Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9)"; Cần Thơ, ngày 4-5/8/2016
DOI: 10.15625/vap.2016.00099

VỀ CẢI TIẾN PHƯƠNG PHÁP FUZZY RANDOM FOREST, ỨNG DỤNG CHO PHÂN LỚP DỮ LIỆU KHÔNG CHẮC CHẮN

Nguyễn Anh Thơ[1], Nguyễn Long Giang[1], Cao Chính Nghĩa[2]
[1]Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam
[2]Khoa Toán – Tin học, Học viện Cảnh sát nhân dân
natho@ioit.ac.vn, nlgiang@ioit.ac.vn, ccnghia@gmail.com

TÓM TẮT—*Các thuật toán khai phá dữ liệu và máy học truyền thống thực hiện phân lớp với dữ liệu đã được xử lý để loại bỏ dữ liệu nhiễu, dữ liệu thiếu chính xác và dữ liệu không đầy đủ, dữ liệu không chắc chắn. Chúng tôi phát hiện ra rằng độ chính xác phân lớp có thể được cải thiện với dữ liệu không chắc chắn khi sử dụng sức mạnh ngẫu nhiên của phương pháp Fuzzy Random Forest (FRF) để tăng sự đa dạng của cây và sự linh hoạt của tập mờ. Chúng tôi mở rộng phương pháp FRF để xử lý với bộ với các giá trị thiếu, dữ liệu không chắc với kỹ thuật cắt tỉa cây trước khi bổ sung vào trong rừng, mà rất có thể cải thiện được độ chính xác phân lớp và kích thước bộ nhớ lưu trữ các cây của FRF.*

Từ khóa— *Cây quyết định mờ, rừng ngẫu nhiên mờ, phân lớp mờ, phân hoạch mờ.*

I. GIỚI THIỆU

Phân lớp luôn luôn là vấn đề thách thức đối với dự liệu hiện nay, tăng cả về số lượng, độ phức tạp và tính đa dạng của dữ liệu. Đã có rất nhiều kỹ thuật và thuật toán giải quyết vấn đề phân lớp [1], [3], [6], [18]. Tuy nhiên, đa số các bài toán phân lớp này được áp dụng trên dữ liệu đầy đủ và được đo đạc chính xác. Nhưng trên thực tế các dữ liệu thu thập được hầu như không hoàn hảo, dữ liệu méo mó, dữ liệu không đầy đủ,... việc xử lý các dạng dữ liệu này rất khó khăn và tốn kém. Hơn nữa các thông tin này thường được điều chỉnh bởi các chuyên gia. Do đó, tính xác thực của dữ liệu trở nên mơ hồ. Vậy nên cần thiết xử lý trực tiếp các dạng thông tin này.

Trong bài báo này, chúng tôi sử dụng kỹ thuật phân lớp mờ [5], [6], [18] để đối phó với dữ liệu không chắc chắn (dữ liệu thiếu giá trị, dữ liệu mờ) bằng cách mở rộng phương pháp rừng ngẫu nhiên mờ (Fuzzy Random Forest - FRF) [14], [15], [16] đươc gọi là Improve Fuzzy Random Forest, viết tắt là IFRF. Phương pháp IFRF có cấu trúc cơ bản dựa trên FRF, nhưng khi phát triển cây quyết định mờ thực hiện phân vùng mờ dữ liệu không đầy đủ và dữ liệu mờ bằng cách sử dụng hàm thuộc hình thang [10] để lựa chọn thuộc tính. Sau đó tối ưu cây quyết định sử dụng phương pháp cắt tỉa cây dựa trên tối ưu giải thuật di truyền [9] trước khi bổ sung cây vào rừng. Mục đích, tăng độ chính xác phân lớp, dự báo và giảm không gian nhớ cần để lưu trữ các nút cũng như giảm hiện tượng overfitting dữ liệu.

Trong mục II chúng tôi trình bày phương pháp học, phân lớp sử dụng FRF[15] và kỹ thuật tổng hợp thông tin trong FRF. Mục III chúng tôi đề xuất mở rộng phương pháp FRF bằng kỹ thuật cắt tỉa cây sử dụng phương pháp tối ưu giải thuật di truyền [9] bằng cách kết hợp toán tử Crossover and Mutation để tạo ra các lai ghép thế hệ mới, hàm Fitness ước lượng giá trị của cá thể để lựa chọn thế hệ tiếp theo. Mục IV thực nghiệm sánh và đánh giá mô hình phân lớp IFRF. Chúng tôi thực hiện thử nghiệm phương pháp IFRF trên bộ dữ liệu không đầy đủ và dữ liệu mờ trong kho dữ liệu chuẩn UCI [4]. Phương pháp đánh giá chéo Cross Validate được sử dụng để kiểm chứng độ chính xác của mô hình phân lớp bằng IFRF. Bên cạnh đó chúng tôi cũng thực hiện so sánh độ chính xác phân lớp của IFRF với các thuật toán phân lớp khác như RF [11], FRF [15] và Boosting. Mục V tổng kết và hướng phát triển. Trong phần này chúng tôi tóm tắt các kết quả đã đạt được, và hướng phát triển trong tương lai. Cuối cùng là tài liệu tham khảo.

