mỗi kết nối từ nút I đến nút J) do
5         Tính trọng số EWA_ij theo (1);
6     end
7     Thực thi thuật toán "đường ngắn nhất" để tìm
      lộ trình từ các nút cảm biến đến BS;
8     Cập nhật thông tin các lộ trình tìm được đến
      các nút cảm biến thông qua các gói điều
      khiển;
9 end
```

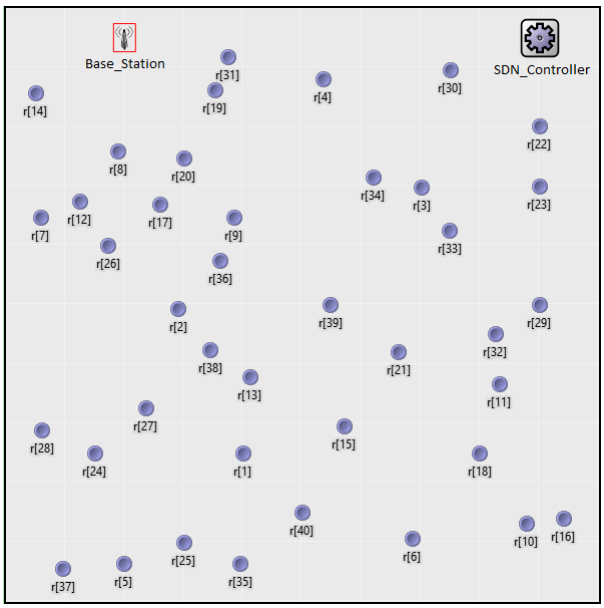
III. ĐÁNH GIÁ THỰC THI BẰNG MÔ PHỎNG

1. Kịch bản mô phỏng

Để đánh giá hiệu quả thực thi của thuật toán định tuyến được đề xuất, SEBR, chúng tôi sử dụng phương pháp mô phỏng, so sánh với hai giao thức định tuyến điển hình thường được sử dụng trong mạng WSN, đó là DSDV (Destination-Sequenced Distance-Vector) và AODV (Ad Hoc On-Demand Distance Vector) [14] về mặt năng lượng tiêu thụ, thời gian hoạt động của các nút và thông lượng trung bình trong toàn mạng. Mô phỏng được thực thi trên OMNeT++ [15] và INET Framwork [16]. Các kịch bản mô phỏng được cài đặt như ở Bảng 1. Chúng tôi thiết lập một mạng WSN trong vùng diện tích 1000×1000 $[m^2]$. Tổng số nút cảm biến biến đổi trong khoảng từ 30 đến 50 tùy theo kịch bản mô phỏng. Cụ thể là sử dụng 5 kịch bản với tổng số nút 30, 35, 40, 45 và 50. Các nút được bố trí ngẫu nhiên trong toàn bộ diện tích mạng. Hình 2 cho thấy một bản chụp giao diện trong quá trình thực thi mô phỏng với kịch bản 40 nút.

