

BỘ GIÁO DỤC  
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC  
VÀ CÔNG NGHỆ VIỆT NAM

**HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ**

---



**Triệu Thu Hương**

**NGHIÊN CỨU ĐỀ XUẤT HỆ HỌC CHUYỂN GIAO MỜ PHỨC  
DỰA TRÊN KỸ THUẬT LẤY MẪU KHÔNG GIAN CON VÀ  
CẤU TRÚC ĐỒ THỊ CÓ HƯỚNG**

**LUẬN ÁN TIẾN SĨ NGÀNH KHOA HỌC MÁY TÍNH**

*Hà Nội - 2024*

BỘ GIÁO DỤC  
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC  
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Triệu Thu Hương

NGHIÊN CỨU ĐỀ XUẤT HỆ HỌC CHUYÊN GIAO MỜ PHỨC  
DỰA TRÊN KỸ THUẬT LẤY MẪU KHÔNG GIAN CON VÀ  
CẤU TRÚC ĐỒ THỊ CÓ HƯỚNG

LUẬN ÁN TIẾN SĨ NGÀNH KHOA HỌC MÁY TÍNH  
Mã số: 9 48 01 01

Xác nhận của Học viện  
Khoa học và Công nghệ

Người hướng dẫn 1  
(Ký, ghi rõ họ tên)

Người hướng dẫn 2  
(Ký, ghi rõ họ tên)

Hà Nội - 2024

## LỜI CAM ĐOAN

Tôi xin cam đoan luận án “**Nghiên cứu đề xuất hệ học chuyển giao mờ phức dựa trên kỹ thuật lấy mẫu không gian con và cấu trúc đồ thị có hướng**” là công trình nghiên cứu của chính mình dưới sự hướng dẫn khoa học của cán bộ hướng dẫn. Luận án có sử dụng thông tin trích dẫn từ nhiều nguồn tham khảo khác nhau và các thông tin trích dẫn được ghi rõ nguồn gốc. Các kết quả nghiên cứu của tôi được viết chung với các tác giả khác đã được sự nhất trí của đồng tác giả khi đưa vào luận án. Các số liệu, kết quả được trình bày trong luận án là hoàn toàn trung thực và chưa từng được công bố trong bất kỳ một công trình nào khác.

Luận án được hoàn thành trong thời gian tôi làm Nghiên cứu sinh tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.

*Hà Nội, ngày 14 tháng 11 năm 2023*

Triệu Thu Hương

## LỜI CẢM ƠN

Luận án này đã hoàn thành nhờ sự nỗ lực không ngừng nghỉ của tác giả cùng với sự hỗ trợ tận tâm từ các thầy giáo hướng dẫn, đồng nghiệp, bạn bè và người thân.

Tác giả muốn bày tỏ lòng biết ơn chân thành và sâu sắc đến thầy giáo hướng dẫn là **PGS.TS Nguyễn Long Giang** và **PGS.TS Lê Hoàng Sơn**. Những lời hướng dẫn, sự động viên và tận tâm của họ dành cho tác giả trong suốt quá trình thực hiện luận án là không thể nào diễn đạt hết.

Tác giả muốn bày tỏ lòng biết ơn sâu sắc đến các giảng viên và cán bộ của phòng quản lý nghiên cứu sinh thuộc Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Sự hỗ trợ nhiệt tình và tạo điều kiện thuận lợi của họ đã đóng góp quan trọng vào việc hoàn thành luận án của tác giả.

Tác giả xin gửi lời cảm ơn tới toàn thể thành viên trong Lab AI 4.0 tại Viện Công nghệ thông tin - Đại học Quốc gia Hà Nội, những người đã đồng hành và hỗ trợ tác giả trong suốt quá trình nghiên cứu tại đó.

Tác giả cũng xin chân thành cảm ơn Ban Giám đốc Học viện Ngân hàng, Ban lãnh đạo Khoa và đồng nghiệp nơi tác giả hiện đang công tác. Họ đã tạo điều kiện rất nhiều để tác giả có thời gian tập trung vào nghiên cứu.

Đặc biệt, tác giả muốn bày tỏ lòng biết ơn sâu sắc đến Bố, Mẹ, Chồng và các chị trong gia đình. Họ đã luôn chia sẻ những khó khăn và động viên tác giả trong quá trình nghiên cứu. Luận án cũng là món quà tinh thần mà tác giả trân trọng gửi đến tất cả thành viên trong gia đình.

Tôi xin trân trọng cảm ơn!

**NCS Triệu Thu Hương**

## MỤC LỤC

<b>LỜI CẢM ƠN</b> . . . . .	ii
<b>DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT</b> . . . . .	vii
<b>MỞ ĐẦU</b> . . . . .	1
<b>CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU VÀ CƠ SỞ LÝ THUYẾT</b>	<b>8</b>
1.1 Suy diễn và hệ suy diễn mờ phức . . . . .	8
1.2 Tổng quan các nghiên cứu liên quan . . . . .	10
1.2.1 Các mô hình suy diễn mờ . . . . .	10
1.2.2 Học chuyển giao . . . . .	19
1.2.3 Học chuyển giao mờ . . . . .	24
1.2.4 Lấy mẫu và các phương pháp lấy mẫu . . . . .	26
1.3 Hạn chế của hệ suy diễn mờ phức và bài toán nghiên cứu . . . . .	29
1.4 Cơ sở lý thuyết . . . . .	31
1.4.1 Tập mờ và tập mờ phức . . . . .	31
1.4.2 Các phép toán trên tập mờ phức . . . . .	32
1.4.3 Hệ suy diễn mờ phức Mamdani . . . . .	35
1.4.4 Lý thuyết học chuyển giao . . . . .	38
1.5 Thực nghiệm . . . . .	40
1.5.1 Bộ dữ liệu thực nghiệm . . . . .	40
1.5.2 Tiền xử lý dữ liệu . . . . .	41
1.5.3 Môi trường thực nghiệm . . . . .	41
1.5.4 Các độ đo đánh giá thực nghiệm . . . . .	42
1.6 Kết luận chương 1 . . . . .	42
<b>CHƯƠNG 2. MÔ HÌNH HỌC CHUYỂN GIAO TRÊN HỆ SUY DIỄN MỜ PHỨC</b>	<b>43</b>

2.1	Một số khái niệm và định nghĩa . . . . .	44
2.2	Mô hình học chuyển giao mờ phức . . . . .	48
2.2.1	Chỉnh sửa tập mờ phức miền nguồn . . . . .	51
2.2.2	Lựa chọn các tập con dữ liệu . . . . .	52
2.2.3	Thích nghi luật mờ phức . . . . .	53
2.2.4	Kết hợp cơ sở luật mờ phức . . . . .	58
2.3	Phân tích độ phức tạp . . . . .	59
2.4	Thực nghiệm . . . . .	60
2.4.1	Kịch bản thực nghiệm . . . . .	60
2.4.2	Kết quả thực nghiệm . . . . .	61
2.5	Kết luận Chương 2 . . . . .	74

### **CHƯƠNG 3. HỆ HỌC CHUYỂN GIAO MỜ PHỨC DỰA TRÊN CẤU**

#### **TRÚC CFRG**

76

3.1	Giới thiệu . . . . .	76
3.2	Một số khái niệm và định nghĩa . . . . .	77
3.3	Các thuật toán trên cấu trúc CFRG . . . . .	83
3.3.1	Thuật toán thêm một luật vào CFRG . . . . .	85
3.3.2	Thuật toán duyệt luật mờ phức trên CFRG . . . . .	87
3.3.3	Thuật toán tìm kiếm một luật trên CFRG . . . . .	91
3.3.4	Thuật toán xóa một luật khỏi CFRG . . . . .	91
3.3.5	Thuật toán suy diễn mờ phức trên CFRG . . . . .	92
3.3.6	Thuật toán sửa một luật trên CFRG . . . . .	94
3.3.7	Thuật toán tách CFRG con từ CFRG cha . . . . .	99
3.3.8	Thuật toán trộn hai CFRG . . . . .	99
3.4	Mô hình học chuyển giao mờ phức dựa trên cấu trúc CFRG (CFRGTL) . . . . .	100
3.4.1	Khởi tạo các cấu trúc CFRG . . . . .	101
3.4.2	Tách các CFRG con từ các cấu trúc CFRG ban đầu . . . . .	102
3.4.3	Thích nghi luật trên cấu trúc CFRG . . . . .	102

3.4.4	Trộn các cấu trúc CFRG . . . . .	103
3.5	Thực nghiệm . . . . .	103
3.5.1	Kịch bản thực nghiệm thứ nhất . . . . .	103
3.5.2	Kịch bản thực nghiệm thứ hai . . . . .	104
3.6	Kết luận Chương 3 . . . . .	117
<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b>		<b>119</b>
<b>CÁC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ</b>		<b>121</b>
<b>TÀI LIỆU THAM KHẢO</b>		<b>123</b>
<b>PHỤ LỤC A. MỘT SỐ VÍ DỤ CHO CÁC THUẬT TOÁN TRÊN CẤU TRÚC CFRG</b>		<b>136</b>

## DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

STT	Từ viết tắt	Từ tiếng Anh	Diễn giải/Tạm dịch
1	ANCFIS	A neurofuzzy architecture employing complex fuzzy set	Một kiến trúc nơ-ron sử dụng tập mờ phức
2	ANFIS	Adaptive Neuro Fuzzy Inference System	Hệ suy diễn mờ nơ-ron thích nghi
3	CANFIS	Complex Neuro-Fuzzy Inference System	Hệ suy diễn mờ nơ-ron thích nghi phức
4	CFIS	Complex Fuzzy Inference System	Hệ suy diễn mờ phức
5	CFL	Complex Fuzzy Logic	Logic mờ phức
6	CFN	Complex fuzzy node	Nút mờ phức
7	FANCFIS	Fast adaptive neuro - complex fuzzy inference system	Hệ thống suy luận mờ phức - nơ-ron thích ứng nhanh
8	CFS	Complex Fuzzy Set	Tập mờ phức
9	CFTL	Complex fuzzy transfer learning	Học chuyển giao mờ phức
10	FIS	Fuzzy Inference System	Hệ suy diễn
11	FKG	Fuzzy Knowledge Graph	Đồ thị tri thức mờ
12	FS	Fuzzy Set	Tập mờ
13	M-CFIS	Mamdani Complex Fuzzy Inference System	Hệ suy diễn mờ phức Mamdani
14	M-CFIS-FKG	Mamdani Complex Fuzzy Inference System Fuzzy-Knowledge Graph	Hệ suy diễn mờ phức Mamdani - Đồ thị tri thức mờ



<b>STT</b>	<b>Từ viết tắt</b>	<b>Từ tiếng Anh</b>	<b>Diễn giải/Tạm dịch</b>
15	M-CFIS-R	Mamdani Complex Fuzzy Inference System Reduce Rule	Hệ suy diễn mờ phức Mamdani - giảm luật
16	TSK	Takagi-Sugeno	Hệ suy diễn mờ Takagi - Sugeno
17	SSL	Semi-Supervised Learning	Học bán giám sát
18	MVL	Multi-View Learning	Học đa hướng
19	MTL	Multi Task Learning	Học đa nhiệm vụ
20	TTL	Transductive Transfer Learning	Học chuyển giao chuyển tiếp
21	ITL	Inductive Transfer Learning	Học chuyển giao quy nạp
22	UTL	Unsupervised Transfer Learning	Học chuyển giao không giám sát
23	TL	Transfer Learning	Học chuyển giao
24	FTL	Fuzzy Transfer Learning	Học chuyển giao mờ
25	DT	Decision Tree	Cây quyết định
26	FDT	Fuzzy Decision Tree	Cây quyết định mờ
27	CFR	Complex Fuzzy Rule	Luật mờ phức
28	CFRs	Complex Fuzzy Rule Set	Tập luật mờ phức
29	AI	Artificial intelligence	Trí tuệ nhân tạo

30	ANCFIS-elm	A machine learning algorithm based on complex fuzzy sets	Thuật toán học máy dựa trên các tập mờ phức
31	CFRG	Complex fuzzy rule graph	Cấu trúc biểu diễn luật mờ phức dựa trên đồ thị
32	CFRGTL	Complex fuzzy rule graph transfer learning	Hệ học chuyển giao mờ phức dựa trên cấu trúc CFRG

## DANH MỤC CÁC HÌNH VẼ

1.1	Mô hình hệ suy diễn mờ tổng quát [47] . . . . .	11
1.2	Cấu trúc của CFIS [43] . . . . .	16
1.3	Phân loại học chuyển giao . . . . .	21
1.4	Các phương pháp lấy mẫu . . . . .	26
1.5	Mô hình bài toán học chuyển giao . . . . .	30
1.6	Một số hàm thuộc phổ biến . . . . .	31
1.7	Cấu trúc hệ M-CFIS [7] . . . . .	35
2.1	Mô hình học chuyển giao trên hệ suy diễn mờ phức . . . . .	49
2.2	Chi tiết mô hình học chuyển giao trên hệ suy diễn mờ phức . . . . .	51
2.3	Đánh dấu luật thích nghi và luật ứng cử . . . . .	56
2.4	Quá trình thêm luật và sửa luật . . . . .	58
2.5	Kịch bản thực nghiệm trên CFTL . . . . .	60
2.6	Kết quả đo lường độ chính xác trên tập dữ liệu Diabetes. . . . .	62
2.7	Kết quả đo lường số lượng luật sinh ra trên tập dữ liệu Diabetes. . . . .	63
2.8	Kết quả đo lường thời gian thực hiện trên tập dữ liệu Diabetes. . . . .	64
2.9	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập Diabetes. . . . .	64
2.10	Kết quả đo lường độ chính xác trên tập dữ liệu BreastCancer. . . . .	65
2.11	Kết quả đo lường số lượng luật trên tập dữ liệu Diabetes. . . . .	66
2.12	Kết quả đo lường thời gian thực hiện trên tập dữ liệu Diabetes. . . . .	67
2.13	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập BreastCancer. . . . .	67
2.14	Kết quả đo lường độ chính xác trên tập dữ liệu CreditCard. . . . .	68
2.15	Kết quả đo lường số lượng luật trên tập dữ liệu CreditCard. . . . .	69

2.16	Kết quả đo lường thời gian thực hiện trên tập dữ liệu CreditCard. . . . .	70
2.17	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập CreditCard. . . . .	70
2.18	Kết quả đo lường độ chính xác trên tập dữ liệu Liver. . . . .	71
2.19	Kết quả đo lường số lượng luật trên tập dữ liệu Liver. . . . .	72
2.20	Kết quả đo lường thời gian thực hiện trên tập dữ liệu Liver. . . . .	73
2.21	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập Liver. . . . .	73
3.1	Cấu trúc một nút mờ phức (a) và ví dụ một nút mờ phức (b) . . . . .	79
3.2	Một cấu trúc CFRG . . . . .	80
3.3	Sự khác nhau giữa cấu trúc CFRG và độ mạnh của cấu trúc CFRG. . .	81
3.4	Một danh sách <i>List_node</i> của cấu trúc CFRG. . . . .	84
3.5	Trường hợp sửa nút thứ nhất . . . . .	96
3.6	Trường hợp sửa nút thứ 2 - nhập nút . . . . .	97
3.7	Trường hợp sửa nút thứ hai -tách nút . . . . .	97
3.8	Mô hình học chuyển giao mờ phức dựa trên cấu trúc CFRG (CFRGTL)	101
3.9	Kết quả đo lường độ chính xác trên tập dữ liệu Diabetes. . . . .	105
3.10	Kết quả đo lường số lượng luật trên tập dữ liệu Diabetes. . . . .	106
3.11	Kết quả đo lường thời gian thực hiện trên bộ dữ liệu Diabetes. . . . .	107
3.12	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập Diabetes. . . . .	108
3.13	Kết quả đo lường độ chính xác trên bộ dữ liệu BreastCancer. . . . .	108
3.14	Kết quả đo lường số lượng luật trên bộ dữ liệu BreastCancer. . . . .	109
3.15	Kết quả đo lường thời gian thực hiện trên bộ dữ liệu BreastCancer. . .	110
3.16	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập BreastCancer. . . . .	111
3.17	Kết quả đo lường độ chính xác trên bộ dữ liệu CreditCard. . . . .	111
3.18	Kết quả đo lường số lượng luật trên bộ dữ liệu CreditCard. . . . .	112

3.19	Kết quả đo lường thời gian thực hiện trên bộ dữ liệu CreditCard. . . . .	113
3.20	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập CreditCard. . . . .	114
3.21	Kết quả đo lường độ chính xác trên bộ dữ liệu Liver. . . . .	114
3.22	Kết quả đo lường số lượng luật trên bộ dữ liệu Liver. . . . .	115
3.23	Kết quả đo lường thời gian thực hiện trên bộ dữ liệu Liver. . . . .	116
3.24	Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập Liver. . . . .	116
1	Thêm luật CFR1 vào CFRG . . . . .	137
2	Thêm luật CFR2 vào CFRG . . . . .	138
3	Duyệt luật mờ phức có chỉ mục 2 theo chiều sâu . . . . .	139
4	Duyệt luật mờ phức có chỉ mục là 2 theo chiều sâu (tiếp) . . . . .	140
5	Duyệt cấu trúc CFRG theo chiều rộng . . . . .	142
6	Suy diễn mờ phức trên cấu trúc CFRG . . . . .	143
7	Tách một cấu trúc CFRG con từ cấu trúc CFRG cha . . . . .	144

## **DANH MỤC CÁC BẢNG BIỂU**

1.1	Phân loại hệ học chuyển giao mờ . . . . .	25
1.2	Danh sách bộ dữ liệu thực nghiệm chuẩn . . . . .	40
1.3	Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh gan . . . . .	41
3.1	Bảng kết quả thực nghiệm theo kịch bản 1 . . . . .	104

## MỞ ĐẦU

### *Tính cấp thiết của luận án nghiên cứu*

Suy diễn là quá trình nhằm mục đích đưa ra các kết luận mới hoặc quyết định giải quyết vấn đề dựa trên việc sử dụng thông tin có sẵn. Suy diễn luôn có vai trò quan trọng và được áp dụng rộng rãi trong cuộc sống hàng ngày cũng như trong nhiều lĩnh vực khác nhau. Trong cuộc sống hàng ngày, suy diễn giúp con người đưa ra các quyết định dựa trên thông tin hiện có, định rõ các phương án và giải quyết các tình huống phức tạp. Trong kinh doanh, nó có thể hỗ trợ rất lớn trong hoạt động dự đoán xu hướng thị trường, nhờ đó đưa ra các quyết định về sản xuất và tiếp thị. Với lĩnh vực khoa học và nghiên cứu, suy diễn giúp xây dựng và kiểm tra các lý thuyết, đồng thời giúp các nhà khoa học đưa ra các kết luận dựa trên dữ liệu và thông tin có sẵn. Trong lĩnh vực trí tuệ nhân tạo (AI), máy tính sử dụng suy diễn để làm việc với dữ liệu, học từ dữ liệu từ đó giúp đưa ra dự đoán hoặc quyết định.

Quá trình suy diễn nói chung có cơ sở là những thông tin, dữ liệu lịch sử, dữ liệu đã có sẵn. Tuy nhiên, cùng với sự phát triển của khoa học công nghệ kéo theo sự thay đổi lớn về dữ liệu. Dữ liệu càng gia tăng cả mặt chất và lượng, kèm theo đó là những thông tin bổ sung trên dữ liệu như tính mơ hồ, không chắc chắn, tính chất chu kỳ và tần suất trong dữ liệu. Chính vì sự gia tăng về độ phức tạp của dữ liệu nên tác động khá nhiều tới chất lượng của quá trình suy diễn.

Để giải quyết vấn đề không chắc chắn và mơ hồ cũng như tính chu kỳ và tần suất có trong dữ liệu, Ramot và cộng sự [1] đã bổ sung thêm thành phần pha thông qua khái niệm tập mờ phức ( Complex fuzzy set - CFS ). Thành phần pha bổ sung nhằm mục đích biểu diễn các hiện tượng thời gian và tính chu kỳ trong dữ liệu. Trên cơ sở đó, các tác giả cũng đề xuất hệ suy diễn mờ phức (CFIS) trên cơ sở hệ suy diễn mờ (FIS) để giải quyết vấn đề ra quyết định. Rất nhiều nghiên cứu phát triển mở rộng từ hệ suy diễn mờ phức như mô hình ANCFIS (Hệ thống suy luận mờ phức thần kinh

thích nghi) [2] và các đề xuất mở rộng ANCFIS-ELM, FANCFIS [3], [4] là sự kết hợp của hệ thống mờ phức với mạng nơ-ron. Gần đây nhất, hệ suy luận mờ phức Mamdani (M-CFIS) [5] đã được giới thiệu với một cấu trúc suy luận dựa trên tập mờ phức và ứng dụng cho hệ hỗ trợ ra quyết định. Dựa trên mô hình M-CFIS, hai cải tiến bao gồm: giảm luật cho hệ M-CFIS [6] và M-CFIS cho đồ thị tri thức [7], đã được đề xuất nhằm cải thiện quá trình huấn luyện và kiểm thử trong M-CFIS. Theo hiểu biết của tôi, các nghiên cứu được đề cập trên đây là những hệ suy luận mờ phức điển hình nhất xử lý dữ liệu có yếu tố chu kỳ trong các hệ thống tri thức.

Học chuyển giao (TL) là quá trình sử dụng tri thức đã có cho một nhiệm vụ học tập mới liên quan hướng tới 2 mục tiêu:

- Tận dụng tri thức đã có, giải quyết các vấn đề về thiếu hụt tri thức và dữ liệu của nhiệm vụ học tập mới.
- Giảm thiểu thời gian học tập nhiệm vụ mới.

Kỹ thuật học chuyển giao được áp dụng trong học máy và trí tuệ nhân tạo đã mang lại hiệu suất tốt hơn trong những tình huống thiếu hụt về dữ liệu và tri thức [8]–[11]. TL đã đạt được những kết quả ấn tượng trong phân loại văn bản [12], xử lý ảnh [13], [14], chẩn đoán y tế [15]–[19], phát hiện đối tượng và nhận diện hành vi [20]–[24], mô hình mạng [25] và dự báo thị trường chứng khoán [26] v.v.

TL đang phát triển mạnh mẽ, tuy nhiên các thông tin không chắc chắn và mơ hồ trong các bài toán vẫn là một hạn chế không nhỏ trong việc huấn luyện và xây dựng các mô hình [26]. Để giải quyết những vấn đề này, Jethro và Simon [27] đã giới thiệu mô hình học chuyển giao mờ (FTL), đây là một khung cho TL kết hợp với lý thuyết mờ để nhằm chuyển giao thông tin không chắc chắn, có tính mơ hồ và được ứng dụng cho các mô hình thông minh. Lý thuyết tập mờ (FS) đã trở thành một công cụ phổ biến và hiệu quả để xử lý dữ liệu có tính mơ hồ và không chắc chắn. Việc kết hợp lý thuyết mờ với học chuyển giao đã giải quyết được những tình huống thiếu hụt thông tin, kèm theo sự mơ hồ và không chắc chắn [26], [28], [29].



Mặc dù có những tiến bộ trong các hệ thống suy diễn mờ phức và học chuyển giao mờ, nhưng vẫn còn một số hạn chế chưa được giải quyết, bao gồm:

- Mô hình M-CFIS là một mô hình điển hình cho xử lý dữ liệu có yếu tố không chắc chắn và tính chu kỳ. Tuy nhiên, nhược điểm chính của mô hình này là cơ sở luật được tạo ra trực tiếp từ toàn bộ dữ liệu mà chưa thực sự học. Các mô hình như vậy thường có khả năng thích ứng kém, khó khăn trong xử lý thông tin phức tạp, nơi mà mối quan hệ giữa dữ liệu không rõ ràng hoặc thay đổi theo thời gian.
- Ngoài ra, khả năng xử lý dữ liệu có yếu tố không chắc chắn mang tính chu kỳ và tần suất của hệ M-CFIS đã làm cho nó bị hạn chế về mặt thời gian xử lý. Chính thành phần pha sinh ra trong quá trình mô tả các yếu tố chu kỳ/định kỳ làm thời gian xử lý tăng lên.
- Trong các trường hợp, khi dữ liệu ngày càng nhiều và được cập nhật liên tục thì việc xây dựng mô hình M-CFIS theo phương án cập nhật luật truyền thống là không khả thi và tốn quá nhiều thời gian.
- Hầu hết các hệ thống FTL hiện tại chỉ dừng lại ở việc kết hợp kỹ thuật TL với logic mờ truyền thống. Trong các nghiên cứu gần đây, có rất ít và thậm chí không có nghiên cứu về FTL trên tập mờ mở rộng, đặc biệt là tập mờ phức.

Lý thuyết và suy diễn mờ phức có ý nghĩa trong giải quyết các bài toán có tính chất mờ hồ, không chắc chắn và chu kỳ/định kỳ. Nhưng hạn chế về phạm vi ứng dụng do yếu tố thời gian. Trong khi đó, kỹ thuật học chuyển giao lại có khả năng giảm thiểu thời gian học tập. Chính vì vậy, luận án này đặt ra nhiệm vụ Nghiên cứu là giải quyết hạn chế của hệ suy diễn mờ phức về mặt thời gian (hoàn thiện các nghiên cứu về hệ suy diễn mờ phức) dựa trên kỹ thuật học chuyển giao. Cụ thể là cải thiện hiệu suất về mặt thời gian cho việc xây dựng các mô hình suy diễn mờ phức cho miền mục tiêu/miền đích (miền dữ liệu cần xây dựng hệ suy diễn mờ phức) và thời gian suy diễn của hệ suy diễn mờ phức.

***Mục tiêu nghiên cứu:****Mục tiêu chung của luận án*

Mục tiêu chung của luận án, là nghiên cứu phát triển hệ học chuyển giao trên mô hình suy diễn mờ phức nhằm tận dụng những tri thức đã có trong những mô hình trước và giảm thiểu thời gian trong xây dựng hệ suy diễn mờ phức cho miền mục tiêu.

*Mục tiêu cụ thể*

Xuất phát từ mục tiêu tổng quát, các mục tiêu cụ thể của luận án bao gồm:

- *Mục tiêu 1:* Đề xuất một mô hình học chuyển giao dựa trên hệ suy diễn mờ phức Mamdani (M-CFIS), ứng dụng xây dựng hệ suy diễn mờ phức cho miền mục tiêu.
- *Mục tiêu 2:* Đề xuất cách biểu diễn tri thức (luật mờ phức) trên cấu trúc dữ liệu mới nhằm mục đích cải tiến hoạt động suy diễn về mặt thời gian và hỗ trợ hoạt động chuyển giao kiến thức nhanh chóng và hiệu quả.

***Nội dung nghiên cứu:***

Để đạt được các mục tiêu nghiên cứu, luận án tập trung vào một số nội dung chính sau:

- Nghiên cứu phát triển hệ suy diễn mờ phức dựa trên kỹ thuật học chuyển giao
- Nghiên cứu cấu trúc đồ thị biểu diễn luật mờ phức cho việc hợp nhất, suy diễn trong quá trình thích nghi luật trên hệ học chuyển giao mờ phức.

***Đối tượng và phạm vi nghiên cứu:****Đối tượng nghiên cứu*

Đối tượng nghiên cứu của luận án là các hệ suy diễn theo tiếp cận tập mờ phức và kỹ thuật học chuyển giao.

*Phạm vi nghiên cứu*

Dựa trên mục tiêu và nội dung nghiên cứu, phạm vi nghiên cứu của luận án được xác định như sau:

- **Lý thuyết:** Nghiên cứu mô hình học chuyển giao trên hệ suy diễn mờ phức trong ngữ cảnh nhiệm vụ nguồn và nhiệm vụ đích giống nhau, nhiệm vụ nguồn và nhiệm vụ đích cùng phân phối, cùng số lượng thuộc tính nhưng khác nhau về khoảng miền giá trị của các thuộc tính.
- **Thực nghiệm:** Thực nghiệm học chuyển giao trên hệ suy diễn mờ phức trong trường hợp nhiệm vụ nguồn và nhiệm vụ đích giống nhau, nhiệm vụ nguồn và nhiệm vụ đích cùng phân phối, cùng số lượng thuộc tính nhưng khác nhau về khoảng miền giá trị của các thuộc tính.

***Phương pháp nghiên cứu:***

Phương pháp nghiên cứu của luận án là nghiên cứu lý thuyết và nghiên cứu thực nghiệm

- Nghiên cứu lý thuyết: Nghiên cứu tổng quan lý thuyết tập mờ phức, hệ suy diễn mờ phức, kỹ thuật và mô hình học chuyển giao, phân tích ưu điểm, nhược điểm và các vấn đề còn tồn tại của các nghiên cứu liên quan. Tổng hợp các nghiên cứu liên quan về tập mờ, tập mờ phức, hệ suy diễn mờ, hệ suy diễn mờ phức và kỹ thuật học chuyển giao, học chuyển giao mờ. Trên cơ sở đó đề xuất mô hình học chuyển giao trên hệ suy diễn mờ phức và kỹ thuật cải tiến nhằm thực hiện mục tiêu được đề ra.
- Nghiên cứu thực nghiệm: Các mô hình và thuật toán đề xuất được cài đặt, chạy thử nghiệm, so sánh, đánh giá với mô hình tương ứng theo phương pháp truyền thống trên tập dữ liệu mẫu từ kho dữ liệu UCI và tập dữ liệu thực tế tại Bệnh viện Gang thép Thái Nguyên nhằm minh chứng về tính hiệu quả của các nghiên cứu về lý thuyết.

Các đóng góp chính của luận án bao gồm các nội dung sau:

- **Đề xuất được mô hình học chuyển giao trên hệ suy diễn mờ phức**

- Mô hình học chuyển giao được đề xuất thực hiện tái sử dụng các tri thức thu được từ CFIS của một miền có liên quan (gọi miền nguồn). Bằng cách kết hợp kỹ thuật học chuyển giao và cơ chế suy diễn của hệ CFIS nhằm làm giảm thời gian xây dựng hệ CFIS cho một miền khác (gọi là miền đích).
- Mô hình đề xuất bao gồm các giai đoạn 1) lựa chọn dữ liệu thích nghi của miền đích, 2) hiệu chỉnh miền đầu vào, 3) thích nghi luật và 4) tổng hợp luật.
- Các kết quả lý thuyết và thực nghiệm cho thấy khả năng cải thiện về mặt thời gian khi xây dựng hệ suy diễn mờ phức cho miền đích bằng kỹ thuật học chuyển giao và cơ chế hệ suy diễn mờ phức. Điều này góp phần không nhỏ trong mở rộng phạm vi ứng dụng của mô hình suy diễn mờ phức trong trường hợp ràng buộc hạn chế về mặt thời gian hay trường hợp dữ liệu lớn và cập nhật liên tục.
- Các đóng góp này được trình bày trong nội dung **Chương 2** của luận án.
- **Đề xuất một cấu trúc dữ liệu mới - CFRG biểu diễn tập luật mờ phức ứng dụng cho mô hình học chuyển giao trên hệ suy diễn mờ phức**
  - Cấu trúc CFRG được đề xuất nhằm biểu diễn tập luật mờ phức cho nhiệm vụ suy diễn mờ phức, giúp giảm thời gian của quá trình suy diễn mờ phức. Tăng khả năng ứng dụng của các mô hình suy diễn mờ phức cho các bài toán trong thực tế, đặc biệt là các bài toán có dữ liệu lớn, dữ liệu được cập nhật liên tục.
  - Cấu trúc CFRG biểu diễn thành phần biên độ và pha của luật trên mỗi nút giúp dễ dàng lựa chọn giá trị khi chỉnh sửa luật trong quá trình thích nghi luật, đẩy nhanh thời gian thích nghi và tăng tính chính xác của mô hình.
  - Các đóng góp này được trình bày chi tiết trong **Chương 3** của luận án.

***Bố cục của luận án:***

Luận án “**Nghiên cứu đề xuất hệ học chuyển giao mờ phức dựa trên kỹ thuật lấy mẫu không gian con và cấu trúc đồ thị có hướng**” gồm có phần mở đầu, 3 chương nội dung, phần kết luận và danh mục các tài liệu tham khảo với các nội dung chính sau:

- **Mở đầu:** Giới thiệu ngữ cảnh của nghiên cứu và đánh giá vai trò khả năng của hệ suy diễn, hệ suy diễn mờ phức, kỹ thuật học chuyển giao cũng như những hạn chế của nó; các vấn đề nghiên cứu; mục tiêu nghiên cứu; hướng tiếp cận và phương pháp nghiên cứu; nội dung nghiên cứu; phạm vi và giới hạn của nghiên cứu; các đóng góp chính và bố cục của luận án.
- **Chương 1:** Trình bày kiến thức cơ sở cho luận án nghiên cứu bao gồm: Khái niệm về tập mờ, tập mờ phức, hệ suy diễn mờ phức, mô hình học chuyển giao, học chuyển giao mờ. Bài toán nghiên cứu, dữ liệu và môi trường thực nghiệm cũng được giới thiệu tại chương này.
- **Chương 2:** Trình bày chi tiết quá trình xây dựng mô hình học chuyển giao trên hệ suy diễn mờ phức (gồm 4 giai đoạn) và các kết quả thực nghiệm trên các bộ dữ liệu UCI và bộ dữ liệu thực cùng với những phân tích đánh giá mô hình đã đề xuất.
- **Chương 3:** Trình bày đề xuất cấu trúc CFRG ứng dụng cho suy diễn mờ phức và thích nghi luật trong mô hình học chuyển giao đã đề xuất ở chương 2, các kết quả thực nghiệm cùng với những phân tích đánh giá hiệu suất.
- **Kết luận và hướng phát triển:** Trình bày các kết quả thực hiện được của luận án, những điểm hạn chế và các hướng phát triển.

# CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU VÀ CƠ SỞ LÝ THUYẾT

Trong chương đầu tiên này, luận án trình bày lý thuyết tổng quan về hệ suy diễn, hệ suy diễn mờ và hệ suy diễn mờ phức - đối tượng nghiên cứu của luận án. Tiếp đó, luận án trình bày kỹ thuật học chuyển giao sẽ được sử dụng trong quá trình nghiên cứu giải quyết hạn chế của hệ suy diễn mờ phức. Đồng thời, tại chương này cũng giới thiệu khái quát về các bộ dữ liệu, môi trường, kịch bản thực nghiệm và thước đo đánh giá trong thực nghiệm.

## 1.1. Suy diễn và hệ suy diễn mờ phức

Suy diễn được coi là một trong những công cụ, kỹ thuật quan trọng của khoa học cũng như cuộc sống. Suy diễn là quá trình rút ra thông tin mới, kết luận từ thông tin đã biết hoặc các nguyên tắc logic. Qua suy diễn, ta có thể kết nối các sự kiện, quy tắc và thông tin để tạo ra một cái nhìn toàn diện và chi tiết hơn về sự vật và hiện tượng. Suy diễn cũng là cơ sở cho việc tạo ra các lập luận logic và hợp lý. Đặc biệt trong trí tuệ nhân tạo và học máy, suy diễn là trọng tâm. Máy tính và hệ thống thông minh được lập trình để suy luận từ dữ liệu nhằm mục đích thực hiện các dự báo và phân tích xu hướng, hiểu và tự động hóa quyết định, hỗ trợ ra quyết định v.v.

Trong ngữ cảnh của trí tuệ nhân tạo và logic, suy diễn là quá trình sử dụng các quy tắc logic, kiến thức, hoặc thông tin đầu vào để tạo ra kết luận mới. Suy diễn giúp chúng ta đi từ thông tin có sẵn đến thông tin mới mà có ý nghĩa hoặc hữu ích.

Suy diễn có nhiều dạng khác nhau, bao gồm suy diễn logic (dựa trên quy tắc logic) [30], [31]; suy diễn thống kê (đưa ra các kết luận dựa trên dữ liệu thống kê và xác suất) [32]; suy diễn trong học máy (sử dụng mô hình máy tính để dự đoán kết quả) [33]; và suy diễn trong xử lý ngôn ngữ tự nhiên (đánh giá và phân tích văn bản ngôn

ngữ tự nhiên để rút ra thông tin) [34].

Hệ suy diễn đóng một vai trò quan trọng trong hệ thống thông minh và lĩnh vực học máy và trí tuệ nhân tạo (AI) bởi khả năng xử lý thông tin và thực hiện quyết định dựa trên luật suy diễn. Hệ suy diễn được sử dụng rộng rãi trong hệ thống gợi ý để đề xuất sản phẩm, dịch vụ hoặc nội dung dựa trên lịch sử và hành vi của người dùng [35], [36]. Hệ suy diễn có khả năng xử lý thông tin mờ và không rõ ràng, giúp cải thiện trải nghiệm người dùng và tăng khả năng tương tác. Không chỉ vậy, hệ suy diễn là một thành phần quan trọng trong các hệ thống hỗ trợ ra quyết định, giúp người ra quyết định xử lý thông tin phức tạp và đưa ra quyết định dựa trên các quy tắc được xác định trước [5], [37], [38].

Trong lĩnh vực quản lý tri thức, hệ suy diễn giúp quản lý tri thức trong các tổ chức và hệ thống thông tin. Nó có thể được sử dụng để tổng hợp, tổ chức và truy cập tri thức từ nhiều nguồn khác nhau [39]. Hệ suy diễn có thể được sử dụng để xây dựng mô hình AI dựa trên quy tắc và luật suy diễn. Điều này có ứng dụng trong việc phát triển hệ thống học máy dựa trên luật (Rule-Based Machine Learning) và hệ thống dựa trên tri thức (Knowledge-Based Systems). Hệ suy diễn cũng có khả năng phân loại đối tượng và dự đoán sự kiện dựa trên dữ liệu đầu vào [26], [40], [41]. Điều này có ứng dụng trong nhận dạng văn bản, hình ảnh và âm thanh, cũng như trong dự đoán xu hướng và biến đổi trong dữ liệu. Trong lĩnh vực y tế, hệ suy diễn được sử dụng để chẩn đoán bệnh lý dựa trên triệu chứng và dữ liệu xét nghiệm. Nó có khả năng kết hợp thông tin từ nhiều nguồn để đưa ra chẩn đoán và điều trị v.v.

Hiệu quả của suy diễn là dựa vào các thông tin, dữ liệu lịch sử. Trong khi đó, thông tin, dữ liệu biến đổi không ngừng và ngày càng chứa nhiều các đặc tính mới. Những đặc tính mới trong dữ liệu như tính mơ hồ, tần suất/chu kỳ đã làm cho logic truyền thống gặp phải khó khăn hay không hiệu quả trong quá trình suy luận. Các nghiên cứu dựa trên suy diễn mờ [42] có khả năng giải quyết các tình huống mà thông tin, dữ liệu có tính mơ hồ, không chắc chắn; hệ suy diễn mờ phức [1], [43] đi kèm với tính chu kỳ/định kỳ.

Các phương pháp suy diễn khác nhau được sử dụng phụ thuộc vào từng bài toán khác nhau. Hệ suy diễn mờ phức [43], còn được gọi là hệ logic mờ phức (CFLS), dựa trên cơ sở là hệ suy diễn mờ và được biết đến với khả năng xử lý các khái niệm ngôn ngữ liên quan đến các tri thức có tính chất tần suất và chu kỳ. Tức là hệ suy diễn mờ phức phù hợp với các bài toán liên quan đến các dữ liệu không chắc chắn, mơ hồ, lại có tính tần suất và chu kỳ [5], [7].

## 1.2. Tổng quan các nghiên cứu liên quan

### 1.2.1. Các mô hình suy diễn mờ

Logic mờ đã và luôn được đề cập như là một công cụ để mô tả những thông tin không chắc chắn, mơ hồ. Nó được áp dụng phổ biến trong việc giải quyết các vấn đề liên quan đến dự báo, điều khiển, phát hiện mẫu và các hệ hỗ trợ ra quyết định với thông tin không chắc chắn. Nó cũng được coi là mô hình tính toán mà có khả năng xử lý đồng thời cả tri thức ngôn ngữ và dữ liệu số. Logic mờ giúp máy tính hiểu và bắt chước suy nghĩ của con người, với mục tiêu làm tăng hiệu quả của quá trình ra quyết định đối với các tri thức mờ hồ, không chắc chắn. Lý thuyết logic mờ đã tạo ra một loạt các hệ thống suy diễn mờ [2], [3], [5], [44], [45] v.v. Mỗi hệ suy diễn mờ được mô tả như là một ánh xạ phi tuyến để đưa ra kết quả dựa trên các lập luận mờ và một tập các luật mờ dạng IF-THEN.

#### **Hệ suy diễn mờ**

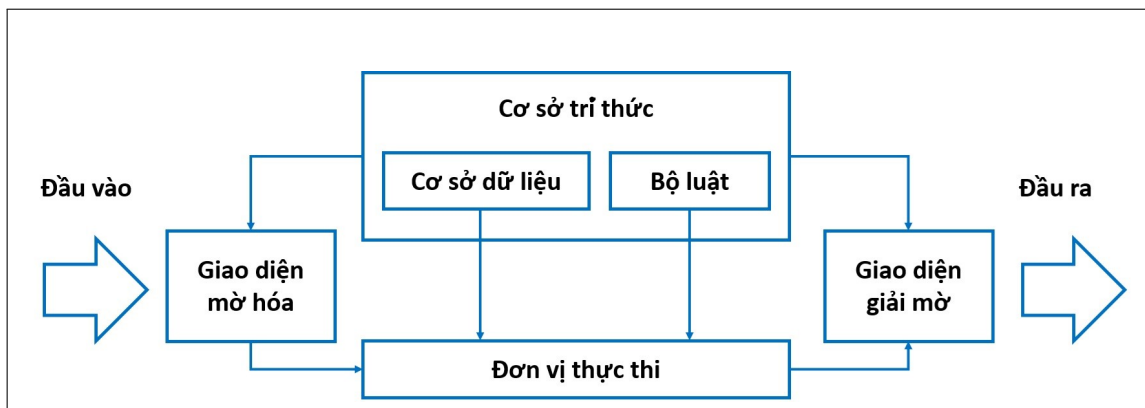
Quá trình suy diễn thực hiện liên kết các tri thức đã có một cách logic và khoa học để suy dẫn ra các tri thức mới. Cơ chế suy diễn thường phụ thuộc nhiều vào cách thức biểu diễn tri thức và không có một phương pháp suy diễn duy nhất cho mọi loại tri thức. Được xây dựng trên cơ sở lý thuyết mờ, FIS là một mô hình suy diễn dựa trên tri thức được mờ hóa, thường được áp dụng trong quá trình hỗ trợ ra quyết định. FIS tỏ ra hiệu quả trong trường hợp tri thức không đầy đủ, bất định hoặc không chính xác.

FIS dựa trên lý thuyết tập mờ đã và đang được sử dụng thành công trong nhiều lĩnh



vực y tế, kinh tế, xử lý ảnh, xử lý ngôn ngữ tự nhiên v.v. FIS được thiết kế để xử lý và giải quyết các vấn đề có sự không chắc chắn, không rõ ràng hoặc không chính xác trong thông tin hoặc dữ liệu đầu vào. Nó thường được ứng dụng trong các tình huống thực tế, nơi mà các hệ thống dựa trên logic truyền thống không thể đáp ứng đầy đủ do sự mơ hồ và bất định trong thông tin [46].

Một trong những điểm mạnh của FIS là khả năng xử lý tri thức mơ hồ thay vì dựa vào giá trị nhị phân (đúng/sai) như các hệ thống logic truyền thống. Điều này cho phép FIS biểu diễn được các dữ liệu có tính chất không rõ ràng và không chắc chắn hiệu quả hơn và FIS thường đem đến kết quả tốt hơn trong các tình huống thực tế.



**Hình 1.1:** Mô hình hệ suy diễn mờ tổng quát [47]

Cấu trúc của một hệ suy diễn mờ cơ bản gồm 5 thành phần :

- Giao diện mờ hóa giúp chuyển đổi các giá trị đầu vào rõ thành các biên độ phù hợp với các giá trị ngôn ngữ;
- Cơ sở dữ liệu với định nghĩa các hàm thuộc của các tập mờ được sử dụng trong các luật mờ;
- Bộ cơ sở luật gồm các luật mờ IF - THEN;
- Đơn vị thực thi quyết định trong đó thực hiện các hoạt động suy diễn của các luật (cơ chế suy diễn);

- Giao diện giải mờ chuyển đổi các giá trị kết quả suy diễn mờ ra các lớp đầu ra.

Thông thường, bộ cơ sở luật và cơ sở dữ liệu liên kết như là một bộ cơ sở tri thức của hệ suy diễn.

Chi tiết các bước suy diễn mờ

- Mờ hoá các biến vào: Với đầu vào có tính chất rõ ràng và có thể đo lường cụ thể, để tham gia vào quá trình suy diễn mờ, thường cần phải thực hiện một quá trình gọi là mờ hoá. Mờ hoá có thể được định nghĩa là một phép ánh xạ từ không gian các giá trị quan sát được (rõ ràng) sang không gian của các từ (tập hợp mờ) trên không gian nền của các biến ngôn ngữ.
- Sử dụng các phép toán mờ (ví dụ: AND hoặc OR) cho các giả thiết trong từng luật (với việc sử dụng các phép toán t - chuẩn hoặc t - đối chuẩn).
- Thực hiện phép toán kép để tính toán các giá trị từ giả thiết đến kết luận trong từng luật.
- Kết hợp kết quả từng luật thành một kết quả duy nhất cho toàn bộ hệ thống qua việc sử dụng toán tử gộp.
- Trích xuất giá trị rõ từ kết quả suy diễn mờ cho ta một số kết quả đoán định sử dụng hàm thuộc của giá trị mờ đầu ra.

Các phương pháp suy diễn mờ được phân loại theo ba phương pháp chính: Hệ suy diễn mờ Mamdani [44], hệ suy diễn mờ Sugeno (hay còn gọi là hệ suy diễn mờ Takagi – Sugeno) [45] và hệ suy diễn mờ Tsukamoto [48].

*Hệ suy diễn mờ Mamdani [44]*

Mô hình Mamdani sử dụng quy tắc suy luận mờ để ánh xạ từ tập mờ đầu vào đến tập mờ đầu ra bằng việc sử dụng các luật IF-THEN mờ. Mô hình Mamdani thích hợp cho các vấn đề có đầu ra mờ, nghĩa là kết quả của hệ thống cũng là biến mờ. Mô hình này thường phức tạp hơn vì nó đòi hỏi việc xử lý các biến mờ và kết hợp chúng bằng các phép toán mờ.