II. PHƯƠNG PHÁP FUZZY RANDOM FOREST (FRF)

Trong Random Forest của Breiman [11], mỗi cây xây dựng với kích thước tối đa và không cắt tỉa. Trong quá trình xây dựng mỗi cây trong rừng, mỗi khi cần tách nút, chỉ có một tập con ngẫu nhiên của tập tất cả các thuộc tính được xem xét và một lựa chọn ngẫu nhiên có hoàn lại được thực hiện cho mỗi phép tách nút. Kích thước của tập con này là tham số duy nhất trong rừng ngẫu nhiên. Kết quả là, một số thuộc tính (bao gồm cả thuộc tính tốt nhất) không được xem xét cho mỗi phép tách nút, nhưng một số thuộc tính được loại trừ lại có thể được sử dụng tách nút khác trong cùng một cây.

Rừng ngẫu nhiên [11] có hai yếu tố ngẫu nhiên, một là bagging được sử dụng lựa chọn tập dữ liệu được sử dụng như dữ liệu đầu vào cho mỗi cây; và hai là tập các thuộc tính được coi là ứng cử viên cho mỗi nút chia. Tính ngẫu nhiên nhằm tăng sự đa dạng của cây và cải thiện chính xác kết quả dự báo sau khi tổng hợp dự báo trên các cây trong rừng. Khi rừng ngẫu nhiên được xây dựng thì 1/3 đối tượng quan sát (exambles) được loại bỏ ra khỏi dữ liệu huấn luyện của mỗi cây trong rừng. Các đối tượng này được gọi là "Out of bag - OOB"[11]. Mỗi cây sẽ có các tập đối tượng OOB khác nhau. Các đối tượng OOB không sử dụng để xây dựng cây và được sử dụng thử nghiệm cho mỗi cây tương ứng [11]

A. Rừng ngẫu nhiên mờ (FRF)

Thuật toán 2.1. Fuzzy Random Forest (FRF)

FRF (***input***: E, Fuzzy Partition; ***output***: Fuzzy Random Forest)
Begin
1. Tạo tập con Sub: Lấy ngẫu nhiên có hoàn lại |E| mẫu từ tập dữ liệu huấn luyện E
2. Xây dựng ***cây quyết định mờ (Fuzzy Decision Tree - FDT)*** từ tập con Sub
3. Lặp lại bước 1 và bước 2 cho tới khi tất cả các cây quyết định mờ (FDT) được xây dựng.

End.

Thuật toán 2.2. Fuzzy Decision Tree
FuzzyDecisionTree(*input*: E, Fuzzy Partition; *output*: Fuzzy Decision Tree)
Begin
1. Khởi tạo các mẫu trong dữ liệu huấn luyện E với giá trị 1 ($\chi_{Fuzzy\_Tree,root}(e)=1$)
2. Đặt M là tập các thuộc tính, tất cả các thuộc tính được phân vùng theo phân vùng mờ (Fuzzy Partition)
3. Chọn thuộc tính để chia tại nút N
3.1. Lựa chọn ngẫu nhiên thuộc tính *e* từ tập các thuộc tính M
3.2. Tính **Information Gain** cho thuộc tính *e*, sử dụng giá trị $\chi_{Fuzzy\_Tree,root}(e)$ mỗi thuộc tính *e* trong nút N
3.3. Chọn thuộc tính *e* có Information Gain lớn nhất
4. Phân hoạch nút N theo thuộc tính *e* được chọn trong bước 3.3 và loại bỏ khỏ M. Đặt E_n là tập dữ liệu của mỗi nút con
5. Lặp lại bước 3 và 4 với mỗi (E_n ,*M*) cho tới khi phù hợp với điều kiện dừng *(stopping criteria)*

End.

Công thức tính giá trị Information Gain dựa trên thuật toán ID3 sử dụng phân vùng mờ hình thang [10]. Tư tưởng chính, mỗi thuộc tính $\{A_1, A_2, .., A_f\}$ được biểu diễn bởi một tập mờ hình thang, vì vậy mỗi nút trong của cây được chia dựa trên phân vùng số thuộc tính tạo ra nút con cho mỗi tập mờ. Phân vùng mờ mỗi thuộc tính đảm bảo đầy đủ (không có điểm trong miền nằm ngoài vùng mờ) và là phân vùng mờ mạnh (thỏa mãn $\forall x \in E, \sum_{i=1}^{f} \mu_{A_i}(x)=1$, với $\{A_1, A_2, .., A_f\}$ là các tập mờ của các phâp hoạch cho bởi hàm thuộc μ_{A_i}).