Bảng I
CÁC THAM SỐ CỦA KỊCH BẢN MÔ PHỎNG

Tham số	**Giá trị**
Vùng diện tích mô phỏng	1000×1000 $[m^2]$
Tổng số nút cảm biến	Từ 30 đến 50
Giao thức MAC	IEEE 802.11ac
Tốc độ dữ liệu	54 [Mbps]
Công suất phát	12 [dBm]
Độ nhạy thu	-76 [dBm]
Thuật toán định tuyến	DSDV, AODV and SEBR
Loại lưu lượng	UDP
Mô hình truyền sóng	Không gian tự do
Tần số sóng mang	2,4 [GHz]
Năng lượng ban đầu của các nút	2 [J]
Năng lượng còn lại khi nút tắt	$\leq 1\%$

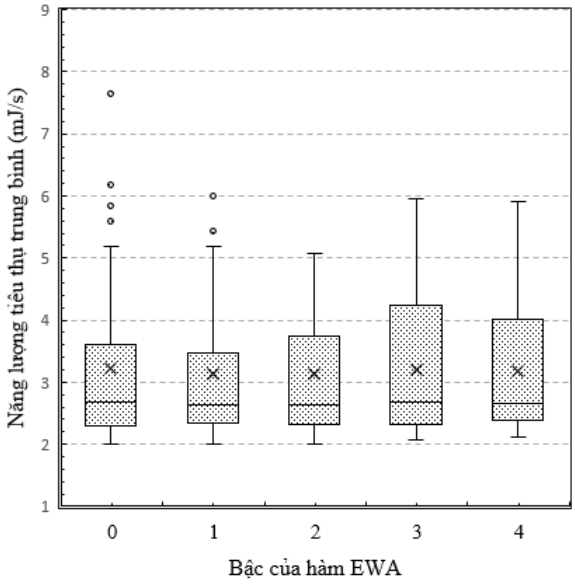


Hình 3. Một bản chụp của giao diện chương trình mô phỏng mạng WSN sử dụng SDN

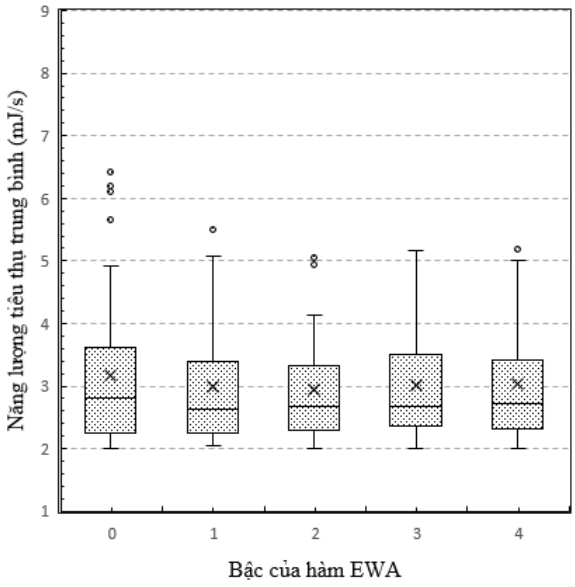
2. Kết quả mô phỏng

a) Xác định bậc n của hàm trọng số EWA_{ij}

Nội dung chính của phần này thực hiện một số kịch bản mô phỏng để xác định bậc n của hàm trọng số EWA_{ij} sao



Hình 4. So sánh mức tiêu thụ năng lượng trung bình trên tất cả các nút khi bậc của hàm trọng số (1) khác nhau khi tổng số nút cảm biến là 35

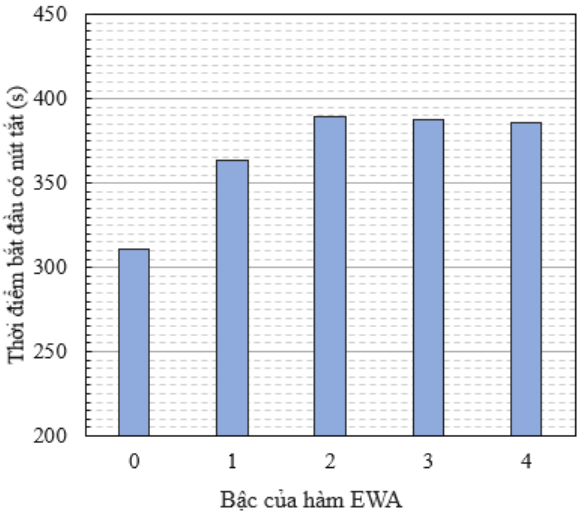


Hình 5. So sánh mức tiêu thụ năng lượng trung bình trên tất cả các nút khi bậc của hàm trọng số (1) khác nhau khi tổng số nút cảm biến là 45

cho hiệu quả sử dụng năng lượng tại các nút cảm biến là tốt nhất. Kết quả ở các Hình 4 và 5 cho thấy rõ sự khác nhau về năng lượng tiêu thụ trung bình tại tất cả các nút cảm biến khi bậc n khác nhau. Xét kịch bản có 35 nút cảm biến (kết quả ở Hình 4), khi $n = 0$ (tương đương với trường hợp định tuyến theo