Trong mô hình suy diễn mờ Mamdani, đầu ra của một luật cụ thể có được thông qua sự kết hợp của giả thiết và kết luận của luật. Hơn nữa, trong các mô hình này toàn bộ đầu ra hệ thống từ một tập các luật được xây dựng bởi việc kết hợp các đầu ra của từng luật riêng biệt. Theo cách này, mỗi luật có dạng:

$$IF U \text{ là } B_i \text{ THEN } V \text{ là } D_i \quad (1.1)$$

Một hệ suy diễn mờ Mamdani có hai đầu vào  $x, y$  và một đầu ra  $z$ . Mỗi đầu vào và đầu ra có hai hàm thành viên, tương ứng là  $A_1, A_2, B_1, B_2$  và  $C_1, C_2$ . Luật thứ  $k$  có dạng:

$$k : \text{If } x \text{ is } A_i^k \text{ and } y \text{ is } B_j^k \text{ then } z \text{ is } C_l^k \quad (1.2)$$

với  $k = 1, \dots, R; i = 1, \dots, N; j = 1, \dots, M$  và  $l = 1, \dots, L$  trong đó  $R$  số lượng luật,  $N, M, L$  là số lượng hàm thuộc của hai biến đầu vào và biến đầu ra. Trong hệ suy diễn này, phương pháp giải mờ thường được sử dụng là lấy cực đại và tính toán điểm trọng tâm.

#### *Hệ suy diễn Sugeno [45]*

Mô hình Sugeno[45] cũng sử dụng các luật mờ IF-THEN, tuy nhiên, nó có một đặc điểm riêng biệt là giá trị đầu ra không phải là một tập hợp mờ mà là một hàm tuyến tính của các biến đầu vào. Mô hình suy diễn Sugeno thường đơn giản hơn và được sử dụng cho các vấn đề có đầu ra cụ thể và dễ dàng áp dụng trong các ứng dụng điều khiển và dự đoán.

Phương pháp suy diễn này dựa trên việc kết hợp các luật có dạng:

$$k : \text{If } x \text{ is } A_i^k \text{ and } y \text{ is } B_j^k \text{ then } z^k = f(x, y)$$

trong đó đầu ra là một hàm phụ thuộc vào biến đầu vào.

Cũng giống như Mamdani,  $k = 1, \dots, R; i = 1, \dots, N$  và  $j = 1, \dots, M$  trong đó  $R$  số lượng luật,  $N, M$  là số lượng hàm thuộc cho biến đầu vào.

Phương pháp giải mờ thường được sử dụng đối với hệ suy diễn Tagaki- Sugeno là phương pháp tính độ mạnh trung bình.

Bởi vì hệ suy diễn Sugeno thường được xem xét là có hiệu suất tính toán cao hơn so với hệ suy diễn Mamdani, do đó, nó thường được sử dụng trong việc xây dựng các mô hình mờ thích ứng. Các kỹ thuật thích ứng này có khả năng tinh chỉnh các hàm thuộc tính để tạo ra mô hình tối ưu nhất dựa trên loại dữ liệu cụ thể.

Hệ suy diễn Sugeno đã được đánh giá cao vì hiệu suất của nó, khả năng làm việc tốt với các kỹ thuật tuyến tính và tối ưu hóa, và khả năng thích hợp cho phân tích toán học. Tuy nhiên, một trong những hạn chế của hệ suy diễn mờ Sugeno là thiếu sự trực quan trong việc xác định các giá trị của các hệ số  $p$ ,  $q$ , và  $r$ . Ngoài ra, trong hệ suy diễn Sugeno, chỉ có đầu ra là rõ ràng và dễ hiểu, còn đầu vào thường không được biểu thị một cách trực quan.

#### *Hệ suy diễn mờ Tsukamoto [48]*

Với mô hình suy diễn Tsukamoto [48] thì các luật trong hệ suy diễn này được đại diện bởi một tập mờ với hàm thuộc đơn điệu. Kết quả suy diễn của từng luật được xác định là một giá trị rõ và được tính toán khi tính độ mạnh của luật. Kết quả cuối cùng thu được bằng cách lấy giá trị trung bình đầu ra của từng luật. Vì mỗi luật suy diễn ra một kết quả rõ, hệ suy diễn Tsukamoto lấy giá trị cuối cùng là trung bình nên quá trình giải mờ diễn ra nhanh chóng.

Với hệ Tsukamoto, mỗi luật mờ được biểu diễn bởi một hàm thành viên đơn điệu (Membership Functions- MF).

$$k : \text{If } x \text{ is } A_i^k \text{ and } y \text{ is } B_j^k \text{ then } z \text{ is } C_l^k$$

Hệ suy diễn Tsukamoto kết hợp đầu ra của các luật theo phương pháp trung bình trọng số. Nó có đặc điểm cụ thể là:

- Mỗi quy tắc suy diễn trong mô hình Tsukamoto được biểu diễn bằng một hàm thành viên đơn điệu.
- Giá trị đầu ra được xác định bằng độ mạnh của quy tắc, tức là mức độ kích hoạt của quy tắc.

- Giá trị đầu ra tổng quát của hệ Tsukamoto được tính bằng cách lấy trung bình có trọng số của các giá trị đầu ra của các quy tắc.

Vì mỗi luật trong hệ Tsukamoto cho ra một giá trị rõ, quá trình giải mờ diễn ra nhanh chóng. Tuy nhiên, hệ suy diễn Tsukamoto thường không được coi là hiệu quả bằng hai hệ suy diễn mờ Mamdani và Sugeno, vì vậy không thường xuyên được ưa chuộng trong ứng dụng thực tế.

### **Hệ suy diễn dựa trên tập mờ phức**

Sau khi Ramot giới thiệu lý thuyết tập mờ phức vào năm 2002 [1], nhiều nhà nghiên cứu đã quan tâm và đề xuất các hệ suy diễn dựa trên nền tảng lý thuyết tập mờ phức. Tập mờ phức có thể được coi là một khái niệm quan trọng trong các hệ thống thông minh, giúp biểu diễn và xử lý các vấn đề mà dữ liệu có tính chu kỳ, định kỳ hoặc dữ liệu cần thêm thông tin bổ sung. Các đặc tính đó ngày càng có nhiều trong thông tin và dữ liệu ngày nay và các tập mờ truyền thống không đủ để mô tả một cách toàn diện.

#### *Hệ logic mờ phức của Ramot*

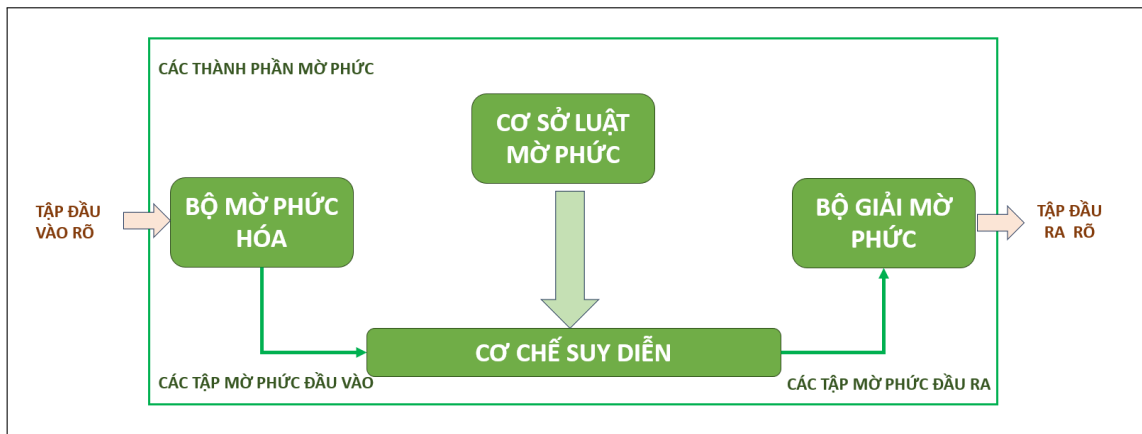
Vào năm 2003, Ramot đã trình bày mô hình logic mờ dựa trên tập mờ phức, được gọi là "logic mờ phức" [43]. Cũng như các hệ thống logic mờ (FLS) thông thường, CFIS/CFLS cũng là một ánh xạ không tuyến tính từ một vector dữ liệu đầu vào sang đầu ra. Một hệ CFIS bao gồm:

- Tập luật mờ phức;
- Bộ mờ phức hóa;
- Cơ chế suy diễn mờ phức;
- Bộ giải mờ phức.

Một hệ CFIS đại diện cho một sự phát triển mô hình Mendel's [49] với việc sử dụng các tập mờ và logic mờ phức như một cách tổng quát hóa của các tập mờ và logic mờ. Hệ CFIS có khả năng hiểu và đáp ứng tính không chắc chắn trong dữ liệu

có tính chất định kỳ hoặc chu kỳ. Tương tự như FIS/FLS truyền thống, hệ CFIS/CFLS hoạt động như một ánh xạ phi tuyến từ dữ liệu đầu vào sang đầu ra với một tỷ lệ cụ thể. Thành phần cốt lõi của một hệ CFIS/CFLS được xác định bởi cơ sở luật mờ phức chứa một tập các luật IF - THEN và các hàm mờ phức.

Mô hình hệ suy diễn mờ phức có thể mô tả như dưới đây (Hình 1.2).



**Hình 1.2:** Cấu trúc của CFIS [43]

*Giai đoạn đầu tiên* là quá trình mờ phức hóa, được dùng để chuyển đổi dữ liệu đầu vào rõ ràng thành dữ liệu đầu vào mờ.

*Giai đoạn thứ hai*, được gọi là suy diễn mờ, sử dụng cơ sở luật mờ phức để biến đổi các tập dữ liệu đầu vào mờ thành các tập dữ liệu đầu ra mờ. Mỗi luật được áp dụng cho các tập dữ liệu đầu vào mờ liên quan đến nó. Sau đó, bằng cách kết hợp các dữ liệu đầu ra mờ phức từ các luật riêng lẻ bằng tích hợp véc tơ, tạo ra một tập dữ liệu đầu ra mờ phức duy nhất.

*Giai đoạn cuối cùng (giai mờ)* là quá trình biến đổi dữ liệu đầu ra mờ phức thành dữ liệu đầu ra rõ. Trong giai đoạn này, bộ giải mờ của tập dữ liệu đầu ra phức sẽ đưa ra một dữ liệu đầu ra rõ ràng. Một cách tiếp cận khác để giải mờ đầu ra mờ phức là bỏ qua tất cả các thành phần pha và chỉ xem xét thành phần biên độ của tập đầu ra. Bất kỳ kỹ thuật giải mờ nào khác được sử dụng trong FIS/FLS truyền thống cũng có thể được áp dụng trong giai đoạn này.

Cấu trúc của CFIS bao gồm các tập dữ liệu đầu vào mờ phức và các luật mờ phức.

CFIS được dùng cho quá trình tổng hợp tri thức từ nguồn và áp dụng nó vào nhiệm vụ cụ thể. Quá trình suy diễn bao gồm việc sử dụng tri thức thu thập từ dữ liệu không có nhãn, kết hợp với kiến thức về tính chất chu kỳ/định kỳ đã học từ trước. Các thành phần riêng biệt của CFIS sẽ biến đổi dữ liệu định kỳ không chắc chắn thành các biến thể. Những biến đổi và thay đổi từ một ngữ cảnh này sang ngữ cảnh khác được lưu giữ bằng cách thay đổi miền của các tập dữ liệu đầu vào mờ phức và thích ứng với cơ sở luật mờ phức.

Quá trình xây dựng CFIS/CFLS được thực hiện thông qua 4 bước cụ thể như sau:

#### Bước 1: Tạo các vùng mờ phức

Trong bước này, ta chia dữ liệu thành các khoảng và mỗi khoảng này đại diện cho một lớp giá trị mờ phức của hàm đối với đầu vào hoặc đầu ra.

#### Bước 2: Tạo tập luật mờ phức

Ở bước này, ta tạo ra các luật mờ phức từ dữ liệu số có tính chất định kỳ/chu kỳ. Các luật này xác định mức độ thuộc về giá trị phức từ mỗi cặp dữ liệu có tính chu kỳ và tạo ra các luật đầu vào - đầu ra. Lớp giá trị phức lớn nhất cho từng cặp dữ liệu đầu vào và đầu ra được xác định từ các bộ dữ liệu chu kỳ khác nhau.

#### Bước 3: Giảm lược cơ sở luật

Từ các luật mờ phức, ta có thể tạo ra một cơ sở luật có kích thước tương đương với tập dữ liệu chu kỳ ban đầu. Mỗi luật mờ phức được gán một mức độ giá trị phức dựa trên tích hợp tối đa của các tập dữ liệu đầu vào và các tập dữ liệu đầu ra riêng biệt. Việc này giúp quản lý kích thước của cơ sở luật mờ phức và loại bỏ các xung đột không cần thiết.

#### Bước 4: Ánh xạ đầu ra thông qua giải mờ

Trong giai đoạn cuối cùng, chúng ta thực hiện ánh xạ giữa đầu vào và đầu ra bằng cách sử dụng giải mờ phức của các dữ liệu đầu vào.

Hệ CANFIS (Complex Neuro-Fuzzy Inference System), được công bố bởi nhóm tác giả Li và Jang [50], là một phương pháp mở rộng của hệ thống suy diễn nơ-ron mờ thích nghi. Nó được phát triển để xử lý đầu vào phức và tạo ra đầu ra phức tương ứng. Trong CANFIS, tất cả các biến đầu vào được định nghĩa dưới dạng biến phức, và các tham số của hệ thống, chẳng hạn như tâm của tập mờ đầu vào và vector trọng số của hàm đầu ra, đều là giá trị phức.

Hệ CANFIS bao gồm năm lớp chức năng: ba lớp ban đầu được gọi là lớp mờ hóa, có nhiệm vụ xác định độ mạnh của mỗi luật mờ; lớp thứ tư thực hiện nhiệm vụ suy diễn, dựa vào các luật để tạo ra các giá trị đầu ra tương ứng; và lớp cuối cùng đảm nhận nhiệm vụ tổng hợp kết quả bằng cách kết nối các giá trị đầu ra từ mỗi luật để đưa ra kết quả đầu ra cuối cùng.

Hệ CANFIS đã được áp dụng phổ biến trong xử lý tín hiệu phức và các hệ thống liên quan đến tín hiệu phi tuyến tính trong lĩnh vực viễn thông. Tuy nhiên, hệ thống này không hoàn toàn phù hợp trên miền phức, vì nó sử dụng các hàm mờ loại 1 riêng biệt cho phần thực và phần ảo của mỗi giá trị đầu vào. Việc phân tách phần thực và phần ảo của các giá trị đầu vào, làm mất đi sự đồng nhất trong việc áp dụng suy diễn trên miền phức. Hơn nữa, việc phân tách này cũng dẫn đến tăng số lượng luật và làm gia tăng độ phức tạp của quá trình tính toán trong hệ thống.

#### *Hệ ANCFIS*

Hệ ANCFIS, được giới thiệu bởi nhóm nghiên cứu Chen và cộng sự vào năm 2010 [2], là hệ thống suy luận mờ phức có khả năng học tập và thích nghi với dữ liệu. Mạng nơ-ron giá trị phức này cho phép dữ liệu đầu vào và đầu ra có dạng giá trị phức (mặc dù nó vẫn cho phép dữ liệu đầu vào và đầu ra là giá trị thực hoặc nhị phân), và các giá trị trọng số trong mô hình có thể mang giá trị phức hoặc giá trị thực.

Mô hình ANCFIS được đề xuất để áp dụng cho dữ liệu chuỗi thời gian, và hệ thống này chỉ yêu cầu một biến đầu vào, tức là một giá trị từ chuỗi dữ liệu thời gian. Điều này nhấn mạnh sự tự nhiên của tập mờ phức, tuy nhiên, trong trường hợp này, họ cần phải phân đoạn dữ liệu chuỗi thời gian sao cho phù hợp với phần thực và phần ảo của



tập mờ phức, đáp ứng yêu cầu của bài toán mà họ nghiên cứu.

### *Hệ M-CFIS*

Hệ suy diễn mờ phức Mamdani [5] lấy ý tưởng thì hệ suy diễn Mamdani truyền thống, bổ sung vào yếu tố phức nhằm giải quyết các hiện tượng theo tần suất, chu kỳ/định kỳ. Nó được phát triển từ mô hình Hệ logic mờ phức do Ramot đề xuất. Mô hình M-CFIS được ứng dụng cho hệ hỗ trợ ra quyết định và thu được kết quả về độ chính xác, chỉ số Precision và Recall cao hơn so với mô hình suy diễn mờ thông thường.

Mô hình M-CFIS thực sự là một hệ thống phức, tức là hệ thống xem xét những thông tin liên quan đến phần pha trong toàn bộ quá trình ra quyết định. Mô hình được đánh giá có khả năng ứng dụng trong ra quyết định chuẩn đoán bệnh và trong lĩnh vực rô bốt.

### **1.2.2. Học chuyển giao**

TL trong học máy lấy cảm hứng từ khả năng học tập chuyển giao của con người khi tận dụng được các kiến thức đã có của những miền liên quan (được gọi là miền nguồn) nhằm cải thiện hiệu suất học tập hoặc giảm thiểu số lượng các mẫu được gán nhãn bắt buộc phải có trong miền đích. Tuy nhiên, không phải học chuyển giao lúc nào cũng đem lại kết quả tốt. Nếu có ít các điểm chung giữa các miền, việc chuyển giao kiến thức có thể không thành công. Chẳng hạn như, việc biết đi xe đạp không làm cho việc học đàn nhanh hơn. Ngoài ra, sự tương đồng giữa các miền không phải lúc nào cũng là điều kiện thuận lợi cho học chuyển giao, đôi khi chính những điểm tương đồng đó lại tạo ra thách thức lớn hơn cho việc học tập. Như trong trường hợp, người đã biết tiếng Tây Ban Nha mà học tiếng Pháp, hai ngôn ngữ này khá giống nhau, chúng đều thuộc nhóm ngôn ngữ Lãng mạn, có ràng buộc chặt chẽ với nhau. Người học tiếng Tây ban Nha có thể gặp khó khăn trong học tiếng Pháp, chẳng hạn như sử dụng từ vựng sai hoặc chia ngữ pháp sai cách. Điều này xảy ra do kinh nghiệm học tập tiếng Tây Ban Nha trước đó cản trở việc hình thành từ, cách sử dụng, cách phát âm, cách

chia, v.v. trong tiếng Pháp. Trong lĩnh vực tâm lý học, hiện tượng kinh nghiệm trước đó ảnh hưởng không tốt đến việc học các nhiệm vụ mới được gọi là chuyển giao tiêu cực [51]. Hiện tượng chuyển giao tiêu cực tương tự cũng xuất hiện trong học chuyển giao của học máy. Học chuyển giao tiêu cực có xảy ra hay không phụ thuộc vào một số yếu tố như mức độ liên quan giữa miền nguồn và miền đích, năng lực của người học trong việc tìm ra kiến thức có thể chuyển giao và có lợi từ các miền.

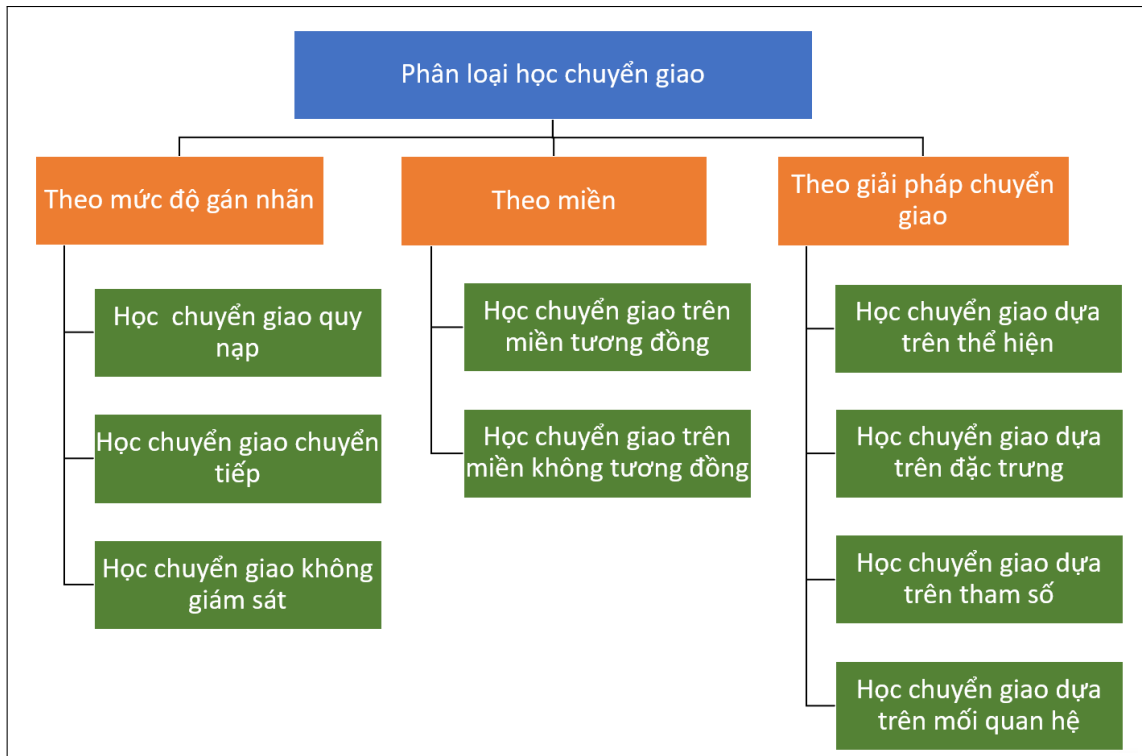
### **Phân loại học chuyển giao**

Các nghiên cứu về TL dựa trên cơ sở khác nhau như bài toán đặt ra, nhãn dữ liệu, dựa trên mô hình. Tuy nhiên, mọi quá trình học chuyển giao đều xem xét đến các vấn đề được chuyển giao. Có 3 câu hỏi luôn được đặt ra trong học chuyển giao đó là:

1. Chuyển giao cái gì?
2. Chuyển giao thế nào?
3. Chuyển giao khi nào?

“Chuyển giao cái gì?” trả lời câu hỏi thành phần kiến thức nào có thể được chuyển giao giữa miền nguồn và đích. Một vài kiến thức được chỉ rõ cho những miền hay nhiệm vụ đơn lẻ. Một vài kiến thức có thể chung giữa các miền nhờ đó chúng giúp cải thiện hiệu suất cho miền đích hay nhiệm vụ đích. Khi đã phát hiện những kiến thức có thể được chuyển giao, các thuật toán học cần được phát triển để thực hiện quá trình chuyển giao, tương ứng đáp ứng cho câu hỏi “Làm thế nào để chuyển giao?”. Đối với câu hỏi “Khi nào chuyển giao?”, là tình huống mà khi đó cần thiết phải thực hiện quá trình chuyển giao. Bởi không phải lúc nào, tình huống nào học chuyển giao cũng cải thiện hiệu suất cho miền đích/nhiệm vụ đích. Trong một vài tình huống cụ thể, khi miền nguồn và miền đích là không liên quan đến nhau, học chuyển giao có thể đem lại kết quả tốt, điều này gây ra hiệu suất học tập kém trên miền đích. Đây sẽ là chuyển giao tiêu cực. Mặc dù vậy, hầu hết các nghiên cứu hiện tại của học chuyển giao chỉ tập trung vào giải quyết hai câu hỏi “Chuyển giao cái gì?” và “Chuyển giao thế nào?”, các nghiên cứu đều ngầm mặc định mối quan hệ giữa miền nguồn và miền đích.

Theo [11] trả lời các câu hỏi “Chuyển giao cái gì?” và “Chuyển giao thế nào?” đã tạo thành một số hướng nghiên cứu về TL có thể mô tả tóm tắt như hình 1.3.



**Hình 1.3:** Phân loại học chuyển giao

#### *Học chuyển giao quy nạp (Inductive Transfer Learning-ITL)*

Trong học chuyển giao quy nạp, nhiệm vụ đích và nhiệm vụ nguồn là khác nhau, không quan tâm đến dữ liệu miền đích và miền nguồn có giống nhau không. Tuy nhiên, một vài dữ liệu được gán nhãn trong miền đích được yêu cầu làm dữ liệu huấn luyện để tạo ra hàm dự đoán trên miền đích. Tùy thuộc vào tình huống miền nguồn, mà có thể chia học chuyển giao quy nạp thành hai trường hợp: 1) dữ liệu được gán nhãn ở miền nguồn là sẵn có và 2) dữ liệu được gán nhãn ở miền nguồn là không có trong khi đó dữ liệu không gán nhãn là có. Gần như toàn bộ các tiếp cận học chuyển giao trong ngữ cảnh này đều tham chiếu đến trường hợp đầu tiên [52].

#### *Học chuyển giao chuyển tiếp (Transductive Transfer Learning - TTL)*

Trong học chuyển giao chuyển tiếp, các tác vụ nguồn và đích giống nhau, trong

khi các miền nguồn và đích khác nhau. Trong trường hợp này, không có dữ liệu được gán nhãn trong miền đích trong khi rất nhiều dữ liệu gán nhãn là có sẵn trong miền nguồn. Ngoài ra, theo các tình huống khác nhau giữa miền nguồn và miền đích, chúng ta có thể phân loại học chuyển giao chuyển tiếp thành hai trường hợp: 1) không gian đặc trưng giữa miền nguồn và miền đích khác nhau; 2) không gian đặc trưng giữa các miền giống nhau, nhưng xác suất phân phối cận biên là khác nhau [52].

#### *Học chuyển giao không giám sát (Unsupervised Transfer Learning -UTL)*

UTL, tương tự như học ITL, nhiệm vụ đích có liên quan đến nhiệm vụ nguồn. Tuy nhiên, học chuyển giao không giám sát tập trung vào việc giải quyết các nhiệm vụ học không giám sát trong miền mục tiêu, chẳng hạn như phân cụm, giảm kích thước và ước tính mật độ v.v. Trong trường hợp này, không có dữ liệu được gán nhãn có sẵn trong cả miền nguồn và miền đích trong quá trình đào tạo [52].

#### *Học chuyển giao trên miền tương đồng và không tương đồng*

Trong [53], nhóm tác giả đã chỉ ra một số vấn đề bất đồng trong quan điểm của học chuyển giao. Xuyên suốt trong các tài liệu học chuyển giao, có một số thuật ngữ không nhất quán. Các cụm từ như học chuyển giao và thích ứng miền được dùng để chỉ các quy trình tương tự. Thích nghi miền, liên quan đến học chuyển giao, là quá trình điều chỉnh một hoặc nhiều miền nguồn để chuyển giao thông tin có ích nhằm tăng cường hiệu suất của việc học miền đích. Quá trình thích nghi miền cố gắng thay đổi miền nguồn làm cho phân phối của miền nguồn gần hơn với miền đích.

Cũng theo nhóm tác giả, phần lớn các giải pháp học chuyển giao đồng nhất sử dụng một trong 3 chiến lược chung bao gồm cố gắng chỉnh sửa sự khác biệt phân phối cận biên trong miền nguồn, cố gắng sửa sự khác nhau về phân phối điều kiện trong miền nguồn, hay cố gắng sửa cả phân phối cận biên và phân phối có điều kiện trong miền nguồn. Nếu phân phối miền là không bằng nhau thì cần thực hiện các bước thích nghi miền là cần thiết.

Với cách nghiên cứu của mình, Kark Weiss [53] cũng đã đưa ra cách phân loại học chuyển giao theo không gian miền nguồn và miền đích. Trong đó, học chuyển giao

tương đồng là học chuyển giao mà trong đó không gian miền nguồn và không gian miền đích là như nhau. Việc áp dụng phương pháp học chuyển giao tương đồng trực tiếp vào môi trường dữ liệu lớn trở nên khả thi hơn, đặc biệt khi nguồn dữ liệu lớn đã sẵn sàng. Điều này mở ra cơ hội sử dụng nguồn tài nguyên dồi dào này cho các nhiệm vụ học máy, mà không cần phải tiến hành thu thập dữ liệu mới, tiết kiệm đáng kể về thời gian và nguồn lực. Học chuyển giao không tương đồng là kịch bản mà miền nguồn và miền đích được biểu diễn trong các không gian đặc trưng khác nhau. Ứng dụng học chuyển giao không tương đồng có mặt rộng rãi trong các lĩnh vực nhận dạng hình ảnh [54]–[58], phân loại văn bản đa ngôn ngữ, phân loại văn bản đơn ngôn ngữ [12], phân loại hiệu quả của thuốc [56], phân loại hoạt động của con người [59] v.v.

#### *Học chuyển giao dựa trên thể hiện*

Một số giả định rằng, sự khác biệt giữa phân phối dữ liệu trong miền nguồn và miền đích có thể dẫn tới một số dữ liệu trong miền nguồn là hữu ích với miền đích, tuy nhiên một số thể hiện lại có thể ảnh hưởng không tốt tới miền đích. Chính vì vậy, hướng tiếp cận này cố gắng chuyển giao các thể hiện tốt của miền nguồn cho miền đích nhằm cải thiện nhiệm vụ học tập miền đích. Một số kỹ thuật được thực hiện như lặp đi lặp lại nhiều lần các mẫu miền nguồn hay lấy mẫu mức độ quang trọng v.v. [52].

#### *Học chuyển giao dựa trên đặc trưng*

Tiếp cận chuyển giao đặc trưng đại diện với mục đích là tìm đặc trưng đại diện “tốt”. Các chiến lược để tìm ra đặc trưng đại diện “tốt” là khác nhau đối với các kiểu dữ liệu nguồn khác nhau. Nếu nhiều dữ liệu được gán nhãn ở miền nguồn là sẵn sàng, phương thức học giám sát có thể được dùng để khởi tạo các đặc trưng đại diện. Nếu không nhãn dữ liệu nào được gán trong miền nguồn, học không giám sát được đề xuất để khởi tạo đặc trưng đại diện [52].

#### *Học Chuyển giao dựa trên tham số*

Hầu hết tiếp cận chuyển giao tham số đều giả định các mô hình riêng lẻ cho các nhiệm vụ liên quan sẽ chia sẻ một tham số hoặc siêu tham số. Học chuyển giao cổ

gắng chia sẻ các tham số này từ mô hình nhiệm vụ nguồn sang các mô hình nhiệm vụ đích liên quan.

### *Học chuyển giao dựa trên mối quan hệ*

Khác biệt với 3 trường hợp trên, tiếp cận chuyển giao kiến thức quan hệ dựa trên ý tưởng giải quyết các vấn đề học chuyển giao trong miền quan hệ, trong đó dữ liệu không độc lập và không có phân phối giống nhau (non-iid) có thể được biểu diễn bằng nhiều quan hệ, như là dữ liệu trên mạng internet hay các nền tảng mạng xã hội. Tiếp cận này không giả sử rằng dữ liệu được rút ra từ mỗi miền độc lập và được phân phối giống nhau (iid) như giả định truyền thống. Nó cố gắng chuyển mối quan hệ giữa các dữ liệu từ miền nguồn sang miền đích, trong bối cảnh này, kỹ thuật học quan hệ thống kê được đề cập để giải quyết. Mặc dù số lượng nghiên cứu theo hướng này là ít hơn nhưng cũng có các kết quả nghiên cứu được ghi nhận như Mihalkova và cộng sự [60], [61], Davis và Domingos [62].

Ngoài cách phân loại học chuyển giao như trên, cũng có các cách phân loại học chuyển giao khác như học chuyển giao theo chiến lược mô hình, học chuyển giao bằng công nghệ học sâu, học chuyển giao mờ v.v.

### **1.2.3. Học chuyển giao mờ**

Dữ liệu là cơ sở của hầu hết các kỹ thuật học máy, trong khi đó, thông tin và dữ liệu trong thế giới thực thường có tính không chắc chắn và có tính chất mơ hồ. Khi thông tin, dữ liệu là không chắc chắn, mờ hồ. Những phương pháp học sẽ thêm vào các kỹ thuật khác nhau để biểu diễn nó nhằm lấy được những tri thức cần thiết cho giai đoạn học. FTL là sự kết hợp giữa lý thuyết mờ và kỹ thuật học chuyển giao nhằm giải quyết các bài toán khan hiếm dữ liệu và dữ liệu không chắc chắn, mơ hồ. Trong khi, kỹ thuật học chuyển giao sẽ giải quyết vấn đề về sự thiếu hụt kiến thức ở miền đích bằng cách tận dụng kiến thức ở các miền nguồn có liên quan. Thì lý thuyết mờ giúp mô tả những kiến thức có tính chất không chắc chắn, mơ hồ. Với ưu điểm này, các kỹ thuật học chuyển giao mờ cũng đã được nghiên cứu và ứng dụng trong nhiều

bài toán khác nhau từ phân lớp, đến hồi quy trong nhiều lĩnh vực như nhận dạng hình ảnh [29], y tế [63], [64] [15], giáo dục, xử lý ngôn ngữ tự nhiên [29], tài chính [65], [26], môi trường thông minh [27] v.v.

### **Phân loại học chuyển giao mờ**

Mục đích của FTL là chuyển giao kiến thức có sẵn trong một môi trường không chắc chắn, mơ hồ. Về cơ bản, FTL được đề xuất trên cơ sở kết hợp các phương pháp học chuyển giao với lý thuyết mờ. Do vậy, các nghiên cứu học chuyển giao mờ cũng có thể phân vào các nhóm của học chuyển giao như chuyển giao quy nạp, chuyển giao không giám sát, chuyển giao đặc trưng v.v. như học chuyển giao truyền thống. Tuy nhiên, học chuyển giao và lý thuyết tập mờ lại được kết hợp mạnh theo chiều hướng dựa trên các hệ suy diễn mờ, nhằm tạo ra các hệ suy diễn mờ cho miền mục tiêu trong những trường hợp thiếu hụt thông tin dữ liệu [26], [28], [66], [67]. Chính vì vậy, có thể phân loại học chuyển giao mờ dựa trên các mô hình suy diễn như Bảng 1.1.

***Bảng 1.1: Phân loại hệ học chuyển giao mờ***

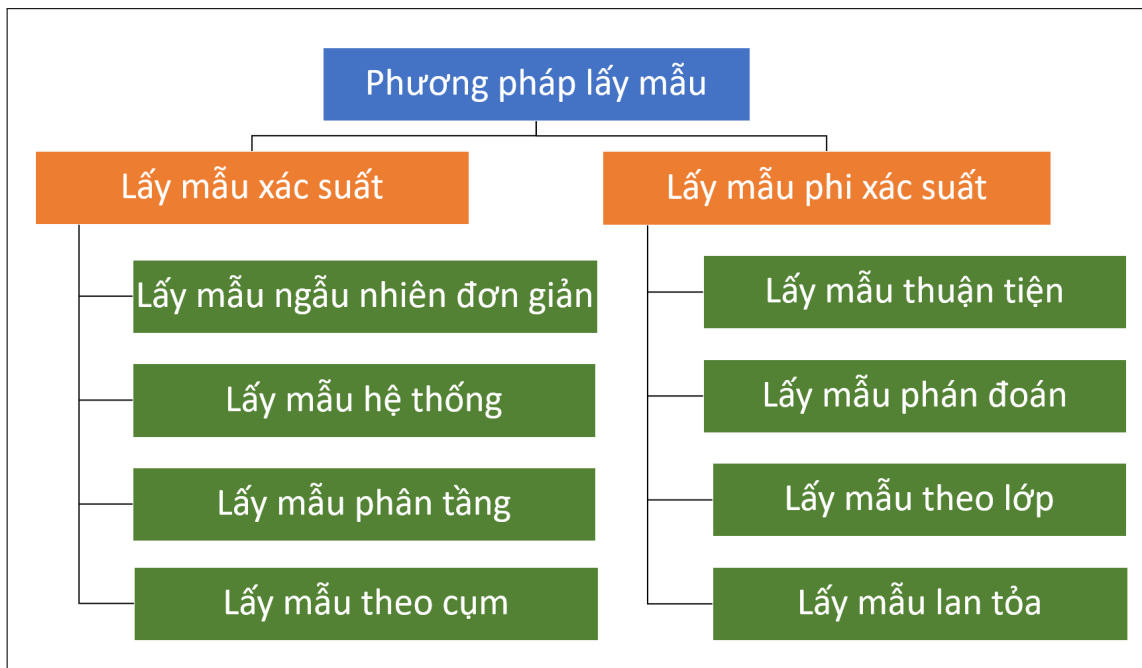
<b>Phương pháp phân loại</b>	<b>Loại</b>	<b>Các nghiên cứu điển hình</b>
Bài toán	Phân lớp	[27], [63]–[65], [67]–[69]
	Hồi quy	[21], [28], [63], [67]–[70]
Mô hình suy diễn	Suy diễn mờ T-S	[26], [28], [63], [64], [67]–[70]
	Suy diễn mờ Mam-dani	[26], [27]
Miền	Đơn miền	[26], [28], [63], [64], [68]–[70]
	Đa miền	[21], [67]

Các kỹ thuật được sử dụng trong các phương pháp học chuyển giao mờ cũng tương tự như các phương pháp học chuyển giao truyền thống bao gồm: chuyển giao thể hiện, chuyển giao đặc trưng thể hiện, chuyển giao tham số và chuyển giao mối quan hệ.

Hầu hết các nghiên cứu về học chuyển giao và học chuyển mờ đều đề cập đến vấn đề thiếu hụt kiến thức và sự không chắc chắn, mơ hồ trong dữ liệu miền đích. Sự thiếu

hạt này cả về nhãn dữ liệu lẫn dữ liệu miền đích. Tuy nhiên việc đề cập đến khía cạnh thời gian trong nghiên cứu về học chuyển giao mờ là khá ít.

#### 1.2.4. Lấy mẫu và các phương pháp lấy mẫu



**Hình 1.4:** Các phương pháp lấy mẫu

Các phương pháp học máy phổ biến là học từ dữ liệu, dữ liệu "tốt" sẽ đem lại hiệu suất tốt cho mô hình học máy. Tuy nhiên, việc thu thập đầy đủ dữ liệu cho quá trình học là bất khả thi về mặt thời gian, chi phí. Chính vì vậy, nghiên cứu về phương pháp lấy mẫu - lấy ra dữ liệu với kích thước nhỏ hơn, đặc trưng và giảm nhiễu là một trong những vấn đề được quan tâm nghiên cứu [71]–[74]. Phương pháp lấy mẫu được phân thành hai loại lấy mẫu xác suất và lấy mẫu phi xác suất [75], [76]. Các phương pháp lấy mẫu xác suất (Probability sampling - PS) là phương pháp mẫu mà trong đó các cá thể được lựa chọn ngẫu nhiên, mỗi cá thể trong quần thể nghiên cứu đều có cơ hội được lựa chọn ngang nhau và không phụ thuộc vào ý kiến chủ quan của người nghiên cứu. Trong khi đó, lấy mẫu phi xác suất (Non-probability sampling - NPS) người nghiên cứu chọn các đối tượng tham gia nghiên cứu một cách chủ định, dựa trên các



cá thể có sẵn khi thu thập số liệu và không tính cỡ mẫu. Chọn mẫu không xác suất có thể là chọn mẫu thuận tiện, chọn mẫu chỉ tiêu hay chọn mẫu có mục đích; nhằm thăm dò hay tìm hiểu sâu một vấn đề vào đó của quần thể (kiến thức, thái độ, niềm tin ...). Các phương pháp lấy mẫu phải có tính hệ thống và được xác định sao cho rút ra được những suy luận có giá trị từ mẫu. Ứng với mỗi phương pháp có kỹ thuật lấy mẫu khác nhau được mô tả chi tiết trong hình dưới đây (Hình 1.4)

*Lấy mẫu ngẫu nhiên đơn giản (Simple random sampling) [77]*

Lấy mẫu ngẫu nhiên đơn giản là kỹ thuật lấy mẫu dựa trên việc đã biết phân bố tổng thể. Phần tử được chọn vào tập mẫu có xác suất như nhau và biết trước. Công cụ áp dụng trong kỹ thuật này như rút thăm, thẻ ngẫu nhiên, hàm ngẫu nhiên... Phương pháp này có ưu điểm là đơn giản dễ thực hiện. Tuy nhiên, phương pháp sẽ không khả thi khi tổng thể mẫu có kích thước lớn. Phương pháp này thường được sử dụng để chọn các phần tử cho các phương pháp lấy mẫu khác như thông thường được sử dụng để khởi tạo cho phương pháp lấy mẫu hệ thống.

*Lấy mẫu hệ thống (Systematic Sampling) [77]*

Trong lấy mẫu có hệ thống, việc chọn đối tượng đầu tiên được thực hiện ngẫu nhiên và sau đó các đối tượng tiếp theo được chọn theo quy trình định kỳ. Lấy mẫu hệ thống giúp khắc phục được nhược điểm phân bố không đều của các phương pháp ngẫu nhiên đơn giản. Tuy nhiên, nhược điểm nếu khung lấy mẫu được xếp theo chu kỳ và tần số trùng với bước nhảy thì mẫu bị chệch.

*Lấy mẫu phân tầng (Stratified Sampling)[77]*

Phương pháp này sẽ chia toàn bộ dữ liệu thành từng nhóm nhỏ theo một tiêu chí nào đó (thu nhập, giới tính, tuổi tác...). Lấy mẫu ngẫu nhiên đơn giản cho từng tổ với kích thước tỉ lệ với độ lớn của nhóm (phương pháp lấy mẫu hệ thống). Tổ hợp hai phương thức này là phương pháp lấy mẫu phân tầng. Phương pháp lấy mẫu phân tầng được sử dụng phổ biến vì tính chính xác và đại diện cao. Tiêu chí phân nhóm ảnh hưởng đến độ chính xác của mẫu.

*Lấy mẫu phân cụm (Cluster Sampling)[77]*

Trong lấy mẫu phân cụm, tổng thể được chia thành nhiều cụm (cluster) như phương pháp phân tầng. Tuy nhiên, các phân cụm này đặc điểm là các phần tử trong cùng một phân cụm có tính khác biệt cao. Vậy nên có thể lại tiếp tục chia từng cụm thành nhiều nhóm nhỏ nữa. Sau đó có thể sử dụng phương pháp ngẫu nhiên đơn giản hay hệ thống để lấy mẫu từ một số nhóm của một phân cụm.

*Lấy mẫu thuận tiện (Convenience Sampling) [77]*

Phương pháp lấy mẫu thuận tiện dựa trên tính “dễ tiếp xúc” và “cơ hội thuận tiện”; nghĩa là người thực hiện nghiên cứu chọn những phần tử có thể tiếp cận được để lấy mẫu. Nhược điểm của biện pháp này là không xác định được sai số lấy mẫu và không thể kết luận cho tổng thể từ kết quả lấy mẫu. Lấy mẫu thuận tiện được sử dụng phổ biến, khi bị giới hạn về thời gian và chi phí thực hiện nghiên cứu.

*Lấy mẫu phán đoán (Judgment Sampling) [77]*

Phương pháp lấy mẫu phán đoán buộc nhà nghiên cứu tự phán đoán sự thích hợp của các phần tử để chọn chúng vào mẫu. Như vậy, tính đại diện của mẫu phụ thuộc vào kiến thức và kinh nghiệm của nhà nghiên cứu.

*Lấy mẫu theo lớp (Quota Sampling) [77]*

Phương pháp lấy mẫu theo lớp áp dụng cho tổng thể đã được phân lớp, nhà nghiên cứu cần chọn đủ số lượng mẫu để đảm bảo tỷ lệ mẫu và các đặc trưng ràng buộc mà không chọn ngẫu nhiên. Có thể sử dụng một hoặc nhiều thuộc tính ràng buộc như độ tuổi, giới tính, thu nhập, loại hình doanh nghiệp v.v. Đây là phương pháp lấy mẫu phi xác suất được sử dụng phổ biến nhất trong thực tiễn nghiên cứu.

*Lấy mẫu lan tỏa (Snow Ball Sampling)[77]*

Phương pháp lấy mẫu lan tỏa là phương pháp lấy mẫu theo yêu cầu nhà nghiên cứu, chọn ngẫu nhiên một số phần tử ban đầu. Sau đó, thông qua các phần tử ban đầu này để xác định các phần tử còn lại của mẫu. Phương pháp này thích hợp khi tổng thể ít phần tử hoặc khó nhận ra đối tượng cần để thu thập dữ liệu.

Việc lựa chọn phương pháp lấy mẫu yêu cầu nhà nghiên cứu phải xem xét nhiều

yếu tố như: câu hỏi nghiên cứu, phương pháp nghiên cứu, kiến thức về tổng thể quan tâm, quy mô tổng thể quan tâm, mức độ tương đồng hoặc khác biệt của các trường hợp trong tổng thể và hạn chế về thời gian và/hoặc tài chính và mức độ tin cậy mong muốn cho các kết luận nghiên cứu cũng như khả năng khái quát hóa [71], [78].

### **1.3. Hạn chế của hệ suy diễn mờ phức và bài toán nghiên cứu**

Các hệ suy diễn mờ phức tỏ ra hiệu quả trong giải quyết các bài toán suy diễn với dữ liệu không chắc chắn, mơ hồ và có yếu tố chu kỳ/định kỳ. Các ứng dụng của hệ suy diễn mờ phức như ra quyết định, dự đoán, dự báo v.v. có những đóng góp trong lĩnh vực nghiên cứu, xử lý tri thức [2], [5]–[7]. Tuy nhiên, quá trình xem xét thành phần pha trong dữ liệu và trong miền luật đã làm cho thời gian thực thi của các mô hình suy diễn này tiêu tốn thời gian đáng kể. Trong thực tế, việc xây dựng những mô hình suy diễn mờ phức trên dữ liệu lớn và cập nhật liên tục thì thời gian cần thiết để cập nhật lại hệ suy diễn là rất lớn. Điều này tạo ra hạn chế của việc ứng dụng rộng rãi mô hình suy diễn mờ phức cho các hệ thống thực, đặc biệt là các hệ thống có yêu cầu cập nhật lại, hoặc đòi hỏi suy diễn trong khoảng thời gian ngắn.