Hàm $\chi_{t,N}(e)$ được gọi là mức của mẫu *e* thỏa mãn điều kiện dừng của cây *t* tại nút *N*. Được xác định như sau:

- $\chi_{t,root}(e)=1$ với $e \in E$ có trong nút gốc của cây *t*
- $\chi_{fuzzy\_se\_partition}(e)>0$ và với $e \in E$ thuộc về một hoặc cả hai nút con. Được xác định như sau:
  - $\chi_{t,childnode}(e)=\chi_{t,node}(e)\times\mu_{fuzzy\_set\_partition}(e)$, nếu giá trị e được xác định
  - Hoặc $\chi_{t,childnode}(e)=\chi_{t,node}(e)\times\frac{1}{number\_output_{split}}$, nếu *e* có giá trị thiếu

Điều kiện dừng trong (*stopping criteria*) cho thuật toán 2 thỏa mãn một trong các trường hợp sau: (1) tất cả các mẫu *e* thuộc một nút; (2) số mẫu *e* thỏa mãn giá trị ngưỡng *x* cho trước; (3) Nút lá rỗng.

B. Phân lớp bằng rừng ngẫu nhiên mờ

Trong phần này miêu tả cách phân lớp sử dụng FRF. Đầu tiên chúng tôi giới thiệu các ký hiệu được sử dụng. Sau đó, chúng tôi xác định hai bước ứng dụng cây quyết định mờ trong FRF để xác định nhãn cho biến mục tiêu của mẫu.

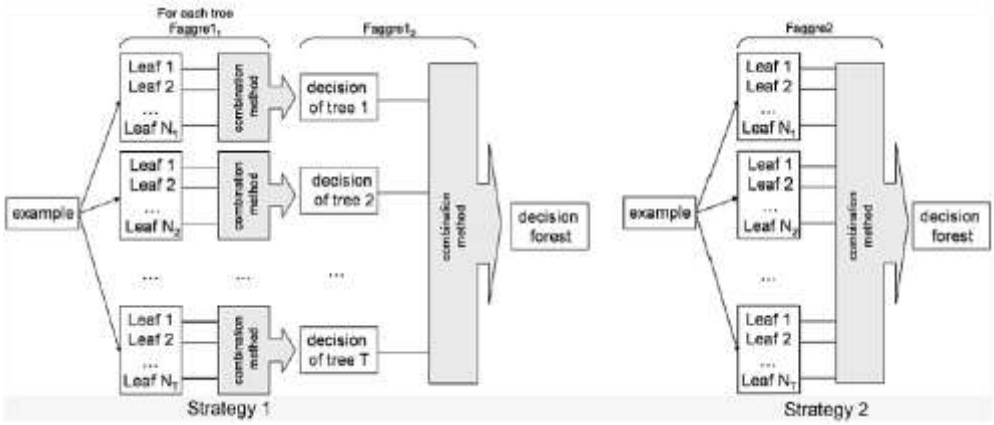
1. Các ký hiệu

- T là số cây trong rừng ngẫu nhiên mờ (FRF)
- N_t là tổng số nút lá trong cây thứ *t* với *t=1,2,...,T*. Đặc tính phân lớp của cây quyết định mờ là một mẫu có thể thuộc về một lá hoặc nhiều lá khác nhau do sự chồng chéo của tập mờ tạo ra một số phân hoạch mà một thuộc tính cùng tồn tại trên các phân hoạch khác nhau.
- I là tổng số lớp của dữ liệu mẫu.
- e là mẫu sử dụng huấn luyện hoặc kiểm tra.
- $\chi_{t,n}(e)$ là độ phụ thuộc mẫu *e* của nút lá *n* trên cây *t*
- *Support* là độ hỗ trợ của lớp *i* trong mỗi, lá bằng $Support(n)=\frac{E_i}{E_n}$ với E_i là tổng mức độ thuộc của các mẫu *e* trong lớp thứ *i* của nút lá *n*, E_n là tổng mức độ thuộc của đối tượng *e* trong nút lá *n*.

- L_FRF là ma trận có kích thước $\left(T \times MAX_{N_t}\right)$, với $MAX_{N_t} = \max\{N_1, N_2, .., N_t\}$, trong đó mỗi phần tử của ma trân là một véctơ kích thước *I* có *Support(i)* bằng độ hỗ trợ của nút lá *n* trên cây *t*. Một số phần tử của ma trận không chứa thông tin vì tất cả các cây không có lá nào đạt MAX_{N_t}. Tuy nhiên ma trận *L_FRF* bao gồm tất cả các thông tin được tạo ra bởi FRF, trong khi các thông tin này được sử dụng để phân lớp các mẫu *e*.
- $L\_FRF_{t,n,i}$ tham chiếu đến phần tử của ma trận chỉ ra độ hỗ trợ lớp *i* của nút lá *n* trên cây *t* .
- $T\_FRF_{t,i}$ là ma trận có kích thước $\left(T \times I\right)$ bao gồm độ chắc chắn (*confidence*) của mỗi cây *t* đối với mỗi lớp *i*.
- D_FRF_i là một véctơ có kích thước I, chỉ độ chắc chắn của *FRF* đối với mỗi lớp i.