trọng số hop-count), mức tiêu thụ năng lượng tại các nút khác nhau rất lớn, phân bố trong khoảng từ 2,01 mJ/s đến 7,75 mJ/s. Đặc biệt là có 4 điểm ngoại lệ, đó là 4 nút có mức tiêu thụ năng lượng cao, các nút này sẽ sớm bị tắt do hết năng lượng. Khi $n = 1$, mức tiêu thụ năng lượng tại các nút cân bằng hơn, phân bổ trong khoảng từ từ 2,01 mJ/s đến 5,99 mJ/s. Tuy nhiên, vẫn còn 2 nút ngoại lệ với mức tiêu thụ năng lượng cao. Xét các trường hợp còn lại của n ta thấy rằng, trường hợp $n = 2$ cho kết quả tốt nhất, mức tiêu thụ năng lượng của tất cả các nút cân bằng nhất với vùng giá trị từ 2,00 mJ/s đến 5,02 mJ/s và không có ngoại lệ nào. Kết quả đạt được hoàn toàn tương tự đối với kịch bản mà tổng số nút cảm biến là 45 (Hình 5), trường hợp $n = 2$ mang lại mức tiêu thụ năng lượng tại tất cả các nút cân bằng nhất.

Một độ đo hiệu năng khác có tác động lớn đến hiệu năng của hệ thống mạng WSN là thời gian hoạt động của các nút cảm biến. Chúng tôi cũng đã phân tích độ đo này để các định bậc n của hàm (1) sao cho thời gian hoạt động của các nút cảm biến là dài nhất. Kết quả thu được như ở Hình 6 cho thấy rằng, trường hợp bậc $n = 2$ mang lại kết quả tốt nhất.

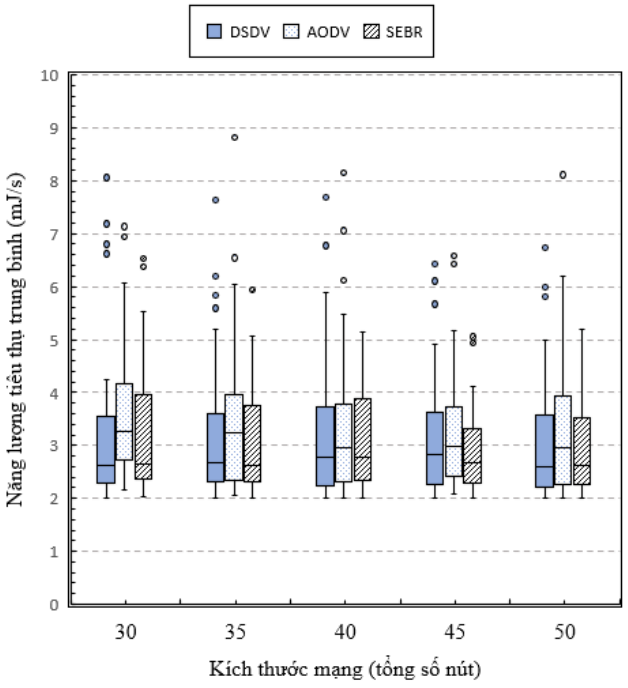
Trên cơ sở các kết quả khảo sát ảnh hưởng của n đến việc sử dụng năng lượng ở trên, chúng tôi chọn bậc n của hàm trọng số (1) là 2, giá trị này được sử dụng cho các kịch bản mô phỏng ở các phần còn lại dưới đây.