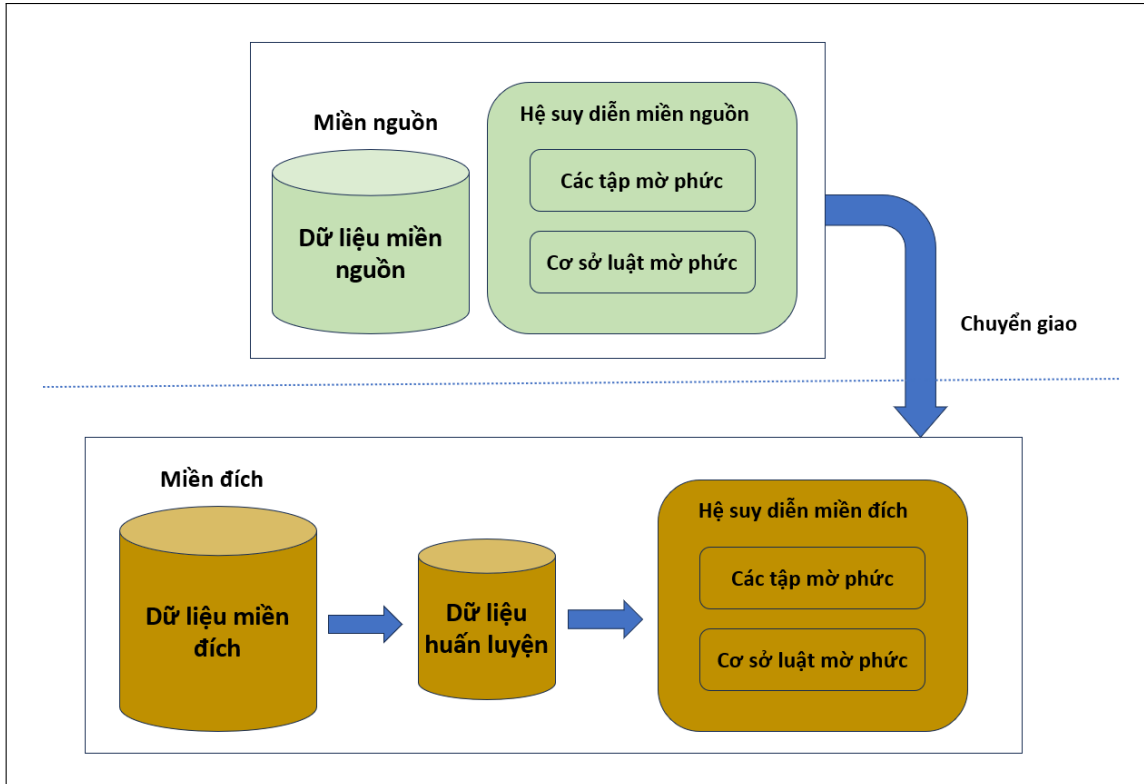
Trong khi đó, một trong những khả năng của TL là giảm thiểu thời gian học tập nhờ sử dụng lại kiến thức của các miền nguồn liên quan. Tuy nhiên, các đề xuất trước đó hầu hết chưa thực sự quan tâm đến điều này, chủ yếu chỉ quan tâm đến vấn đề về thiếu hụt tri thức trong miền đích [11].

Với phân tích đó, luận án đề xuất nghiên cứu kỹ thuật học chuyển giao cho mục tiêu giảm thời gian tạo ra hệ suy diễn mờ phức cho miền mục đích dựa trên hệ suy diễn mờ phức miền nguồn có sẵn.

#### **Bài toán nghiên cứu**

Ngữ cảnh đặt ra cho nghiên cứu là, cần xây dựng một hệ M-CFIS cho miền mục đích với kích thước  $N_{\psi}$  (rất lớn) trong khoảng thời gian ngắn hơn so với phương pháp xây dựng truyền thống. Bằng cách tận dụng hệ M-CFIS của miền khác tạm gọi là

miền nguồn với kích thước  $N_O$  ( $N_O \ll N_\Psi$ ) có sẵn. Miền đích và miền nguồn có mối quan hệ ràng buộc với nhau. Bài toán nghiên cứu có thể được mô tả như hình 1.5.



**Hình 1.5:** Mô hình bài toán học chuyển giao

Hình 1.5 cho thấy rằng, mục tiêu cụ thể cần nghiên cứu và giải quyết ở đây là:

1. Sử dụng kỹ thuật học chuyển giao để chuyển giao các tri thức từ miền nguồn (các tập mờ phức, cơ sở luật mờ phức) sang miền đích.
2. Kết hợp với một lượng dữ liệu nhỏ được lựa chọn từ miền đích.

Nhằm thu được hệ suy diễn M-CFIS cho miền đích với thời gian ngắn hơn so với việc xây dựng hệ suy diễn miền đích trực tiếp từ toàn bộ dữ liệu miền đích.

## 1.4. Cơ sở lý thuyết

### 1.4.1. Tập mờ và tập mờ phức

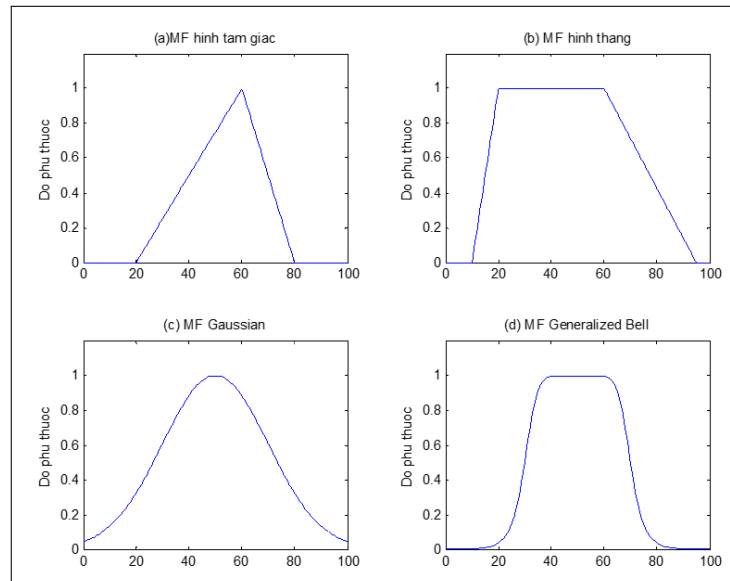
Lý thuyết tập mờ (FS) được L.A. Zadeh đưa ra vào năm 1965 [42]. Khái niệm nổi bật trong tập mờ của Zadeh là xuất phát từ những ý tưởng trừu tượng về ngữ nghĩa của thông tin mờ, đồng thời làm cho tính không chắc chắn trở nên quan trọng.

**Định nghĩa 1.1.** [42] Cho một tập vũ trụ  $U$ , tập  $H$  được cho bởi:  $H = \{(u, \mu_H(u)) \in [0, 1] | u \in U\}$  được gọi là một tập mờ trên không gian vũ trụ  $U$ .

Biến  $u \in U$  được gọi là biến cơ sở, tập  $U$  còn được gọi là tập tham chiếu hay miền cơ sở. Hàm  $\mu_H : U \rightarrow [0, 1]$  được gọi là hàm thuộc (membership function) và giá trị  $\mu_H(u)$  tại  $u$  được gọi là độ thuộc của phần tử  $u$  thuộc về tập mờ  $H$ .

Ví dụ 1.1. Xét tập  $U$  gồm 5 đối tượng là  $x_1, x_2, \dots, x_5$  tương ứng có khoảng cách đến một đối tượng  $O$  là 5, 25, 50, 55, 70 và  $H$  là tập các đối tượng “Gần”  $O$ . Hàm thuộc của tập Gần  $O$  có dạng như sau:  $\mu_H(O) = \frac{x}{100}$ , khi đó độ thuộc có thể như sau:

$\mu_{\text{Gần}}(x_1) = 0.95$ ,  $\mu_{\text{Gần}}(x_2) = 0.75$ ,  $\mu_{\text{Gần}}(x_3) = 0.5$ ,  $\mu_{\text{Gần}}(x_4) = 0.45$ ,  $\mu_{\text{Gần}}(x_5) = 0.3$  và tập mờ  $H = \frac{0.95}{x_1} + \frac{0.75}{x_2} + \frac{0.5}{x_3} + \frac{0.45}{x_4} + \frac{0.3}{x_5}$



**Hình 1.6:** Một số hàm thuộc phổ biến

Các hàm thuộc có nhiều loại khác nhau, một số hàm thuộc phổ biến hay được sử dụng như hàm thuộc bậc nhất, hàm thuộc tam giác, hàm thuộc hình thang, hàm thuộc Gaussian, hàm hàm sigma (Hình 1.6)

**Định nghĩa 1.2.** [1] Một cặp  $(l, \mu_H(l))$  được định nghĩa là một tập mờ phức trên không gian vũ trụ  $U$ , trong đó hàm  $\mu_H(l)$  là hàm liên thuộc giá trị phức cho bất kỳ  $l \in U$  với dạng sau:

$$\mu_H(l) = \vartheta_H(l) \cdot e^{j\varpi_H(l)} \quad (1.3)$$

với  $j = \sqrt{-1}$ , thành phần biên độ  $\vartheta_H(l) \in [0, 1]$  và thành phần pha  $\varpi_H(l) \in (0, 2\pi]$ .

Một tập mờ phức được định nghĩa bởi hàm  $\mu_H(l)$  với giá trị phức, trong đó phạm vi của hàm này tạo thành một đường tròn đơn vị trên mặt phẳng phức. Thêm vào đó, phần pha cung cấp thêm thông tin mở rộng về yếu tố chu kỳ và thời gian trong tập mờ, dựa trên biên độ.

Theo Ramot [1], [43], tập mờ phức được coi là một công cụ mô tả một cách hiệu quả cho các bài toán và đối tượng có tính biến đổi theo thời gian hoặc các vấn đề có tính chu kỳ, định kỳ.

Trong khi tập mờ thường có phạm vi giới hạn trong khoảng  $[0, 1]$ , tập mờ phức mở rộng phạm vi của hàm thành vòng tròn đơn vị trên mặt phẳng phức. Do đó, tập mờ phức cung cấp nền tảng toán học để biểu diễn hàm dưới dạng số phức. Vì thế, tập mờ là một trường hợp cụ thể của tập mờ phức khi thành phần pha bằng 0 [1].

#### 1.4.2. Các phép toán trên tập mờ phức

##### Phần bù của tập mờ phức

Cho  $H$  là một CFS với hàm thuộc mờ phức tương ứng là:  $\mu_H(l) = \vartheta_H(l) e^{j\varpi_H(l)}$ .

**Định nghĩa 1.3.** [1] Phần bù của tập mờ phức  $H$  (kí hiệu  $\bar{H}$ ) có thể được xác định như sau:

$$\bar{H} = \{(l, \mu_{\bar{H}}(l)) | l \in U\} = \{(l, \vartheta_{\bar{H}}(l) e^{j\varpi_{\bar{H}}(l)}) | l \in U\} \quad (1.4)$$

Với  $\vartheta_{\bar{H}}(l) = 1 - \vartheta_H(l)$  và  $\bar{\omega}_H(l) = 2\pi - \omega_H(l)$ .

Theo [1], phép toán phần bù mờ phức như sau:

$$\bar{H} = (1 - \vartheta_H(l)) \cdot e^{j(-\omega_H(l))} \quad (1.5)$$

$$\bar{H} = (1 - \vartheta_H(l)) \cdot e^{j(\omega_H(l))} \quad (1.6)$$

$$\bar{H} = (1 - \vartheta_H(l)) \cdot e^{j(\omega_H(l)+\pi)} \quad (1.7)$$

Ví dụ 1.2. Cho tập mờ phức  $A = 0.6e^{j1.2\pi}/x + 1.0e^{j2\pi}/y + 0.8e^{j1.6\pi}/z$ .

Khi đó,  $\bar{A} = 0.4e^{j0.8\pi}/x + 0/y + 0.2e^{j0.4\pi}/z$

### ***Phép hợp và phép giao của hai tập mờ phức***

Ramot [1] đã giới thiệu về phép hợp và giao trên tập mờ phức cùng với các toán tử được áp dụng cho thành phần pha của độ thuộc mờ phức.

Cho  $H$  và  $M$  là hai CFS với hàm thuộc mờ phức tương ứng là:  $\mu_H(l) = \vartheta_H(l)e^{j\omega_H(l)}$  và  $\mu_M(l) = \vartheta_M(l)e^{j\omega_M(l)}$ , khi đó, các phép toán hợp và giao là:

**Định nghĩa 1.4.** [1] Phép hợp hai tập mờ phức  $H$  và  $M$  (kí hiệu  $H \cup M$ ) được định nghĩa như sau:

$$\begin{aligned} H \cup M &= \{(l, \mu_{H \cup M}(l)) | l \in U\} \\ &= \{(l, \vartheta_{H \cup M}(l)e^{j\omega_{H \cup M}(l)}) | l \in U\} \\ &= \{(l, [\vartheta_H(l) \oplus \vartheta_M(l)]e^{j\omega_{H \cup M}(l)}) | l \in U\} \end{aligned} \quad (1.8)$$

Với phép  $\oplus$  có thể là phép t-đối chuẩn, với  $\vartheta_{H \cup M}(l) = \max\{\vartheta_H(l), \vartheta_M(l)\}$  và  $\bar{\omega}_{H \cup M} = \bar{\omega}_H + \bar{\omega}_M$ .

**Định nghĩa 1.5.** [1] Phép giao hai tập mờ phức  $A$  và  $B$  (kí hiệu  $H \cap M$ ) được xác định

bởi

$$\begin{aligned}
H \cap M &= \{(l, \mu_{H \cap M}(l)) | l \in U\} \\
&= \{(l, \vartheta_{H \cap M}(l) e^{j\overline{\omega}_{H \cap M}(l)}) | l \in U\} \\
&= \{(l, [\vartheta_H(l) \otimes \vartheta_M(l)] e^{j\overline{\omega}_{H \cap M}(l)}) | l \in U\}
\end{aligned} \tag{1.9}$$

Với  $\vartheta_{H \cap M}(l) = \min\{\vartheta_H(l), \vartheta_M(l)\}$  và  $\overline{\omega}_{H \cap M}(l) = \min(\overline{\omega}_H(l), \overline{\omega}_M(l))$ .

Trong đó, phép  $\otimes$  biểu diễn hàm T-chuẩn, ví dụ như toán tử Min hoặc phép nhân đại số. Khi  $\vartheta_H$  và  $\vartheta_M$  là giá trị thực, các toán tử max và min đều có thể được sử dụng.

Theo Ramot [1], giá trị thành phần pha của phép hợp và phép giao của tập mờ phức  $\overline{\omega}_{H \cap M}(l)$  và  $\overline{\omega}_{H \cup M}(l)$  được chọn tùy thuộc vào ngữ cảnh ứng dụng. Các phép toán được sử dụng với  $\overline{\omega}_{H \cap M}(l)$  thì cũng được dùng với  $\overline{\omega}_{H \cup M}(l)$  và có thể có các dạng như sau:

$$Sum : \overline{\omega}_{H \cup M} = \overline{\omega}_H + \overline{\omega}_M \tag{1.10}$$

$$Max : \overline{\omega}_{H \cup M} = \max(\overline{\omega}_H, \overline{\omega}_M) \tag{1.11}$$

$$Min : \overline{\omega}_{H \cup M} = \min(\overline{\omega}_H, \overline{\omega}_M) \tag{1.12}$$

$$WinnerTakeAll : \overline{\omega}_{H \cup M} = \begin{cases} \overline{\omega}_H & \vartheta_H > \vartheta_M \\ \overline{\omega}_M & \vartheta_H < \vartheta_M \end{cases} \tag{1.13}$$

$$WeightedAverage : \overline{\omega}_{H \cup M} = \frac{\vartheta_H \cdot \overline{\omega}_H + \vartheta_M \cdot \overline{\omega}_M}{\vartheta_H + \vartheta_M} \tag{1.14}$$

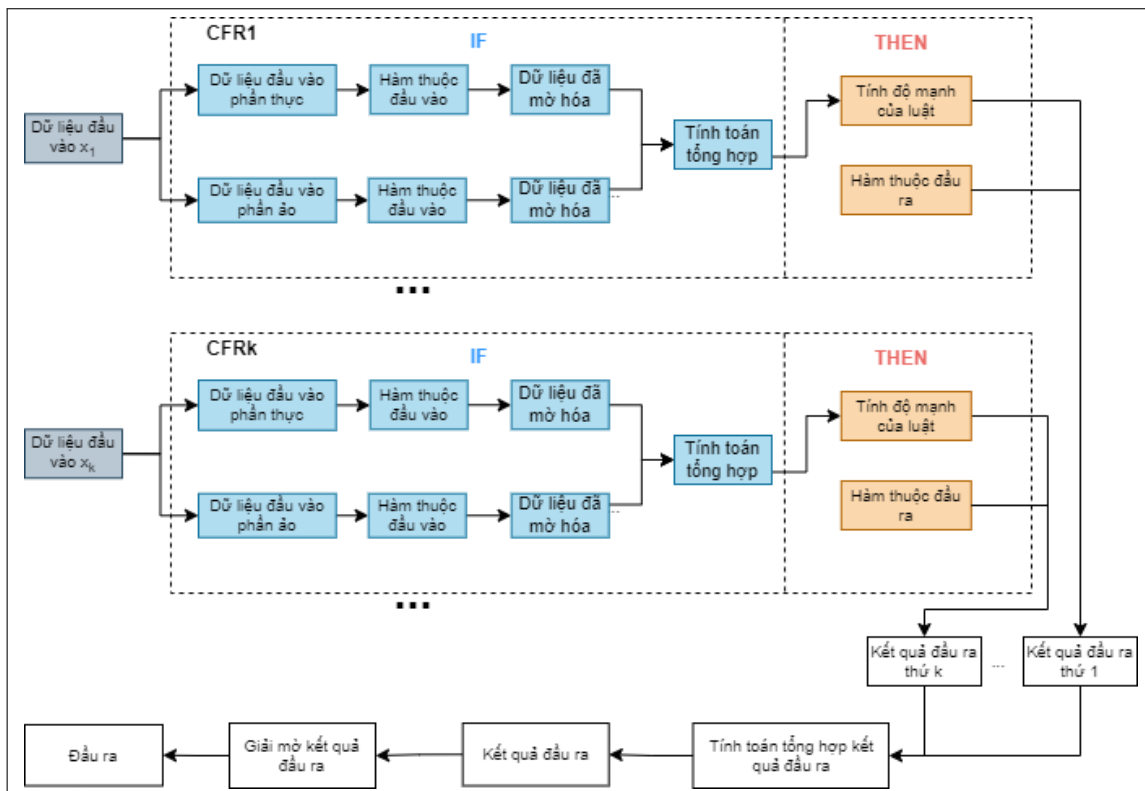
$$Average : \overline{\omega}_{H \cup M} = \frac{\overline{\omega}_H + \overline{\omega}_M}{2} \tag{1.15}$$

$$Difference : \overline{\omega}_{H \cup M} = \overline{\omega}_H - \overline{\omega}_M \tag{1.16}$$



### 1.4.3. Hệ suy diễn mờ phức Mamdani

M-CFIS [7], được đề xuất trên cơ sở cơ chế suy diễn Mamdani cổ điển và logic mờ phức là một công cụ hiệu quả để giải quyết các bài toán không chỉ giới hạn ở các giá trị của một thời điểm nhất định mà còn bao gồm tất cả yếu tố chu kỳ/định kỳ trong một khoảng thời gian nhất định. Một mô hình M-CFIS có cấu trúc như Hình 1.7.



Hình 1.7: Cấu trúc hệ M-CFIS [7]

Các giai đoạn suy luận mờ phức theo phương pháp M-CFIS được thực hiện như sau:

Cho  $\{l_1, l_2, \dots, l_m\} \in \mathbb{C}$  là tập dữ liệu đầu vào.

Bước 1: Tạo tập luật mờ phức (CFRs)

$CFR_1 : \text{If } l_{1,1} \text{ is } H_{1,1} \mathbf{O}_{1,1} \ l_{1,2} \text{ is } H_{1,2} \mathbf{O}_{1,2} \ \dots \ \mathbf{O}_{1,m_1-1} \ l_{1,m_1} \text{ is } H_{1,m_1} \text{ then } Z_1$

$CFR_2 : \text{If } l_{2,1} \text{ is } H_{2,1} \mathbf{O}_{2,1} \ l_{2,2} \text{ is } H_{2,2} \mathbf{O}_{2,2} \ \dots \ \mathbf{O}_{2,m_2-1} \ l_{2,m_2} \text{ is } H_{2,m_2} \text{ then } Z_2$

...

$CFR_k$  : If  $l_{q,1}$  is  $H_{q,1}$   $\mathbf{O}_{q,1}$   $l_{q,2}$  is  $H_{q,2}$   $\mathbf{O}_{q,2}$  ...  $\mathbf{O}_{k,m_k-1}$   $l_{k,m_k}$  is  $H_{k,m_k}$  then  $Z_t$

Với tất cả  $u, v$  thỏa mãn:

- (1)  $(u, v)_i \in \{1, 2, \dots, n\}$ , với  $1 \leq (u, 1) < (u, 2) < \dots < (u, m_u) \leq n$
- (2)  $\mu_{H_{u,v}}(l_{u,v}) = \vartheta_{H_{u,v}}(l_{u,v}) e^{j\varpi_{H_{u,v}}(l_{u,v})}$ , với  $\vartheta_{H_{u,v}} : \mathbb{C} \rightarrow [0, 1]$  và  $\varpi_{H_{u,v}} : \mathbb{C} \rightarrow (0, 2\pi]$
- (3)  $\mu_{Z_u}(w) = \vartheta_{Z_u}(w) e^{j\varpi_{Z_u}(w)}$ , với  $\vartheta_{Z_u} : \mathbb{C} \rightarrow [0, 1]$  và  $\varpi_{Z_u} : \mathbb{C} \rightarrow (0, 2\pi]$
- (iv)  $O_{u,v} = \text{and}$  nếu  $N_{u,v} = T_0$ ,  $T_0$  là một T - chuẩn
- (v)  $O_{u,v} = \text{or}$  nếu  $N_{u,v} = S_0$ ,  $S_0$  là T - đối chuẩn .

Bước 2: Mờ phức hóa dữ liệu đầu vào bằng cách sử dụng hàm thành viên mờ phức

$$\mu_H(l) = \vartheta_H(l) \cdot e^{j\varpi_H(l)}; l \in \mathbb{C}$$

Bước 3: Tính toán độ mạnh của luật

Xác định độ mạnh  $\varpi_u$  của mỗi luật theo công thức:  $\varpi_u = \tau_u e^{j\xi_u}$

Trong đó:

$$\tau_u = N_{u,n_u-1}(\dots N_{u,2}(N_{u,1}(\vartheta_{H_{u,1}}(l_{(u,1)}), \vartheta_{H_{u,2}}(l_{(u,2)})), \vartheta_{H_{u,3}}(l_{(u,3)})), \dots, \vartheta_{H_{u,n_u}}(l_{(u,n_u)}))$$

$$\xi_u = f_u(\varpi_{H_{u,1}}(l_{(u,1)}), \varpi_{H_{u,2}}(l_{(u,2)}), \varpi_{H_{u,3}}(l_{(u,3)}), \dots, \varpi_{H_{u,n_u}}(l_{(u,n_u)}))$$

Bước 4: Tính toán các kết quả đầu ra của từng CFR

Kết luận của CFR được tính toán bằng việc sử dụng luật kéo theo Mamdani.

Chọn hàm  $U_0 : [0, 1]^2 \rightarrow [0, 1]$  với  $U_0(1, 1) = 1$ , và hàm  $g_0 : (0, 2\pi]^2 \rightarrow (0, 2\pi]$  với  $g_0(2\pi, 2\pi) = 2\pi$ .

Hàm đầu ra tương ứng đối với mỗi luật mờ phức  $CFR_u$  được tính theo công thức:

$$\Gamma_u(z) = U_0(\tau_u, \vartheta_{C_u}(z)) e^{jg_0(\xi_u, \varpi_{C_u}(z))}.$$

Ở bước này thì việc chọn hàm  $U_0$  và  $g_0$  tùy thuộc vào ngữ cảnh và vấn đề nghiên cứu.

Bước 5: Tổng hợp kết quả đầu ra của các luật mờ phức

Quá trình tổng hợp kết quả đầu ra được thực hiện theo:

$$D(z) = \Gamma_1(z) + \Gamma_2(z) + \dots + \Gamma_q(z). \quad (1.17)$$

Với  $D = F(\mathbb{C}, \mathbb{C})$

Mục đích sử dụng phép toán tổng hợp là để tính toán đến sự tác động của thành phần trong các dữ liệu đầu vào một cách được chính xác nhất.

Bước 6: Giải mờ phức

Hàm  $\phi : F(\mathbb{C}, \mathbb{C}) \rightarrow \mathbb{C}$ , cho giá trị đầu ra theo:  $y_{op} = \phi(D)$

*Các lựa chọn sử dụng trong hệ suy diễn mờ phức Mamdani*

Các lựa chọn đối với mô hình hệ suy diễn mờ phức Mamdani được đề xuất gồm có: hàm thuộc mờ phức, phương pháp xác định độ tăng trưởng luật và toán tử tổ hợp. Cụ thể như sau:

Hàm thuộc mờ phức

Trong các hệ thống phát triển dựa trên lý thuyết tập mờ phức đã có thường sử dụng hàm thuộc Gauss và hàm Sin. Tuy nhiên, với mục đích xây dựng Hệ suy diễn mờ phức Mamdani là sự khái quát từ hệ suy diễn mờ thường Mamdani tức hàm thuộc phải đảm bảo sao cho giá trị  $\mu(l) \in [0, 1]$ . Và hệ thống đề xuất phải đảm bảo theo mô hình tổng quát của hệ logic mờ phức của Ramot nên hàm thuộc mờ phức của hệ thống M-CFIS đề xuất sẽ có dạng hàm thuộc mờ phức có sẵn như trong mô hình hệ logic mờ phức của Ramot, cụ thể như sau:  $\mu(l) = \vartheta(l) \cdot e^{j\varpi(l)}$  với thành phần pha  $\varpi(l) \in (0, 2\pi]$  and thành phần biên độ  $\vartheta(l) \in [0, 1]$ . Công thức hàm thuộc Gauss có dạng [79]:

$$\begin{aligned} \text{Re}(c\text{Gaussian}(l, c, \sigma)) &= \exp \left[ -0.5 \left( \frac{l-c}{\sigma} \right)^2 \right]; \\ \text{Im}(c\text{Gaussian}(l, c, \sigma)) &= -\exp \left[ -0.5 \left( \frac{l-c}{\sigma} \right)^2 \right] \times \left( \frac{l-c}{\sigma^2} \right) \end{aligned} \quad (1.18)$$

Các toán tử sử dụng trong hệ M-CFIS

Trong mô hình hệ M-CFIS, các toán tử được xác định như sau:

- Toán tử T- chuẩn Minimum được sử dụng để tính toán độ mạnh của luật mờ phức với phép toán AND được dùng để liên kết các điều kiện của luật.
- Toán tử T-đối chuẩn Maximum được sử dụng để tính toán độ mạnh của luật mờ phức với phép toán OR được dùng để liên kết các điều kiện luật.
- Luật kéo theo Mamdani được dùng để tính toán kết quả của mỗi luật mờ phức sử dụng phép tích vô hướng có dạng như sau:

$$\mu_{H \rightarrow M}(x, y) = (\vartheta_H(x) \cdot \vartheta_M(y)) \cdot e^{j2\pi \left( \frac{\omega_H(x)}{2\pi} \cdot \frac{\omega_B(y)}{2\pi} \right)} \quad (1.19)$$

Ở đây, phép tích vô hướng đối với các vector giá trị phức là phép toán khái quát hóa của phép tích vô hướng với các vector giá trị thực. Thêm nữa, phép tích T-chuẩn được sử dụng như trong ngữ cảnh hệ suy diễn Mamdani thường.

Chú ý: độ mạnh của luật và hệ quả của các luật mờ phức là các giá trị phức.

Vec tơ tổ hợp đối với tập mờ phức

Phép toán tổ hợp được coi là phép tích vô hướng giữa các vector giá trị phức có dạng như sau:

$$\begin{aligned} w_p \cdot \mu_{A_p}(x) &= w_p \cdot \overline{\mu_{A_p}(x)} \\ &= \left( r'_p e^{i\omega'_p} \right) \left( r_{A_p}(x) e^{-i\omega_{A_p}(x)} \right) \\ &= r'_p r_{A_p}(x) e^{i(\omega'_p - \omega_{A_p}(x))} \\ &= r'_p r_{A_p}(x) \left( \cos \left( \omega'_p - \omega_{A_p}(x) \right) + i \sin \left( \left( \omega'_p - \omega_{A_p}(x) \right) \right) \right). \end{aligned}$$

#### 1.4.4. Lý thuyết học chuyển giao

TL [11] được đề xuất trong rất nhiều nghiên cứu nhằm chuyển giao kiến thức giữa các miền, rút ngắn khoảng cách kiến thức của miền đích dựa trên kiến thức miền

nguồn có liên quan.

Khi xem xét về học chuyển giao, người ta đề cập đến các vấn đề như miền chuyển giao, nhiệm vụ miền nguồn và miền đích, kỹ thuật chuyển giao v.v.

**Định nghĩa 1.6.** (Miền) [80]

Miền được xác định bởi  $\zeta = \{\mathbf{L}, T(L)\}$ , trong đó  $L$  biểu diễn đặc trưng miền và  $T(L)$  mô tả phân phối xác suất của các phần tử  $L = \{l_1, \dots, l_m \in \mathbf{L}\}$ .

**Định nghĩa 1.7.** (Nhiệm vụ) [80]

Một nhiệm vụ  $E$  được cho bởi  $E = \{W, \rho\}$ , trong đó  $W$  không gian nhãn và  $\rho$  là một hàm dự đoán (là hàm ẩn có thể được học từ dữ liệu mẫu)

Một vài mô hình học máy, có đầu ra là các nhãn phân phối có điều kiện được dự đoán trong học máy:

$$\rho(l_i) = \{T(w_k | l_i) | w_k \in W; k = 1, \dots, |W|\} \quad (1.20)$$

Trong thực tế, một miền bao gồm cả những dữ liệu được gán nhãn và những dữ liệu không được gán nhãn. Ví dụ, cho một miền nguồn  $\zeta_O$  và một nhiệm vụ nguồn  $E_O$  của miền nguồn thường được quan sát thông qua cặp (phần tử, nhãn) là:  $\zeta_O = \{(l, w) | l_i \in L_O\}; w_k \in W_O; i = 1, \dots, m_{iO}; \text{ and } k = 1, \dots, m_{kO}$ . Cùng với miền đích, bao gồm một vài mẫu dữ liệu có hoặc không gán nhãn.

**Định nghĩa 1.8.** (Học chuyển giao) [80]

Cho  $\zeta_O, E_O, \zeta_\psi$  và  $E_\psi$  là miền nguồn và nhiệm vụ nguồn, miền đích và nhiệm vụ đích tương ứng. Học chuyển giao sử dụng những kiến thức thu được từ miền nguồn cho việc tái sử dụng cho hàm dự đoán  $\rho$  trong miền đích  $\zeta_\psi$  trong đó  $\zeta_O \neq \zeta_\psi$  hoặc  $E_O \neq E_\psi$ .

## 1.5. Thực nghiệm

Trong luận án này, các bộ dữ liệu thực nghiệm được sử dụng gồm 4 bộ dữ liệu từ kho dữ liệu chuẩn UCI và bộ dữ liệu về bệnh viêm gan thu thập từ Bệnh viện Gang thép và Bệnh viện Đa khoa Thái Nguyên. Việc chọn lọc những bộ dữ liệu này trên cơ sở các tiêu chí như thời gian thu thập hoặc thành phần bổ sung trong dữ liệu đảm bảo vai trò của thành phần pha của dữ liệu. Điều này nhấn mạnh sự quan trọng của việc phân tích các thành phần cụ thể và bổ sung đối với từng thuộc tính.

### 1.5.1. Bộ dữ liệu thực nghiệm

#### *Bộ dữ liệu chuẩn*

Những bộ dữ liệu chuẩn đã được thu thập từ kho dữ liệu học máy UCI [81], và được tổng hợp trong bảng dưới đây:

**Bảng 1.2:** Danh sách bộ dữ liệu thực nghiệm chuẩn

Số thứ tự	Bộ dữ liệu	Số thuộc tính	Số bản ghi	Số nhãn
1	Bộ dữ liệu ung thư vú - BreastCancer	9	680	2
2	Bộ dữ liệu tiểu đường - Diabetes	5	390	2
3	Bộ dữ liệu Thẻ tín dụng - Creditcard	16	8636	7

#### *Bộ dữ liệu thực- Bệnh gan (Liver)*

Dữ liệu về bệnh gan đã được tạo ra bằng cách tổng hợp và trích xuất từ hồ sơ bệnh án liên quan đến kết quả xét nghiệm (bao gồm sinh hóa máu và công thức máu) và các chuẩn đoán bệnh do các bác sĩ tại Bệnh viện Gang Thép và Bệnh viện Đa khoa

Thái Nguyên [6].

**Bảng 1.3:** Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh gan

Số thứ tự	Thuộc tính	Mô tả
1	Tuổi (Age)	Tuổi tính đến ngày xét nghiệm
2	Giới tính (Gender)	0: nam; 1: nữ
3	3 AST	Chỉ số men AST
4	ALT	Chỉ số men ALT
5	GGT	Chỉ số GGT
6	Albumin	Chỉ số Albumin
7	TB	Chỉ số Total Bilirubin
8	DB	Chỉ số Direct Bilirubin
9	DB/TB	Tỷ số DB/TB

Tập dữ liệu bệnh gan thực nghiệm gồm 4156 hồ bệnh nhân. Trong đó, 1202 hồ sơ bệnh nhân có chuẩn đoán là mắc bệnh viêm gan. Chi tiết về bộ dữ liệu này được trình bày trong bảng 1.3

### 1.5.2. Tiền xử lý dữ liệu

Đối với từng bản ghi đầu vào, với từng giá trị thuộc tính, việc sinh phần thực và ảo cho dữ liệu được thực hiện theo suy tắc sau: Phần thực được lấy chính giá trị gốc đầu vào của thuộc tính; phần ảo được tính bằng cách tính tổng phương sai theo cột và theo hàng ( $\text{Var.R}(\text{bản ghi}) + \text{var.A}(\text{thuộc tính})$ ) [82]. Kết quả của quá trình này là một tập dữ liệu mới. Tập dữ liệu gốc được gọi là dữ liệu phần thực, trong khi tập dữ liệu mới được tạo ra được gọi là dữ liệu ảo. Hai tập dữ liệu này sẽ được sử dụng làm dữ liệu đầu vào cho các thí nghiệm tiếp theo.

### 1.5.3. Môi trường thực nghiệm

Các kết quả thử nghiệm được cài đặt và chạy trên máy tính laptop Lenovo với bộ xử lý Core i7 bằng ngôn ngữ lập trình Python.

#### **1.5.4. Các độ đo đánh giá thực nghiệm**

Các độ đo được sử dụng nhằm đánh giá mô hình được đề xuất trong luận án so với phương pháp truyền thống xây dựng mô hình M-CFIS.

- Độ chính xác (Accuracy): là tỉ lệ giữa số mẫu được phân loại đúng trên tổng số mẫu.
- Thời gian thực hiện (Times): bao gồm thời gian huấn luyện và thời gian suy diễn kiểm thử.
- Số lượng luật (Number of Rules): số lượng luật được tạo ra được sử dụng cho mô hình suy diễn trên miền đích.

### **1.6. Kết luận chương 1**

Trong khuôn khổ nội dung chương 1, luận án đã trình bày những nghiên cứu liên quan về suy diễn và hệ suy diễn mờ phức với trọng tâm là hệ suy diễn mờ phức Mamdani - cho việc giải quyết các vấn đề suy diễn dựa trên thông tin không chắc chắn, mờ hồ và có tính chu kỳ. Đồng thời luận án cũng chỉ ra những hạn chế cũng trong hướng nghiên cứu tính đến hiện tại.

Cũng trong chương này, luận án cũng trình bày kỹ thuật học chuyển giao, các phương pháp học chuyển giao, học chuyển giao mờ. Với đặc trưng của kỹ thuật học chuyển giao có thể sử dụng để giải quyết vấn đề thiếu hụt kiến thức và rút ngắn thời gian học tập. Đây là các kiến thức nền tảng, sẽ được sử dụng trong các chương tiếp sau của luận án cho việc đề xuất các mô hình giải quyết các khoảng trống nghiên cứu đã nêu.

Kết quả nghiên cứu này được công bố trong công trình [1, 2] của luận án.



## CHƯƠNG 2. MÔ HÌNH HỌC CHUYỂN GIAO TRÊN HỆ SUY DIỄN MỜ PHỨC

Để giải quyết bài toán nghiên cứu được nêu ra trong chương 1, trong chương này của luận án đề xuất một mô hình học chuyển giao dựa trên hệ suy diễn mờ phức. Cụ thể như sau:

- Đề xuất mô hình CFTL, dựa trên FTL và M-CFIS, cho nhiệm vụ rút ngắn thời gian xây dựng hệ suy diễn mờ phức Mamdani.
- Giới thiệu mô hình CFTL gồm 4 bước: Điều chỉnh miền nguồn; Lựa chọn các tập con trong miền đích; Thích nghi luật; và Tổng hợp luật cho tập luật thích nghi cuối cùng.
- Đề xuất một định nghĩa mới về luật thích nghi và luật ứng cử để chọn những luật quan trọng nhất trong cơ sở luật cho quá trình thích nghi luật với dữ liệu miền đích.
- Kiểm chứng khả năng của mô hình CFTL trong việc xử lý các tình huống chuyển giao tri thức với ngữ cảnh thiếu thông tin trong miền mục tiêu, thông qua việc thực hiện các kịch bản thử nghiệm.
- Chứng minh hiệu suất của mô hình mới đề xuất qua các thử nghiệm thực hiện trên dữ liệu từ UCI [81] cũng như các tập dữ liệu thực tế, đánh giá về độ chính xác, số lượng luật, và thời gian thực thi.

Bản chất của mô hình CFTL đề xuất nằm ở cơ chế suy luận theo mô hình suy luận mờ phức Mamdani. Điểm mới là hệ thống CFTL được bổ sung tính năng học chuyển giao mờ. Bên cạnh đó, theo sự hiểu biết của NCS, hầu hết các hệ thống CFIS hiện tại không thực tận dụng được thành phần pha. Ngược lại, mô hình CFTL được đề xuất

thực hiện các tính toán trên cả hai phần biên độ và pha trong toàn bộ quá trình ra quyết định.

## 2.1. Một số khái niệm và định nghĩa

Trong lý thuyết về học chuyển giao, mối quan hệ tương đồng giữa dữ liệu miền nguồn và miền đích đóng vai trò quan trọng. Nó giúp xác định khả năng xảy ra chuyển giao tiêu cực và là cơ sở quyết định phương pháp, kỹ thuật chuyển giao. Lợi dụng những yếu tố tương đồng này, nhà nghiên cứu có thể xây dựng những kỹ thuật học chuyển giao nhằm thu được lợi điểm về mặt thời gian. Nghiên cứu này, với ngữ cảnh đặt ra về mối quan hệ giữa miền nguồn và miền đích là miền nguồn và miền đích có số lượng thuộc tính và nhiệm vụ đầu ra tương ứng giống nhau nhưng khác nhau khoảng miền giá trị của thuộc tính. Ngữ cảnh này chính là câu trả lời cho câu hỏi "Khi nào thì chuyển giao?" trong mô hình được đề xuất. Để rõ ràng hơn dưới đây là định nghĩa trình bày về định nghĩa học chuyển giao trên hệ suy diễn mờ phức.

**Định nghĩa 2.1.** Học chuyển giao trên hệ suy diễn mờ phức

Cho  $\zeta_O$ ,  $E_O$ ,  $\zeta_\psi$  và  $E_\psi$  là miền nguồn và nhiệm vụ nguồn, miền đích và nhiệm vụ đích tương ứng. Học chuyển giao trên hệ suy diễn mờ phức sử dụng những kiến thức thu được từ miền nguồn, tái sử dụng cho hàm dự đoán  $\rho$  trong miền đích  $\zeta_\psi$ . Với ràng buộc miền nguồn và miền đích như sau:

- $\zeta_O, \zeta_\psi$  giống nhau về phân phối, số lượng thuộc tính nhưng khác nhau về khoảng miền giá trị của các thuộc tính.
- $E_O = E_\psi$ , giống nhau về nhiệm vụ.

Định nghĩa 2.1 về ngữ cảnh chuyển giao của mô hình CFTL, có thể cho rằng phương pháp học chuyển giao được sử dụng ở đây là chuyển giao chuyển tiếp, với nhiệm vụ nguồn và nhiệm vụ đích là giống nhau, trong khi miền nguồn và đích là khác nhau về khoảng miền giá trị của các thuộc tính.

Định nghĩa 2.1, chính là điều kiện cần để thực hiện mô hình CFTL (Hình 2.1). Bên

cạnh đó, để hoạt động học chuyển giao hiệu quả, đồng thời đo lường trước khả năng học chuyển giao thành công trên CFTL, một phép đo tương tự giữa hai miền được đề xuất trong Định nghĩa 2.2.

**Định nghĩa 2.2.** Độ tương tự giữa hai miền cho CFTL.

Giả sử  $\zeta_U$  và  $\zeta_V$  là hai miền, độ tương tự của miền  $\zeta_V$  với miền  $\zeta_U$  được xác định bởi công thức sau:

$$DI(\zeta_U, \zeta_V) = \frac{\sum_{i=1}^m S_i}{m} \quad (2.1)$$

Trong đó:

$$S_i = \begin{cases} 1, & \text{if } x_i^V \geq L^U \text{ and } x_i^V \leq R^U \\ \frac{1}{|L^U - x_i^V|}, & \text{if } x_i^V < L^U \\ \frac{1}{|x_i^V - R^U|}, & \text{if } x_i^V > R^U \end{cases} \quad (2.2)$$

với  $x_i^V$  là giá trị phần tử thứ  $i$  của miền  $\zeta_V$ ,  $L^U = \min_{j=1,n} x_j^U$ ; và  $R^U = \max_{j=1,n} x_j^U$ ;  $m, n$  là số lượng dòng dữ liệu trong  $\zeta_V, \zeta_U$  tương ứng.

**Bổ đề 1:** Với độ tương tự  $DI$  giữa hai miền  $\zeta_U$  và  $\zeta_V$ , giá trị  $DI$  thỏa mãn các yêu cầu sau:

1.  $DI(\zeta_U, \zeta_V) \in [0, 1]$
2.  $DI(\zeta_U, \zeta_V) = 1, \text{ IF } \zeta_V \subseteq \zeta_U$
3.  $DI(\zeta_U, \zeta_V) \rightarrow 0 \text{ IF } \forall x_i^V \ll L^U \text{ hoặc } \forall x_i^V \gg R^U$

**Chứng minh:**

1.  $\forall S_i \in [0, 1] \Rightarrow \frac{\sum_{i=1}^m S_i}{m} \in [0, 1] \Rightarrow DI(\zeta_U, \zeta_V) \in [0, 1]$
2. Cho  $\zeta_V \subseteq \zeta_U$  then  $L^V \leq L^U$  và  $R^V \leq R^U \Rightarrow \forall x_i^V \in [L^U, R^U] \Rightarrow S_i = 1, \forall i = 1, \bar{m} \Rightarrow DI(\zeta_U, \zeta_V) = 1.$

$$3. \text{ Cho } \forall x_i^V \ll L^U \Rightarrow \frac{1}{|L^U - x_i^V|} \rightarrow 0 \text{ hoặc } \forall x_i^V \gg R^U \Rightarrow \frac{1}{|x_i^V - R^U|} \rightarrow 0 \\ \Rightarrow S_i = 1, \forall i = 1, \bar{m} \rightarrow 0 \Rightarrow DI(\zeta_U, \zeta_V) \rightarrow 0$$

Nếu độ tương tự giữa hai miền gần với 1, nghĩa hai miền có độ tương tự cao, quá trình chuyển giao sẽ thu được kết quả tốt. Ngược lại nếu độ tương tự giữa hai miền gần với 0, hai miền khác nhau nhiều, quá trình chuyển giao sẽ tốn nhiều tài nguyên và chi phí và kết quả chuyển giao có thể không tốt.

Kỹ thuật học chuyển giao được sử dụng trong mô hình đề xuất nhằm mục tiêu chuyển giao tri thức (cơ sở luật, tập mờ phức) từ miền nguồn sang miền đích. Tuy nhiên, không phải luật nào cũng được lựa chọn để chuyển giao sang miền đích. Để xác định các luật được chuyển giao sang miền đích, một định nghĩa về các luật phù hợp (bao gồm luật thích nghi và luật ứng cử) được trình bày. Nhưng trước hết, việc xác định này dựa trên cơ chế suy diễn của hệ M-CFIS và quá trình đánh giá độ mạnh được đề xuất dưới đây (Định nghĩa 2.3).

**Định nghĩa 2.3.** Độ mạnh của luật mờ phức cho lựa chọn luật chuyển giao.

Cho luật mờ phức R, độ mạnh của luật được xác định bởi công thức sau:

$$\omega_R = \min_{k=1, n} \left( \sqrt{(F_A \cdot \cos(F_P))^2 + (F_A \cdot \sin(F_P))^2} \right) \quad (2.3)$$

trong đó  $n$  là số lượng các đầu vào của luật,  $F_A = \vartheta_H(l_k)$  là độ mạnh phần biên độ, và  $F_P = w_H(l_k)$  là độ mạnh phần pha của mỗi đầu vào  $l_k$  là các thuộc tính đầu vào.

Khác với phương pháp đánh giá độ mạnh của luật đã đề xuất trong [5]. Độ mạnh của luật được đề xuất ở được gộp lại từ hai phần biên độ và pha ở mỗi thuộc tính đầu vào. Với mục tiêu sử dụng độ mạnh của luật và độ mạnh biên độ, độ mạnh pha cho việc xác định các luật thích nghi, luật ứng cử.

Để xác định luật được giữ lại (luật thích nghi, luật phù hợp) và các luật cần chỉnh sửa (luật ứng cử) một định nghĩa về luật thích nghi và luật ứng cử được phát biểu như sau:

**Định nghĩa 2.4.** Luật thích nghi và luật ứng cử.

Cho luật mờ phức R và dòng dữ liệu t

(i) Luật R là luật thích nghi trên dữ liệu t khi:

$$- R.label = t.label$$

$$- \omega_R > \varepsilon$$

(ii) Luật R gọi là luật ứng cử trên dòng dữ liệu t nếu nó thỏa mãn:

$$- R.label = t.label$$

$$- \frac{n(F_A > \varepsilon)}{n(F_A)} \geq \alpha$$

$$- \text{Với } \forall (F_A \leq \varepsilon) \text{ thì } (F_P \geq \beta)$$

Trong đó:

- R.label là nhãn của đầu ra luật;
- t.label là nhãn của dòng dữ liệu t;
- $F_A$  là độ mạnh thành phần biên độ;
- $F_P$  là độ mạnh thành phần pha;
- $\omega_R$  là độ mạnh của luật;
- $n(F_A > \varepsilon)$  là số lượng phần biên độ lớn hơn  $\varepsilon$ ;
- $n(F_A)$  là số lượng phần biên độ;
- $\varepsilon, \alpha, \beta \in [0, 1]$ : các tham số được xác định trước.