2. Phân lớp trong rừng ngẫu nhiên mờ

Phân lớp mờ được P. Bonissone và các cộng sự [15] đưa ra hai dạng mô hình được gọi là *Strategy 1* và *Strategy 2* như sau:



Hình 2.1. Mô hình phân lớp mờ [15]

a) *Mô hình 1 (kí hiệu Strategy 1*)

Tổng hợp thông tin từ các lá trong mỗi cây quyết định khác nhau. Sau đó tổng hợp cây quyết định thì tạo được một rừng. Hàm $Faggre1_1$ sử dụng tổng hợp thông tin từ các lá trên mỗi cây, hàm $Faggre1_2$ sử dụng tổng hợp thông tin từ các cây quyết định. Mô hình phân lớp Strategy 1 được thực hiện bởi thuật toán 2.3 như sau:

Thuật toán 2.3. FRF Classification (Strategy 1)

FRFClassification(Input e, Fuzzy Random Forest; Output c)
Begin
DecisionsOfTrees(in: e,Fuzzy Random Forest; out: T_FRF);
DecisionOfForest(in: T_FRF; out: c);
End;
DecisionsOfTrees(in: e,Fuzzy Random Forest; out: T_FRF)
Begin
1) Tạo ma trận *L_FRF*
2) For each tree *t* do {For each class *i* do $T\_FRF_{t,i} = Faggre1_1\left(t, i, L\_FRF\right)$ }

End;
DecisionOfForest(in: T_FRF; out: c)
Begin
1) For each class i do $D\_FRF_i = Faggre1_2\left(i, T\_FRF\right)$
2) $c = \arg\max_{i,i=1..I}\left\{D\_FRF_i\right\}$

End;
Ma trận *L_FRF* và hàm tổng hợp thông tin *Faggre* được xác định như sau:
- Ma trận *L_FRF* được tạo ra bằng cách quét mâu *e* trên các cây *t*
- Các hàm tổng hợp thông tin *Faggre* coi như trọng số của cây trong *FRF* và xác định như sau:

$$Faggre1_1\left(t, i, L\_FRF\right) = \begin{cases} 1 \; if \; i = \arg\max\limits_{j; j=1,..,I}\left\{\sum\limits_{n=1}^{N_t} L\_FRF_{t,n,j}\right\} \\ 0 \; otherwise \end{cases} \quad (2.1)$$

$$Faggre1_2\left(i, T\_FRF\right) = \sum_{t=1}^{T} \mu\left(\frac{errors\left(OOB_t\right)}{size\left(OOB_t\right)}\right) \times T\_FRF_{t,i} \quad (2.2)$$

Với μ là hàm thuộc được xác định :

$$\mu(x)=\begin{cases}1 & 0\le x\le pmin+marg\\ \dfrac{(pmax+marg)-x}{pmax-pmin} & pmin+marg\le x\le pmax+marg\\ 0 & pmax+marg\le x\end{cases} \quad (2.3)$$

Trong đó: $pmax=max_{t=1,..,T}\left\{\dfrac{errors(OOB_t)}{size(OOB_t)}\right\}$ là tỷ lệ lỗi lớn nhất trong các cây của rừng, $\dfrac{errors(OOB_t)}{size(OOB_t)}$ tỷ lệ lỗi của cây *t*, $errors(OOB_t)$ số lỗi khi thực hiện phân lớp thực hiện trên cây *t* sử dụng dữ liệu kiểm thử *OOB*, $size(OOB_t)$ kích thước của dữ liệu kiểm tra *OOB* của cây *t*. $pmin$ là tỷ lệ lỗi của cây *t* và $marg=\dfrac{pmax-pmin}{4}$.

Các cây trong FRF bao giờ cũng có trọng số lớn hơn 0. Trọng số thể hiện tỷ lệ lỗi, vì thế cây có tỷ lệ lỗi thấp nhất thì có trọng số là 1.

b) *Mô hình 2 (kí hiệu Strategy 2)*:

Tổng hợp thông tin từ tất cả các lá trên tất cả các cây tạo thành rừng. Hàm Faggre2 được sử dụng tổng hợp thông tin từ tất cả các lá. Phân lớp theo mô hình Strategy 2 được thực hiện bởi thuật toán 2.4.

Thuật toán 2.4. FRF Classification (Strategy 2)

FRFclassification(in: e, Fuzzy Random Forest; out: c)

Begin

1. Tạo ma trận *L_FRF*
2. For each class i do $D\_FRF_i=Faggre2(i,L\_FRF)$
3. $c=\arg\max_{i,i=1..I}\{D\_FRF_i\}$

End;

Trong thuật toán này thì ma trận *L_FRF* được tạo ra thông qua chay mẫu *e* trên cây trong rừng và hàm tổng hợp thông tin Faggre 2 được xác định bởi công thức sau:

$$Faggre2(i,T\_FRF)=\sum_{t=1}^{T}\mu\left(\frac{errors(OOB_t)}{size(OOB_t)}\right)\times\sum_{n=1}^{N_t}T\_FRF_{t,n,i} \quad (2.4)$$

Với hàm thuộc $\mu\left(\dfrac{errors(OOB_t)}{size(OOB_t)}\right)$ được xác định tương tự thuật toán 2.3.