Hình 6. So sánh thời điểm bắt đầu có nút tắt do hết năng lượng khi bậc của hàm trọng số (1) khác nhau trong trường hợp tổng số nút cảm biến là 45

b) Phân tích mức tiêu thụ năng lượng tại mỗi nút

Trong phần này, chúng tôi so sánh mức tiêu thụ năng lượng tại tất cả các nút cảm biến của thuật toán được đề xuất so với các thuật toán DSDV và AODV. Kết quả thu được như cho thấy ở Hình 7. Khi sử dụng các thuật toán DSDV và AODV, có một số nút tiêu thụ năng lượng rất lớn, đó là các điểm ngoại lệ của các đồ thị hộp. Ví dụ, với kịch bản 30 nút và việc định tuyến được thực hiện bởi giao thức DSDV, có 4 ngoại lệ với giá trị trung bình là 6,61, 6,82, 7,23 và 8,11 mJ/s. Với kịch bản 35 nút cũng có 4 nút ngoại lệ tiêu thụ năng lượng lớn với mức trung bình là 5,51, 5,72, 6,15 và 7,82 mJ/s. Kết quả cũng hoàn toàn tương tự khi giao thức AODV được sử dụng để tìm tập lộ trình, tất cả các kịch bản mô phỏng với giao thức định tuyến này đều tồn tại các nút có mức tiêu thụ năng lượng cao. Cụ thể, với kịch bản 40 nút, có 3 nút tiêu thụ năng lượng lớn. Mức tiêu thụ trung bình của ba nút này là 8,16, 7,05 và 6,11 mJ/s. Các nút này sẽ bị tắt sớm do hết năng lượng, điều này ảnh hưởng đến việc truyền dữ liệu trong mạng. Nguyên nhân dẫn đến tình trạng này là do các nút trên (nút có mức tiêu thụ năng lượng là các điểm ngoại lệ trong đồ thị hộp) chịu tải lưu lượng lớn, bao gồm cả tải lưu lượng do chính nút đó phát ra và tải lưu lượng từ các nút khác truyền đến để chuyển tiếp về BS. Vì vậy, các nút này phải thường xuyên hoạt động với cường độ cao, dẫn đến tiêu thụ nhiều năng lượng. Với thuật toán SEBR, có nhiều điểm ngoại lệ đã được giải quyết. Điều này đồng nghĩa với

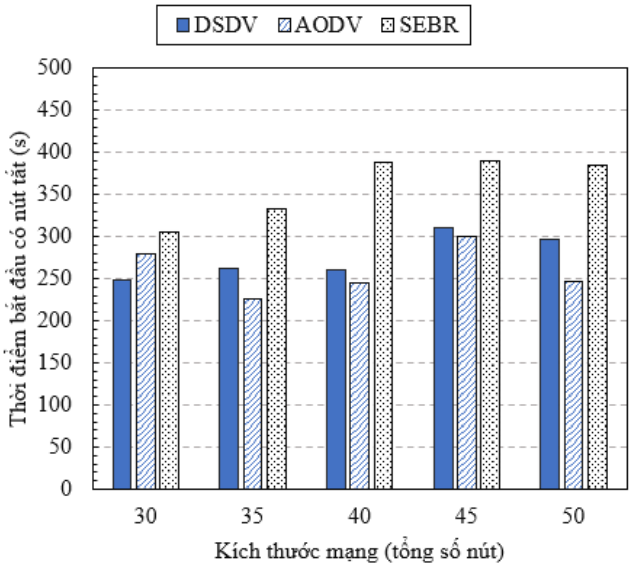


Hình 7. So sánh mức tiêu thụ năng lượng trên tất cả các nút khi sử dụng các thuật toán định tuyến DSDV, AODV và SEBR

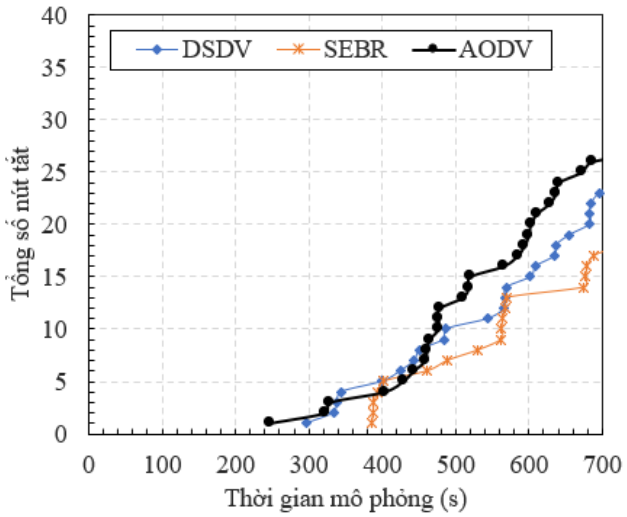
việc năng lượng tiêu thụ tại các nút đã cân bằng hơn. Cụ thể, với các kịch bản mô phỏng 40 và 50 nút thì không còn điểm ngoại lệ nào. Kịch bản 35 nút chỉ còn một nút ngoại lệ, kịch bản 45 nút còn 2 nút ngoại lệ. Tuy nhiên, mức tiêu thụ năng lượng tại các nút ngoại lệ này cũng đã giảm đi rõ ràng. Kết quả này đạt được là do thuật toán SEBR đã xem xét đến năng lượng còn lại tại các nút mỗi khi tính toán, cập nhật lại bảng định tuyến, dẫn đến tập lộ trình truyền dữ liệu trong mạng hạn chế đi qua các nút có năng lượng còn lại thấp hơn.