Ví dụ 2.1. Cho 4 luật mờ phức:  $CFR_1, CFR_2, CFR_3, CFR_4$  và sau khi tính toán độ mạnh của luật trên dòng dữ liệu thứ t, ta nhận được các giá trị độ mạnh luật, như dưới đây:

$CFR_1$  : If  $X_1$  is  $0.89e^{j0.011}$  and  $X_2$  is  $0.99e^{j0.02}$  and  $X_3$  is  $0.81e^{j0.066}$  and  $X_4$  is  $0.86e^{j0.065}$  then  $y_1$ , với  $\omega_{CFR_1} = 0.81$

$CFR_2$  : If  $X_1$  is  $0.56e^{j0.07}$  and  $X_2$  is  $0.92e^{j0.05}$  and  $X_3$  is  $0.81e^{j0.004}$  and  $X_4$

is  $0.86e^{j0.016}$  then  $y_1$ , với  $\omega_{CFR_2} = 0.56$

$CFR_3$  : If  $X_1$  is  $0.89e^{j0.011}$  and  $X_2$  is  $0.69e^{j0.02}$  and  $X_3$  is  $0.98e^{j0.003}$  and  $X_4$  is  $0.43e^{j0.2}$  then  $y_1$ , với  $\omega_{CFR_3} = 0.69$

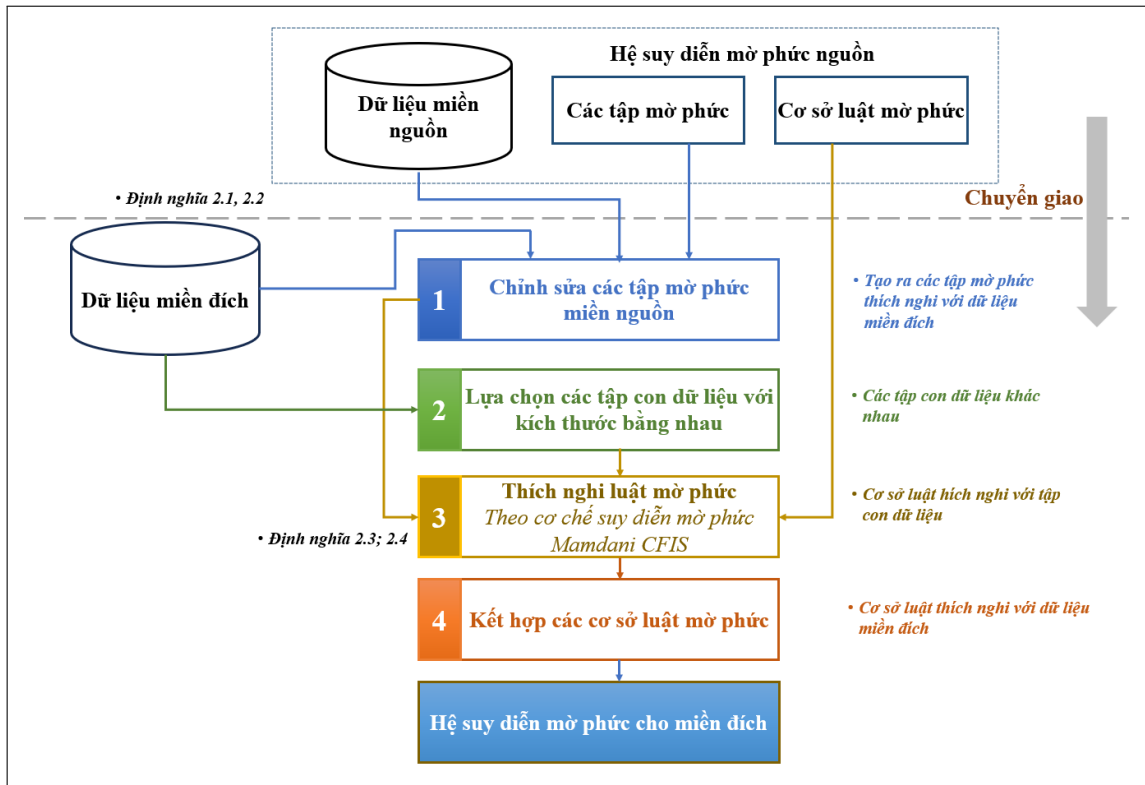
$CFR_4$  : If  $X_1$  is  $0.87e^{j0.07}$  and  $X_2$  is  $0.80e^{j0.02}$  and  $X_3$  is  $0.83e^{j0.07}$  and  $X_4$  is  $0.53e^{j0.08}$  then  $y_2$ , với  $\omega_{CFR_4} = 0.53$

Theo định nghĩa về luật thích nghi và luật ứng cử (xem Định nghĩa. 2.4), với  $\varepsilon \geq 0.8$ ,  $\alpha \geq 0.75$ ,  $\beta \geq 0.05$ , ta có :

- $CFR_1$  là luật thích nghi trên bản ghi dữ liệu thứ t, do  $\omega_{CFR_1} = 0.81 > \varepsilon$ ;
- $CFR_2$  là luật ứng cử do  $\omega_{CFR_2} = 0.56 < \varepsilon$ ,  $\frac{n(F_A > \varepsilon)}{n(F_A)} = \frac{3}{4} = 0.75 \geq \alpha$ , với phần biên độ  $F_A = 0.56 < \varepsilon$  thì có phần pha  $F_P = 0.07 > \beta$ ;
- $CFR_3$  không đáp ứng các điều kiện luật thích nghi hay luật ứng cử vì  $\frac{n(F_A > \varepsilon)}{n(F_A)} = \frac{3}{4} = 0.75 < \alpha$ ;
- Và  $CFR_4$  là luật ứng cử do  $\omega_{CFR_4} = 0.53 < \varepsilon$ ,  $\frac{n(F_A > \varepsilon)}{n(F_A)} = \frac{3}{4} = 0.75 \geq \alpha$ , với phần biên độ  $F_A = 0.53 < \varepsilon$  thì có phần pha  $F_P = 0.08 > \beta$ ;

## 2.2. Mô hình học chuyển giao mờ phức

Mô hình CFTL được đề xuất được thiết kế như một mô hình chuyển giao tri thức trong đó miền nguồn và miền đích có mối quan hệ chuyển giao (Định nghĩa 2.1). Trong mô hình này, các nhiệm vụ miền nguồn và đích có cùng kích thước nhưng có giá trị khoảng miền thuộc tính là khác nhau. CFTL cố gắng thu hẹp khoảng cách khác biệt phân phối trong suốt quá trình học tập và điều chỉnh việc học từ ngữ cảnh này sang ngữ cảnh khác. Sự thay đổi bối cảnh có thể do thay đổi miền, thiếu thông tin, tình huống, v.v. CFTL được đề xuất chuyển kiến thức từ không gian này sang không gian khác để giảm thời gian xử lý nhằm tạo ra cơ sở luật mới với chất lượng dự đoán chấp nhận được.



**Hình 2.1:** Mô hình học chuyển giao trên hệ suy diễn mờ phức

### Thuật toán 2.1 Các bước học chuyển giao trên hệ suy diễn mờ phức

**Input:** Dữ liệu nguồn; Dữ liệu đích; Tập cơ sở luật miền nguồn:  $CFR_s$ ; tỉ lệ học:  $\eta$ ; kích thước mỗi  $D_{sub}$ :  $K$ .

**Output:**  $R_{ad}(Final)$ : cơ sở luật mờ phức thích nghi với miền đích.

- 1: Chỉnh sửa các tập mờ phức miền nguồn;
- 2:  $k=1$ ;
- 3:  $N_d = \eta * N_\psi$ ; { $N_\psi$  là số lượng dữ liệu trong miền đích}
- 4:  $N_{D_{sub}} = N_d / K$ ; {Số lượng tập con  $D_{sub}(k)$ }
- 5: **while**  $k \leq N_{D_{sub}}$  **do**
- 6:    $R_{ad}(k) \leftarrow CFR_s$ ;
- 7:   Lựa chọn tập con  $D_{sub}(k)$  từ dữ liệu miền đích;
- 8:   **for all**  $t$  in  $D_{sub}(k)$  **do**
- 9:     Thích nghi  $R_{ad}(k)$  với dòng dữ liệu  $t$ ;
- 10:   **end for**
- 11:    $k=k+1$ ;
- 12: **end while**
- 13: Kết hợp các cơ sở luật mờ phức  $R_{ad}(1), R_{ad}(2), \dots, R_{ad}(N_d)$  tạo ra  $R_{ad}(Final)$ ;

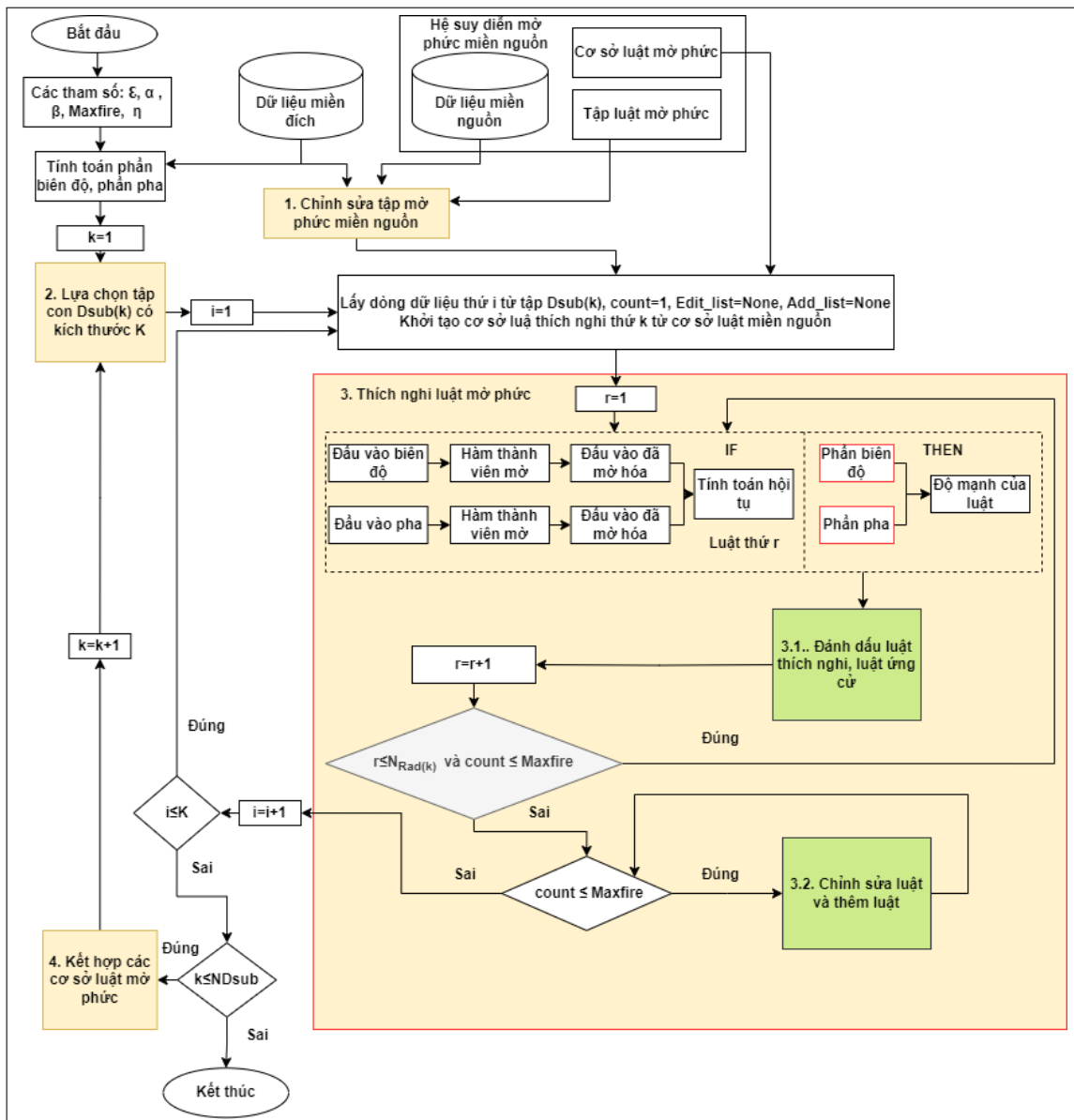
Vấn đề đầu tiên cần xem xét là quyết định có chuyển kiến thức để hỗ trợ nhiệm vụ dự đoán trong miền mục tiêu hay không. Quá trình xem xét này sử dụng một độ tương tự giữa hai miền (Định nghĩa 2.2). Nếu giá trị thu được của việc đo độ tương tự tương đối cao hoặc gần bằng một thì hai miền có độ tương tự cao. Ngược lại, hai miền có độ tương tự thấp. CFTL được sử dụng trong bối cảnh này sẽ cải thiện kết quả của nhiệm vụ trong miền mục tiêu.

Dựa vào luật mờ phức (CFR) và tập mờ phức (CFS) nhận được từ mô hình M-CFIS trước đó, CFTL sẽ điều chỉnh các thành phần mờ phức cho miền mục tiêu. Quá trình thích nghi sử dụng kiến thức thu được từ thông tin trước đó (miền nguồn) để cải thiện nhiệm vụ trong miền mục tiêu. Mô hình này cho phép điều chỉnh các thành phần riêng lẻ của M-CFIS để giải thích cho những thay đổi trong dữ liệu. Các thay đổi từ tình huống này sang tình huống khác được thực hiện bằng cách thay đổi miền CFS và điều chỉnh CFR.

Thiết kế của CFTL (Hình 2.1) bao gồm 4 giai đoạn. Đầu tiên, giai đoạn điều chỉnh khoảng miền sẽ thay đổi thông tin CFS trong miền nguồn để phù hợp với khoảng dữ liệu miền đích. Tiếp theo, một quy trình để chọn ngẫu nhiên các tập hợp con dữ liệu  $D_{sub}$  từ dữ liệu miền đích được thực hiện. Số lượng tập hợp con  $D_{sub}$ , được ký hiệu là  $N_{D_{sub}}$ , đại diện cho dữ liệu của miền đích. Sau đó, mỗi bản ghi dữ liệu trong các tập con  $D_{sub}(k)$  này được sử dụng để thích ứng nghi các luật; mỗi tập hợp con  $D_{sub}$  sẽ tạo ra một tập các CFR thích nghi, ký hiệu là  $R_{ad}$ . Một phương pháp đánh dấu các luật thích nghi và ứng cử được sử dụng trong quy trình điều chỉnh để giảm số lượng luật cần sửa đổi hoặc bổ sung. Cuối cùng, các CFR thích nghi được kết hợp và tạo ra các tập luật thích nghi cuối cùng -  $R_{ad}(Final)$ . Đây chính là tập luật mờ phức được sử dụng để suy luận trong miền đích.

CFTL được thực thi theo các bước trong Thuật toán 2.1 và được mô tả chi tiết trong Hình 2.2.





**Hình 2.2:** Chi tiết mô hình học chuyển giao trên hệ suy diễn mờ phức

### 2.2.1. Chỉnh sửa tập mờ phức miền nguồn

Do khoảng miền dữ liệu là khác nhau trong miền nguồn và miền đích, phần này đưa ra phương pháp điều chỉnh để chuyển giao các tập mờ miền nguồn sang miền đích. Miền dữ liệu của đầu vào/đầu ra trong miền nguồn hoặc miền đích là khoảng, biểu diễn biến ngôn ngữ của các CFS. Mỗi thuộc tính dữ liệu được chia thành các phần có kích thước tương đương hoặc không tương đương và được xác định bởi trọng

tâm tương ứng.

Để điều chỉnh các miền đầu vào nguồn cho phù hợp với các miền đầu vào đích, chúng ta cần thay đổi các khoảng đầu vào nguồn thành các khoảng đầu vào đích. Các thành phần biên độ và pha giữa các thuộc tính miền nguồn và miền đích được so sánh. Biên khoảng miền của các thuộc tính miền nguồn sẽ mở rộng hoặc thu hẹp do dữ liệu từ các khoảng thuộc tính trong miền đích. Mỗi thành phần biên độ và pha trong đầu vào miền đích được so sánh tương ứng với các khoảng biên độ và pha trong miền nguồn. Nếu nó nhỏ hơn biên bên trái, thì biên trái bị thu hẹp; mặt khác, biên phải được mở rộng nếu nó lớn hơn biên phải. Kết quả thu được một tập hợp các trung tâm mới thích ứng với đầu vào miền đích.

Không dễ để xác định một phương pháp cụ thể cho việc thu được kết quả tốt nhất. Việc chọn một chiến lược phù hợp để điều chỉnh dữ liệu nguồn dựa trên vấn đề và phụ thuộc vào bộ dữ liệu. Thuật toán tổng thể để so khớp dữ liệu nguồn với dữ liệu đích được đưa ra trong Thuật toán 2.2.

Đầu vào của thuật toán gồm:

- Dữ liệu miền nguồn (bao gồm phần thực hay phần ảo):  $Source(m_s, n)$
- Dữ liệu miền đích (bao gồm phần thực hay phần ảo):  $Target(m_T, n)$
- Các tâm cụm  $C_{ij}$  của các tập mờ miền nguồn: tâm cụm  $j$  của thuộc tính thứ  $i$ ;  $i = 1, n; j = 1, m_{c_i}$  với  $m_{c_i}$  là số lượng cụm của thuộc tính thứ  $i$ .

Đầu ra của thuật toán:

- Các tâm cụm mới thích nghi với dữ liệu miền đích.

### **2.2.2. Lựa chọn các tập con dữ liệu**

Việc chọn các tập con của dữ liệu đích là điều cần thiết để giảm thông tin vô ích của các mẫu đích cần có trong quy trình thích nghi luật. Mỗi tập hợp con  $D_{sub}$  được lựa chọn có kích thước  $K$  là riêng biệt, tách rời.

Quá trình lựa chọn các tập con  $D_{sub}$  có thể được mô tả như sau: Với dữ liệu trong

---

**Thuật toán 2.2** Chỉnh sửa các tập mờ miền nguồn
 

---

**Input:** Dữ liệu miền nguồn, Dữ liệu miền đích, Tâm cụm của các tập mờ phức miền nguồn:  $C_{ij}$ .

**Output:** Tâm cụm của các tập mờ  $C_{ij}$  của miền đích

```

1:  $R\_Source = \max_{i=1,n}(Source)$ ;           {Tính giá trị lớn nhất theo từng thuộc tính}
2:  $L\_Source = \min_{i=1,n}(Source)$ ;           {Tính giá trị nhỏ nhất theo từng thuộc tính}
3:  $R\_Target = \max_{i=1,n}(Target)$ ;          {Tính giá trị lớn nhất theo từng thuộc tính}
4:  $L\_Target = \min_{i=1,n}(Target)$ ;          {Tính giá trị nhỏ nhất theo từng thuộc tính}
5: for  $i=1$  to  $n$  do
6:    $dr = R\_Target[i] - R\_Source[i]$ ;           {Độ lệch biên phải}
7:    $dl = L\_Target[i] - L\_Source[i]$ ;
8:    $ddr = \text{abs}(dr)/(m_{c_i})$ ;
9:    $ddl = \text{abs}(dl)/(m_{c_i})$ ;
10:  for  $j=1$  to  $m_{c_i}$  do
11:    if  $dr \geq 0$  then
12:       $C[i, j] = C[i, j] + (j + 1) * ddr$ ;           {Mở rộng biên phải}
13:    end if
14:    if  $dr < 0$  then
15:       $C[i, n - j - 1] = C[i, n - j - 1] - (j + 1) * ddl$ ;   {Mở rộng biên trái}
16:    end if
17:    if  $dl \geq 0$  then
18:       $C[i, j] = C[i, j] - (j + 1) * ddr$ ;           {Thu hẹp biên phải}
19:    end if
20:    if  $dl < 0$  then
21:       $C[i, n - j - 1] = C[i, n - j - 1] + (j + 1) * ddl$ ;   {Thu hẹp biên trái}
22:    end if
23:  end for
24: end for
25: return  $C_{ij}$ ;

```

---

miền đích, ta cần chọn ra  $N_d$  ( $N_d = \eta * N_\psi$ ,  $\eta$  là tỷ lệ học) bản ghi được chia thành  $N_{Dsub}$  ( $N_{Dsub} = N_d / K$ ) tập con dữ liệu có kích thước  $K$ . Các bản ghi được chọn là ngẫu nhiên, không chọn lại.

### 2.2.3. Thích nghi luật mờ phức

Trong Hình 2.2, mỗi tập con  $D_{sub}$  đã chọn sẽ được sử dụng để tạo tập luật thích nghi (gọi là  $R_{ad}$ ) thông qua điều chỉnh các luật mờ phức. Việc điều chỉnh CFR dựa

trên cơ chế suy luận của M-CFIS, sử dụng CFR của miền nguồn làm cơ sở luật khởi tạo cho mỗi  $R_{ad}$ . Cơ sở luật này được kiểm tra lặp đi lặp lại để tìm ra các luật thích nghi nhiều nhất với độ mạnh lớn hơn. Điều này giúp cải thiện tốt hơn cho miền mục tiêu.

Mỗi bản ghi dữ liệu  $t$  trong tập con  $D_{sub}$  sẽ tạo ra một sự thích nghi trên tập luật mờ phức  $R_{ad}$ , tức là tạo ra một quá trình hiệu chỉnh hay thêm luật mới nhằm mục đích thích nghi với bản ghi dữ liệu  $t$ . Tham số  $Maxfire$ , được gọi là tốc độ học. Số lượng luật thích nghi ứng với mỗi bản ghi phụ thuộc vào tham số này.

Với mô hình được đề xuất, một quá trình chọn ngẫu nhiên dữ liệu  $N_d$  thay vì lấy toàn bộ dữ liệu miền mục tiêu để tạo luật. Điều đó có nghĩa là  $N_d$  bản ghi dữ liệu này sẽ đại diện cho  $N_\psi$  bản ghi dữ liệu trong miền đích. Giả sử một bản ghi dữ liệu đã tạo ra một luật, thì đối với  $N_\psi$  dữ liệu, tạo ra  $N_\psi$  luật. Trong khi đó  $N_d$  được sử dụng để đại diện cho  $N_\psi$  nhằm giúp tạo cơ sở luật miền mục tiêu. Do đó, với dữ liệu  $N_d$  sau khi thích nghi tối đa sẽ tạo ra  $N_\psi$  luật và mỗi bản ghi sẽ tạo ra  $N_\psi/N_d$  luật. Đây là ý tưởng cơ bản để xác định  $Maxfire$  khi thích nghi cơ sở luật.

Thuật toán thích nghi luật với bản ghi dữ liệu  $t$  được trình bày chi tiết trong thuật toán 2.3. Quá trình thích nghi luật có thể chia thành hai giai đoạn, giai đoạn đầu là đánh dấu các luật thích nghi, luật ứng cử. Giai đoạn sau là chỉnh sửa các luật ứng cử và bổ sung các luật.

---

**Thuật toán 2.3** Thuật toán thích nghi luật mờ phức với bản ghi dữ liệu t
 

---

**Input:** Bản ghi dữ liệu:  $t$ ; Hệ số:  $Maxfire, \alpha, \beta, \varepsilon$ ; Cơ sở luật khởi tạo:  $R_{ad}$ .

**Output:**  $R_{ad}$  thích nghi với bản ghi dữ liệu t.

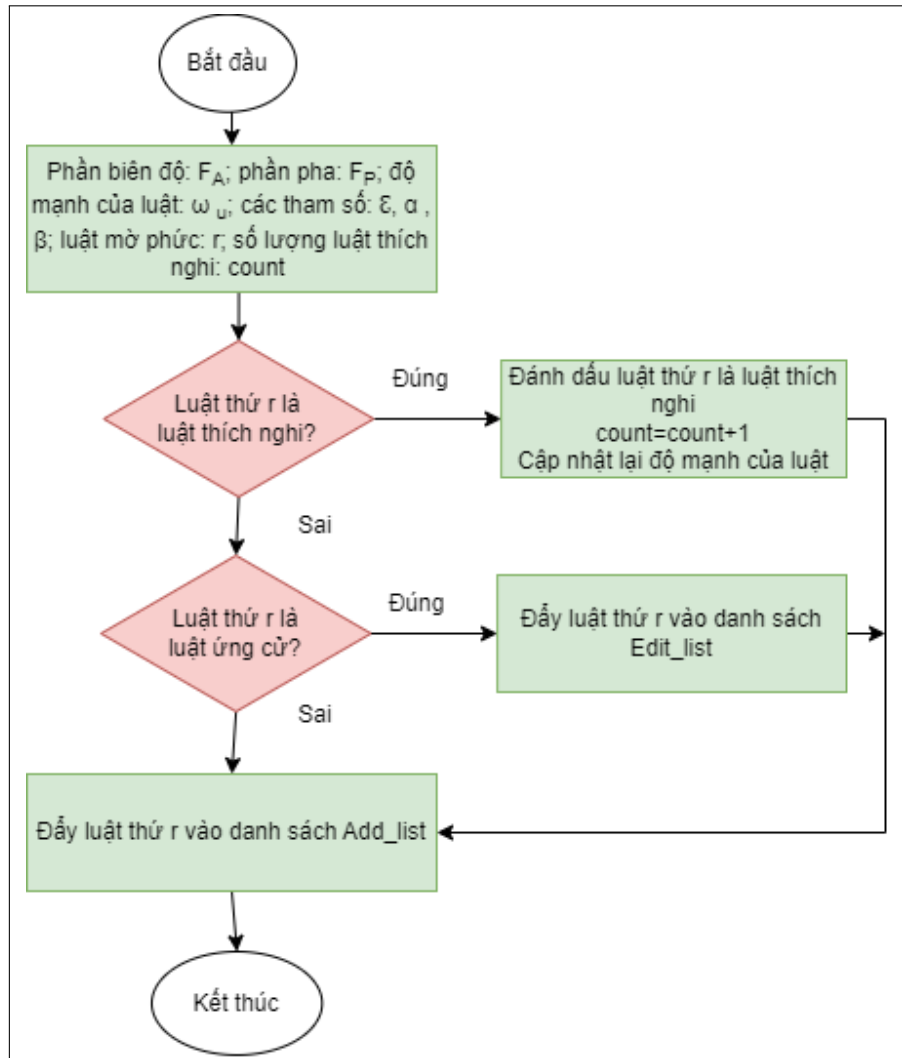
```

1: r=1;                                     { Xét luật đầu tiên }
2: count=1;                                 { Đếm số lượng luật thích nghi }
3: Edit_list=None;
4: Add_list=None;
5:  $fuzzy\_re, fuzzy\_im = Fuzzy(t)$ ;        { Mờ hóa dữ liệu t hàm Gauss }
6: while  $r \leq N_{R_{ad}}$  do
7:   Tính  $F_A, F_B, \omega_r$ ;                { Theo công thức 2.3 }
8:   if  $\omega_r \geq \varepsilon$  and r.label=t.label then
9:     count=count+1;
10:  else
11:    if r.label=t.label and  $\frac{n(F_A > \varepsilon)}{n(F_A)} \geq \alpha$  and  $( (\forall F_A \leq \varepsilon), (F_P \geq \beta) )$  then
12:      Đưa thứ tự luật r vào Edit_list;
13:    else
14:      Đưa thứ tự luật r vào Add_list;
15:    end if
16:  end if
17:  r=r+1;
18: end while
19: j=1;
20: while  $count \leq Maxfire$  and Edit_list is not Empty do
21:    $r_e = R_{ad}[Edit\_list[j]]$ ;             { Lấy thứ tự luật trong Edit_list }
22:   Chính sửa luật  $r_e$ ;
23:   Cập nhật luật  $R_{ad}[Edit\_list[j]]$  là  $r_e$ ;
24:   j=j+1;
25:   count=count+1;
26: end while
27: j=1;
28: while  $count \leq Maxfire$  and Add_list is not Empty do
29:    $r_a = R_{ad}[Add\_list[j]].copy$ ;
30:   Chính sửa luật  $r_a$ ;
31:   Thêm luật  $r_a$  vào  $R_{ad}$ ;
32:   j=j+1;
33:   count=count+1;
34: end while
35: return  $R_{ad}$ ;

```

---

### 2.2.3.1. Xác định luật thích nghi và luật ứng cử



**Hình 2.3:** Đánh dấu luật thích nghi và luật ứng cử

Đánh dấu các luật thích nghi hoặc luật ứng cử nhằm chỉ ra liệu luật đó có phù hợp với dữ liệu thử nghiệm hay không. Việc xác định các luật thích nghi và ứng cử dựa trên tổng hợp độ mạnh của thành phần biên độ và pha. Các luật thích nghi, ứng cử còn được xác định dựa trên thông tin thành phần pha dựa vào ngưỡng  $\beta$  cho trước như được nêu trong Định nghĩa 2.4. Các luật được đánh dấu là luật ứng cử sẽ được sắp trong một danh sách các luật ứng cử gọi là *Edit\_list*. Các luật không là luật thích nghi hay không là luật ứng cử được tiếp tục đánh giá dựa trên độ cháy phần biên độ, phần

pha rồi đưa vào danh sách *Add\_list* cho việc lựa chọn luật để thêm mới.

Một điểm mới và đặc biệt ở đây là việc dùng thành phần pha trong luật mờ phức cho việc lựa chọn luật thích nghi và luật ứng cử. Lý thuyết mờ phức đã nêu bật vai trò của thành phần pha thể hiện tính chất chu kỳ/định kỳ của dữ liệu. Thành phần pha càng nhỏ cho ta một gợi ý về sự lặp lại của yếu tố biên độ càng cao. Đây chính là cơ sở để xác định các thành phần thuộc tính có khả năng được chỉnh sửa trên các luật, cho việc thích nghi các luật với dữ liệu miền đích khi phân phối dữ liệu thay đổi.

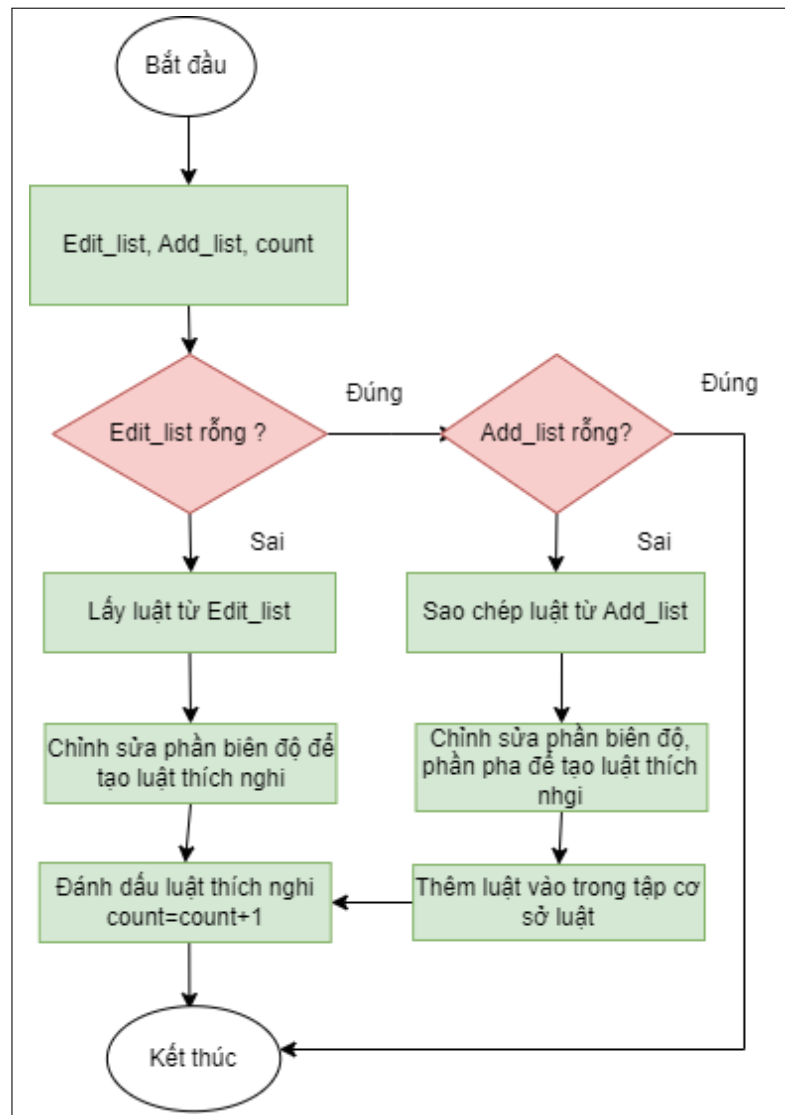
Lưu ý rằng việc, sửa đổi và thêm luật sẽ xảy ra nếu số lượng luật thích nghi chưa đạt đến tỷ lệ *Maxfire* nhất định.

#### 2.2.3.2. Quy trình sửa đổi, bổ sung luật

Quá trình chỉnh sửa luật sẽ diễn ra khi số lượng luật thích nghi chưa đạt đến giá trị *Maxfire* đã cho. Một số luật từ *Edit\_list* sẽ sửa đổi để thích ứng với dữ liệu đầu vào, nghĩa là thay đổi luật thành luật thích nghi. Quá trình chỉnh sửa sẽ thay đổi từng điểm dữ liệu có biên độ mạnh nhỏ hơn  $\varepsilon$ . Sau mỗi lần dịch chuyển, độ mạnh biên độ, độ mạnh của pha và độ mạnh của luật được tính toán lại. Ở bất kỳ thời điểm nào mà giá trị luật thích nghi lớn hơn hoặc bằng  $\varepsilon$ , quá trình chỉnh sửa luật sẽ dừng lại và luật được đánh dấu là luật thích nghi.

Nếu toàn bộ *Edit\_list* đã được kiểm tra và số lượng luật thích nghi chưa đạt đến *Maxfire*, quá trình bổ sung luật mới sẽ được thực thi. Danh sách các luật có thể sử dụng cho mục đích thêm luật nằm trong *Add\_list* theo thứ tự ưu tiên mức độ mạnh của luật. Các biên độ và pha có độ mạnh nhỏ hơn ngưỡng cho trước sẽ được hiệu chỉnh để thu được luật thích nghi. Quá trình dịch chuyển được thực hiện giống như chỉnh sửa biên độ ở chỉnh sửa luật. Sau đó, bổ sung đầu ra của luật đó bằng nhân của bản ghi dữ liệu  $t$  và đánh dấu là luật thích nghi. Luật thích nghi được tạo ra này sẽ được bổ sung vào tập luật thích nghi hiện tại. Quá trình thêm luật diễn ra cho đến khi số lượng luật thích nghi (thích nghi) với bản ghi dữ liệu  $t$  đạt đến *Maxfire*.

Các bước chi tiết của quy trình chỉnh sửa luật được trình bày trong Hình 2.4.



**Hình 2.4:** Quá trình thêm luật và sửa luật

#### 2.2.4. Kết hợp cơ sở luật mờ phức

Sau khi có được tập các CFR thích nghi  $R_{ad}(1), R_{ad}(2), \dots, R_{ad}(k)$ , chúng sẽ được tổng hợp bằng cách loại bỏ các CFR giống nhau và các CFR yếu để có được tập hợp các CFR thích nghi cuối cùng (được đặt tên là  $R_{ad}(Final)$ ) làm tập CFR cho dữ liệu đích.



### 2.3. Phân tích độ phức tạp

Độ phức tạp tính toán của mô hình CFTL được tính như sau:

1. Sự phức tạp tính toán của giai đoạn Chỉnh sửa tập mờ miền nguồn:

- Xác định ranh giới trái phải của từng thuộc tính của miền nguồn và miền đích:

$$O(n_O \cdot (N_\Psi + N_O))$$

- Dịch chuyển vectơ trọng tâm cho mỗi thuộc tính:

$$O(n_c \cdot n_O) \Rightarrow O(n_O(N_\Psi + N_O + n_c)) \approx O(n_O \cdot N_\Psi)$$

2. Sự phức tạp tính toán của giai đoạn Lựa chọn tập con Dsub(k) với K dữ liệu và Thích nghi luật với  $N_{Rad}$  luật:

- Độ phức tạp tính toán của Lựa chọn tập con Dsub(k):  $O(N_\Psi)$

- Độ phức tạp tính toán của Hệ suy diễn mờ phức Mamdani:  $O(n_O \cdot n_c)$

- Độ phức tạp tính toán của Đánh dấu luật thích nghi và luật ứng cử:  $O(10)$

- Độ phức tạp tính toán của Chỉnh sửa luật và thêm luật được xác định:

$$\begin{aligned} & O(n_O + Maxfire \cdot n_O) \\ \Rightarrow & O(K \cdot N_{Rad} \cdot (N_\Psi + n_O \cdot n_c + 10 + Maxfire \cdot n_O)) \\ \approx & O(K \cdot N_\Psi \cdot N_\Psi) = O(K \cdot N_\Psi^2) \end{aligned}$$

- Độ phức tạp tính toán của tổng hợp cơ sở luật mờ phức:  $O(N_\Psi^2)$

Cuối cùng, độ phức tạp tính toán của mô hình đề xuất:

$$O(n_O \cdot N_\Psi + N_{Dsub} \cdot N_\Psi + N_{Dsub} \cdot K \cdot N_\Psi^2 + N_{Dsub} \cdot N_\Psi^2) \approx O(N_\Psi^2)$$

trong đó:

$n_O$ : Số lượng thuộc tính trong Dữ liệu nguồn

$n_c$ : Số cụm

$K$ : Số hàng dữ liệu trong mỗi tập hợp con Dsub(k)

$Maxfire$ : Tỷ lệ luật thích nghi

$N_{\psi}$ : Số hàng dữ liệu trong Dữ liệu đích

$N_O$ : Số hàng dữ liệu trong Dữ liệu nguồn

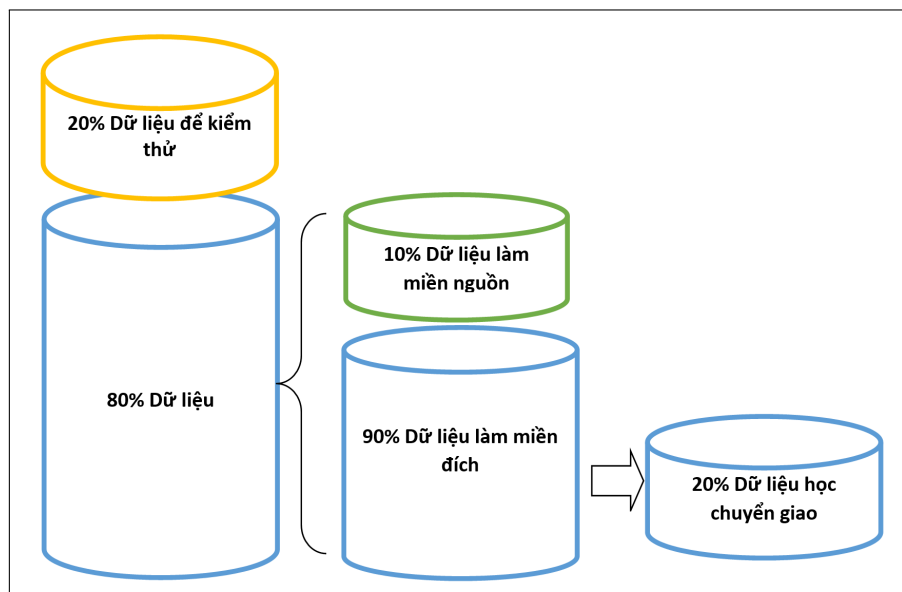
$N_{Rad}$ : Số luật trong CFR thích ứng

$N_{Dsub}$ : Số tập hợp con Dsub

## 2.4. Thực nghiệm

### 2.4.1. Kịch bản thực nghiệm

Với mục đích đánh giá hiệu năng của CFTL, CFTL được sử dụng cho việc tạo ra mô hình suy diễn mờ phức cho miền mục tiêu trên một phần dữ liệu nhỏ thay vì sử dụng toàn bộ dữ liệu miền đích. Kịch bản thực nghiệm trên CFTL được thực hiện như Hình 2.5.



**Hình 2.5:** Kịch bản thực nghiệm trên CFTL

- Từ một tập dữ liệu ban đầu sau khi tách ra ngẫu nhiên một lượng dữ liệu kiểm thử (20%), dữ liệu còn lại (80%), tiếp tục tách ra một phần coi là miền nguồn và một phần coi là miền đích. Để đảm bảo ngữ cảnh giống với bài toán đã đặt

ra (chương 1), trong thực nghiệm, chỉ lấy một lượng dữ liệu nhỏ (10%) là miền nguồn còn lại (90%) là miền đích. Việc tách ra dữ liệu coi là miền nguồn được thực hiện theo phương pháp lấy mẫu ngẫu nhiên đơn giản đảm bảo dữ liệu miền nguồn và đích có cùng phân bố nhưng có thể khác nhau về giá trị khoảng miền của các thuộc tính.

- Trong quá trình thực hiện học chuyển giao trên mô hình CFTL một lượng dữ liệu miền đích tiếp tục được lấy cho quá trình thích nghi luật theo tỉ lệ học (20%), phần dữ liệu này được lấy ngẫu nhiên chia thành các tập Dsub cho quá trình học chuyển giao trên mô hình đề xuất.

Mô hình CFTL được so sánh với phương pháp mô hình M-CFIS truyền thống (nơi mà toàn bộ 80% dữ liệu sẽ được sử dụng để sinh luật cho hệ suy diễn miền đích), các kết quả thực nghiệm được đánh giá dựa trên ba chỉ số: thời gian tính toán, độ chính xác và số lượng luật.

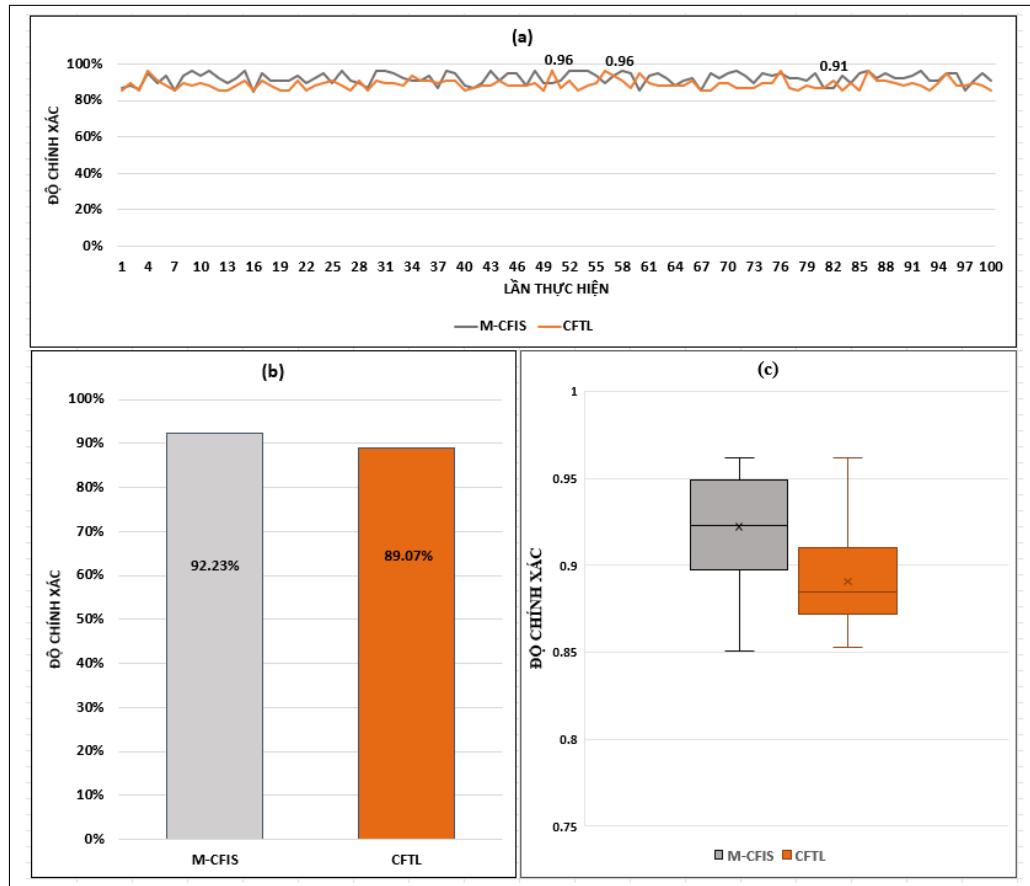
#### ***2.4.2. Kết quả thực nghiệm***

Sau khi tiến hành các thử nghiệm, kết quả thu được trên 4 bộ dữ liệu tương ứng với hai kịch bản được trình bày lần lượt trong Hình 2.6 đến Hình 2.21.

##### ***2.4.2.1. Bộ dữ liệu Diabetes***

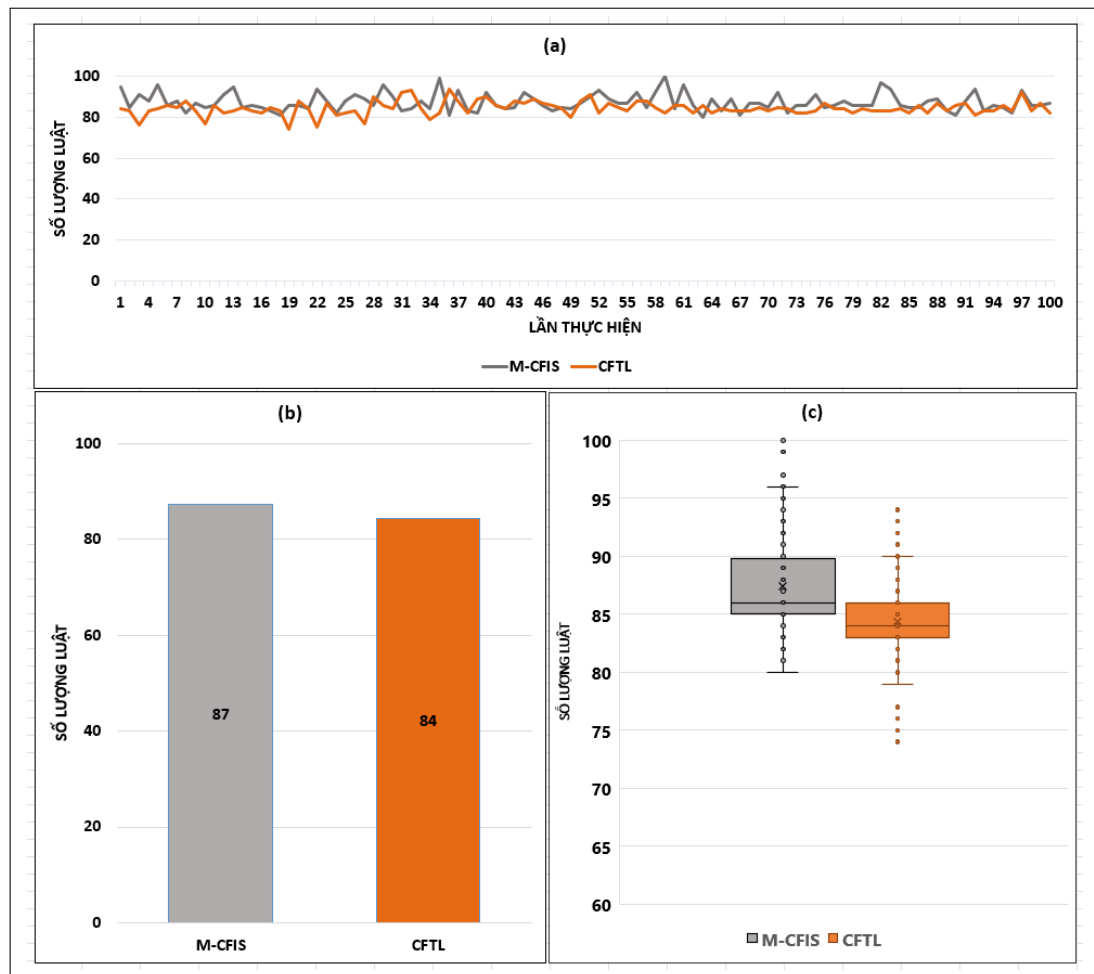
Hình 2.6 mô tả các kết quả thử nghiệm về độ chính xác của mô hình đề xuất CFTL và M-CFIS trên bộ dữ liệu Diabetes. Như kết quả được biểu diễn trong Hình 2.6 (a), Hình 2.6 (b) và Hình 2.6 (c), độ chính xác của mô hình CFTL thấp hơn so với mô hình M-CFIS. Trong khi mô hình M-CFIS có độ chính xác trung bình là 92,23 % thì độ chính xác của CFTL là 89,07%. Tuy nhiên, có một số lần thực hiện, độ chính xác của CFTL cao hơn M-CFIS như trong Hình 2.6 (a). Độ chính xác lớn nhất cả hai mô hình đạt được là tương tự nhau và trong tất cả các lần thực hiện (100 lần) thì độ chính xác

của mô hình CFTL đều lớn hơn độ chính xác thấp nhất của mô hình M-CFIS (Hình 2.6 (c)).