III. ĐỀ XUẤT PHƯƠNG PHÁP IFRF

Trong phần này chúng tôi đề xuất giải pháp mở rộng rừng ngẫu nhiên mờ được gọi là Improve Fuzzy Random Forest, viết tắt là IFRF. Phương pháp rừng ngẫu nhiên mờ FRF [15] dự trên RF [11]. Do vậy, FRF tạo cây theo mục tiêu lấy mẫu ngẫu nhiên có hoàn lại, cây không cắt tỉa, càng nhiều cây khác nhau càng tốt. Phương pháp FRF [11] được phát triển dựa trên RF sử dụng hàm thuộc trong lý thuyết mờ để xác định trọng số tổng hợp cây. Do đó, cây được tạo ra trong FRF cũng là cây không cắt tỉa. Cây không cắt tỉa là nguyên nhân dẫn đến sự mất cân bằng trên cây, ảnh hưởng đến độ chính xác phân lớp và dự báo, mất thời gian tìm kiếm và không gian lưu trữ các nút và gây ra hiện tương overfitting dữ liệu. Do đó, để cải thiện các vấn đề nêu trên chúng tôi đề xuất giải nháp cải tiến bằng cách cắt tỉa cây quyết định mờ (FDT) trước khi bổ sung vào FRF. Phương pháp được trình bày trong thuật toán 3.1 và 3.2 dưới đây:

Thuật toán 3.1. Improve Fuzzy Random Forest (EFRF)
IFRF(***input***: E, Fuzzy Partition; ***output***: Fuzzy Random Forest)
Begin

1. Tạo tập con sub data set(SDT): Lấy ngẫu nhiên có hoàn lại |E| mẫu từ tập dữ liệu huấn luyện E
2. Xây dựng ***cây quyết định mờ (Fuzzy Decision Tree - FDT)*** từ tập con SDT
3. Cây được cắt tỉa từ FDT gọi là FDTp
4. Lặp lại bước 1 và bước 3 cho tới khi tất cả các cây quyết định mờ (FDT) được xây dựng.

End.

Thuật toán 3.1 thực hiện kỹ thuật cắt tỉa cây sau khi xây dựng cây quyết định mờ (FDT). Do vậy, đây là kỹ thuật cắt tỉa sau khi xây dựng cây (Postpruning). Phương pháp cắt tỉa này không phụ thuộc vào giới hạn của cây, và được thực hiện cắt tỉa theo một điều kiện hoặc một phương pháp heuristic nào đó.

Brieman's với phương pháp cost-complexity pruning (CCP), và J. R. Quinlan với phương pháp Pessimistic Error Pruning (PEP) là kỹ thuật Postpruning đã chỉ ra rằng quá trình cắt tỉa làm giảm số cây con từ cây quyết định ban đầu và hiệu quả hơn các phương pháp pre-pruning.

Trong bài báo này, phương pháp tối ưu giải thuật di truyền [10], được ứng dụng để phát hiện cây con cần cắt tỉa bằng cách biểu diễn cây như chuỗi gen gồm các bít 0 (không cắt) hoặc 1 (cắt) được gọi là trọng số nhánh của cây. Sau đó, sử dụng các toán tử là Crossover và Mutation để lai tạo ra các thế hệ tiếp theo. Tiếp theo thực hiện lựa chọn cá thể trong quần thể để thực hiện lai tạo (sinh ra các cá thể cho thế hệ kế cận) trong các quần thể bằng cách xây dựng hàm Fitness. Fitness là một hàm ước lượng giá trị trọng số mỗi các thể trong quần thể. Cá thể được chọn theo một điều kiện trọng số nào đó. Từ các yếu tố trên chúng tôi đề xuất phương pháp cắt tỉa như sau:

Thuật toán 3.2. Cắt tỉa cây quyết định mờ

PruningFuzzyDecisionTree (input : T;Output: T')

Begin

1) Tạo ngẫu nhiên *h[P]* giả thuyết; Khởi tạo quần thể P
2) Tính hàm $Fitness(h_i) = \alpha N(T) + \beta E(T)$, Với N(T) là số nút của cây quyết định mờ T; E(T) là số lỗi của cây quyết định mờ T; α, β là hai trọng số chỉ kích cỡ và số lỗi của cây quyết định mờ
3) Tạo một thế hệ mới *Ps*
   a. Tính xác suất $Pr(h_i)$ giả thuyết h_i trong quần thể *P* theo công thức

$$Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_i)} \quad (3.1)$$

   b. Crossover: Chọn cặp giả thuyết có cùng giá trị $Pr(h_i)$ từ P. Ví dụ chọn cặp (h_1,h_2)có cùng giá trị xác xuất $Pr(h1)=Pr(h2)$. Sau đó tạo ra các con cặp (h_1,h_2) bằng cách áp dụng toán tử Crossover. Thêm tất cả các con vào Ps.
   c. Mutate: Chọn *m* phần trăm số giả thuyết của Ps có cùng một xác suất. Mỗi một giả thuyết chọn ngẫu nhiên một bít để nghịch đảo.
4) Cập nhật *Ps = P*
5) Lặp lại bước 2 đến 4 cho tới khi $Fitness(h) \le \delta$ (δ là gí trị ngưỡng có trước). Thu được cây T có các cạnh được gán giá trị trọng số là 1 tối đa.
6) Loại bỏ các cạnh có trọng số 1 có cây cắt tỉa T'