c) Phân tích thời gian hoạt động của các nút

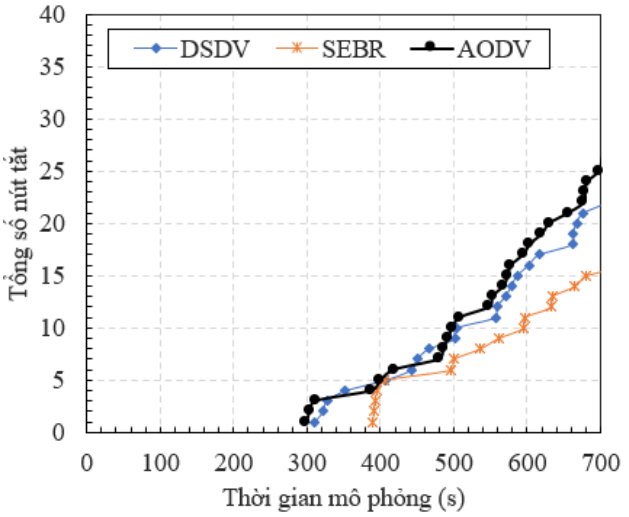
Vì mức sử dụng năng lượng tại các nút của thuật toán SEBR cân bằng hơn so với DSDV và AODV, nên thời gian hoạt động của các nút cũng kéo dài hơn. Kết quả này cho thấy rõ ở Hình 8, chúng tôi so sánh thời điểm bắt đầu có nút tắt do hết năng lượng của các thuật toán DSDV, AODV và SEBR. Ta thấy rằng, trong cả 5 kịch bản mà tổng số nút là 30, 35, 40, 45 và 50, thuật toán SEBR mang lại thời điểm có nút bắt đầu tắt lâu hơn so với các thuật toán DSDV và AODV. Cụ thể, với kịch bản mô phỏng 30 nút, tại thời điểm 248,1 giây bắt đầu có nút tắt nếu sử dụng thuật toán DSDV. Nếu sử dụng AODV, thời điểm bắt đầu có nút tắt là 280,15 giây. Trong khi đó, tại thời điểm 306,1 giây mới bắt đầu có nút tắt nếu sử dụng thuật toán SEBR. Như vậy, thuật toán SEBR đã kéo dài thời gian hoạt động ổn định của hệ thống mạng thêm 26 giây và 58 giây nếu so với các thuật toán AODV và DSDV theo thứ tự tương ứng. Trường hợp thời gian được kéo dài nhiều nhất là kịch bản 40 nút, thời gian kéo dài trong kịch bản này lên đến 128.2 giây khi so với thuật toán DSDV.