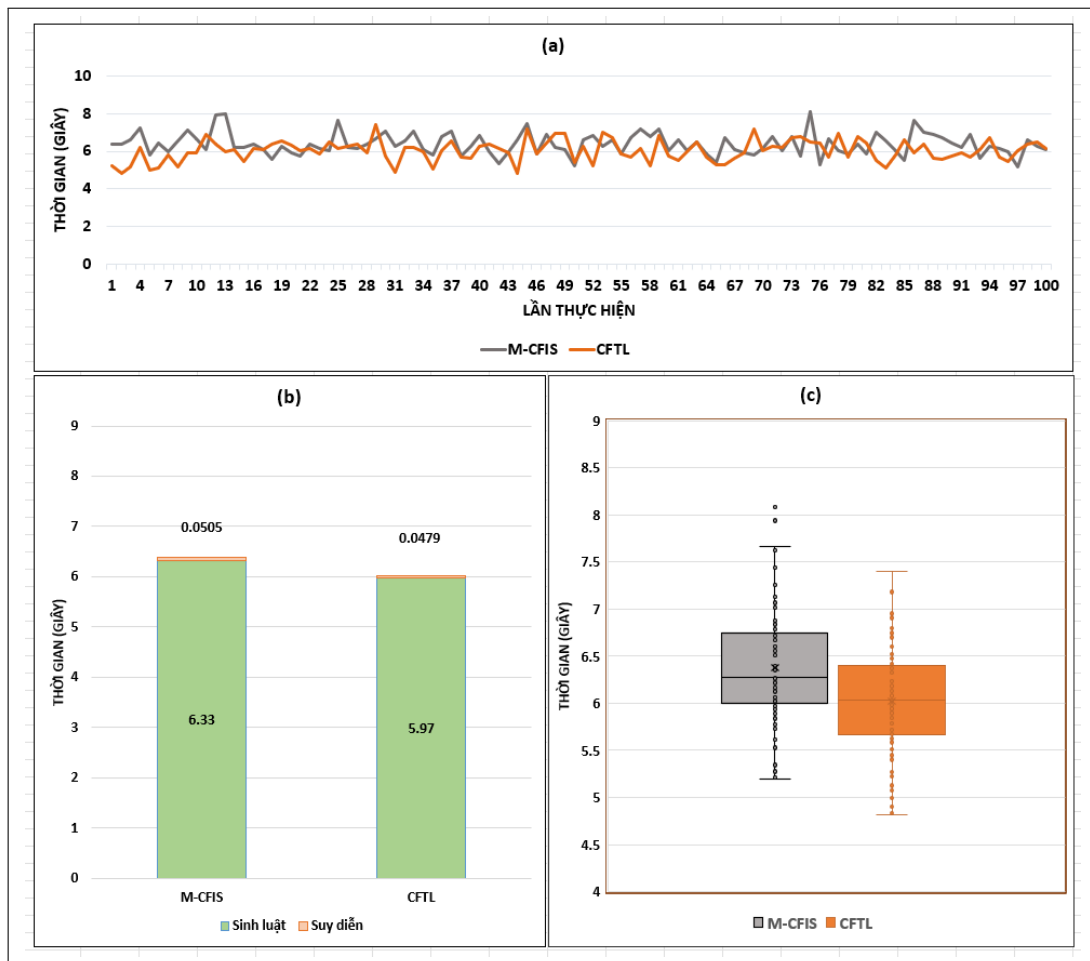
**Hình 2.6:** Kết quả đo lường độ chính xác trên tập dữ liệu Diabetes.

Hình 2.7 cho thấy số lượng luật giữa hai mô hình. Có thể thấy số lượng luật của CFTL ít hơn một chút so với M-CFIS (trung bình chỉ thấp hơn 3 luật - Hình 2.7 (b)). Kết quả này là dễ hiểu và nó phù hợp với độ chính xác tương ứng của hai mô hình. Trong một số trường hợp, số lượng luật sinh ra bởi mô hình CFTL là ít hơn khá nhiều 75 luật (Hình 2.7 (c)). Khi xem xét kết hợp cùng với độ chính xác thu được ở trên, số lượng luật ít thì độ chính xác thu được vẫn lớn hơn độ chính xác thấp nhất của MCFIS.



**Hình 2.7:** Kết quả đo lường số lượng luật sinh ra trên tập dữ liệu Diabetes.

Điều đặc biệt là, thời gian tạo cơ sở luật của CFTL nhỏ hơn so với phương pháp tạo cơ sở luật của M-CFIS truyền thống và tương tự đối với quá trình suy diễn (Hình 2.8 (a), (b) và (c)). Nhưng, mức độ chênh lệch thời gian của hai mô hình vẫn là không đáng kể (0,0505 và 0,0479 ở M-CFIS và CFTL tương ứng - Hình 2.8 (b)). Kết quả này là do tổng số dữ liệu của tập dữ liệu nhỏ, chỉ có 390 bản ghi. Vì vậy, các tập Dsub cũng có kích thước nhỏ, kiến thức để huấn luyện và kiểm thử còn hạn chế. Kết quả thực nghiệm so sánh hai mô hình trên bộ dữ liệu cho thấy không có nhiều khác biệt giữa CFTL và M-CFIS. Một trong những nguyên nhân dẫn đến kết quả đó là do số lượng bản ghi trong tập dữ liệu quá ít.



**Hình 2.8:** Kết quả đo lường thời gian thực hiện trên tập dữ liệu Diabetes.

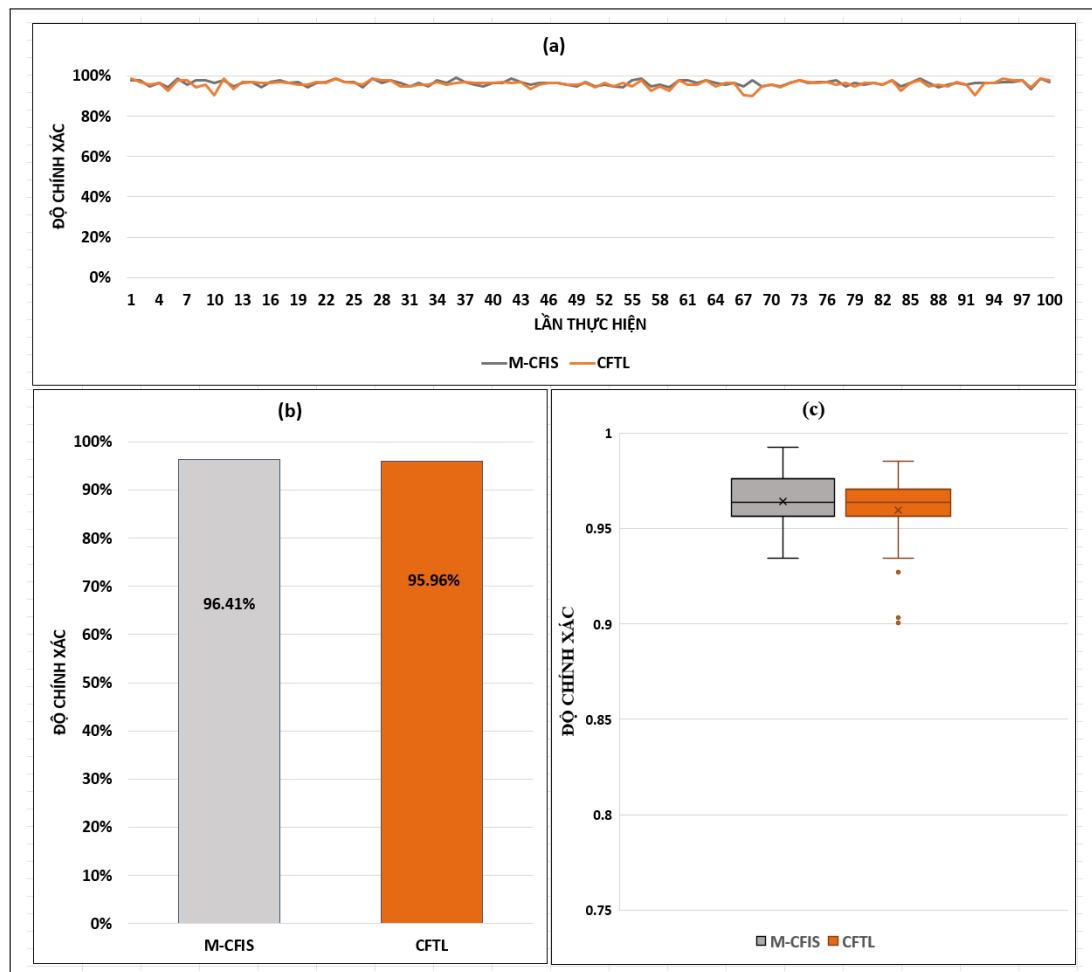
Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 2.9. Kết quả đo lường cho thấy mô hình CFTL ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so với M-CFIS.

	M-CFIS	CFTL
Phương sai	0.34156365	0.31573403
Độ lệch chuẩn	0.58443447	0.56190216

**Hình 2.9:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập Diabetes.

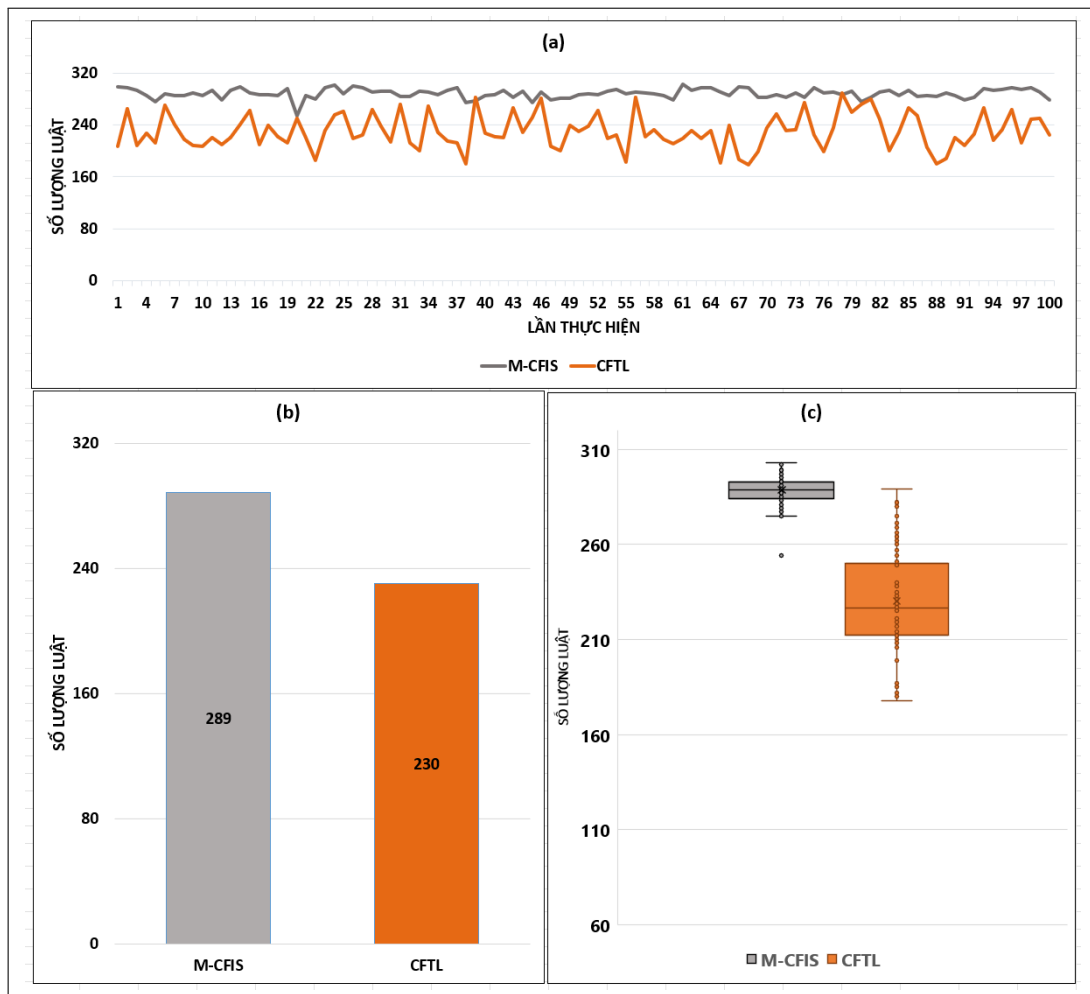
#### 2.4.2.2. Bộ dữ liệu BreastCancer

Kết quả thử nghiệm mô hình CFTL trên bộ dữ liệu BreastCancer được trình bày trong Hình 2.10, Hình 2.11, Hình 2.12. Độ chính xác của M-CFIS và CFTL trong Hình 2.10 (a), (b) và (c) với giá trị trung bình độ chính xác tương ứng là 95.96% trên mô hình CFTL và 96,43% trên mô hình M-CFIS. Như vậy, độ chính xác của hai mô hình có thể được giả định là tương đương nhau.



**Hình 2.10:** Kết quả đo lường độ chính xác trên tập dữ liệu BreastCancer.

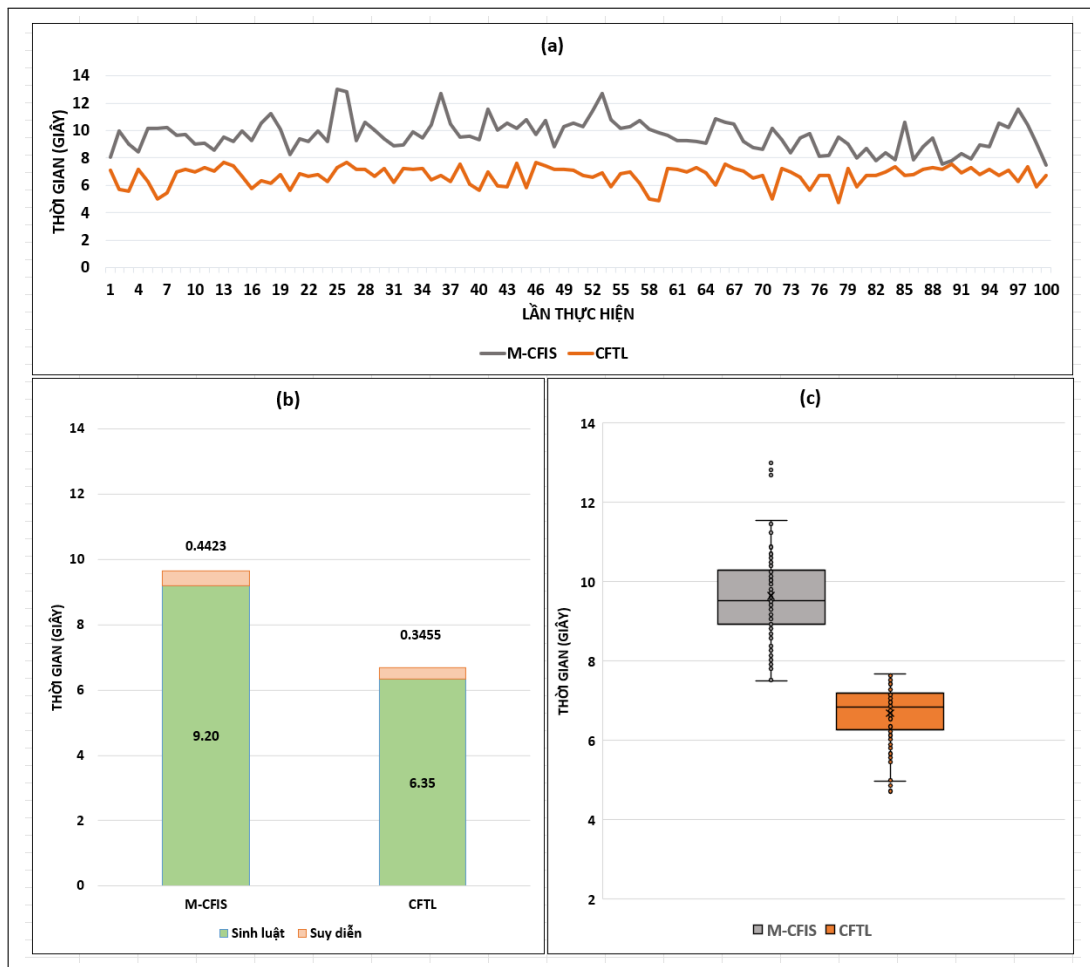
Tuy nhiên, số lượng luật trung bình trong Hình 2.11(b) của CFTL là 230, ít hơn 59 luật so với kết quả của M-CFIS 2.11(b). Do đó, cơ sở luật của CFTL có ít luật hơn M-CFIS.



**Hình 2.11:** Kết quả đo lường số lượng luật trên tập dữ liệu Diabetes.

Hơn nữa, mức tiêu thụ thời gian của CFTL thấp hơn 35% so với M-CFIS (Hình 2.12 (a), 2.12 (b) và 2.12 (c)). Trung bình, tổng thời gian sinh luật là 6,35 và 9,20 tương ứng với CFTL và M-CFIS, trong khi giá trị độ chính xác là như nhau. Những kết quả thử nghiệm này cho thấy CFTL tốt hơn đáng kể trong bộ dữ liệu BreastCancer. Như vậy, đối với bộ dữ liệu BreastCancer - có số lượng bản ghi lớn, mô hình CFTL đã thể hiện rõ ưu thế về thời gian thực hiện. Thời gian tính toán trong mỗi lần thực hiện gần như luôn ít hơn nhiều so với M-CFIS. Ngoài ra, với những lần thử nghiệm mà CFTL có thời gian chạy cao hơn M-CFIS, tương ứng với những thời điểm số luật nhiều hơn và độ chính xác của CFTL trong những lần đó cao hơn M-CFIS.





**Hình 2.12:** Kết quả đo lường thời gian thực hiện trên tập dữ liệu Diabetes.

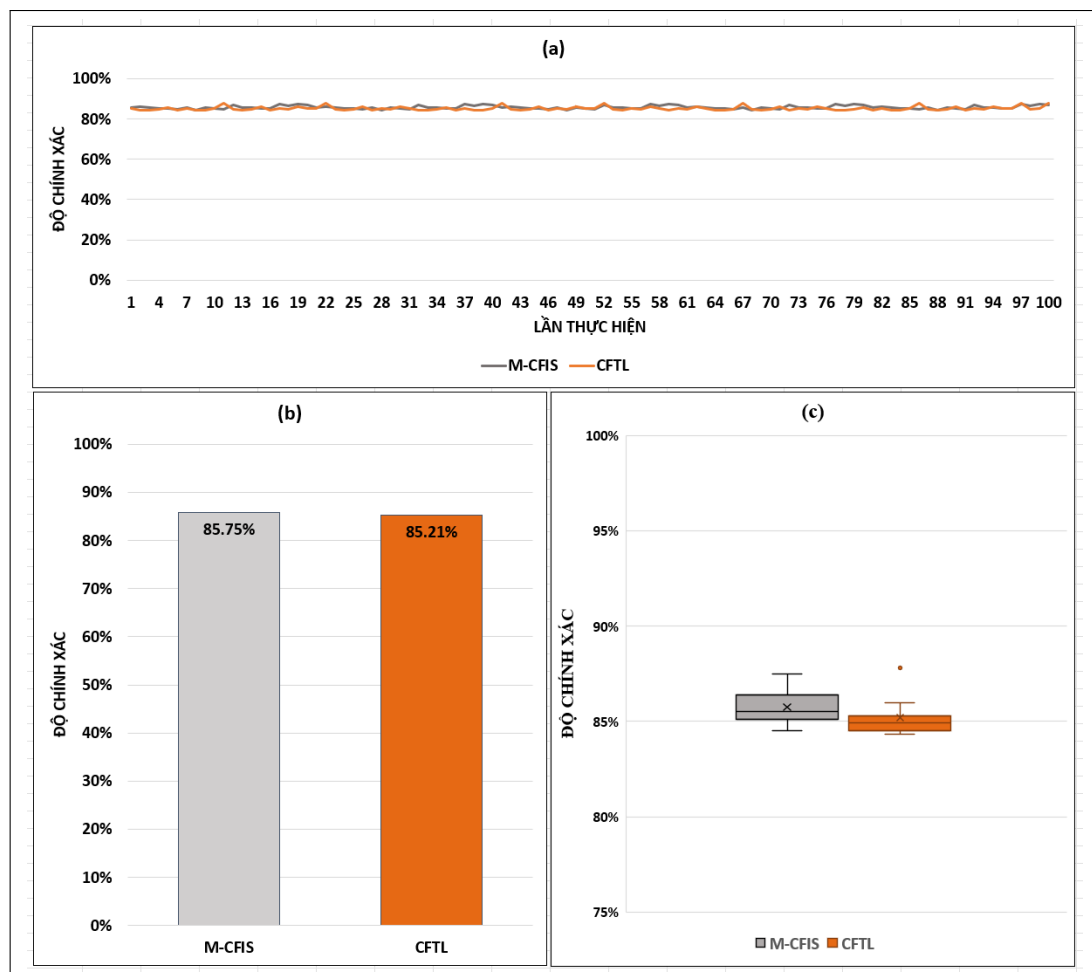
Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 2.13. Kết quả đo lường cho thấy mô hình CFTL ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so với M-CFIS.

	M-CFIS	CFTL
Phương sai	1.268646981	0.461671166
Độ lệch chuẩn	1.126342302	0.679463881

**Hình 2.13:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập BreastCancer.

### 2.4.2.3. Bộ dữ liệu CreditCard

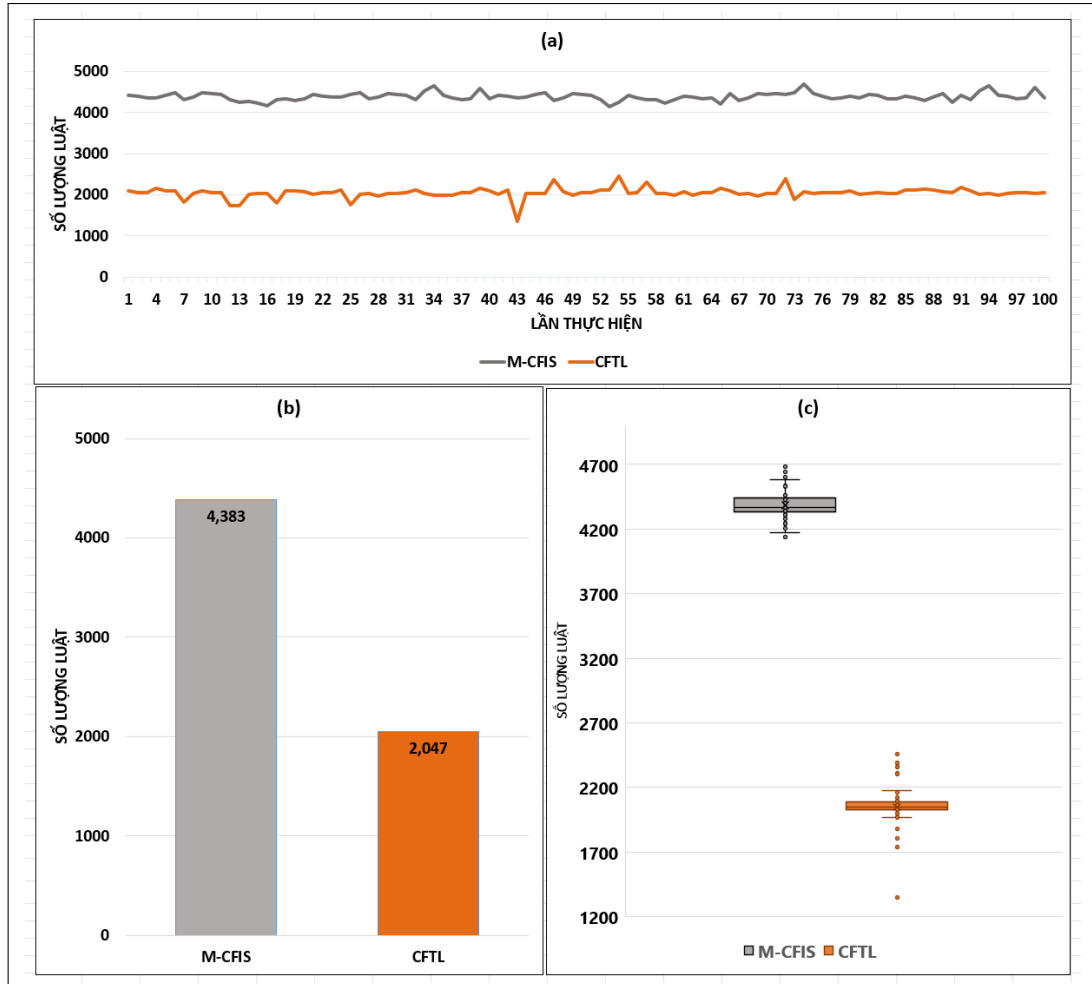
Với bộ dữ liệu CreditCard, hiệu suất thu được trên CFTL và M-CFIS được trình bày trong Hình 2.14, Hình 2.15, Hình 2.16. Giống với kết quả thử nghiệm trên bộ dữ liệu BreastCancer, không có nhiều khác biệt về độ chính xác giữa chúng (cụ thể là CFTL - 85,21% và M-CFIS - 85,75%).



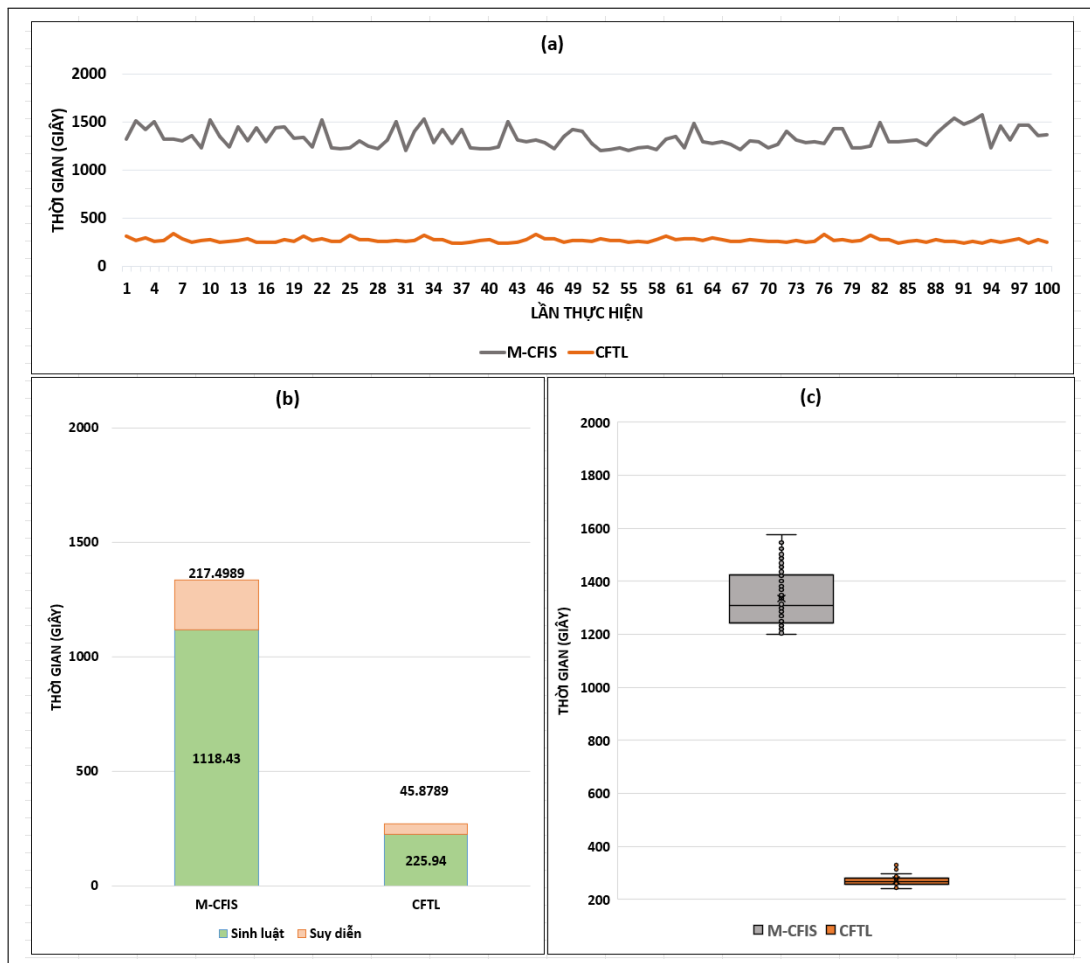
**Hình 2.14:** Kết quả đo lường độ chính xác trên tập dữ liệu CreditCard.

Đặc biệt trong Hình 2.15 và Hình 2.16, số lượng luật và thời gian thực hiện của M-CFIS cao hơn đáng kể so với CFTL. Số lượng luật trung bình của CFTL giảm 48% so với M-CFIS (nghĩa là 2047 luật và 4403 luật tương ứng như trong Hình 2.15 (b)). Và tỷ lệ thời gian của M-CFIS và CFTL (trung bình) cao hơn 4,148 lần như trong

Hình 2.16 (b)). CFTL tiếp tục cho thấy khả năng hoạt động tốt hơn trên bộ dữ liệu có kích thước bản ghi lớn.



**Hình 2.15:** Kết quả đo lường số lượng luật trên tập dữ liệu CreditCard.



**Hình 2.16:** Kết quả đo lường thời gian thực hiện trên tập dữ liệu CreditCard.

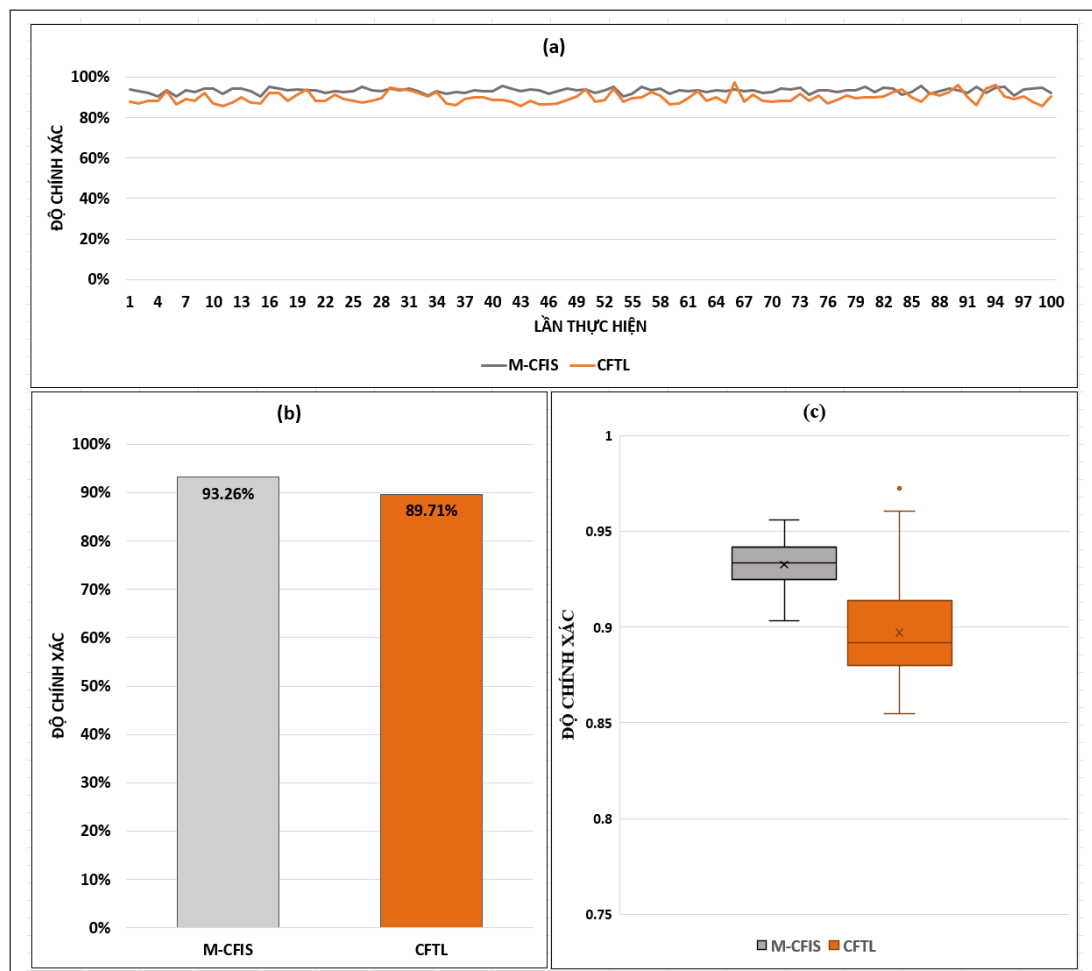
Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 2.17. Kết quả đo lường cũng cho thấy mô hình CFTL ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so với M-CFIS.

	M-CFIS	CFTL
Phương sai	9885.061651	460.8195573
Độ lệch chuẩn	99.42364734	21.46670812

**Hình 2.17:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập CreditCard.

#### 2.4.2.4. Bộ dữ liệu Liver

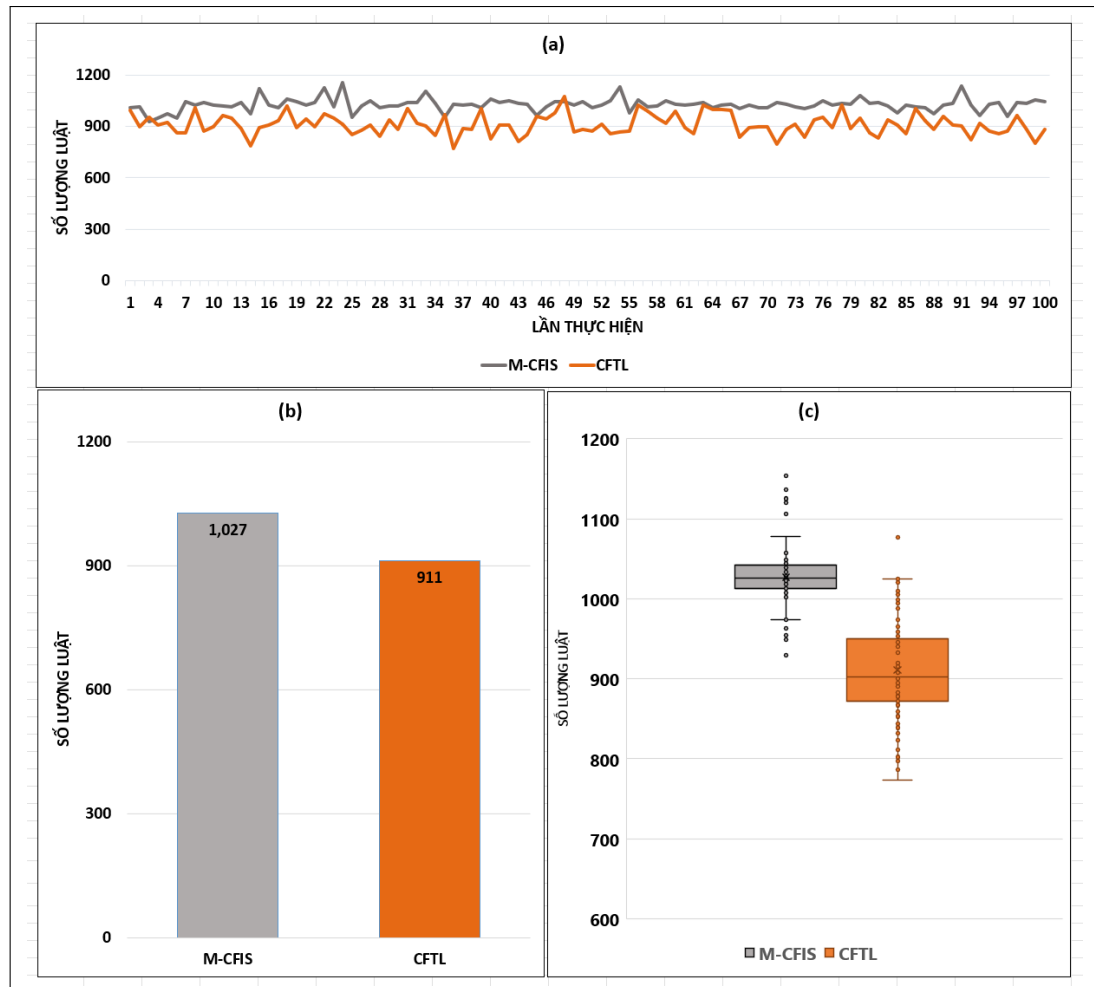
Độ chính xác, số lượng luật và thời gian thực hiện trên mô hình CFTL và M-CFIS khi với tập dữ liệu thực tế (Liver) được trình bày trong Hình 2.18, Hình 2.19, Hình 2.20. Kết quả thu được có đặc điểm tương tự như trong các bộ dữ liệu trước đó. Như thể hiện trong Hình 2.18(b), giá trị của độ chính xác CFTL thấp hơn khá nhiều so với M-CFIS. Các giá trị độ chính xác tương ứng với CFTL và M-CFIS trung bình là 89,71% và 93,26%. Tuy nhiên, vẫn có những lần thực hiện mà độ chính xác của mô hình CFTL là cao hơn so với mô hình M-CFIS (Hình 2.18(a))



**Hình 2.18:** Kết quả đo lường độ chính xác trên tập dữ liệu Liver.

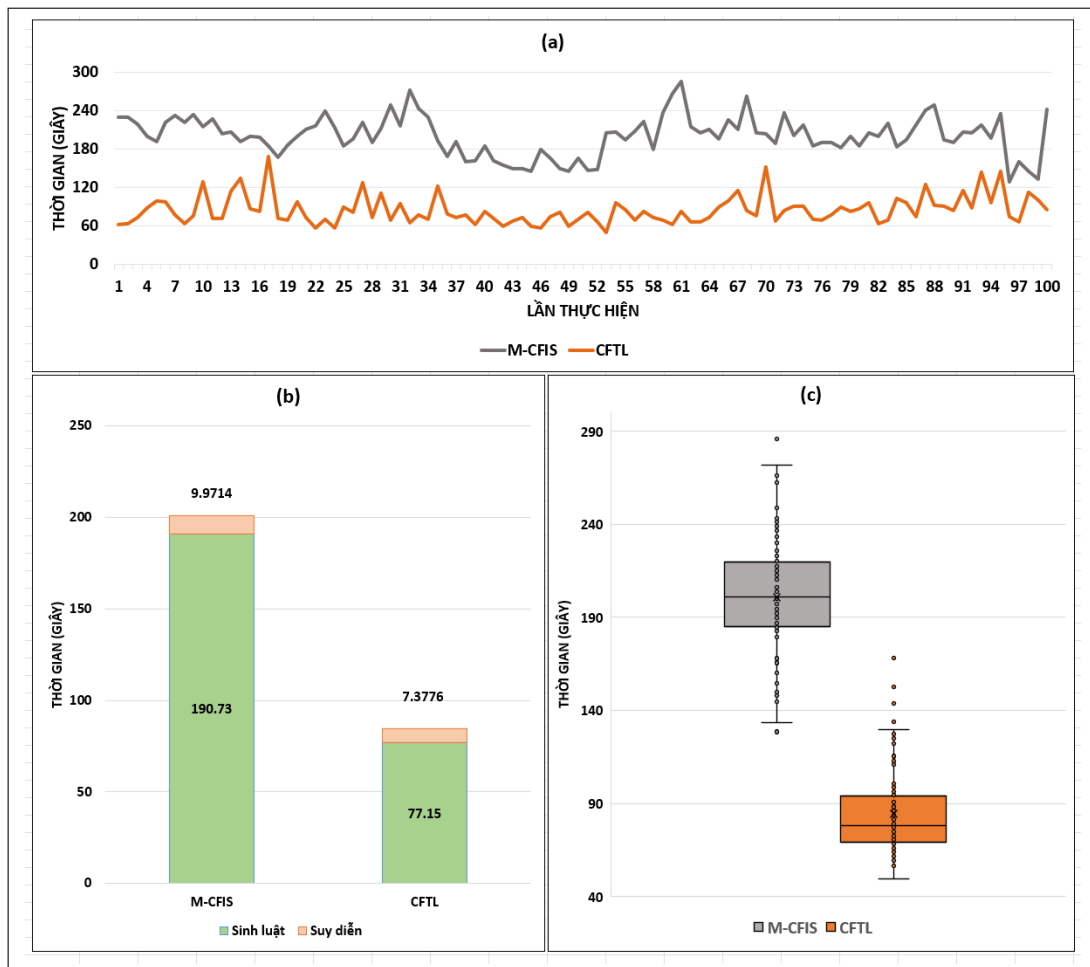
Tương tự như bộ dữ liệu BreastCancer và CreditCard trước đây, số lượng luật của

CFTL thấp hơn nhiều so với M-CFIS (trung bình có 911 luật và 1027 luật tương ứng như trong Hình 2.19 (b)).



**Hình 2.19:** Kết quả đo lường số lượng luật trên tập dữ liệu Liver.

Trong khi đó, thời gian tính toán của CFTL cũng giảm trung bình 2,37 lần bao gồm cả thời gian sinh luật và suy diễn (tương ứng 84.52 giây và 200.70 giây).



**Hình 2.20:** Kết quả đo lường thời gian thực hiện trên tập dữ liệu Lvier.

Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 2.13. Kết quả đo lường, cho thấy mô hình CFTL ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so với M-CFIS.

	M-CFIS	CFTL
Phương sai	963.3746765	498.7194385
Độ lệch chuẩn	31.0382776	22.33202719

**Hình 2.21:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian thực hiện trên tập Liver.

Các kết quả thực nghiệm chỉ ra rằng mô hình CFTL được đề xuất đã cải thiện mặt thời gian và số lượng luật trên những bộ dữ liệu có kích thước lớn (số lượng bản ghi và số lượng thuộc tính). Hơn nữa, thực nghiệm cũng minh chứng khả năng CFTL hoạt động hiệu quả trong trường hợp thiếu kiến thức trong miền mục tiêu.

Kết quả thực nghiệm trên cả hai loại dữ liệu: dữ liệu chuẩn và dữ liệu thực tế đều cho thấy tính hiệu quả và ý nghĩa về mặt thời gian thực hiện của CFTL.

## 2.5. Kết luận Chương 2

Trong chương này, một mô hình học chuyển giao mờ phức dựa trên hệ thống suy diễn mờ phức (CFIS) đã được đề xuất nhằm giảm thời gian xây dựng mô hình cho miền mục tiêu thông qua việc học kiến thức từ miền liên quan. Hệ thống được đề xuất đã chọn dữ liệu từ miền đích để điều chỉnh cơ sở luật miền nguồn theo cơ chế của hệ suy diễn M-CFIS. Kích thước dữ liệu được chọn ít hơn nhiều so với toàn bộ dữ liệu trong miền đích để giảm thời gian tính toán. Kỹ thuật học chuyển giao mờ đã được áp dụng để chuyển giao hệ suy diễn mờ phức miền nguồn (tập luật mờ phức miền nguồn và cơ sở luật mờ phức miền nguồn) sang miền đích bằng giai đoạn điều chỉnh thích nghi miền nguồn và thích nghi luật mờ phức.

Kết quả thực nghiệm trên bộ dữ liệu UCI và bộ dữ liệu thực tế cho thấy hệ thống CFTL mang lại kết quả như mong muốn trong với tình huống cần xây dựng một CFIS miền đích (với dữ liệu khá lớn) trong khoảng thời gian hạn chế bằng cách tái sử dụng hệ thống suy diễn mờ phức của miền tương tự có sẵn (miền nguồn).

Kết quả nghiên cứu này được công bố trong công trình [3] của luận án.

Mô hình CFTL đã rút ngắn thời gian tạo ra hệ M-CFIS cho miền mục tiêu nhờ khả năng chuyển giao luật, kết hợp với dữ liệu hạn chế trong miền đích. Trong quá trình thích nghi luật với dữ liệu mới, mô hình đề xuất cũng đã sử dụng thành phần pha trong việc lựa chọn luật giữ lại và luật chỉnh sửa. Điều này cho thấy vai trò của yếu tố bổ sung (chu kỳ/định kỳ) của dữ liệu trong các mô hình suy diễn. Tuy nhiên, mô



hình học chuyển giao CFTL đề xuất chưa tận dụng được thành phần Pha này trong quá trình chỉnh sửa các luật. Bên cạnh đó, mặc dù đã rút ngắn được thời gian tạo cơ sở luật mờ phức cho hệ suy diễn mờ phức của miền đích, nhưng thời gian suy diễn mờ phức lại chưa được xem xét đến. Đây chính là những hạn chế của mô hình CFTL đã đề xuất và cũng là những vấn đề nghiên cứu được đặt ra cho nghiên cứu tiếp theo tại chương 3.

## CHƯƠNG 3. HỆ HỌC CHUYỂN GIAO MỜ PHỨC DỰA TRÊN CẤU TRÚC CFRG

### 3.1. Giới thiệu

Giảm thời gian suy diễn trên hệ suy diễn mờ có thể thực hiện bằng nhiều phương pháp khác nhau như tối ưu hóa luật suy diễn, sử dụng biểu đồ quan hệ, chia nhỏ hệ thống v.v. Việc giảm thời gian suy diễn trên hệ suy diễn mờ phức là rất cần thiết, vì thành phần pha đã làm tăng thời gian suy diễn của hệ suy diễn mờ phức lên đáng kể. Bên cạnh đó, việc giảm thời gian suy diễn trên hệ suy diễn mờ phức cũng làm giảm thời gian cho mô hình học chuyển giao CFTL đã đề xuất tại chương 2, do quá trình thích nghi luật của mô hình CFTL được xây dựng dựa trên cơ chế suy diễn mờ phức. Ngoài ra, một cơ chế gợi ý lựa chọn giá trị phần biên độ và pha khi chỉnh sửa luật trong giai đoạn thích nghi luật dễ dàng hơn, cũng sẽ giúp giảm thời gian tìm kiếm giá trị biên độ hay pha mới. Và tổng thể cũng giúp thời gian học chuyển giao trên hệ suy diễn mờ phức sẽ cải tiến về mặt thời gian hơn nữa.