End;

IV. THỰC NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH PHÂN LỚP IFRF

Trong phần này, chúng tôi tiến hành thử nghiệm mô hình phân lớp IFRF trên 8 bộ dữ liệu trong kho dữ liệu UCI[4] được mô tả chi tiết trong bảng 4.1, với |E| là số mẫu, M là số thuộc tính, I là số lớp và Abbr là tên viết tắt của dữ liệu. Thực nghiệm được thực hiện đối với trường hợp dữ liệu mất giá trị và dữ liệu mờ, cho biết độ chính xác của mô hình bằng cách sử dụng phương pháp kiểm tra chéo Cross validation, số nút của IFRF trước và sau khi cắt tỉa.

Bảng 4.1. Dữ liệu thử nghiệm UCI [4]

Data set	Abbr (abbreviation)	|E|	M	I
Appendicitis	APE	106	7	2
Wisconsin breast C.	BCW	683	9	2
German credit	GER	1000	24	2
Glass	GLA	214	9	7
Ionosphere	ION	351	34	2
Iris plants	IRP	150	4	3
Pima Indian diabetes	PIM	768	8	2
Wine	WIN	178	13	3

Các tham số được thiết lập cho mô hình phân lớp IFRF như sau: Số cây là T(100,150); Số thuộc tính được chọn ngẫu nhiên $\log_2(|M|+1)$ với $|M|$ là số thuộc tính; Mỗi cây quyết định mờ của IFRF được xây dựng tối đa (nút có các mẫu cùng thuộc một lớp hoặc tập các biến thuộc tính là rỗng) và không cắt tỉa; a% (5%, 15% và 30%) giá trị không chắc chắn (giá trị thiếu hoặc giá trị mờ); Dữ liệu huấn luyện được lấy ngẫu nhiên bằng $a\% \times |E| \times |M|$ mẫu từ tập dữ liệu D $\left(DataTraina = Randomsiz(D, a\% \times |E| \times |M|)\right)$ và dữ liệu huấn luyên là phần còn lại sau khi đã lấy dữ liệu huấn luyện ra khỏi tập dữ liệu D $\left(DataTest = D(|E| - |DataTrain|)\right)$.

Để thấy được tính hiệu quả của phương pháp mở rộng IFRF đối với dữ liệu không chắc chắn (Dữ liệu mất giá trị và dữ liệu mờ). Chúng tôi sử dụng dữ liệu kiểm tra (DataTest) để đánh giá mô hình phân lớp của IFRF. Dữ liệu

kiểm tra được chia làm hai trường hợp: (1) Các giá trị bị mất có cả trên thuộc tính liên tục (thuộc tính số) và thuộc tính rời rạc; (2) Chuyển các thuộc tính số sang dạng dữ liệu mờ sử dụng hàm thuộc hình vuông [11] khác nhau cho các thuộc tính khác nhau.

Phương pháp sử dụng để đánh giá mô hình phân lớp IFRF là phương pháp kiểm tra chéo (Cross Validation) bằng cách chia tập dữ liệu thành 10 phần như nhau (10-fold cross validation) và thực hiện lặp 5 lần (5x10-fold cross validation). Độ chính xác phân lớp và số nút của mô hình bằng trung bình của 5 lần lặp. Kết quả thực nghiệm được miêu tả trong bảng 4.2 và bảng 4.3.

Bảng 4.2. Kết quả thử nghiệm với dữ liệu thiếu

Dữ liệu	Không cắt tỉa				Cắt tỉa			
	Số nút	Độ chính xác			Số nút	Độ chính xác		
		5%	15%	30%		5%	15%	30%
APE	12	90.31	90.1	90.92	8	91.13	90.35	86.42
BCW	165	97.19	96.52	94.39	89	97.31	95.12	92.89
GER	274	75.98	72.82	71.52	165	76.68	71.86	71.25
GLA	52	71.04	66.71	60.46	29	77.66	71.05	70.01
ION	86	95.47	93.75	90.32	58	96.41	93.18	91.79
IRP	13	96.1	93.22	80.62	5	97.33	96.03	94.38
PIM	145	76.32	74.57	69.67	55	77.14	75.55	73.58
WIN	9	93.46	91.6	83.66	7	97.87	96.01	93.47

Bảng 4.3. Kết quả thử nghiệm với dữ liệu mờ

Dữ liệu	Không cắt tỉa				Cắt tỉa			
	Số nút	Độ chính xác			Số nút	Độ chính xác		
		5%	15%	30%		5%	15%	30%
APE	15	91.13	90.52	90.76	8	90.92	91.34	91.97
BCW	150	97.31	96.61	93.51	78	97.73	96.89	93.63
GER	254	76.68	76.89	76.62	145	76.76	76.6	76.36
GLA	48	77.66	73.74	70.67	29	76.58	73.74	71.98
ION	85	96.41	95.42	93.35	52	96.94	95.88	94.29
IRP	13	97.33	96.02	92.09	5	98.64	96.02	92.09
PIM	142	77.14	76.45	73.57	53	77.66	76.62	75.06
WIN	9	97.87	97.67	94.28	7	97.58	97.16	95.03