Hình 8. So sánh thời điểm bắt đầu có nút tắt do hết năng lượng của các thuật toán DSDV, AODV và SEBR



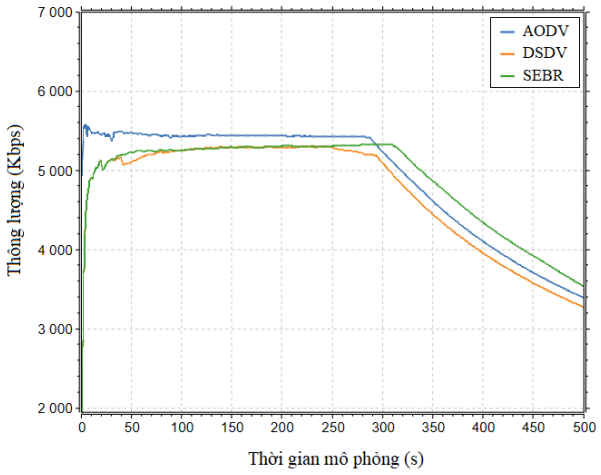
Hình 9. So sánh tổng số nút tắt theo thời gian mô phỏng của thuật toán DSDV, AODV và SEBR khi tổng số nút là 50



Hình 10. So sánh tổng số nút tắt theo thời gian mô phỏng của thuật toán DSDV, AODV và SEBR trong trường hợp tổng số nút là 45

Tiếp theo, chúng tôi khảo sát chi tiết về tổng số nút bị tắt do hết năng lượng theo thời gian mô phỏng. Các kết quả thu được như ở Hình 9 và Hình 10, tương ứng với các kịch bản mà tổng số nút là 45 và 50. Các kết quả thu được đã cho thấy rằng, với cả ba thuật toán SEBR, DSDV và AODV, tổng số nút bị tắt tăng dần theo thời gian mô phỏng. Điều này là hiển nhiên vì các nút hoạt động càng lâu thì năng lượng càng cạn kiệt. Tuy nhiên, nếu so sánh giữa các thuật toán SEBR, DSDV và AODV, trường hợp của thuật toán SEBR luôn có tổng số nút bị tắt nhỏ hơn hoặc bằng hai thuật toán còn lại.

Từ các kết quả khảo sát về việc tiêu thụ năng lượng tại các nút ở trên, chúng tôi có nhận xét rằng, thuật toán SEBR



Hình 11. So sánh thông lượng khi sử dụng thuật toán định tuyến AODV, DSDV và SEBR

hoạt động tốt hơn các thuật toán DSDV và AODV nếu xét về mặt hiệu quả sử dụng năng lượng. Cụ thể là cân bằng mức năng lượng được sử dụng tại các nút, kéo dài thời gian hoạt động ổn định của hệ thống mạng, giảm số nút bị tắt do hết năng lượng.

d) Phân tích thông lượng

Trong phần cuối cùng này, chúng tôi khảo sát thông lượng trung bình nhận được tại BS qua thời gian mô phỏng. Với độ đo này, thuật toán SEBR cũng thực thi hiệu quả hơn so với các thuật toán DSDV và AODV. Nhận định này thể hiện rõ bằng kết quả mô phỏng ở Hình 11. Trong đó, chúng tôi đo thông lượng trung bình nhận được tại BS theo thời gian mô phỏng. Các đồ thị trên Hình 11 cho thấy rằng, khi thời gian mô phỏng nhỏ hơn 250 giây, thông lượng trung bình của cả hai thuật toán DSDV và SEBR gần như là giống nhau, thuật toán AODV mang lại thông lượng cao nhất. Nguyên nhân là vì trong khoảng thời gian này, tình trạng một số nút bị tắt do hết năng lượng chưa xảy ra. Khi thời gian mô phỏng lớn hơn 250 giây, thông lượng của các thuật toán DSDV và AODV bắt đầu giảm. Nguyên nhân là do một số nút đã bị tắt, ảnh hưởng đến việc truyền dữ liệu trong toàn mạng. Đến thời điểm 321 giây, thông lượng của cả ba thuật toán DSDV, AODV và SEBR đều bắt đầu giảm, cũng với lý do một số nút đã bị tắt do hết năng lượng. Tuy nhiên, so sánh giữa ba thuật toán, thuật toán SEBR mang lại thông lượng trung bình cao nhất.

