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