Cấu trúc dữ liệu cây luôn là một cấu trúc dữ liệu tốt nhằm biểu diễn dữ liệu có mối quan hệ thứ bậc, dễ dàng cho truy xuất dữ liệu nhanh. Đặc biệt là cây quyết định - một trong những kỹ thuật nổi tiếng giúp phân loại từ dữ liệu [83]–[85] và các biến thể của nó như RandomForest [86], [87], Gradient Boosting Trees [88], cây quyết định mờ (Fuzzy Decision Tree - FDT) [85], [89]. Trong FDT, sự mờ hoá được áp dụng cho dữ liệu và quyết định, tạo điều kiện cho nó hoạt động hiệu quả hơn trong các tình huống khi dữ liệu không luôn luôn rõ ràng và chính xác [90]–[98]. Cấu trúc dữ liệu cây cũng đã được tích hợp vào các phương pháp học chuyển giao [11], [99], [100]. Sự kết hợp này đã tạo ra những kết quả tuyệt vời xét về độ phức tạp tính toán và khả năng diễn giải [101], [102].

Một cấu trúc khác cũng hiệu quả trong biểu diễn dữ liệu, tri thức là đồ thị. Trong

đó, đồ thị có hướng (Directed Acyclic Graph - DAG) là một cấu trúc dữ liệu rất quan trọng trong lĩnh vực khoa học máy tính và các lĩnh vực khác. DAG là một loại đồ thị mà có các đỉnh và các cạnh chỉ định hướng di chuyển từ một đỉnh này đến một đỉnh khác, và không có chu trình nào tồn tại trong đồ thị [103]. DAG có mặt trong nhiều nghiên cứu thuộc các lĩnh vực nghiên cứu khác nhau như y tế [104], [105], dự đoán rủi ro [106] v.v. Cấu trúc đơn giản, có hướng, không chu trình của DAG đáp ứng nhu cầu phân tích dữ liệu, quy trình, truy cập nhanh v.v.

Việc sử dụng cây và đồ thị để biểu diễn thông tin có nhiều ưu điểm vì chúng phản ánh cấu trúc và mối quan hệ giữa các phần tử trong dữ liệu một cách rõ ràng, tăng khả năng tìm kiếm truy xuất dữ liệu nhanh chóng, có thể dễ dàng mở rộng và giúp linh hoạt trong việc mô tả sự phức tạp của thông tin.

Tiếp nối nghiên cứu từ chương 2, chương này đề xuất sử dụng cấu trúc CFRG (dựa theo cấu trúc DAG và Tree) để biểu diễn luật mờ phức nhằm mục đích:

- Biểu diễn trực quan các luật mờ phức bao gồm phần biên độ và phân pha trên cấu trúc CFRG.
- Đề xuất một số thuật toán trên cấu trúc CFRG.
- Cải thiện hiệu suất của quá trình suy diễn mờ phức về mặt thời gian.
- Ứng dụng cấu CFRG để cải tiến mô hình CFTL đã đề xuất tại chương 2.

Để xây dựng cấu trúc CFRG, một số khái niệm và định nghĩa được giới thiệu như dưới đây.

### **3.2. Một số khái niệm và định nghĩa**

Với mục tiêu định nghĩa một cấu trúc CFRG để biểu diễn tập luật mờ phức bao gồm giá trị biên độ, giá trị pha và đáp ứng những mục đích đã nêu, một cấu trúc nút mờ phức được giới thiệu trong Định nghĩa 3.1.

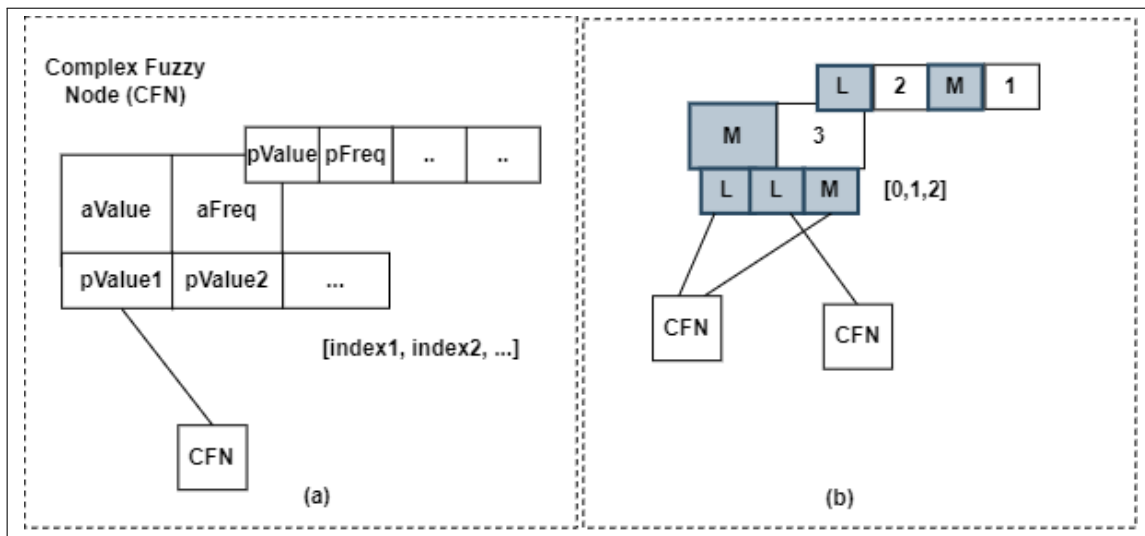
**Định nghĩa 3.1.** Nút mờ phức (Complex fuzzy node - CFN)

Một nút mờ phức  $CFN$  là một cấu trúc bao gồm 3 phần  $(p_A, p_P, p_I)$ . Trong đó:

- Phần thứ nhất  $p_A$  là một cặp  $(aValue, aFreq)$  trong đó  $aValue$  là biến ngôn ngữ phần biên độ và  $aFreq$  tần suất xảy ra của nó trên cùng một thuộc tính.
- Phần thứ hai  $p_P$  là một danh sách các cặp  $(pValue, pFreq)$ , mỗi cặp gồm biến ngôn ngữ phần pha  $pValue$  và tần suất xảy ra của nó trên cùng một thuộc tính gọi là  $pFreq$ .
- Phần thứ ba  $p_I$  là danh sách các liên kết để kết nối đến các nút con theo thứ tự luật và phần pha.

Hình 3.1 (a) biểu diễn cấu trúc của một nút mờ phức theo Định nghĩa 3.1, và Hình 3.1 (b) là một ví dụ của nút mờ phức bao gồm:

- Phần thứ nhất  $p_A$ , giá trị biên độ  $aValue = M$  và tần suất là  $aFreq = 3$ .
- Phần thứ hai  $p_P$ , một hai cặp giá trị pha: cặp thứ nhất  $(pValue = L, pFreq = 2)$ , cặp thứ hai  $(pValue = M, pFreq = 1)$ .
- Phần thứ ba  $p_I$ , là một danh sách các liên kết đến các nút con thông qua phần pha  $[L, L, M]$  với chỉ số thứ tự tương ứng là  $[0, 1, 2]$ .



**Hình 3.1:** Cấu trúc một nút mờ phức (a) và ví dụ một nút mờ phức (b)

Từ định nghĩa nút mờ phức trên, định nghĩa về cấu trúc CFRG được phát biểu như sau:

**Định nghĩa 3.2.** Cấu trúc CFRG (Complex fuzzy rule Graph - CFRG)

Một cấu trúc CFRG là một cấu trúc tựa cây biểu diễn một tập các luật mờ phức. Trong đó:

- Mỗi nút trên CFRG là một CFN mà biểu diễn thuộc tính của một hay nhiều luật có cùng biến ngôn ngữ phần biên độ nhưng có thể khác nhau về phần pha.
- Mỗi cạnh biểu diễn toán tử T-norm (AND hoặc OR).
- Các phần pha của nút thuộc cùng một luật sẽ có cùng chỉ số.

Cấu trúc CFRG được xây dựng dựa theo cấu trúc cây và cấu trúc DAG. Theo định nghĩa về CFRG thì nút lá (leaf) là nút không có nút con hay trong trường hợp này nút lá là nút có giá trị phần pha và danh sách các con là None.

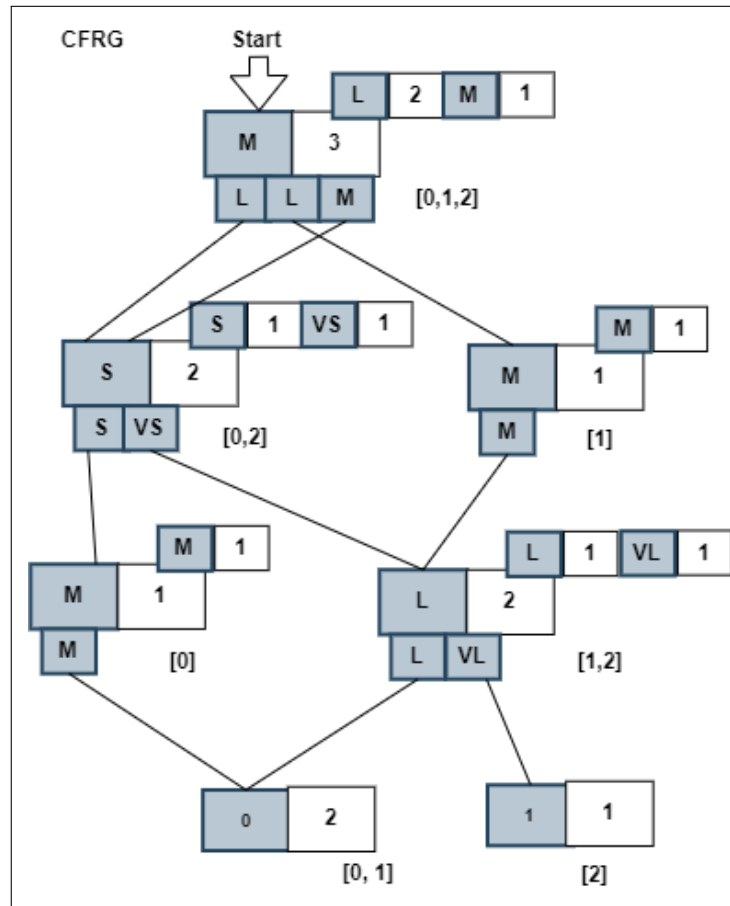
Một tập luật mờ phức sẽ tạo thành  $n$  CFRG theo định nghĩa như trên, trong đó  $n$  là số lượng biến ngôn ngữ của thuộc tính đầu tiên của luật. Để rõ ràng hơn, Hình 3.2 biểu diễn một cấu trúc CFRG với 3 luật mờ phức như sau:

$CFR_1 : If (Ax_1 \text{ is } M \text{ and } Px_1 \text{ is } L) \text{ and } (Ax_2 \text{ is } S \text{ and } Px_2 \text{ is } S) \text{ and } (Ax_3 \text{ is } M \text{ and}$

$Px_3$  is M ) then 0

$CFR_2$  : If ( $Ax_1$  is M and  $Px_1$  is L) and ( $Ax_2$  is M and  $Px_2$  is M) and ( $Ax_3$  is L and  $Px_3$  is L ) then 0

$CFR_3$  : If ( $Ax_1$  is M and  $Px_1$  is M) and ( $Ax_2$  is S and  $Px_2$  is VS) and ( $Ax_3$  is L and  $Px_3$  is VL ) then 1



**Hình 3.2:** Một cấu trúc CFRG

Ngoài hai định nghĩa về nút mờ phức và cấu trúc CFRG, một số định nghĩa để đo lường mức độ khác nhau giữa hai CFRG và đánh giá độ mạnh của CFRG cũng được giới thiệu.

**Định nghĩa 3.3.** Độ đo khác nhau giữa hai cấu trúc CFRG

Cho hai cấu trúc  $CFRG_1$  và  $CFRG_2$  cùng được trích rút ra từ một cấu trúc

CFRG cha nào đó. Sự khác nhau giữa  $CFRG_1$  và  $CFRG_2$  được tính toán như sau:

$$DI = \sum_i \frac{DI_i(CFRG_1, CFRG_2)}{h * (n(CFRG_1.CFN) + n(CFRG_2.CFN))_i}; \quad (3.1)$$

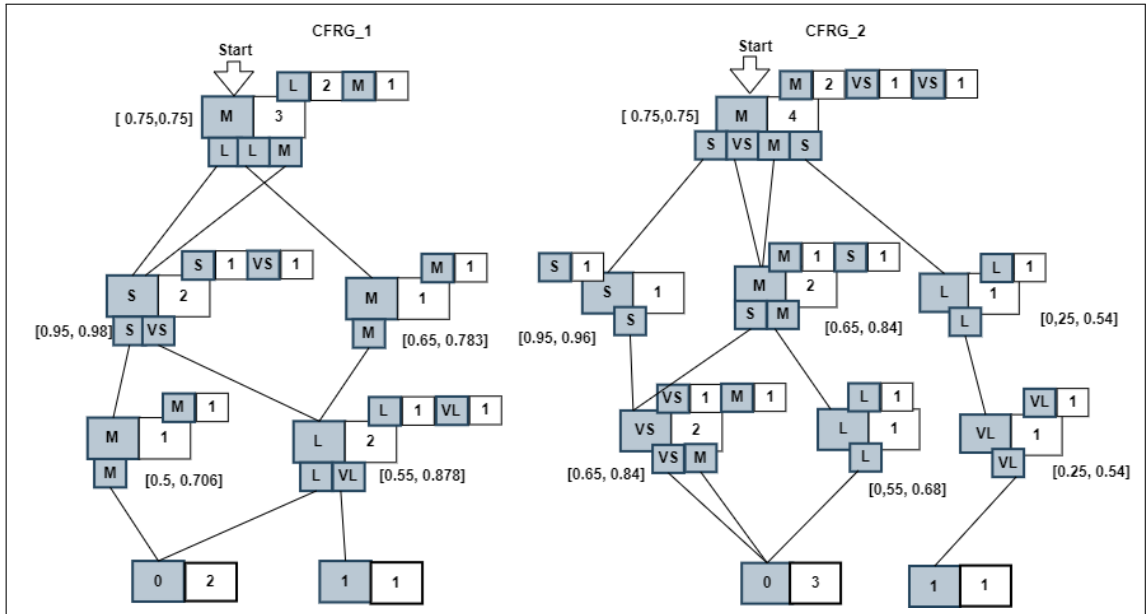
Với:

-  $DI_i(CFRG_1, CFRG_2) = \sum_j |CFRG_1.CFN_j.freq - CFRG_2.CFN_j.freq|_i$ , với  $j$  là biến ngôn ngữ.

-  $i \in [1, h)$ ,  $h$  mức lớn nhất trên cấu trúc CFRG;

-  $n(CFRG_1.CFN), n(CFRG_2.CFN)$  là tổng tần suất các nút trên cùng một mức của cấu trúc  $CFRG_1, CFRG_2$  tương ứng.

Ví dụ 3.1. Để rõ ràng hơn về độ đo sự khác nhau giữa hai cấu trúc CFRG, chúng ta cùng xem xét hai cấu trúc  $CFRG_1$  và  $CFRG_2$  trong Hình 3.3 dưới đây.



**Hình 3.3:** Sự khác nhau giữa cấu trúc CFRG và độ mạnh của cấu trúc CFRG.

- Ở mức thứ nhất, chúng ta có:  $DI_0 = \frac{|CFRG_1.M.3 - CFRG_2.M.4|}{CFRG_1.M.3 + CFRG_2.M.4} = \frac{1}{7}$

- Ở mức thứ hai:

$$DI_1 = \frac{|CFRG_1.S.2 - CFRG_2.S.1| + |CFRG_1.M.1 - CFRG_2.M.2| + |CFRG_1.L.0 - CFRG_2.L.1|}{CFRG_1.S.2 + CFRG_1.M.1 + CFRG_2.S.1 + CFRG_2.M.2 + CFRG_2.L.1} = \frac{3}{7}$$

- Ở mức thứ ba:  $DI_2 = \frac{5}{7}$

- Ở mức thứ ba:  $DI_3 = \frac{1}{7}$

Độ đo sự khác nhau giữa hay cấu trúc CFRG là:

$$DI = \frac{1}{4} * (DI_0 + DI_1 + DI_2) = \frac{1}{4} * \left(\frac{1}{7} + \frac{3}{7} + \frac{5}{7} + \frac{1}{7}\right) = \frac{10}{28}.$$

Để đo mức độ mạnh của cấu trúc CFRG, độ mạnh của nút trên cấu trúc CFRG được trình bày dưới đây.

**Định nghĩa 3.4.** Độ mạnh của nút mờ phức

Độ mạnh của nút  $CFN$  tại mức thứ  $k$  ký hiệu là  $(S_{CFN_k})$ , được cho bởi công thức:

$$S_{CFN_k} = \sqrt{(SA_{CFN_k} \cdot \cos(SP_{CFN_k}))^2 + (SA_{CFN_k} \cdot \sin(SP_{CFN_k}))^2} \quad (3.2)$$

Với:

$$SA_{CFN_k} = \mu(CFN_k.aValue) + \left(1 - \mu(CFN_k.aValue)^{\frac{CFN_k.afreq}{n(CFN_k)}}\right) \quad (3.3)$$

$$SP_{CFN_k} = \max_{i=1,n} \left( \omega(CFN_k.pValue_i) + \left(1 - \omega(CFN_k.pValue_i)^{\frac{CFN_k.pValue_i.pfreq}{CFN_k.afreq}}\right) \right) \quad (3.4)$$

Trong đó:

- $SA_{CFN_k}, SP_{CFN_k}$  là độ mạnh phần biên độ, độ mạnh phần pha của nút.
- $\mu(CFN_k.aValue)$  phần biên độ,  $\omega(CFN_k.pValue_i)$  phần pha của mỗi đầu vào của luật.
- $n$  số lượng phần pha của nút.
- $n(CFN_k)$  là tổng tần suất của các nút ở mức thứ  $k$ .

Ví dụ 3.2. Cho một nút  $CFN$  có giá trị mờ phần biên độ là 0.65, tần suất của nút là 4; danh sách giá trị mờ phần pha và tần suất tương ứng là: (0.02, 1), (0.05, 3); số lượng nút tại cấp thứ  $k$  là 10. Độ mạnh của nút được tính toán theo Định nghĩa 3.4 là:

$$SA(CFN_k) = 0.65 + \left(1 - 0.65^{\frac{4}{10}}\right)$$

$$SP(CFN_k) = 0.05 + \left(1 - 0.05^{\frac{3}{4}}\right)$$



$$S(CFN_k) \approx 0.808$$

**Định nghĩa 3.5.** Độ mạnh của cấu trúc CFRG

Cho một CFRG, độ mạnh của một CFRG được tính như sau:

$$S_{CFRG} = \prod_{k=1, \bar{h}-1} (\max_{1,n}(S_{CFN_k})) \quad (3.5)$$

Trong đó:

- $S_{CFN_k}$  là độ mạnh của nút ở cấp thứ  $k$ ;
- $n$  số lượng nút ở cấp độ thứ  $k$ ;
- $h$  mức lớn nhất của CFRG.

Trong ví dụ về hai CFRG như trong Hình 3.3, mỗi nút có giá trị độ mạnh riêng là sự kết hợp giữa độ mạnh biên độ với giá trị tần suất theo định nghĩa 3.4. Ở mỗi mức (level) giá trị độ mạnh lớn nhất của nút được xác định. Ở mức thứ nhất, độ mạnh tại mức này trên CFRG\_1 là lớn hơn độ mạnh tương ứng trên CFRG\_2. Và mức thứ hai, CFRG\_2 mạnh hơn. Kết hợp độ mạnh tất cả các mức trên cấu trúc CFRG theo định nghĩa 3.5 ta có, CFRG\_1 sẽ có độ mạnh là 0,63 và CFRG\_2 sẽ có độ mạnh là 0,60.

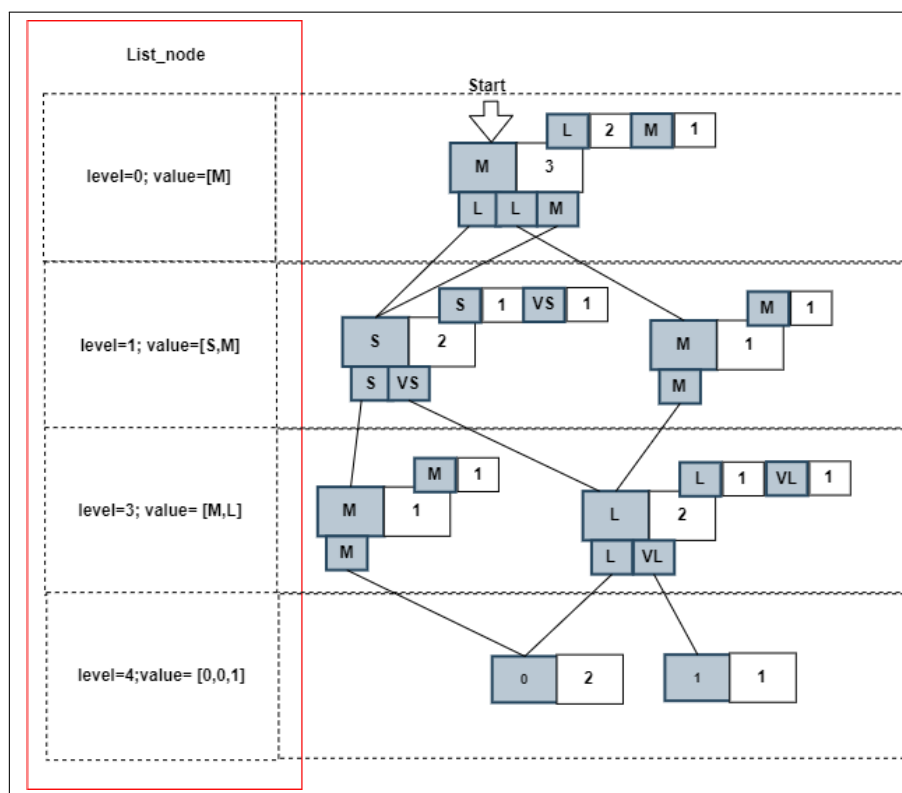
Cấu trúc CFRG được đề xuất nhằm biểu diễn tập luật mờ phức, đồng thời phải đảm bảo các thao tác ứng với tập luật mờ phức. Dưới đây sẽ trình bày các thao tác (thuật toán) cơ bản trên cấu trúc CFRG, nhằm đáp ứng cả các thao tác trên tập luật mờ phức.

### 3.3. Các thuật toán trên cấu trúc CFRG

Với mục đích ứng dụng cấu trúc CFRG vào việc cải tiến hiệu suất quá trình suy diễn mờ phức, thích nghi luật của mô hình CFTL đề xuất tại Chương 2. Một số thuật toán dưới đây cho phép thao tác trên cấu trúc CFRG được giới thiệu như sau:

- Thuật toán thêm một luật vào CFRG.
- Thuật toán duyệt luật mờ phức trên CFRG.
- Thuật toán tìm kiếm một luật trên CFRG.

- Thuật toán xóa một luật khỏi CFRG.
- Thuật toán suy diễn mờ phức trên CFRG.
- Thuật toán chỉnh sửa luật trên CFRG.
- Thuật toán tách CFRG con từ CFRG cha.
- Thuật toán trộn hai cấu trúc CFRG.



**Hình 3.4:** Một danh sách *List\_node* của cấu trúc CFRG.

Để biểu diễn, truy xuất dễ dàng và thực hiện các thao tác trên CFRG, một vài kỹ thuật tổ chức trên cấu trúc CFRG được bổ sung và mô tả như sau:

1. Cũng như cấu trúc cây thông thường, mỗi một cấu trúc CFRG sẽ được quản lý bởi nút gọi là *Start*.
2. Cấu trúc CFRG sẽ quản lý các luật thông qua một chỉ mục/chỉ số (*index*), ban đầu

chỉ mục này được khởi tạo là 0. Khi kết thúc quá trình thêm một luật (thêm nút lá) thì chỉ số này sẽ được tăng lên 1 đơn vị.

3. Để thực hiện truy xuất các nút và thao tác trên CFRG một cách hiệu quả hơn, một danh sách *List\_node* (Hình 3.4) được tổ chức nhằm lưu trữ các nút trên cùng một mức. Tức là mỗi CFRG sẽ tồn tại một danh sách các nút được quản lý theo mức và mỗi nút sẽ có thêm thuộc tính mức để cho biết nút đó thuộc mức nào. Các *list\_node* được xây dựng sử dụng kiểu *dictionary*.
4. Để thêm luật vào cấu trúc CFRG, cấu trúc CFRG có sử dụng thêm con trỏ *pAdd* để quản lý đối tượng nút vừa được thêm vào. Khi khởi tạo cấu trúc con trỏ *pAdd* có giá trị None, mỗi khi kết thúc một luật được thêm con trỏ *pAdd* được gán lại nút *Start*.

### 3.3.1. Thuật toán thêm một luật vào CFRG

Như đã định nghĩa (Định nghĩa 3.2), một cấu trúc CFRG là một cấu trúc có thứ tự với nút gốc biểu diễn thuộc tính đầu tiên của luật, vì vậy những luật có biến ngôn ngữ của thuộc tính thứ nhất giống nhau sẽ được biểu diễn trên cùng một CFRG. Mỗi cặp gồm biến ngôn ngữ phần biên độ và biến ngôn ngữ phần pha, biểu diễn điều kiện của một thuộc tính, trên mỗi luật, sẽ tạo thành một nút trên CFRG. Những thuộc tính trên các luật khác nhau mà có thứ tự giống nhau và chung biến ngôn ngữ phần thực, sẽ chia sẻ cùng một nút.

Mỗi CFR sẽ tạo ra một danh sách các nút CFN, trong đó số lượng các nút đúng bằng số lượng thuộc tính của CFR. Một CFR được biểu diễn (thêm vào) trên CFRG bằng cách thêm liên tiếp các nút CFN tương ứng với các thuộc tính trên luật vào CFRG. Quá trình thêm một luật sẽ kết thúc khi thêm vào nút lá, là nhãn đầu ra của luật.

Trong phần này, thuật toán thêm một nút CFN vào CFRG được trình bày trên cơ sở làm nền tảng cho thuật toán thêm một CFR vào CFRG. Thuật toán thêm một nút

---

**Thuật toán 3.1** Thêm một nút CFN vào một CFRG
 

---

**Input:** CFN *node*, CFRG *T***Output:** *T* chứa *node*.

```

1: if T.Start is None then
2:   T.Start  $\leftarrow$  node;
3:   T.pAdd  $\leftarrow$  node;
4:   flag  $\leftarrow$  true;                                {Đánh dấu vừa thêm nút gốc}
5: else
6:   if T.pAdd is T.Start and not flag then
7:     T.pAdd.freq = T.pAdd.freq + 1 ;
8:     phase  $\leftarrow$  node.pValue1;
9:     Thêm phase vào T.pAdd;
10:    flag  $\leftarrow$  true;
11:  else
12:    if node.pValue1 rỗng then
13:      Thêm node là con của pAdd;
14:      T.pAdd  $\leftarrow$  T.Start;
15:      flag  $\leftarrow$  false;
16:    else
17:      child  $\leftarrow$  T.pAdd.find_node(node.aValue; {Tìm con của pAdd có giá trị
        bằng giá trị nút thêm vào}
18:      if child is None then
19:        node is child of T.pAdd;
20:        T.pAdd  $\leftarrow$  node;
21:      else
22:        Thêm child là con của T.pAdd;
23:        Thêm node.pValue1 là pha của child ;
24:        child.freq=child.freq+1;
25:        T.pAdd  $\leftarrow$  child;
26:      end if
27:    end if
28:  end if
29: end if

```

---

CFN vào CFRG được trình bày chi tiết dưới đây (Thuật toán 3.1).

Thêm luật mờ phức vào CFRG là quá trình thêm liên tiếp các nút CFN vào CFRG cho đến khi gặp nút lá (nhãn của luật) thì dừng. Mỗi luật được thêm vào sẽ có một chỉ số (thứ tự) sử dụng để theo dõi các nút của cùng một luật. Khi quá trình thêm một luật kết thúc thì chỉ số (thứ tự) index của CFRG sẽ tăng lên. Chi tiết các bước thêm luật

vào CFRG được mô tả trong Thuật toán 3.2.

---

**Thuật toán 3.2** Thêm một luật mờ phức CFR vào CFRG

---

**Input:** CFRG  $T$ , CFR  $R(rule, ruleIm)$

**Output:**  $T$  chứa  $R$ .

```

1:  $index \leftarrow T.get\_index();$                                 {Lấy ra chỉ số của luật}
2: for all  $pair(r_1, Im_1) \in R$  do
3:    $node \leftarrow CFN(r_1);$ 
4:   if  $ruleIm$  not is None then
5:      $p \leftarrow phase(Im_1);$ 
6:   else
7:      $p \leftarrow phase(None);$ 
8:   end if
9:   Thêm  $p$  là pha của  $node$  theo  $index$ ;
10:  Thêm  $node$  vào  $T$ ;                                         {Thuật toán 3.1}
11: end for
12:  $T.index = T.index + 1;$ 

```

---

### 3.3.2. Thuật toán duyệt luật mờ phức trên CFRG

Một tập các luật CFR có thể được biểu diễn trên một hay nhiều CFRG, việc duyệt các cấu trúc CFRG phải trả về tập luật mờ phức ban đầu. Hai phương pháp duyệt luật mờ phức trên CFRG được thực hiện theo chiều sâu và theo chiều rộng được giới thiệu dưới đây.

#### 3.3.2.1. Duyệt luật mờ phức theo chiều sâu

Để thực hiện duyệt toàn bộ CFRG luật theo chiều sâu, một thuật toán để duyệt một luật theo một thứ tự (chỉ số) theo chiều sâu được giới thiệu. Do cấu trúc lưu trữ các nút con theo chỉ số của luật, nên tại mỗi nút khi duyệt tới sau khi lấy ra giá trị phần biên độ và giá trị phần pha của nút để tạo thành các cặp điều kiện trong luật, nút con có chỉ số tương ứng với chỉ số của luật sẽ được xác định để duyệt tiếp theo. Quá trình duyệt một luật theo chiều sâu kết thúc khi nút lá được duyệt đến.

Các bước duyệt một luật với chỉ số  $index$  như sau:

Bước 1. Bắt đầu từ nút gốc;

Bước 2. Thăm nút

- Nếu là nút lá, lấy giá trị biên độ và kết thúc.

- Ngược lại, lấy ra giá trị biên độ của nút, lấy giá trị pha tương ứng chỉ số *index* đã chỉ định.

Bước 3. Xác định nút con cần duyệt đến bằng chỉ số *index* và phần pha của nút đó theo *index*

Bước 4. Duyệt nút con; Quay lại bước 2;

Thuật toán duyệt một luật theo chiều sâu được trình bày chi tiết tại Thuật toán 3.3. Quá trình duyệt một luật kết thúc ta sẽ thu được một luật mờ phức bao gồm phần biên độ và phần pha là các cặp điều kiện của luật và lá là nhãn đầu ra.

---

**Thuật toán 3.3** Duyệt một luật có thứ tự *index* theo chiều sâu

---

**Input:** CFRG  $T$ , thứ tự luật *index*;

**Output:** CFR  $R$ .

```

1:  $rule\_re \leftarrow None$ ;
2:  $rule\_im \leftarrow None$ ;
3:  $pTravel \leftarrow T.Start$ ;
4: while  $pTravel$  not is None do
5:   Thêm  $pTravel.aValue$  vào  $rule\_re$ ;
6:   if  $pTravel.pValue$  is None then
7:     Lấy  $phase$  từ  $pTravel$  theo  $index$ ;
8:     Thêm  $phase$  vào  $rule\_im$ ;
9:   end if
10:  Lấy  $child$  từ  $pTravel$  theo  $index$ ;
11:   $pTravel \leftarrow child$ ;
12: end while
13: return  $rule\_re, rule\_im$ ;

```

---

Dựa theo thuật toán duyệt một luật có chỉ số *index* (Thuật toán 3.3), thuật toán duyệt toàn bộ CFRG luật được trình bày dưới đây. Ý tưởng của thuật toán là:

Bước 1. Bắt đầu từ nút Start của CFRG.

Bước 2. Lấy danh sách chỉ số các luật tại nút gốc.

Bước 3. Duyệt danh sách chỉ số luật đã lấy từ bước 2.

- Duyệt luật với chỉ số *index* từ danh sách chỉ số

Chi tiết các bước của thuật toán duyệt toàn bộ luật trên CFRG theo chiều sâu được trình bày chi tiết trong thuật toán 3.4.

---

**Thuật toán 3.4** Duyệt CFRG theo chiều sâu

---

**Input:** CFRG  $T$

**Output:** CFRs  $R_s$ .

```

1:  $rules\_re \leftarrow None$ ;
2:  $rules\_im \leftarrow None$ ;
3: for all  $index \in T.Start$  do
4:    $rule \leftarrow None$ ;
5:    $ruleim \leftarrow None$ ;
6:    $rule, ruleim =$  Duyệt luật có thứ tự  $index$ ;           {Thuật toán 3.3}
7:   Thêm  $rule$  vào  $rules\_re$ ;
8:   Thêm  $ruleim$  vào  $rules\_im$ ;
9: end for
10: return  $rules\_re, rules\_im$ ;
```

---

Các bước duyệt luật theo chiều sâu được cụ thể hóa trong phụ lục A

### 3.3.2.2. Duyệt luật mờ phức theo chiều rộng

Như đã đề cập ở trên, để thuận tiện cho truy cập các nút trên CFRG, một danh sách *list\_node* đã được thêm vào khi tạo CFRG (Hình 3.4). Tất cả các nút trên cùng một mức của CFRG được lưu trữ như một phần tử của từ điển *list\_node*. Trong đó, khóa là mức của CFRG và mỗi phần tử của từ điển là một danh sách các nút ở mức đó (Hình 3.4).

Mỗi luật được thêm vào CFRG sẽ có một chỉ số (như đã trình bày ở trên), và các nút được thêm vào sẽ được xác định theo chỉ số của luật và phần pha. Kết quả là ở mọi mức của CFRG ta luôn có một danh sách có thứ tự các luật theo phần pha. Còn phần biên độ được truy cập theo số lượng tần suất tương ứng của nó. Với cách này, tần suất truy cập nút trên CFRG ít hơn nhiều so với cách thức duyệt luật theo chiều sâu. Chính vì thế thời gian duyệt CFRG luật theo chiều rộng nhanh hơn so với duyệt CFRG luật

theo chiều sâu.

Các bước duyệt luật theo chiều rộng như sau:

Bước 1.  $level=0$ ;

Bước 2. Duyệt các nút ở mức  $level$ :

- Lấy phần biên độ theo tần suất của nút;
- Lấy phần pha theo chỉ số;

Bước 3. Tăng  $level=level+1$ ;

Bước 4. Nếu  $level=h$  thì kết thúc ngược lại sang Bước 2;

Thuật toán duyệt luật theo chiều rộng được trình bày chi tiết trong thuật toán 3.5 dưới đây.

---

### Thuật toán 3.5 Duyệt luật mờ phức CFR trên CFRG theo chiều rộng

---

**Input:** CFRG  $T$

**Output:** CFRs  $R_s$ .

```

1:  $list\_node \leftarrow T.list\_node$ ;
2:  $rules\_re \leftarrow None$ ;
3:  $rules\_im \leftarrow None$ ;
4: for all  $key, list\_level\_node \in list\_node$  do
5:    $rule\_re \leftarrow None$ ;
6:    $rule\_im \leftarrow None$ ;
7:   for all  $node \in list\_level\_node$  do
8:     for all  $index, phase \in node$  do
9:        $rule[index] \leftarrow node.aValue$ ;
10:       $rule\_im[index] \leftarrow phase$ ;
11:     end for
12:   end for
13:   Thêm  $rule$  vào  $rules\_re$ ;
14:   Thêm  $rule\_im$  vào  $rules\_im$ ;
15: end for
16: return  $rules\_re, rules\_im$ ;
```

---

Duyệt CFRG theo chiều rộng, cũng được thực hiện từ nút gốc xuống nút lá. Tuy nhiên, trong quá trình đi xuống, các nút theo từng mức của CFRG sẽ được duyệt ra trước hết, rồi mới chuyển sang mức kế tiếp.



Để duyệt toàn bộ luật trên CFRG thì phương pháp duyệt theo chiều rộng nhanh hơn so với phương pháp duyệt theo chiều sâu. Tuy nhiên, trong nhiều trường hợp, để truy cập đến một luật với chỉ số index ta có thể sử dụng phương pháp duyệt theo chiều sâu.

### **3.3.3. Thuật toán tìm kiếm một luật trên CFRG**

Ý tưởng của việc tìm kiếm một luật trên CFRG là tương tự như thêm một luật vào CFRG, nhưng quá trình thêm là thêm thử nghiệm (giả thêm). Nếu mọi nút của luật được thêm trên CFRG đều không làm phát sinh thêm con mà chỉ tăng tần suất của nút con (trùng nút), điều này chứng tỏ luật vừa thêm vào đã có trên CFRG đó.

Thuật toán tìm kiếm một luật trên CFRG (Thuật toán 3.7) với đầu vào là một luật CFR cần tìm, đầu ra là True nếu tìm thấy và False nếu không thấy. Thuật toán được xây dựng dựa thuật toán thêm một luật vào CFRG có bổ sung vào thao tác đếm số lần tăng tần suất nút con (gọi là trùng nút trên CFRG - Thuật toán 3.6), nếu số lần tăng tần suất bằng đúng số lượng thuộc tính của luật thì luật đó là luật đã tồn tại trên CFRG.

### **3.3.4. Thuật toán xóa một luật khỏi CFRG**

Xóa một luật khỏi CFRG có thể có nhiều tiêu chí xóa khác nhau. Tuy nhiên, trong khuôn khổ nghiên cứu này, xóa một luật trên CFRG được thực hiện dựa trên chỉ số của luật cần xóa.

Việc xóa luật theo chỉ số có thể hoàn toàn được thực hiện giống thao tác duyệt một luật với chỉ số theo chiều sâu (Thuật toán 3.3). Nhưng thay vì chỉ lấy ra giá trị (biên độ, pha) của nút để tạo thành luật, sẽ đồng thời giảm bớt tần suất phân biên độ, xóa các phần pha tương ứng hoặc xóa nút khỏi CFRG mỗi khi duyệt qua một nút.

---

**Thuật toán 3.6** Xác định nút trùng trên CFRG
 

---

**Input:** CFN *node*, CFRG *T***Output:** True-trùng nút, False-không trùng nút.

```

1: flag_dup ← False;
2: if T.Start is None then
3:   T.Start ← node;
4:   T.pAdd ← node;
5:   flag ← True;                                     {Đánh dấu vừa thêm nút gốc}
6: else
7:   if T.pAdd is T.Start and not flag then
8:     T.pAdd.afreq = T.pAdd.afreq + 1;
9:     phase ← node.pValue;
10:    Thêm phase vào pAdd;
11:    flag ← True;
12:    flag_dup ← True;
13:  else
14:    if node.pvalue is None then
15:      Thêm node là con của T.pAdd;
16:      T.pAdd ← T.Start;
17:      flag ← False;
18:    else
19:      T.pAdd.find_node(node.aValue);   {Tìm con của pAdd có giá trị bằng
                                         giá trị nút thêm vào}
20:    if child is None then
21:      Thêm node là con của T.pAdd;
22:      T.pAdd ← node;
23:    else
24:      Thêm child là con của T.pAdd ;
25:      Thêm node.pValue là pha của child ;
26:      child.afreq = child.afreq + 1;
27:      T.pAdd ← child;
28:      flag_dup ← True;
29:    end if
30:  end if
31: end if
32: end if
33: return flag_dup;

```

---

**3.3.5. Thuật toán suy diễn mờ phức trên CFRG**

Mục tiêu của việc xây dựng cấu trúc CFRG hướng tới cải thiện hiệu suất của quá trình suy diễn mờ phức và thích nghi luật theo hệ suy diễn mờ phức trong mô hình

---

**Thuật toán 3.7** Tìm kiếm một luật mờ phức CFR trên CFRG
 

---

**Input:** CFRG  $T$ , CFR  $R(rule, ruleIm)$ 
**Output:** True- nếu tồn tại, False- nếu không tồn tại

```

1:  $T1 \leftarrow T.copy()$ ;
2:  $count \leftarrow count + 1$ ;
3:  $Flag \leftarrow False$ ;
4:  $index \leftarrow T1.get\_index()$ ;
5: for all  $pair(r, Im) \in R$  do
6:    $node \leftarrow CFN(r)$ ;
7:   if  $ruleIm$  not is None then
8:      $p \leftarrow phase(Im)$ ;
9:   else
10:     $p \leftarrow phase(None)$ ;
11:    Thêm  $p$  là pha của  $node$  theo  $index$ ;
12:   end if
13:   if Xác định nút trùng trên CFRG( $node, T1$ ) then
14:      $count \leftarrow count + 1$ ;
15:   end if
16: end for
17: if  $count == len(rule)$  then
18:    $Flag \leftarrow True$ ;
19: end if
20: return  $Flag$ ;

```

---

học chuyển giao mờ phức đã đề xuất trước đó (chương 2). Do đó, thuật toán suy diễn mờ phức trên CFRG được giới thiệu trong phần này.

Suy diễn mờ phức trên CFRG được thực hiện tương tự quá trình duyệt CFRG theo chiều sâu hoặc duyệt CFRG theo chiều rộng, tuy nhiên thay vì lấy giá trị phần biên độ và phân pha ở mỗi nút để lấy ra luật thì một ánh xạ đến dữ liệu mờ hóa đầu vào được thực hiện nhằm thu được độ cháy phần biên độ và phân pha tương ứng. Quá trình tổng hợp giá trị đầu ra được thực hiện theo cột, vì mỗi cột biểu diễn giá trị cháy trên mỗi luật. Để cải thiện thời gian suy diễn mờ phức, phần này đề cập đến phương pháp suy diễn mờ phức trên CFRG theo phương pháp duyệt CFRG theo chiều rộng.

Thuật toán duyệt luật theo chiều rộng được trình bày chi tiết trong thuật toán 3.8. Ví dụ về suy diễn mờ phức trên cấu trúc CFRG được trình bày trong phụ lục A

---

**Thuật toán 3.8** Suy diễn mờ phức trên CFRG theo chiều rộng
 

---

**Input:** CFRG  $T$ , dòng dữ liệu  $t$

**Output:** Nhãn đầu ra  $label$ .

```

1:  $list\_node \leftarrow T.list\_node$ ;
2:  $rules\_re \leftarrow None$ ;
3:  $rules\_im \leftarrow None$ ;
4:  $fuzzy\_re, fuzzy\_im =$ Mờ phức hóa ( $t$ )  {Thực hiện mờ phức hóa dữ dòng dữ liệu
    $t$ }
5: for all  $key, list\_level\_node \in list\_node$  do
6:    $rule\_re \leftarrow None$ ;
7:    $rule\_im \leftarrow None$ ;
8:   for all  $node \in list\_level\_node$  do
9:     for all  $index, phase \in node$  do
10:       $rule[index] \leftarrow fuzzy\_re[index][node.aValue]$ ;
11:      if phase not is None then
12:         $rule\_im[index] \leftarrow fuzzy\_im[index][phase]$ ;
13:      end if
14:    end for
15:  end for
16:  Thêm  $rule\_re$  vào  $rules\_re$ ;
17:  if  $rule\_im$  not is None then
18:    Thêm  $rule\_im$  vào  $rules\_im$ ;
19:  end if
20: end for
21: Transpose  $rules\_re, rules\_im$ ;  {Chuyển đổi danh sách giá trị ánh xạ luật thành
   hàng ngang tương ứng}
22: Tính toán tổng hợp độ mạnh của luật;
23: Tổng hợp kết quả, giải mờ tạo nhãn đầu ra  $label$ ;  {theo mô hình M-CFIS}
24: return  $label$ ;

```

---

### 3.3.6. Thuật toán sửa một luật trên CFRG

Việc thích nghi luật, đòi hỏi phải chỉnh sửa luật nhằm đáp ứng ngữ cảnh mới. Chỉnh sửa luật là chỉnh sửa các giá trị thuộc tính của luật mà mỗi thuộc tính đó được biểu diễn thành các nút trên CFRG. Do đó, quá trình chỉnh sửa luật mờ phức có cốt lõi là chỉnh sửa các một CFN. Sửa một nút CFN bao gồm sửa giá trị phần biên độ và giá trị phần pha của một nút CFN. Tuy nhiên một nút CFN cần chỉnh sửa có thể là gộp của nhiều thuộc tính trên các luật khác nhau. Do đó, việc chỉnh sửa một nút, chỉ được

phép chỉnh sửa một cặp giá trị (biên độ, pha) trong nút đó thay vì chỉnh sửa tất cả các cặp giá trị của nút đó.

---

**Thuật toán 3.9** Sửa nút thuộc luật thứ *index* và mức thứ *level*

---

**Input:** CFRG *T*, thứ tự luật: *index*, mức: *level*, giá trị biên độ mới: *aValue\_new*, giá trị pha mới: *pValue\_new*

**Output:** *T* với luật thứ *index* chứa nút ở mức *level* mang giá trị biên độ là *aValue\_new* và giá trị pha *pValue\_new*.

```

1: list_node = T.list_node[level];           {Lấy danh sách các nút tại mức level}
2: node = list_node[index];                 {Lấy nút tại chỉ số index}
3: if node.freq==1 then
4:   node.aValue ← aValue_new;
5:   node.pValue ← pValue_new;
6: else
7:   sibling ← list_node.find_node(aValue_new); {Tìm kiếm nút anh em liền kề
   có giá trị bằng với giá trị mới cần sửa}
8:   if sibling is None then
9:     new_node = Node(aValue_new, pValue_new);
10:    new_node.level ← node.level;
11:    parent ← node.parent theo index;
12:    Xóa node.parent theo index;
13:    Thêm con của parent là new_node theo index;
14:    child ← node.child theo index;
15:    Xóa node.child theo index;
16:    Thêm new_node là con của child theo index;
17:  else
18:    parent ← node.parent theo index;
19:    Xóa node.parent;
20:    Thêm sibling là con của parent theo index;
21:    child ← node.child theo index;
22:    Xóa node.child theo index;
23:    Thêm sibling là con của child theo index;
24:    sibling.aFreq = sibling.aFreq + 1;
25:  end if
26:  node.aFreq = node.aFreq - 1;
27: end if

```

---

Như vậy, chỉnh sửa một nút có thể chia thành nhiều trường hợp:

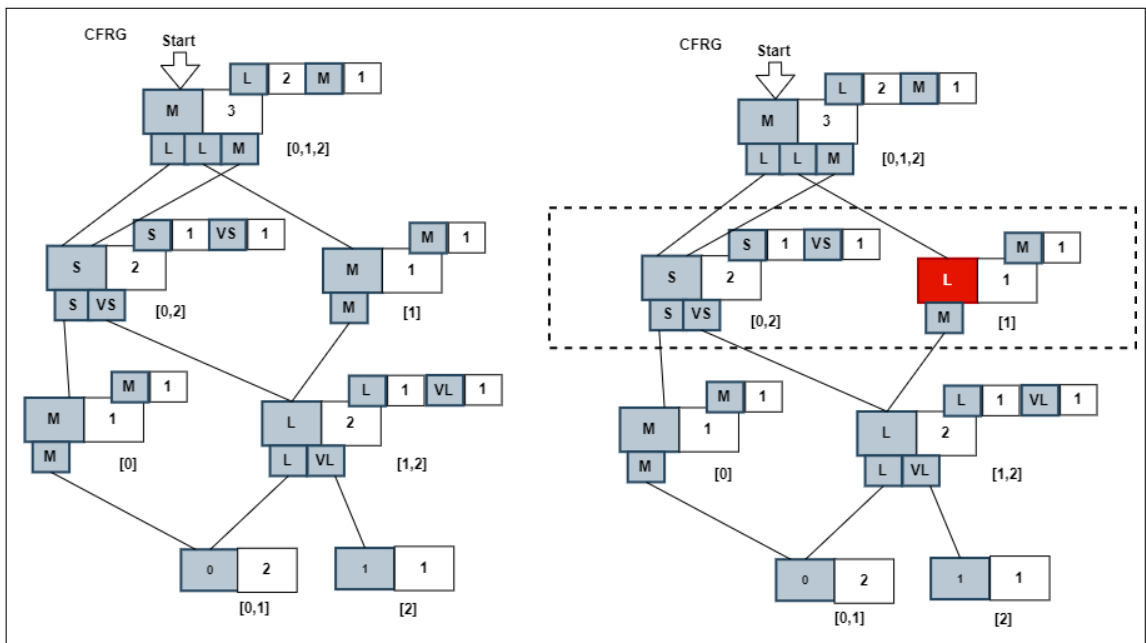
TH1: Nếu nút cần sửa có tần suất biên độ là 1, thì sửa trực tiếp tại nút (Hình 3.5).