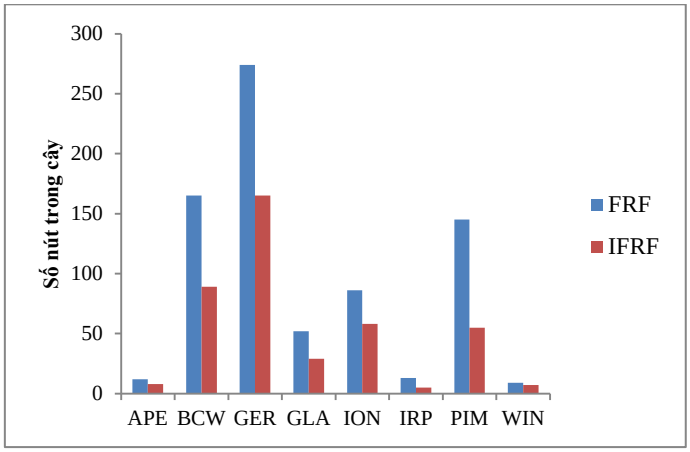
Để chứng minh tính hiệu quả phương pháp mở rộng IFRF, chúng tôi tiến hành thử nghiệm so sánh độ chính xác của thuật toán IFRF với một số thuật toán phân lớp mờ FRF và một số thuật toán phân lớp khác đó là RF và Boosting có cùng tham số đã thiết lập trên. Kết quả cho bảng 4.4.

Bảng 4.4. Kết quả thử nghiệm với dữ liệu thiếu 5%

Data Set	NoTree	RF	Boosting	FRF	IFRF
APE	140	89.15	87.35	90.31	91.13
BCW	125	97.07	94.51	97.30	97.73
GER	200	72.68	65.79	72.97	76.68
GLA	120	78.85	74.89	78.38	77.66
ION	175	93.45	94.09	94.66	95.79
IRP	120	95.33	96.67	97.33	98.38
PIM	150	75.26	66.18	76.53	76.58
WIN	150	98.03	97.20	97.48	98.47

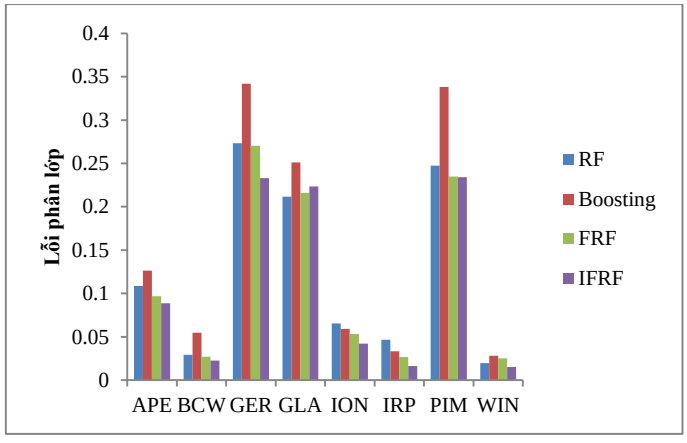
V. TỔNG KẾT

Trong bài báo này, chúng tôi đã đề xuất một phương pháp mở rộng FRF được gọi là IFRF bằng cách cắt tỉa cây quyết định mờ trước khi bổ sung vào tập cây trong rừng. Chiến lược cắt tỉa cây dựa trên giải thuật di truyền. Cách tiếp cận này của chúng tôi đã cho thấy được hiệu quả phân lớp, mà cụ thể độ chính xác phân lớp tốt hơn hẳn các phương pháp phân lớp quần thể khác như RF, Boosting và FRF hình 5.2. Điều này đã được chứng minh qua thử nghiệm trên các bộ dữ liệu thiếu giá trị và dữ liệu mờ. Đặc biệt thực nghiệm cho thấy số nút sử dụng cho cây giảm từ 20% đến 60% so với trước khi thực hiện cắt tỉa hình 5.1.



Hình 5.1. Biểu đồ so sánh số nút trong cây giữa FRF và IFRF

Biểu đồ hình 5.1 cho thấy số nút phương pháp mở rộng của chúng tôi sử dụng ít hơn rất nhiều so với phương pháp FRF. Điều này chứng tỏ bộ nhớ cần sử dụng để lưu trữ các nút trong cây của phương pháp mở rộng IFRF ít hơn phương pháp FRF.