IV. KẾT LUẬN

Việc sử dụng năng lượng sao cho hiệu quả để kéo dài thời gian hoạt động của các nút cảm biến trong mạng WSN là điều hết sức cần thiết, đặc biệt là đối với các trường hợp mà WSN được triển khai cho các ứng dụng với tải lưu lượng lớn, yêu cầu độ tin cậy cao. Bài báo đã đề xuất một thuật toán định tuyến có xét đến mức tiêu thụ năng lượng giữa các nút cảm biến, được đặt tên là SEBR. Mục tiêu của thuật toán SEBR là cân bằng mức tiêu thụ năng lượng, giảm thiểu số nút phải tiêu thụ nhiều năng lượng nhằm tăng thời gian hoạt động của chúng. Phương pháp của chúng tôi là xây dựng một hàm trọng số cho các kết nối không dây có chứa các tham số về năng lượng còn lại tại các nút. Sau đó, sử dụng cơ chế định tuyến tập trung dựa trên kiến trúc mạng điều khiển bằng phần mềm (SDN) để tìm lộ trình có trọng số tốt nhất để truyền dữ liệu. Kết quả mô phỏng trên OMNeT++ cho thấy rằng, thuật toán được đề xuất cho phép tăng thời gian hoạt động của các nút, tăng thông lượng mạng so với các thuật toán DSDV và AODV.

Trong xu thế phát triển của công nghệ mạng không dây hiện nay, WSN tích hợp với mạng 5G (5G-based WSN) và điện toán đám mây (Cloud-based WSN) đang làm một xu hướng phát triển của công nghệ mạng WSN. Việc phát triển thuật toán cho các mô hình WSN này là điều cần thiết. Đây cũng là hướng phát triển tiếp theo của công trình nghiên cứu này.

TÀI LIỆU THAM KHẢO

[1] H. Mostafaei and M. Menth, "Software-defined wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 119, pp. 42–56, 06 2018.

[2] K. M. Modieginyane, B. B. Letswamotse, R. Malekian, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for efficient network management: A survey," *Computers Electrical Engineering*, vol. 66, pp. 274–287, 2018.

[3] B. B. Letswamotse, R. Malekian, C.-Y. Chen, and K. M. Modieginyane, "Software defined wireless sensor networks (sdwsn): A review on efficient resources, applications and technologies," *Journal of Internet Technology*, vol. 19, pp. 1303–1313, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:69822630

[4] S. Wang, A. Hawbani, X. Wang, O. Busaileh, and L. Ping, "Heuristic routing for software defined wireless sensor network," in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, 2018, pp. 39–44.

[5] S. Manisekaran and R. Venkatesan, "An analysis of software-defined routing approach for wireless sensor networks," *Computers Electrical Engineering*, vol. 56, 07 2016.

[6] R. Huang, Y. Dong, G. Bao, Y. Liu, M. Wei, J. Lu, and Y. Huo, "A new topology control algorithm in software defined wireless rechargeable sensor networks," *IEEE Access*, vol. 9, pp. 101 003–101 012, 2021.

[7] Z. Geng, W. Xia, W. Cao, T. Wu, F. Yan, L. Shen, and J. Pang, "An energy-efficient hierarchical topology control algorithm in software-defined wireless sensor network," in *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2021, pp. 1–6.

[8] Q. Liu, L. Cheng, R. Alves, T. Ozcelebi, F. Kuipers, G. Xu, J. Lukkien, and S. Chen, "Cluster-based flow control in hybrid software-defined wireless sensor networks," *Computer Networks*, vol. 187, p. 107788, 2021.

[9] B. Cao, S. Deng, H. Qin, and Y. Tan, "A novel method of mobility-based clustering protocol in software defined sensor network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, 04 2021.

[10] M. Alenazi and S. Monti, "Software-defined network-based energy-aware routing method for wireless sensor networks in industry 4.0," *Applied Sciences*, vol. 12, p. 10073, 10 2022.