TH2: Nếu nút cần sửa có tần suất biên độ >1:

- Nếu giá trị nút mới trùng với anh em liền kề quá trình nhập nút sẽ diễn ra (Hình 3.6).

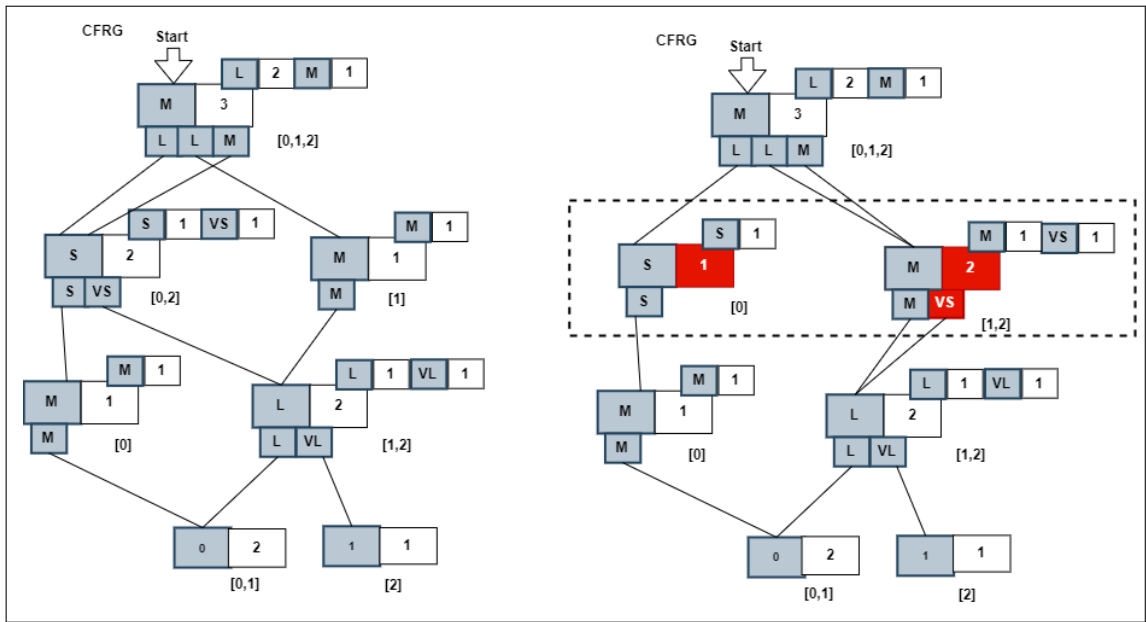
- Nếu giá trị nút mới không trùng anh em liền kề quá trình tách nút sẽ diễn ra (Hình 3.7).

Chi tiết các bước sửa nút được trình bày trong thuật toán 3.9

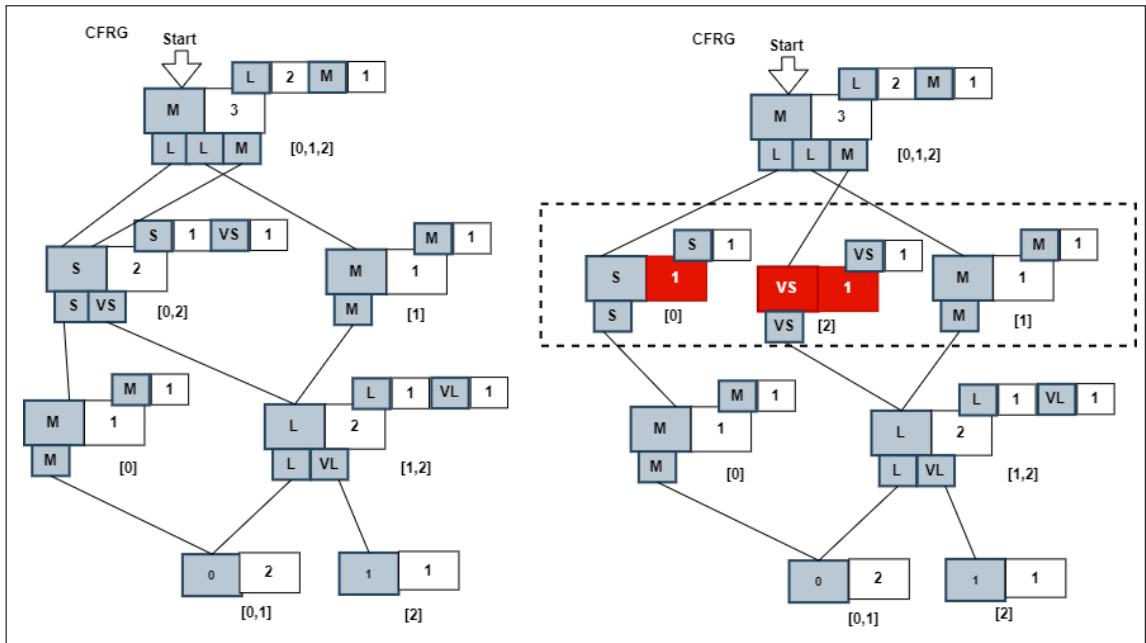


**Hình 3.5:** Trường hợp sửa nút thứ nhất

Hình 3.5 biểu diễn trường hợp khi cần sửa luật có chỉ số index là 1, với yêu cầu sửa thuộc tính thứ 2 với giá trị biên độ là M thành biên độ có giá trị L. Do thuộc tính thứ 2 của luật nằm trên nút có giá trị tần suất là 1, nên việc sửa nút này được quy vào TH1 như đã nêu ở trên. Và việc chỉnh sửa diễn ra đơn giản bằng cách thay thế giá trị biên độ cũ (M) thành giá trị biên độ mới (L) trên nút.



Hình 3.6: Trường hợp sửa nút thứ 2 - nhập nút



Hình 3.7: Trường hợp sửa nút thứ hai - tách nút

Hình 3.6 biểu diễn trường hợp khi cần sửa luật có chỉ số index là 2, với yêu cầu sửa thuộc tính thứ 2 với giá trị biên độ là S thành biên độ có giá trị M. Do thuộc tính thứ 2

của luật nằm trên nút có giá trị tần suất là 2, và giá trị mới là M trùng với anh em liền kề cùng mức, nên việc sửa nút này được quy vào TH2.1 như đã nêu ở trên. Và việc chỉnh sửa diễn ra bằng cách nhập nút bao gồm tăng tần suất, cập nhật lại các mối liên kết cha, con.

Hình 3.7 biểu diễn trường hợp khi cần sửa luật có chỉ số index là 2, với yêu cầu sửa thuộc tính thứ 2 với giá trị biên độ là S thành biên độ có giá trị VS. Do thuộc tính thứ 2 của luật nằm trên nút có giá trị tần suất là 2, và giá trị mới là M không trùng với anh em liền kề cùng mức, nên việc sửa nút này được quy vào TH2.2 như đã nêu ở trên. Và việc chỉnh sửa diễn ra bằng cách tách nút bao gồm tăng tần suất, cập nhật lại các mối liên kết cha, con.

---

### Thuật toán 3.10 Tách CFRG con từ CFRG cha

---

**Input:** CFRG  $T$ ,  $n$ : Số lượng luật cần tách ra từ CFRG

**Output:**  $CFRG\_child$  gồm  $n$  luật

```

1:  $CFRG\_child.Start \leftarrow None$ ;
2:  $list\_node \leftarrow T.list\_node$ ;
3:  $list\_number\_get \leftarrow None$ ;           {Danh sách số nút cần tách, có chỉ số là node}
4:  $list\_index\_get \leftarrow None$ ;         {Danh sách các chỉ số luật cần tách ra}
5: Xác định  $level\_max\_node$ ;               {là mức có số lượng nút nhiều nhất}
6: Xác định  $list\_number\_get$ ;               {Thuật toán 3.10 }
7: for all  $node \in list\_node[level\_max\_node]$  do
8:    $k \leftarrow 0$ ;
9:   for all  $index, phase \in node$  do
10:    if  $k == list\_number\_get[node]$  then
11:      break;
12:    end if
13:  end for
14:  Thêm  $index$  vào  $list\_index\_get$ ;
15:   $k \leftarrow k + 1$ ;
16: end for
17: for all  $index \in list\_index\_get$  do
18:   $rule\_re, rule\_im = T.delete\_CFR(index)$ ;   {Thuật toán xóa luật trên CFRG
theo chỉ số}
19:  Thêm  $rule\_re, rule\_im$  vào  $CFRG\_child$ ;     {Thuật toán 3.2}
20: end for
21: return  $CFRG\_child$ ;

```

---



### 3.3.7. Thuật toán tách CFRG con từ CFRG cha

Ý tưởng chính của thuật toán tách CFRG con từ CFRG cha là tìm mức có nhiều nút nhất trên CFRG cha và tính số nút con sẽ được tách ra từ mỗi nút ở mức này. Từ số con này ta sẽ lấy được các pha tương ứng trên mỗi nút và cũng xác định được danh sách chỉ số của luật cần lấy ra. Dựa trên danh sách chỉ số của luật cần lấy, thực hiện xóa các luật trong danh sách này từ CFRG cha và tạo CFRG mới từ danh sách luật xóa ta được CFRG con.

Input của thuật toán gồm:  $n$ : số lượng luật tách ra,  $T$ : cấu trúc CFRG cha.

Output: Cấu trúc CFRG con: *CFRG\_child*

Các bước của thuật toán như sau:

Bước 1. Xác định mức có số lượng nút nhiều nhất trên  $T$ , *level\_max\_node*

Bước 2. Xác định danh sách số lượng nút *list\_number\_get* cần tách ra trên các nút con tại mức đã xác định trong bước 1 (Thuật toán 3.11)

Bước 3. Duyệt các nút tại mức *level\_max\_node*, tại mỗi nút lấy ra chỉ số *index* của nút cần tách (theo thứ tự ưu tiên trái) với số lượng được xác định tại bước 2, đưa vào danh sách *list\_index\_get*

Bước 4. Duyệt các chỉ số *index* trong danh sách *list\_index\_get*

- Xóa luật với chỉ số *index* trên CFRG cha

-Thêm luật vừa xóa vào *CFRG\_child*

Chi tiết các bước của thuật toán được trình bày trong Thuật toán 3.10.

Quá trình xác định danh sách số lượng nút *list\_number\_get* cần tách ra trên mỗi nút thuộc mức *level\_max\_node* được xác định theo Thuật toán 3.11

### 3.3.8. Thuật toán trộn hai CFRG

Việc trộn (gộp) hai cấu trúc CFRG được tiến hành theo cách duyệt một cấu trúc CFRG và thêm vào cấu trúc CFRG còn lại. Tuy nhiên, các luật của CFRG thứ hai mà

---

**Thuật toán 3.11** Xác định danh sách số lượng nút cần tách *list\_number\_get*


---

**Input:** CFRG  $T$ ,  $n$ : Số lượng luật cần tách ra từ CFRG, *level\_max\_node*: mức

**Output:** *list\_number\_get*

```

1: list_number_get  $\leftarrow$  None;
2: list_node  $\leftarrow$   $T.list\_node$ ;
3: for all node  $\in$  list_node[level_max_node] do
4:   list_number_get[node] =  $int(\frac{n}{n(T)} * node.afreq)$ ;   {  $n(T)$  là số lượng luật trên
   CFRG }
5: end for
6: while  $sum(list\_number\_get) < n$  do
7:   for all node, number  $\in$  list_number_get do
8:     if number  $<$  node.afreq then
9:       number = number + 1;
10:    end if
11:  end for
12: end while

```

---

đã có trong CFRG thứ nhất sẽ không được thêm vào. Như vậy quá trình hợp nhất hai cấu trúc CFRG sẽ sử dụng thuật toán duyệt CFRG luật (Thuật toán 3.4) rồi thêm từng luật vào CFRG luật thứ nhất (Thuật toán 3.2), đồng thời sẽ sử dụng thêm thuật toán tìm kiếm luật trên CFRG (Thuật toán 3.7) trước khi thêm luật vào CFRG còn lại.

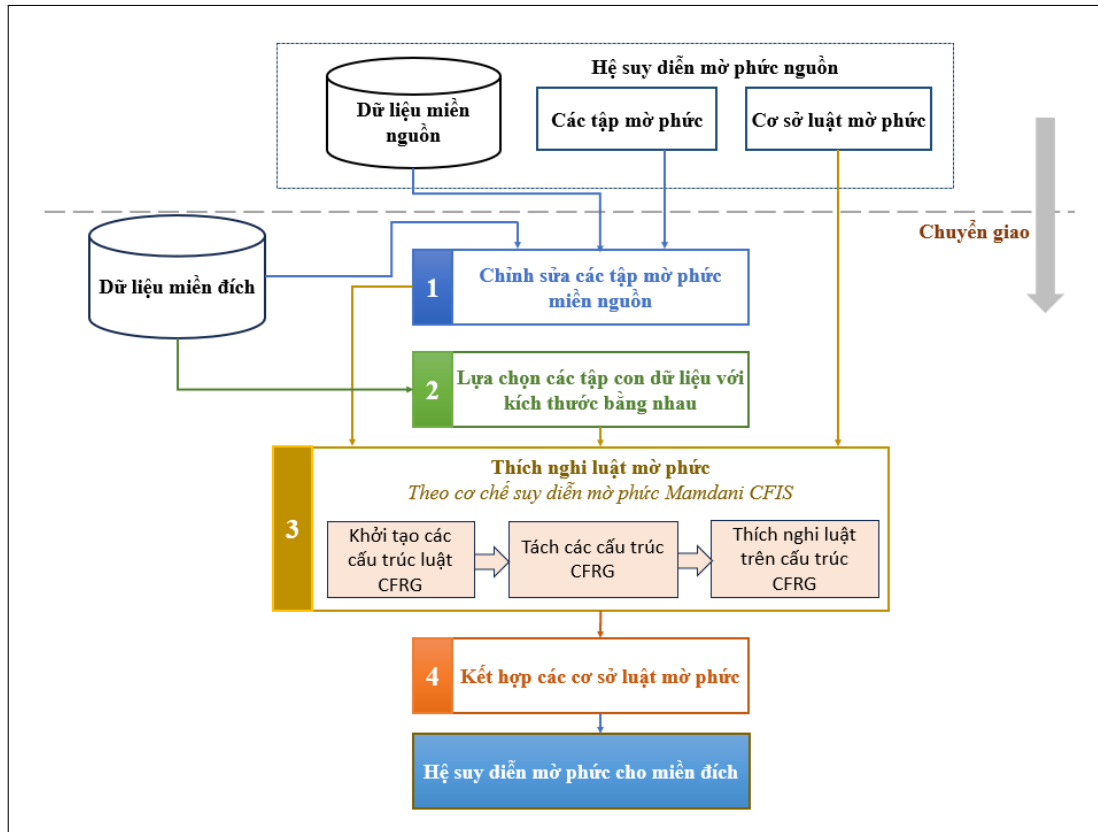
### 3.4. Mô hình học chuyển giao mờ phức dựa trên cấu trúc CFRG (CFRGTL)

Để tăng hiệu suất của mô hình học chuyển giao đã đề xuất ở chương 2, một biểu diễn luật mờ phức trên CFRG luật mờ (Định nghĩa 3.2) được giới thiệu. Điều này làm thay đổi cơ bản các thao tác trong những giai đoạn của mô hình học chuyển giao.

So với mô hình trước đó (Hình 2.2), mô hình mới được đề xuất ở mục này, đã bổ sung vào một số giai đoạn quan trọng (Hình 3.8):

- Khởi tạo các CFRG từ CFRs của miền nguồn;
- Tách cấu trúc CFRG con ra khỏi cấu trúc CFRG cha;

- Thích nghi trên các cấu trúc CFRG;
- Cuối cùng hợp nhất các cấu trúc CFRG sau sửa đổi.



**Hình 3.8:** Mô hình học chuyển giao mờ phức dựa trên cấu trúc CFRG (CFRGTL)

### 3.4.1. Khởi tạo các cấu trúc CFRG

Giống như mô hình học chuyển giao được đề xuất ở chương 2, cơ sở luật mờ phức miền nguồn sẽ được sử dụng làm cơ sở luật mờ phức khởi tạo cho quá trình học chuyển giao. Tuy nhiên trong mô hình mới được đề xuất ở chương này, các luật mờ phức miền nguồn sẽ được khởi tạo thành các cấu trúc CFRG tương ứng sử dụng thuật toán 3.2.

Số lượng các biến ngôn ngữ trong thuộc tính đầu tiên của tập luật mờ phức miền nguồn sẽ xác định số lượng cấu trúc CFRG được tạo ra. Thuộc tính đầu tiên hỗ trợ xác định một luật sẽ được thêm vào CFRG nào. Mỗi cấu trúc CFRG sẽ biểu diễn các luật có thuộc tính đầu tiên mang giá trị như nhau.

### **3.4.2. Tách các CFRG con từ các cấu trúc CFRG ban đầu**

Đánh dấu các luật thích nghi, luật ứng cử phù hợp với dữ liệu mới trong giai đoạn thích nghi luật của mô hình học chuyển giao đã đề xuất ở chương 2, là tốn thời gian vì nó phải duyệt qua toàn bộ các luật nguồn ứng từng dòng dữ liệu đích được lựa chọn trong các Dsub. Thêm vào đó, ở những lần thích nghi tiếp theo, khi số lượng luật tăng lên thì với mỗi dòng dữ liệu đều phải duyệt qua toàn bộ tập luật.

Một vấn đề nữa cần xem xét đến, khi các luật mới được thêm vào nằm kề cận nhau có thể khá tương đồng và có thể bỏ qua khi duyệt tới. Điều này, chưa được đề cập và giải quyết trong mô hình cũ. Để giải quyết mặt hạn chế này, một đề xuất tách các CFRG đã khởi tạo thành các CFRG con để: 1. Thực hiện quá trình suy diễn đồng thời trên nhiều CFRG, 2. Xem xét bỏ qua các CFR tương đồng không cần phải duyệt tới. Việc tách các CFRG con được thực hiện theo thuật toán 3.5.

### **3.4.3. Thích nghi luật trên cấu trúc CFRG**

Một thay đổi lớn trong việc thích nghi luật trên CFRG được đề xuất trong mô hình mới này, thay thế quá trình thích nghi suy diễn mờ phức thông thường trong mô hình trước đó. Cụ thể, thích nghi luật sẽ được tiến hành đồng thời (song song) trên các cấu trúc CFRG đã được tách ra ở bước trên. Thực hiện thích nghi luật vẫn tuân theo cơ chế suy diễn theo mô hình M-CFIS (Thuật toán 3.9) và tuân theo cơ chế thích nghi của mô hình chọc chuyển giao đã đề xuất ở chương 2.

Suy diễn mờ phức trên CFRG thực hiện theo phương pháp duyệt theo chiều rộng, giúp giảm thời gian suy diễn. Việc đánh dấu các nút thích nghi và nút ứng cử nhằm xác định luật thích nghi và luật ứng cử cũng giống với cách thực hiện trong mô hình đề xuất ở chương 2 nhưng sẽ diễn ra nhanh hơn trên CFRG. Nhiệm vụ chỉnh các luật ở bước này là khác biệt so với mô hình đề xuất trước đó. Tại mỗi nút cần chỉnh sửa, giá trị biên độ mới được đề xuất bởi các giá trị pha tại nút, dựa theo tần suất pha. Điều mà trước đây, trong mô hình cũ được thực hiện bằng quá trình tìm kiếm vét cạn cho

đến khi đạt được giá trị nút thích nghi. Cách thức thực hiện lựa chọn giá trị biên độ theo phân pha này cũng giúp giảm thời gian cho việc chỉnh sửa các luật và giảm thời gian học tập của cả mô hình học chuyển giao nói chung.

#### **3.4.4. Trộn các cấu trúc CFRG**

Sau cùng, các cấu trúc CFRG thích nghi với dữ liệu sẽ được hợp nhất và bổ sung vào cấu trúc CFRG cơ sở ban đầu để chuẩn bị cho bước thích nghi tập con dữ liệu tiếp theo. Quá trình được lặp lại cho đến khi tất cả các tập con dữ liệu cần được điều chỉnh, chúng ta sẽ thu được một CFRG cơ sở luật mờ phức mới đại diện cho tập hợp luật mờ phức cho hệ thống suy diễn miền đích.

### **3.5. Thực nghiệm**

Nhằm mục đích xem xét khả năng của cấu trúc CFRG đã đề xuất ở trên, hai kịch bản thực nghiệm được đề xuất: 1. Thực nghiệm đánh giá thời gian suy diễn trên CFRG so với Mamdani - CFIS truyền thống; 2. Thực nghiệm đánh giá khả năng ứng dụng của CFRG thông qua việc đánh giá hiệu suất của mô hình học chuyển giao mờ phức dựa trên cấu trúc CFRG (CFRGTL).

#### **3.5.1. Kịch bản thực nghiệm thứ nhất**

Để chứng minh về khả năng suy diễn nhanh trên CFRG, 4 bộ dữ liệu trong Bảng 1.2, được thực hiện sinh luật theo cách M-CFIS truyền thống [7]. Từ bộ luật mờ phức được sinh ra, tôi thực hiện suy diễn mờ phức theo hai cách: cách 1, suy diễn trên mảng thông thường; cách 2, cấu trúc thành theo CFRG và suy diễn trên CFRG. Kết quả thu được như bảng dưới đây (Bảng 3.1).

**Bảng 3.1:** Bảng kết quả thực nghiệm theo kịch bản 1

STT	Tập dữ liệu	Số lượng bản ghi	Số lượng luật tạo ra	Số lượng kiểm tra	Thời gian suy diễn M-CFIS	Thời gian suy diễn CFRGTL
1	Diabetes	390	101	78	0.1099	0.0786
2	Breast-Cancer	683	288	137	0.7483	0.5821
3	CreditCard	8636	4419	1727	240.223	126.243
4	Liver	4156	1044	831	23.4976	11.7312

Kết quả thực nghiệm trong bảng 3.1 cho thấy việc cấu trúc luật trên CFRG cho phép suy diễn nhanh hơn cách tổ chức trên mảng hai chiều thông thường (M-CFIS). Với hai bộ dữ liệu có kích thước nhỏ hơn là Diabetes và BreastCancer thì thời gian suy diễn trên CFRG nhanh hơn so với M-CFIS thông thường khoảng 27% - 28%. Tuy nhiên với hai bộ dữ liệu Credit Card và Liver với kích thước lớn hơn (cả về số lượng bản ghi lẫn số lượng thuộc tính) thì thời gian suy diễn đã giảm thiểu khá nhiều 45% - 50%. Điều này cho thấy khả năng xử lý của CFRG trên các bộ dữ liệu có kích thước lớn.

### 3.5.2. Kịch bản thực nghiệm thứ hai

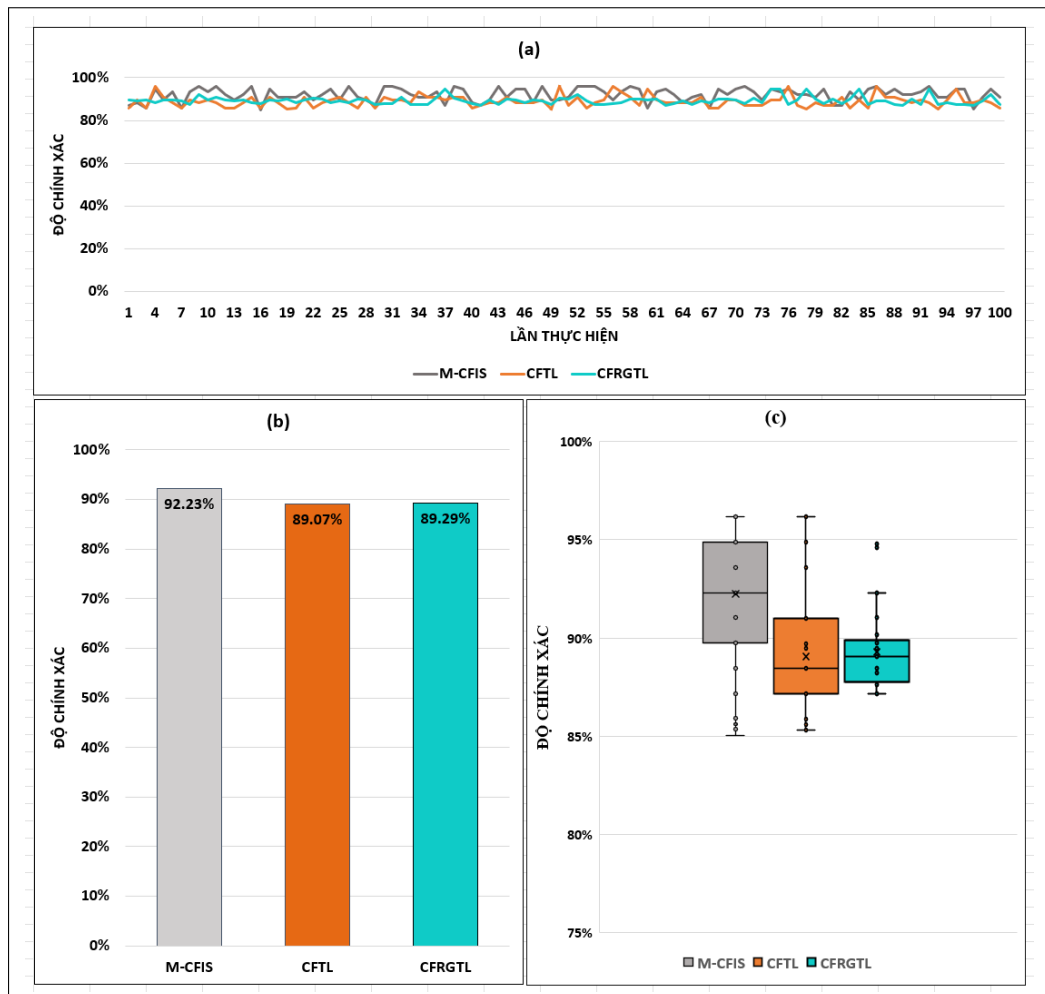
Nhằm đánh giá khả năng ứng dụng của CFRG, CFRGTL được đề xuất tiếp tục được thực nghiệm trên các tập dữ liệu UCI và dữ liệu thực tế và so sánh nó với phương pháp liên quan. Có ba tập dữ liệu (Diabetes, BreastCancer và CreditCar) từ UCI [81] và dữ liệu y tế (Bệnh gan) - Liver từ Bệnh viện Thái Nguyên. Thông tin chi tiết về bộ dữ liệu thử nghiệm được trình bày trực quan trong Bảng 1.2.

Để tìm ra khả năng cải thiện hiệu suất của CFRGTL dựa trên cấu trúc CFRG luật, kết quả thực nghiệm tiếp tục được xem xét trên ba chỉ số: thời gian tính toán, độ chính xác và số lượng luật. Hai trường hợp thực nghiệm (tương tự như kịch bản thực nghiệm

của chương 2) được thực hiện để so sánh chúng với phương pháp liên quan M-CFIS và CFTL trước đó. Mỗi thí nghiệm sẽ được thực hiện 100 lần nhằm mục đích kiểm tra tính bền vững của mô hình.

Kết quả thu được trên 4 bộ dữ liệu tương ứng với hai trường hợp thử nghiệm được trình bày lần lượt từ Hình 3.9 đến Hình 3.13.

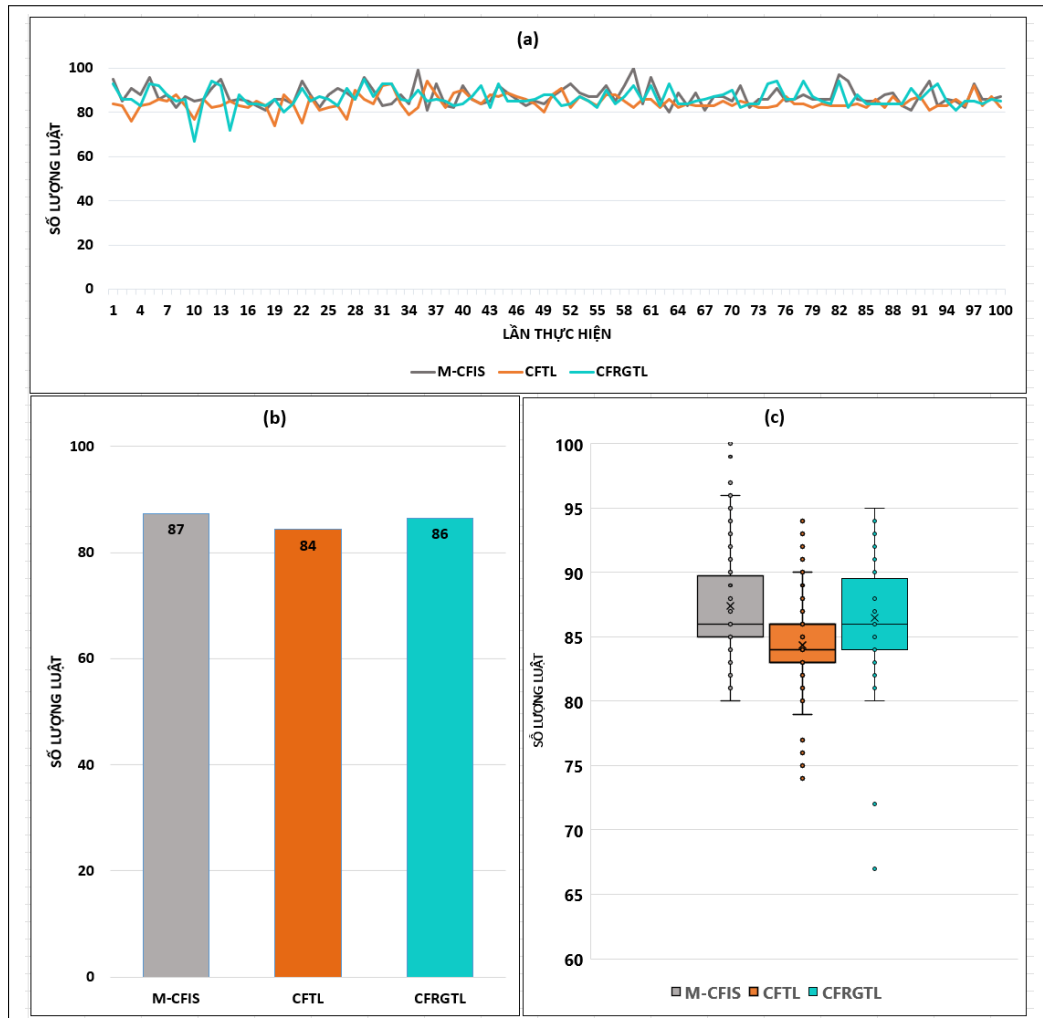
*Kết quả thực nghiệm trên bộ dữ liệu Diabetes*



**Hình 3.9:** Kết quả đo lường độ chính xác trên tập dữ liệu Diabetes.

Kết quả thực nghiệm trên bộ dữ liệu Diabetes (Hình 3.9) cho thấy mô hình học chuyển giao mờ phức mới CFRGTL có độ chính xác tốt hơn mô hình CFTL, tuy nhiên độ chính xác tốt hơn không đáng kể (89,07% trên CFTL và 89,29% trên CFRGTL -

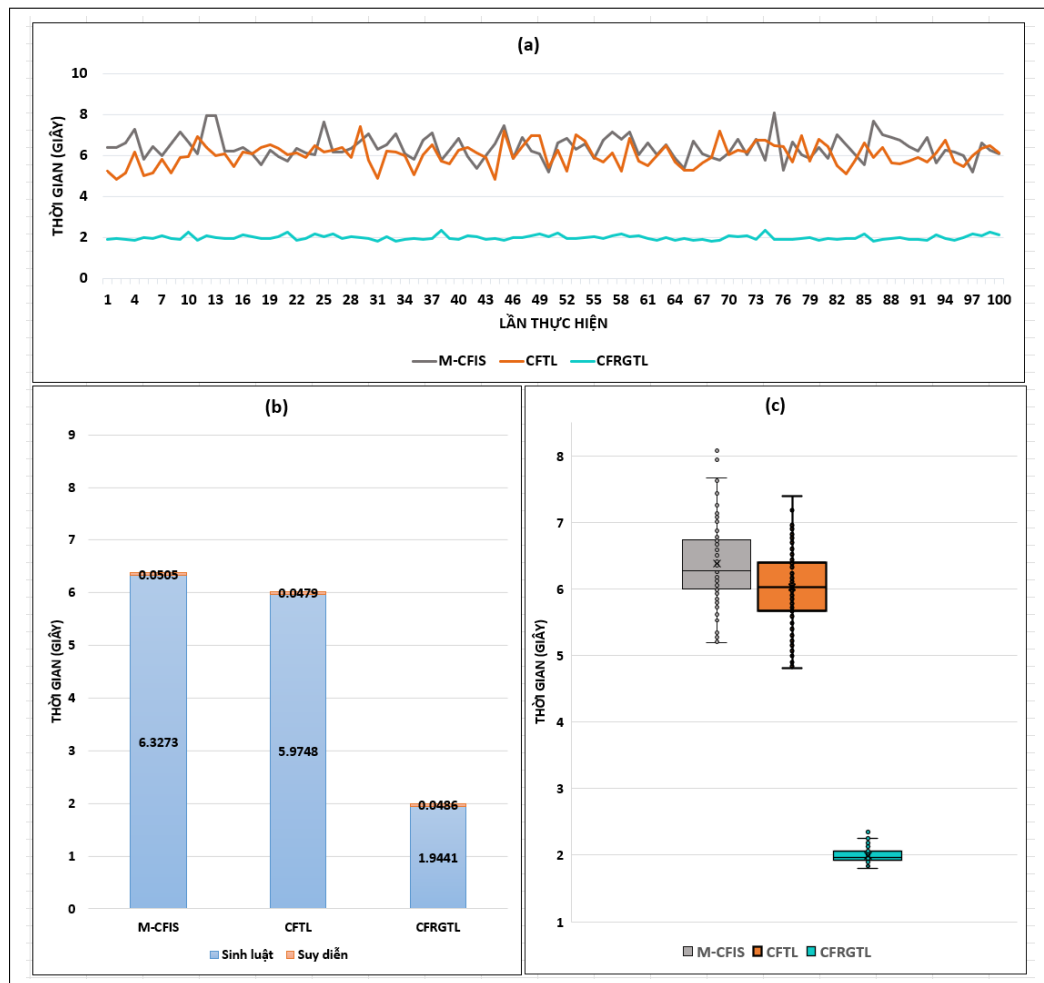
Hình 3.4(b)). Điểm đáng chú ý là, phương sai của giá trị độ chính xác đo được trong các lần thử nghiệm là nhỏ hơn so với phương pháp CFTL (Hình 3.9 (a, c)) cho thấy tính ổn định của phương pháp mới.



**Hình 3.10:** Kết quả đo lường số lượng luật trên tập dữ liệu Diabetes.

Hình 3.10 (b) đưa ra thông tin số lượng luật trung bình tạo ra của ba mô hình M-CFIS, CFTL và CFRGTL. Mặc dù số lượng luật tạo ra trong mô hình CFRGTL vẫn ít hơn so với mô hình M-CFIS, nó lại lớn hơn so với CFTL. Điều này là dễ hiểu vì mức độ dao động về số lượng luật của CFRGTL thấp hơn so với CFTL (Hình 3.10 (c)) và độ chính xác cao hơn (như đã đề cập ở trên).





**Hình 3.11:** Kết quả đo lường thời gian thực hiện trên bộ dữ liệu Diabetes.

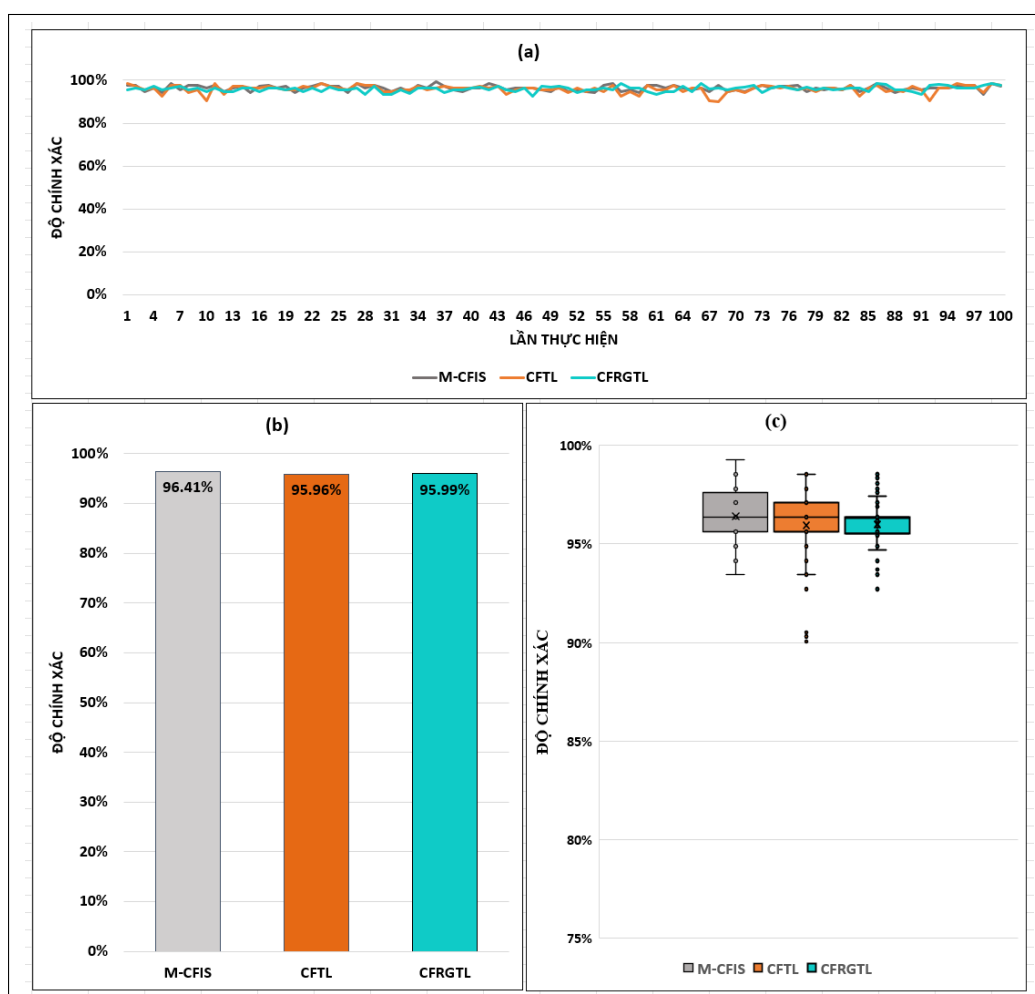
Hình 3.11 là kết quả đo lường thời gian thực hiện (thời gian tạo ra mô hình suy diễn mờ phức cho miền đích) của mô hình CFRGTL nhanh hơn so với mô hình CFTL cũ và dĩ nhiên nhanh hơn nhiều so với mô hình M-CFIS. Do số lượng luật sinh ra ở CFRGTL là nhiều hơn (Hình 3.10) nên thời gian suy diễn trên dữ liệu kiểm thử cũng nhiều hơn. Mặc dù tổng thời gian sinh luật và suy diễn trên mô hình mới là nhỏ hơn.

Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 3.12). Kết quả đo lường cho thấy mô hình CFRGTL ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so với CFTL.

	M-CFIS	CFTL	CFRGTL
<b>Phương sai</b>	0.341563647	0.315734034	0.013267774
<b>Độ lệch chuẩn</b>	0.584434468	0.561902157	0.115185824

**Hình 3.12:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập Diabetes.

*Kết quả thực nghiệm trên bộ dữ liệu BreastCancer*

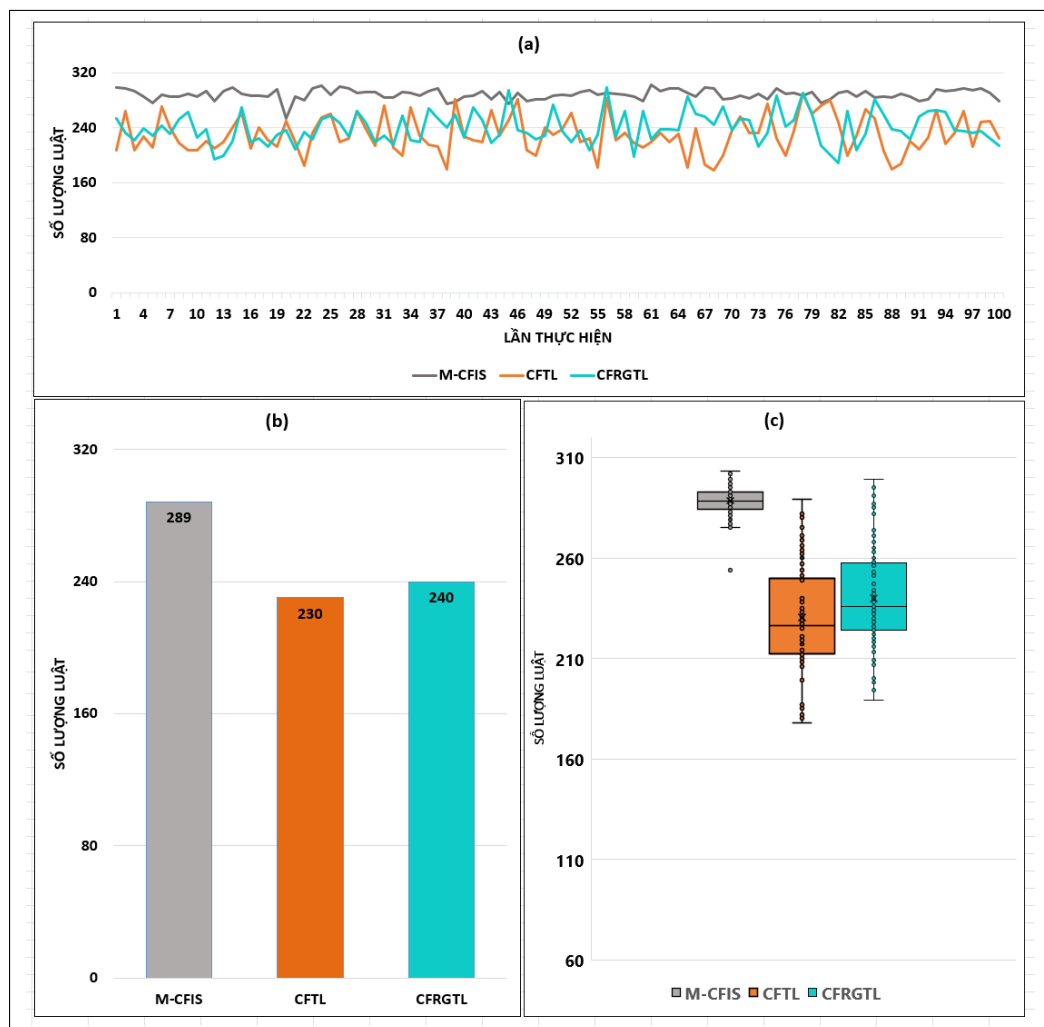


**Hình 3.13:** Kết quả đo lường độ chính xác trên bộ dữ liệu BreastCancer.

Tương tự với bộ dữ liệu BreastCancer, độ chính xác trung bình của mô hình CFRGTL tốt hơn chút ít với mô hình cũ CFTL tương ứng là 95,99% và 95,96%.

Điều đặc biệt là, phương sai của độ chính xác ổn định hơn trong mỗi lần huấn luyện (Hình 3.13 (a, c)). Giá trị độ chính xác thấp nhất mà mô hình CFRGTL đạt được lớn hơn so với mô hình CFTL (Hình 3.13 (c)).