Hình 5.2. Biểu đồ so sánh độ chính xác phân lớp

Kết quả hình 5.1 và hình 5.2. cho thấy phương pháp mở rộng IFRF của chúng tôi có độ chính xác tốt hơn các phương pháp phân lớp khác, và dung lượng sử dụng để lưu trữ cây thấp hơn hẳn so với các phương pháp phân lớp khác như FRF, RF và Boosting đối với dữ liệu không chắc chắn. Tuy nhiên, độ chính xác chưa được cải thiện nhiều, đây cũng là một khía cạnh mà chúng tôi quan tâm trong tương lai. Trong thực nghiệm này chúng tôi mới thực hiện thử nghiệm trên dữ liệu thiếu và dữ liệu mờ. Một khía cạnh nữa của dữ liệu không chắc chắn dữ liệu nhiễu và dữ liệu ngoại lai cũng luôn luôn xuất hiện trong quá trình thu thập và xử lý dữ liệu thực tế. Đây cũng là nhóm dữ liệu cần quan tâm xử lý trong tương lai.

VI. LỜI CẢM ƠN

Kết quả nghiên cứu này được tài trợ bởi Đề tài nghiên cứu mã số CS.16.16, cấp Viện CNTT, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.

TÀI LIỆU THAM KHẢO

[1] Amir Hussain, Erfu Yang "A Novel Classification Algorithm Based on Incremental Semi-Supervised Support Vector Machin", PLOS ONE | DOI:10.1371/journal.pone.0135709 August 14, 2015.

[2] Adriano Donato De Matteis; Francesco Marcelloni; Armando Segatori "A new approach to fuzzy random forest generation" Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on, 2015.

[3] Data Set UCI: https://archive.ics.uci.edu/ml/datasets/.

[4] Eyke Hüllermeier "Does machine learning need fuzzy logic", Fuzzy Sets and Systems 281(2015)292–299.

[5] Fernández-Delgado, Manuel, Eva Cernadas, Senén Barro, and Dinani Amorim "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" The Journal of Machine Learning Research 15, 2014 .

[6] Jesús Alcalá-Fdez, Rafael Alcalá,María José Gacto,Francisco Herrera "Learning the member ship function contexts formining fuzzy association rules by using genetic algorithms", Fuzzy Setsand Systems, 2009.

[7] Jooyeol Yun, Jun Won Seo, and Taeseon Yoon "The New Approach on Fuzzy Decision Forest", Lecture Notes on Software Engineering, Vol. 4, No. 2, May 2016.

[8] Jie Chen, Xizhao Wang, Junhai Zhai, "Pruning Decision Tree Using Genetic Algorithms", International Conference on Artificial Intelligence and Computational Intelligence, 2009.

[9] L. Breiman. Random forests. *Machine learning*, 45(1):5–32,2001.

[10] M. Zeinalkhani, M.Eftekhari "Comparing different stopping critetia for fuzzy decision tree induction through IDFID3", Iranian Journal of Fuzzy Systems Vol. 11, No. 1, (2014) pp. 27-48.

[11] Nikita Patel, Saurabh Upadhyay "Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA", International Journal of Computer Applications (0975 – 8887) Volume 60– No.12, December 2012.

[12] P. P. Bonissone, J. M. Cadenas, M. C. Garrido, R. A. D´ıaz-Valladares "A Fuzzy Random Forest: Fundamental for Design and Construction" Proceedings of IPMU'08, pp. 1231- 1238 Torremolinos (Malaga), June 22-27, 2008

[13] Piero Bonissone, José M. Cadenas, M. Carmen Garrido, R. Andrés Díaz-Valladares "A fuzzy random forest", International Journal of Approximate Reasoning 51 (2010) 729–747.

[14] P. P. Bonissone, J. M. Cadenas, M. C. Garrido, R. A. D´ıaz-Valladares, R. Mart´ınez "Weighted decisions in a Fuzzy Random Forest", IFSA-EUSFLAT 2009.

[15] Pragati Pandey, Minu Choudhary "Uncertain Data Management and Mining", IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), Vol. 2, No.6, December 2012.

[16] Renuka D. Suryawanshi, D. M. Thakore "Decision Tree Classification Implementation with Fuzzy Logic", IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.10, October 2012.

[17] S. Meenakshi, V. Venkatachalam "FUDT: A Fuzzy Uncertain Decision Tree Algorithm for Classification of Uncertain Data", research article - computer engineering and computer science, Arab J Sci Eng (2015) 40:3187–3196.

[18] Vitaly LEVASHENKO, Penka MARTINCOVÁ "Fuzzy decision tree for parallel processing support", Journal of Information, Control and Management Systems, Vol. 3, (2005), No. 1.

ABOUT IMPROVE FUZZY RANDOM FOREST METHODS, APPLICATIONS FOR CLASSIFICATIN UNCERTAIN DATA

Nguyen Anh Tho, Nguyen Long Giang, Cao Chinh Nghia

ABSTRACT*— The algorithms of data mining and machine learning to achieve classifiers with the data that has been processed to remove noise data, data inaccuracies, incomplete data and uncertain data. We recognize that classification accuracy could be improved with uncertain data when use random power of Fuzzy Random Forest method (FRF) to increase the diversity of plants and the flexibility of fuzzy sets. We expand the method FRF to handle the set with missing values, the data is not sure with techniques of tree pruning before adding into the forest, which can greatly improve the accuracy of classification and size of storage memory of FRF trees.*