[11] V. Duong Thi Thuy and L. Binh, "Irsml: An intelligent routing algorithm based on machine learning in software defined wireless networking," *ETRI Journal*, vol. 44, 08 2022.
[12] M. U. Younus, M. K. Khan, and A. R. Bhatti, "Improving the software-defined wireless sensor networks routing performance using reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3495–3508, 2022.
[13] S. Roy, R. Dutta, N. Ghosh, and P. Ghosh, "Adaptive motif-based topology control in mobile software defined wireless sensor networks," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE Press, 2021, p. 1–6. [Online]. Available: https://doi.org/10.1109/CCNC49032.2021.9369601
[14] Perkins, et. al., *Ad hoc On-Demand Distance Vector (AODV) Routing*. Network Working Group, 2003. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3561.html
[15] András Varga and OpenSim Ltd,, *OMNeT++ Simulation Manual - Version 6.x*, 2022. [Online]. Available: https://http://www.omnetpp.org/documentation/
[16] A. Virdis and M. Kirsche, *Recent advances in network simulation - The OMNeT++ environmentand its ecosystem*. Springer Nature Switzerland AG, 2019. [Online]. Available: https://http://www.omnetpp.org/documentation/

SƠ LƯỢC VỀ TÁC GIẢ

Lê Đức Huy tốt nghiệp Cử nhân ngành CNTT tại Trường Đại học Kinh doanh và Công nghệ Hà Nội, năm 2012. Tốt nghiệp Thạc sĩ ngành Khoa học máy tính tại Trường Đại học Công nghệ thông tin và Truyền thông Thái Nguyên, năm 2015. Hiện là nghiên cứu sinh tại Học viện Khoa học và Công nghệ (GUST), Viện Hàm lâm Khoa học và Công nghệ Việt Nam. Hiện công tác tại Khoa CNTT, Trường Đại học Kinh doanh và Công nghệ Hà Nội.
Lĩnh vực nghiên cứu: các giao thức điều khiển trong mạng máy tính và Viễn thông, an ninh mạng.
Email: huyld@hubt.edu.vn

Lê Hữu Bình tốt nghiệp Kỹ sư ngành Điện tử - Viễn thông tại Trường Đại học Bách Khoa, Đại học Đà Nẵng năm 2001, tốt nghiệp Thạc sĩ ngành Khoa học máy tính tại Trường Đại học Khoa học, Đại học Huế, năm 2007, bảo vệ thành công luận án Tiến sỹ năm 2020 tại Học viện Khoa học và Công nghệ (GUST), Viện Hàm lâm Khoa học và Công nghệ Việt Nam, ngành Máy tính, chuyên ngành Hệ thống thông tin. Hiện nay, ông đang công tác tại Khoa CNTT, Trường Đại học Khoa học, Đại học Huế.
Lĩnh vực nghiên cứu: công nghệ mạng không dây thế hệ mới, SDN (Software Defined Networking), mô phỏng và thiết kế mạng.
Email: lhbinh@hueuni.edu.vn

Đỗ Thành Công tốt nghiệp Cử nhân ngành CNTT tại Trường Đại học Công nghệ thông tin - Đại học Quốc gia Thành phố Hồ Chí Minh năm 2010, tốt nghiệp Thạc sĩ ngành Khoa học máy tính tại Học viện Kỹ thuật Quân sự năm 2013. Hiện là giảng viên khoa CNTT, trường Đại học Kinh doanh và Công nghệ Hà Nội.
Lĩnh vực nghiên cứu: ứng dụng AI trong xử lý và nhận diện hình ảnh, công nghệ mạng cảm biến không dây.
Email: congdt@hubt.edu.vn

Nguyễn Đỗ Hoàng Giang hiện là học sinh chuyên toán niên khóa 2021 – 2024 tại Trường Trung học Phổ thông chuyên Khoa học Tự nhiên, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội.
Lĩnh vực nghiên cứu: trí tuệ nhân tạo, máy tính nhúng và mạng máy tính.
Email: giangnguyen.061106@gmail.com