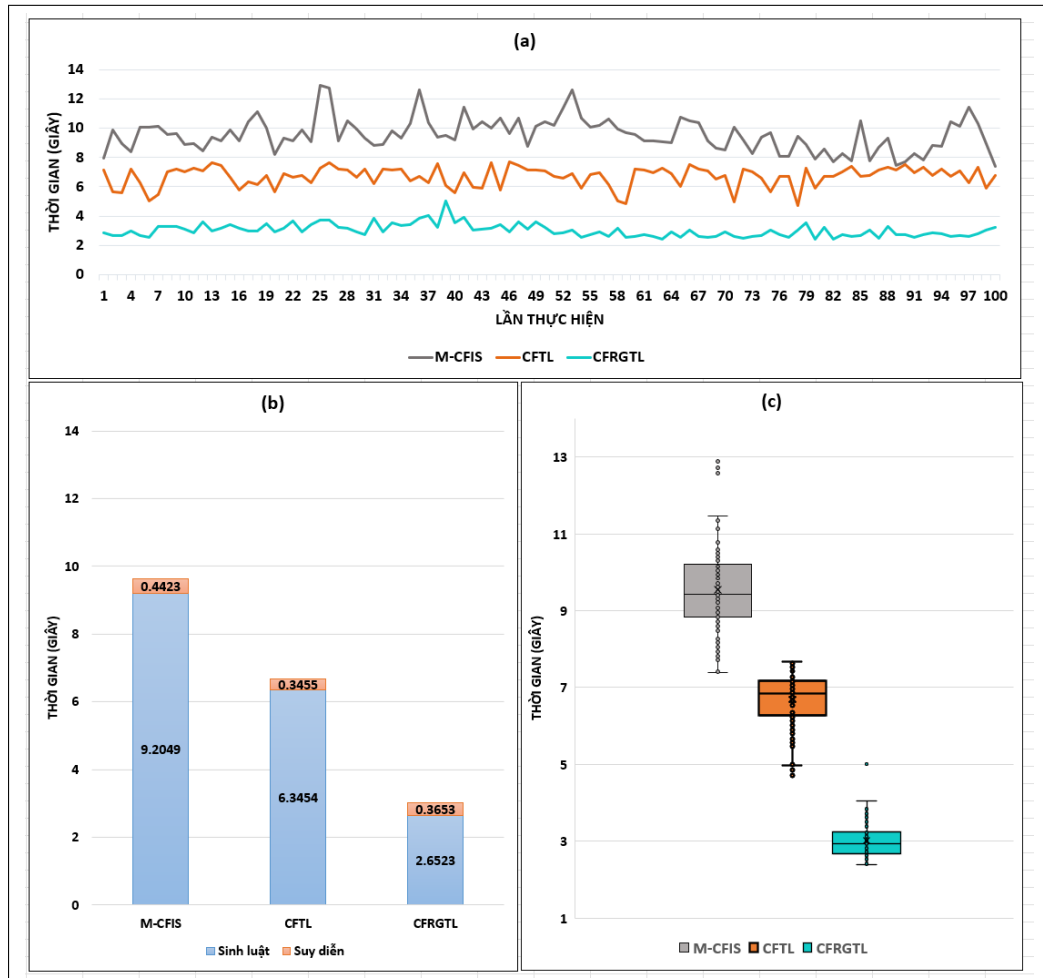
Số lượng luật trung bình của mô hình CFRGTL là lớn hơn so với mô hình CFTL mặc dù nó vẫn thấp hơn so với mô hình M-CFIS (Hình 3.14 (b)). Tương tự phương sai, số lượng luật trong mỗi lần thử nghiệm trên CFRGTL cũng nhỏ hơn so với CFTL (Hình 3.14 (c)).



**Hình 3.14:** Kết quả đo lường số lượng luật trên bộ dữ liệu BreastCancer.

Thời gian huấn luyện trung bình trên mô hình CFRGTL ít hơn đến 2,52 lần so với mô hình học CFTL và ít hơn 3,59 lần so với mô hình M-CFIS (Hình 3.15 (b)). Đặc

biệt phương sai thời gian huấn luyện cũng giảm so với mô hình CFTL (Hình 3.15 (a, c)).



**Hình 3.15:** Kết quả đo lường thời gian thực hiện trên bộ dữ liệu BreastCancer.

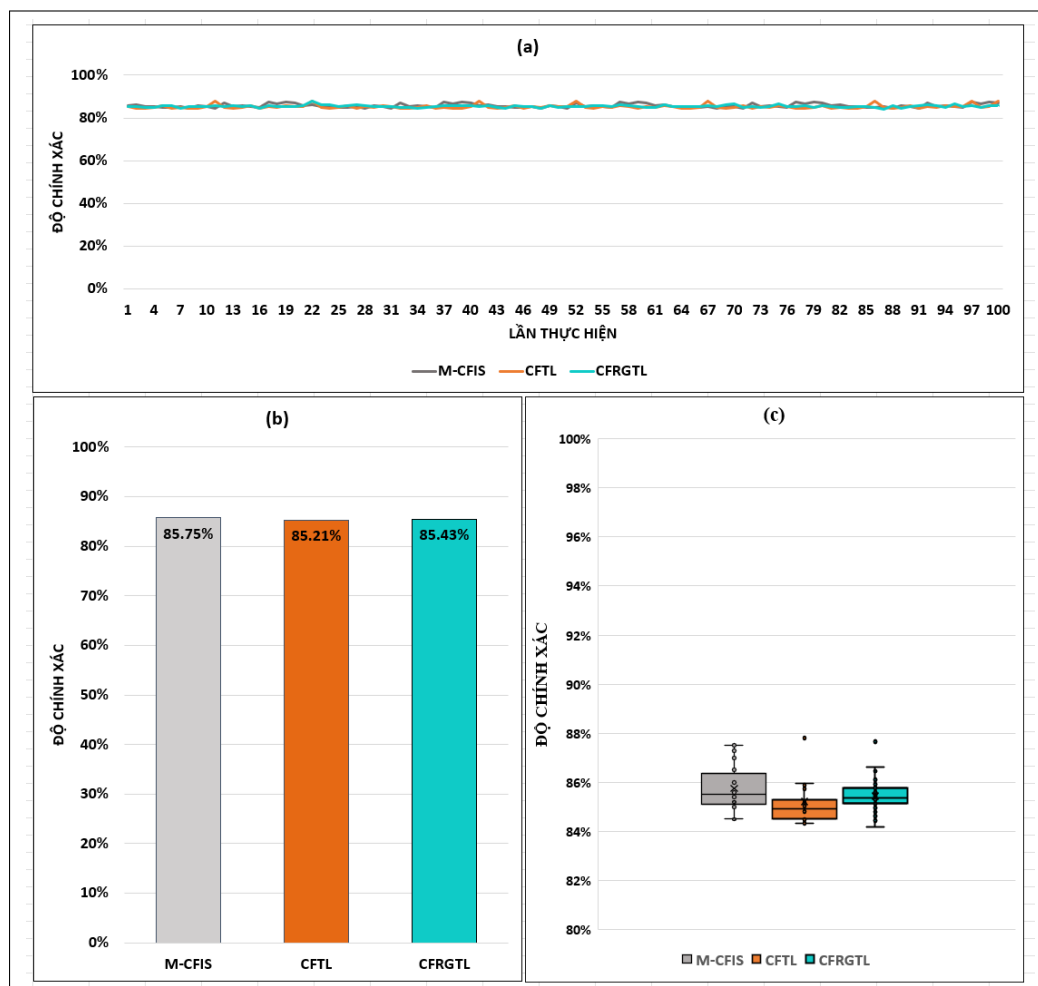
Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 3.16). Kết quả đo lường cho thấy mô hình CFRGTL ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so với CFTL.

	M-CFIS	CFTL	CFRGTL
<b>Phương sai</b>	1.268646981	0.461671166	0.185177656
<b>Độ lệch chuẩn</b>	1.126342302	0.679463881	0.430322735

**Hình 3.16:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập BreastCancer.

*Kết quả trên bộ dữ liệu CreditCard*

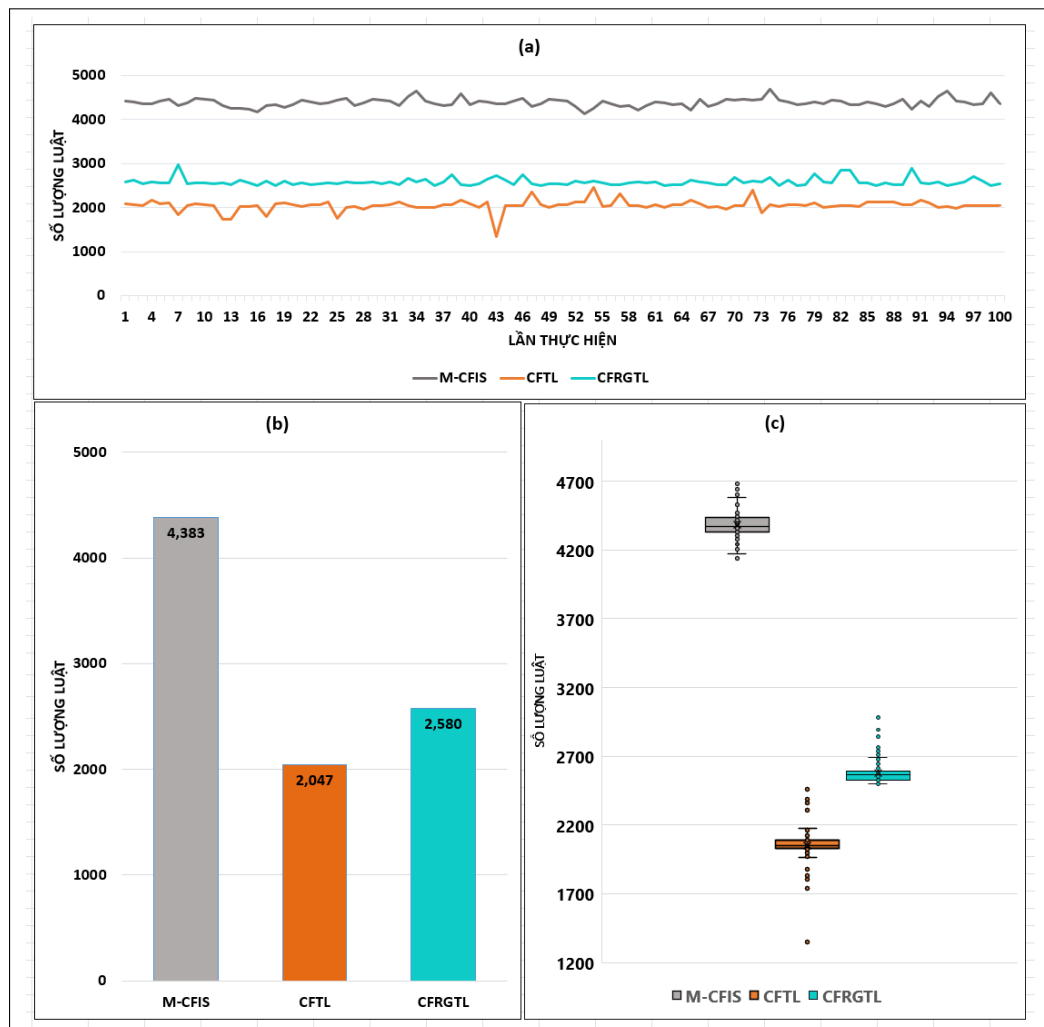
Khả năng vượt trội của CFRGTL mới đề xuất thể hiện rõ trong bộ dữ liệu CreditCard với kích thước dữ liệu lớn và nhiều đầu ra.



**Hình 3.17:** Kết quả đo lường độ chính xác trên bộ dữ liệu CreditCard.

Mô hình CFRGTL trên bộ dữ liệu CreditCard tiếp tục cho thấy khả năng cải thiện độ chính xác so với CFTL, tương ứng trung bình là 85,43% và 85,21% (Hình 3.17 (b)). Phương sai của độ chính xác cũng tiếp tục được cải thiện (Hình 3.17 (a, c)).

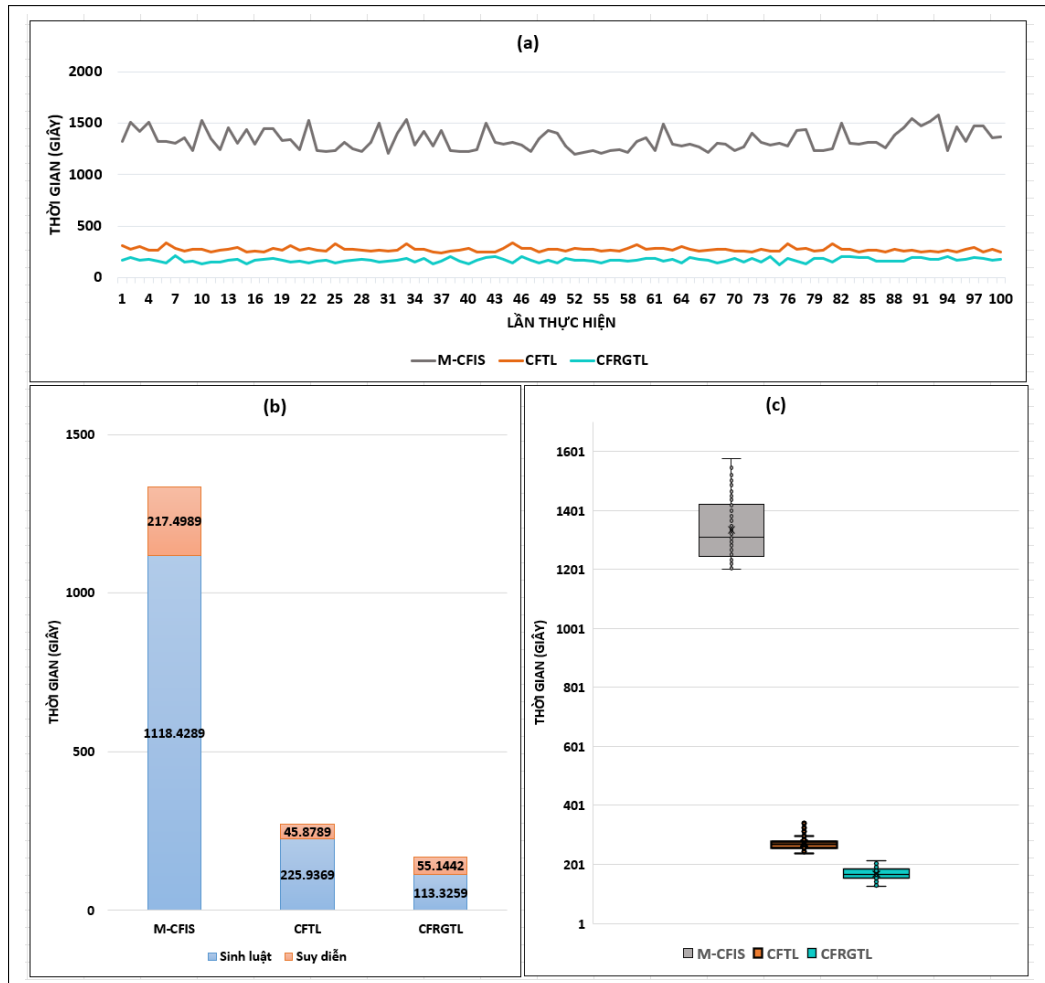
Đặc biệt với bộ dữ liệu có kích thước lớn như bộ dữ liệu CreditCard, số lượng luật giảm là một trong yếu tố quan trọng quyết định đến thời gian suy diễn. Mặc dù, số lượng luật trung bình của mô hình CFRGTL thu được cao hơn so với mô hình CFTL nhưng vẫn giảm nhiều so với mô hình M-CFIS (Hình 3.18 (b)). Ngoài ra, Hình 3.18 (a, c) cho mức độ ổn định về số lượng luật của mô hình CFRGTL hơn so với CFTL.



**Hình 3.18:** Kết quả đo lường số lượng luật trên bộ dữ liệu CreditCard.

Mô hình CFRGTL tiếp tục cho thấy tính ổn định của mô hình trên yếu tố thời gian

(Hình 3.19). CFRGTL với thời gian huấn luyện sinh luật rút ngắn hơn nhiều so với mô hình CFTL gốc. Cụ thể với mô hình CFTL thì tổng thời gian sinh luật và suy diễn giảm so với mô hình M-CFIS khoảng 4.91 lần thì mô hình mới CFRGTL giảm khoảng 7,92 lần (Hình 3.19 (b)). Hình 3.19 (a, c) cũng cho thấy phương sai thời gian thực hiện của mô hình CFRGTL cũng thấp hơn so với mô hình M-CFIS.



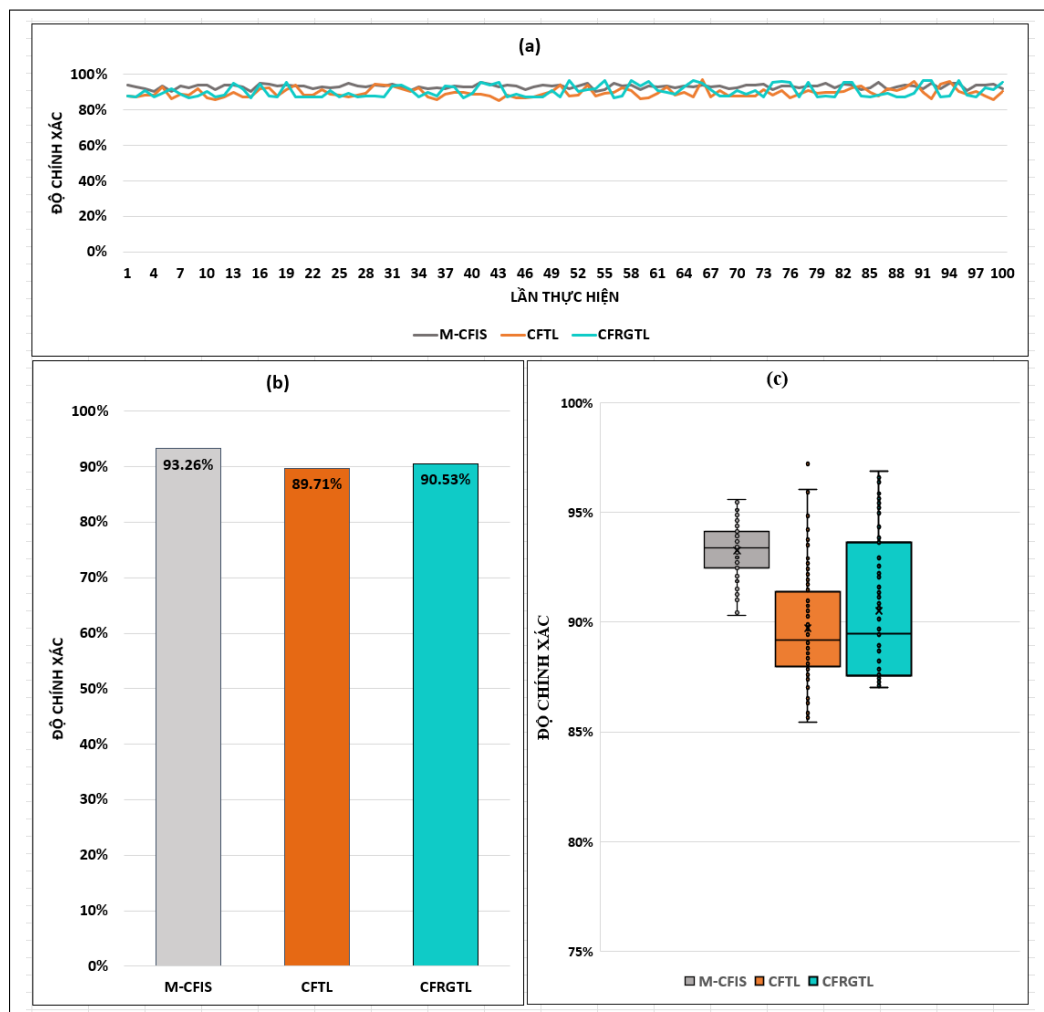
**Hình 3.19:** Kết quả đo lường thời gian thực hiện trên bộ dữ liệu CreditCard.

Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 3.20). Kết quả đo lường cho thấy mô hình CFRGTL tiếp tục ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so với CFTL.

	M-CFIS	CFTL	CFRGTL
<b>Phương sai</b>	9885.061651	460.8195573	402.6103345
<b>Độ lệch chuẩn</b>	99.42364734	21.46670812	20.06515224

**Hình 3.20:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập CreditCard.

*Kết quả trên bộ dữ liệu Liver*

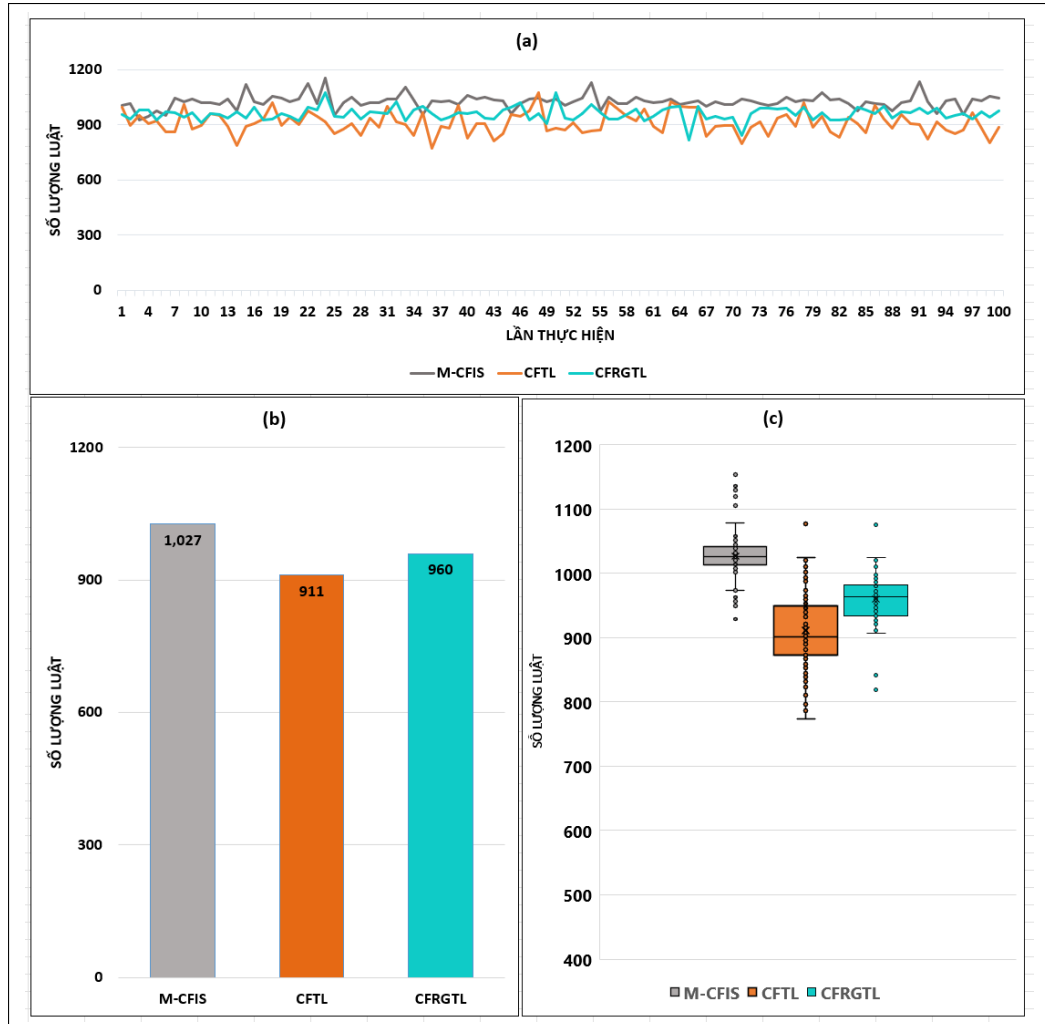


**Hình 3.21:** Kết quả đo lường độ chính xác trên bộ dữ liệu Liver.

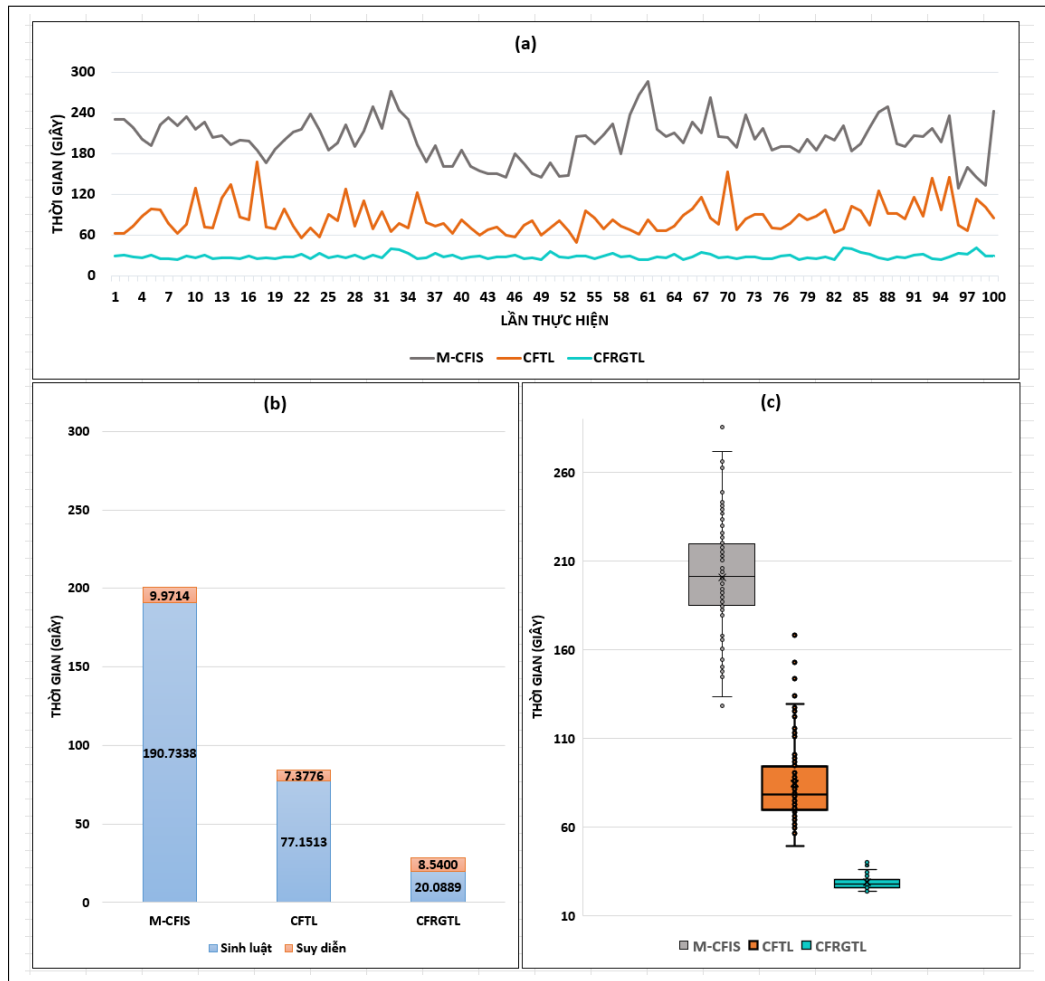
Cuối cùng là kết quả thực nghiệm trên mô hình dữ liệu thực được trình bày trong Hình 3.21, Hình 3.22 và Hình 3.23), cũng thu được kết quả thực nghiệm theo quy luật



tương tự. Các kết quả trên mô hình CFRGTL đều ổn định hơn mô hình CFTL cũ và thời gian huấn luyện sinh luật cũng rút ngắn hơn.



**Hình 3.22:** Kết quả đo lường số lượng luật trên bộ dữ liệu Liver.



**Hình 3.23:** Kết quả đo lường thời gian thực hiện trên bộ dữ liệu Liver.

Để xem xét mức độ ổn định của mô hình về mặt thời gian, kết quả đo lường phương sai và độ lệch chuẩn trên 100 mẫu thời gian thực hiện ở cả hai mô hình được biểu diễn trong Hình 3.24). Kết quả đo lường cho thấy mô hình CFRGTL tiếp tục ổn định hơn vì Phương sai và Độ lệch chuẩn nhỏ hơn so sử dụng FTL.

	M-CFIS	CFTL	CFRGTL
<b>Phương sai</b>	963.3746765	498.7194385	14.57840692
<b>Độ lệch chuẩn</b>	31.0382776	22.33202719	3.818168006

**Hình 3.24:** Kết quả đo lường phương sai và độ lệch chuẩn của thời gian trên tập Liver.

### **Nhận xét**

Thứ nhất, việc cấu trúc luật trên cấu trúc CFRG - một biểu diễn chưa từng có trước đây giúp quá trình duyệt luật trên CFRG nhanh hơn. Làm cho quá trình suy diễn mờ phức trên CFRG cũng nhanh hơn.

Thứ hai, trong giai đoạn chỉnh sửa luật việc lựa chọn biên độ mới theo đề xuất phần pha giúp giảm thời gian lựa chọn biên độ.

Thứ ba, việc tách từng CFRG con từ CFRG ban đầu và thực hiện kỹ thuật đa luồng cho quá trình suy diễn chỉnh sửa cũng giúp ngắn thời gian so với mô hình đề xuất ở chương 2 .

### **3.6. Kết luận Chương 3**

Trong chương này, tác giả đã đề xuất một cấu trúc CFRG mới cho phép lưu trữ cả phần biên độ, phần pha và tần suất của chúng trên mỗi nút. Lợi điểm của cách thức lưu trữ này, giúp trực quan yếu tố pha của mỗi nút luật, thấy rõ hơn khả năng ảnh hưởng của chúng đối với luật trong quá trình suy diễn hay chỉnh sửa.

Rõ ràng, tính ưu việt của cấu trúc CFRG đã được thể hiện rõ trong kết quả thực nghiệm thứ nhất, khi sử dụng để suy diễn mờ phức đã rút ngắn thời gian suy diễn đầu ra của các hệ M-CFIS. Thêm vào đó, việc sử dụng cấu trúc CFRG cho quá trình thích nghi luật trong học chuyển giao mờ phức cũng rút ngắn thời gian thích nghi do hai yếu tố: 1) lựa chọn giá trị chỉnh sửa mới dựa trên phần pha trên mỗi nút; 2) có thể thực hiện đa luồng thông qua việc tách các cấu trúc con CFRG.

Kết quả nghiên cứu này được công bố trong công trình [6] của luận án

Cũng từ kết quả nghiên cứu của chương này, một số câu hỏi nghiên cứu tiếp tục được đặt ra như sau: 1) Cấu trúc CFRG có thể hỗ trợ tổng hợp luật tốt hơn làm tăng độ chính xác hay giảm số lượng luật hay không? 2) Ngữ cảnh của mô hình học chuyển giao có thể thực hiện mở rộng trong các trường hợp khác hay không? 3) Các giai đoạn khác của mô hình học chuyển giao gốc ở chương 2 có thể tiếp tục phát triển làm tăng

hiệu suất của mô hình không?

Đây cũng chính là mục tiêu phát triển mô hình đề xuất ở các nghiên cứu tiếp theo

## KẾT LUẬN

### Những kết quả chính của luận án

Với mục tiêu nghiên cứu là tập trung nghiên cứu đề xuất xây dựng Mô hình học chuyển giao trên hệ suy diễn mờ phức nhằm cải thiện thời gian học tập (thời gian xây dựng hệ suy diễn) cho miền đích:

- **Thứ nhất**, luận án đã đề xuất một Mô hình học chuyển giao trên hệ suy diễn mờ phức Mamdani.
  - Phương pháp đề xuất đã sử dụng kỹ thuật học chuyển giao trong quá trình xây dựng hệ suy diễn mờ phức Mamdani cho miền mục đích với mục tiêu rút ngắn thời gian so với phương pháp trước đó.
- **Thứ hai**, luận án cũng đã đề xuất một cách biểu diễn tập luật mờ phức trên cấu trúc CFRG:
  - Biểu diễn trực quan luật mờ phức trên cấu trúc CFRG
  - Cải thiện thời gian suy diễn mờ phức của các mô hình suy diễn mờ phức Mamdani. Trên cơ sở đó tiếp tục cải thiện thời gian thích nghi luật (đánh dấu luật thích nghi, ứng cử và Lựa chọn giá trị chỉnh sửa) của mô hình học chuyển giao đã đề xuất trước đó.

Luận án một lần nữa minh chứng khả năng của kỹ thuật học chuyển giao trong việc rút ngắn thời gian học tập. Đồng thời cũng khẳng định vai trò của các yếu tố bổ sung trong dữ liệu đối với các nhiệm vụ học tập và suy diễn. Như trong trường hợp này là thành phần Pha.

### Một số hạn chế của luận án

Bên cạnh các kết quả nghiên cứu đã đạt được, những nghiên cứu trong luận án vẫn còn tồn tại một số điểm hạn chế như:

- Luận án mới chỉ thực nghiệm trên các bộ dữ liệu số, còn dữ liệu không gian chưa được thực hiện.
- Các bước thực hiện trong mô hình học chuyển giao mờ đề xuất còn đơn giản.
- Ngưỡng cảnh cho mô hình học chuyển giao còn nhiều ràng buộc và trong điều kiện quá lý tưởng.
- Chưa khai thác đầy đủ khả năng của cấu trúc CFRG đã đề xuất, như khả năng xử lý dữ liệu khuyết, khả năng tổng hợp và tối ưu hóa bộ luật trên cấu trúc...

## **Hướng phát triển của luận án**

Trong tương lai, định hướng phát triển tiếp theo của luận án có thể thực hiện theo các hướng nghiên cứu sau:

- Thử nghiệm thuật toán trên mô hình dữ liệu lớn
- Bổ sung thêm các chiều của dữ liệu để thể hiện tinh ưu việt của mô hình;
- Tiếp tục thử nghiệm các mô hình đề xuất trong luận án với nhiều bộ dữ liệu phức tạp hơn trong các lĩnh vực khác nhau trong cuộc sống như: y tế, kinh tế, địa lý v.v.
- Áp dụng, triển khai và tích hợp các nghiên cứu đã đề xuất cho các hệ thống trong thực tiễn như dự báo thời tiết, thiên tai, dự báo bão, v.v.;

## CÁC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ

### Công trình đã công bố

- [CT1] **Triệu Thu Hương**, Lương Thị Hồng Lan, Nguyễn Long Giang, Phạm Ngọc Đức, Lê Minh Tuấn and Phùng Thế Huân. (2020) “Hệ hỗ trợ ra quyết định dựa trên tập mờ phức và ứng dụng chấm điểm tín dụng”. *Kỷ yếu Hội thảo quốc gia lần thứ XXIII về Một số vấn đề chọn lọc của công nghệ thông tin & truyền thông*, 125-130.
- [CT2] Lương Thị Hồng Lan, **Triệu Thu Hương**, Nguyễn Long Giang, Lê Hoàng Sơn, Vũ Thị Khánh Trinh (2022). Một mô hình học chuyển giao mờ và ứng dụng. *Kỷ yếu Hội thảo Quốc gia lần thứ XXV - VNICT 2022*, Hà Nội, tr 87-91.
- [CT3] **Triệu Thu Hương**, Lương Thị Hồng Lan, Nguyễn Long Giang, Nguyễn Mỹ Bình, Bay Vo, Lê Hoàng Sơn (2023). A novel transfer learning model on complex fuzzy inference system. *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 3, p 3733-3750. (ISSN: 1064-1246, SCIE, 2023, IF=2.0), DOI = <https://doi.org/10.3233/JIFS-222582>)
- [CT4] Chu Thi Hong Hai, **Triệu Thu Hương** (2022). Research Complex Fuzzy Inference System in Early Warning Credit Risk at Commercial Banks in Viet Nam. *Intelligent Systems and Networks: Selected Articles from ICISN 2022, Vietnam* (pp. 519-525). Singapore: Springer Nature Singapore.
- [CT5] **Triệu Thu Hương**, Lương Thị Hồng Lan, Lê Trường Giang, Nguyễn Long Giang, Phạm Huy Thông, Nguyễn Thị Mỹ Bình (2021). Về một mô hình học chuyển giao trên hệ suy diễn mờ phức. *Kỷ yếu Hội thảo Quốc gia lần thứ XXIV - VNICT 2021*, Thái Nguyên, tr 616-621.

## **Công trình đã được nhận đăng**

[CT6 ] **Trieu Thu Huong**, Luong Thi Hong Lan (2023). The Novel Tree - Based Complex Fuzzy Transfer Learning System. Journal of Computer Science and Cybernetics. Accepted.



## TÀI LIỆU THAM KHẢO

### Tiếng Anh:

- [1] D. Ramot, R. Milo, M. Friedman, and A. Kandel, “Complex fuzzy sets,” *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 171–186, 2002.
- [2] Z. Chen, S. Aghakhani, J. Man, and S. Dick, “Ancfis: A neurofuzzy architecture employing complex fuzzy sets,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 305–322, 2010.
- [3] O. Yazdanbakhsh and S. Dick, “Ancfis-elm: A machine learning algorithm based on complex fuzzy sets,” in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2016, pp. 2007–2013.
- [4] O. Yazdanbakhsh and S. Dick, “Fancfis: Fast adaptive neuro-complex fuzzy inference system,” *International Journal of Approximate Reasoning*, vol. 105, pp. 417–430, 2019.
- [5] G. Selvachandran, S. G. Quek, L. T. H. Lan, *et al.*, “A new design of mamdani complex fuzzy inference system for multiattribute decision making problems,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 4, pp. 716–730, 2019.
- [6] T. M. Tuan, L. T. H. Lan, S.-Y. Chou, *et al.*, “M-cfis-r: Mamdani complex fuzzy inference system with rule reduction using complex fuzzy measures in granular computing,” *Mathematics*, vol. 8, no. 5, p. 707, 2020.
- [7] L. T. H. Lan, T. M. Tuan, T. T. Ngan, N. L. Giang, V. T. N. Ngoc, P. Van Hai, *et al.*, “A new complex fuzzy inference system with fuzzy knowledge graph and extensions in decision making,” *IEEE Access*, vol. 8, pp. 164 899–164 921, 2020.

- [8] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, “Transfer learning using computational intelligence: A survey,” *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.
- [9] X. Chen, “Simulation of english speech emotion recognition based on transfer learning and cnn neural network,” *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 2, pp. 2349–2360, 2021.
- [10] S. Niu, Y. Liu, J. Wang, and H. Song, “A decade survey of transfer learning (2010–2020),” *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020.
- [11] F. Zhuang, Z. Qi, K. Duan, *et al.*, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [12] C. Wang and S. Mahadevan, “Heterogeneous domain adaptation using manifold alignment,” in *Twenty-second international joint conference on artificial intelligence*, 2011.
- [13] A. Li, Z. Deng, Q. Lou, K.-S. Choi, H. Shen, and S. Wang, “A novel tsk fuzzy system incorporating multi-view collaborative transfer learning for personalized epileptic eeg detection,” *arXiv preprint arXiv:2111.08457*, 2021.
- [14] N. Wagner, L. H. Son, M. Joo, *et al.*, “Complex evolutionary artificial intelligence in cognitive digital twinning,” *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 2, pp. 2013–2016, 2021.
- [15] T.T.Ngân, N. Dung, T. Tuấn, and L. Lan, “Mô hình học chuyển giao mờ trong hỗ trợ chẩn đoán bệnh xơ gan,” *TNU Journal of Science and Technology*, vol. 189, no. 13, pp. 93–98, 2018.
- [16] S. H. Kassani, P. H. Kassani, M. J. Wesolowski, K. A. Schneider, and R. Deters, “Breast cancer diagnosis with transfer learning and global pooling,” in

*2019 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 2019, pp. 519–524.

- [17] M. Maqsood, F. Nazir, U. Khan, *et al.*, “Transfer learning assisted classification and detection of alzheimer’s disease stages using 3d mri scans,” *Sensors*, vol. 19, no. 11, p. 2645, 2019.
- [18] X. Tang, B. Du, J. Huang, Z. Wang, and L. Zhang, “On combining active and transfer learning for medical data classification,” *IET Computer Vision*, vol. 13, no. 2, pp. 194–205, 2019.
- [19] H. T. Anh, “Kết hợp các mô hình trong học chuyển giao để xây dựng mô hình phân lớp dự đoán bệnh về da,” pp. 75–81, 2021.
- [20] J. Xu, S. Ramos, D. Vázquez, and A. M. López, “Domain adaptation of deformable part-based models,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2367–2380, 2014.
- [21] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, “Transfer learning for driver model adaptation in lane-changing scenarios using manifold alignment,” *IEEE transactions on intelligent transportation systems*, vol. 21, no. 8, pp. 3281–3293, 2019.
- [22] Y. Liu, P. Lasang, S. Pranata, S. Shen, and W. Zhang, “Driver pose estimation using recurrent lightweight network and virtual data augmented transfer learning,” *IEEE transactions on intelligent transportation systems*, vol. 20, no. 10, pp. 3818–3831, 2019.
- [23] S. Bansod and A. Nandedkar, “Transfer learning for video anomaly detection,” *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 1967–1975, 2019.
- [24] T.T.P.Thanh and N. Nghe, “Nhận dạng bệnh trên lá lúa bằng phương pháp học chuyển giao,” *Tạp chí Khoa học Đại học Cần thơ*, vol. 58, no. 4, pp. 1–7, 2022.

- [25] H. Thang, N. Tuan, D. Son, T. Van, and T. Anh, “Phân loại luồng dữ liệu dựa trên học chuyển giao đa nguồn trong hệ thống mạng sdn phân tán,” *Journal of Science and Technology on Information security*, pp. 59–60, 2023.
- [26] S. S. Pal and S. Kar, “Fuzzy transfer learning in time series forecasting for stock market prices,” *Soft Computing*, pp. 1–12, 2022.
- [27] J. Shell and S. Coupland, “Fuzzy transfer learning: Methodology and application,” *Information Sciences*, vol. 293, pp. 59–79, 2015.
- [28] H. Zuo, J. Lu, G. Zhang, and F. Liu, “Fuzzy transfer learning using an infinite gaussian mixture model and active learning,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 2, pp. 291–303, 2018.
- [29] P. Xu, Z. Deng, J. Wang, Q. Zhang, K.-S. Choi, and S. Wang, “Transfer representation learning with tsf fuzzy system,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 649–663, 2019.
- [30] S. Hussain, Y.-S. Kim, S. Thakur, and J. G. Breslin, “Optimization of waiting time for electric vehicles using a fuzzy inference system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 396–15 407, 2022.
- [31] N. Murakami, M. Ishida, Y. Takahashi, H. Yanaka, and D. Bekki, “Knowledge injection for disease names in logical inference between japanese clinical texts,” in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 2023, pp. 108–117.
- [32] C. Wu, “Statistical inference with non-probability survey samples,” *Surv. Methodol*, vol. 48, pp. 283–311, 2022.
- [33] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, “Membership inference attacks on machine learning: A survey,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

- [34] A. Feder, K. A. Keith, E. Manzoor, *et al.*, “Causal inference in natural language processing: Estimation, prediction, interpretation and beyond,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 1138–1158, 2022.
- [35] Y. Wang, D. Liang, L. Charlin, and D. M. Blei, “Causal inference for recommender systems,” in *Proceedings of the 14th ACM Conference on Recommender Systems*, 2020, pp. 426–431.
- [36] C. Gao, Y. Zheng, W. Wang, F. Feng, X. He, and Y. Li, “Causal inference in recommender systems: A survey and future directions,” *arXiv preprint arXiv:2208.12397*, 2022.
- [37] T. M. Alam, K. Shaukat, A. Khelifi, *et al.*, “A fuzzy inference-based decision support system for disease diagnosis,” *The Computer Journal*, vol. 66, no. 9, pp. 2169–2180, 2023.
- [38] H. Humaira, R. Rasyidah, and I. Rahmayuni, “Designing mamdani fuzzy inference systems for decision support systems,” in *2019 2nd International Conference on Applied Information Technology and Innovation (ICAITI)*, IEEE, 2019, pp. 111–115.
- [39] N. Masri, Y. A. Sultan, A. N. Akkila, *et al.*, “Survey of rule-based systems,” *International Journal of Academic Information Systems Research (IJASIR)*, vol. 3, no. 7, pp. 1–23, 2019.
- [40] X. Gu, P. P. Angelov, and Q. Shen, “Self-organizing fuzzy belief inference system for classification,” *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 12, pp. 5473–5483, 2022.
- [41] F. Aghaeipoor, M. Sabokrou, and A. Fernández, “Fuzzy rule-based explainer systems for deep neural networks: From local explainability to global understanding,” *IEEE Transactions on Fuzzy Systems*, 2023.

- [42] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.
- [43] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, "Complex fuzzy logic," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 450–461, 2003.
- [44] E. H. Mamdani, "Applications of fuzzy algorithms for control of simple dynamic plant," *Proc. Iee*, vol. 121, pp. 1585–1588, 1974.
- [45] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE transactions on systems, man, and cybernetics*, no. 1, pp. 116–132, 1985.
- [46] M. F. Azeem, *Fuzzy inference system: theory and applications*. BoD–Books on Demand, 2012.
- [47] S. Chaudhari, M. Patil, and J. Bambhori, "Study and review of fuzzy inference systems for decision making and control," *American International Journal of Research in Science, Technology, Engineering & Mathematics*, vol. 14, no. 147, pp. 88–92, 2014.
- [48] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing—a computational approach to learning and machine intelligence [book review]," *IEEE Transactions on automatic control*, vol. 42, no. 10, pp. 1482–1484, 1997.
- [49] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on systems, man, and cybernetics*, vol. 22, no. 6, pp. 1414–1427, 1992.
- [50] Y. Li and Y.-T. Jang, "Complex adaptive fuzzy inference systems," in *Soft Computing in Intelligent Systems and Information Processing. Proceedings of the 1996 Asian Fuzzy Systems Symposium*, IEEE, 1996, pp. 551–556.

- [51] G. Salomon and D. N. Perkins, “Transfer of learning,” *Computer Science*, 1992.
- [52] S. Pan and Q. Yang, *A survey on transfer learning. iee transaction on knowledge discovery and data engineering*, 22 (10), 2010.
- [53] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.
- [54] L. Duan, D. Xu, and I. Tsang, “Learning with augmented features for heterogeneous domain adaptation,” *arXiv preprint arXiv:1206.4660*, 2012.
- [55] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” in *CVPR 2011*, IEEE, 2011, pp. 1785–1792.
- [56] X. Shi, Q. Liu, W. Fan, S. Y. Philip, and R. Zhu, “Transfer learning on heterogenous feature spaces via spectral transformation,” in *2010 IEEE international conference on data mining*, IEEE, 2010, pp. 1049–1054.
- [57] G.-J. Qi, C. Aggarwal, and T. Huang, “Towards semantic knowledge propagation from text corpus to web images,” in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 297–306.
- [58] W. Li, L. Duan, D. Xu, and I. W. Tsang, “Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1134–1148, 2013.
- [59] M. Harel and S. Mannor, “Learning from multiple outlooks,” *arXiv preprint arXiv:1005.0027*, 2010.
- [60] L. Mihalkova, T. Huynh, and R. J. Mooney, “Mapping and revising markov logic networks for transfer learning,” in *Aaai*, vol. 7, 2007, pp. 608–614.

- [61] L. Mihalkova and R. J. Mooney, “Transfer learning by mapping with minimal target data,” in *Proceedings of the AAAI-08 workshop on transfer learning for complex tasks*, 2008.
- [62] J. Davis and P. Domingos, “Deep transfer via second-order markov logic,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 217–224.
- [63] Z. Deng, Y. Jiang, L. Cao, and S. Wang, “Knowledge-leverage based tsf fuzzy system with improved knowledge transfer,” in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2014, pp. 178–185.
- [64] A. Salami, M. B. Khodabakhshi, and M. H. Moradi, “Fuzzy transfer learning approach for analysing imagery bci tasks,” in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, IEEE, 2017, pp. 300–305.
- [65] V. Behbood, J. Lu, and G. Zhang, “Fuzzy bridged refinement domain adaptation: Long-term bank failure prediction,” *International Journal of Computational Intelligence and Applications*, vol. 12, no. 01, p. 1 350 003, 2013.
- [66] J. Shell, “Fuzzy transfer learning,” 2013.
- [67] Z. Deng, Y. Jiang, H. Ishibuchi, K.-S. Choi, and S. Wang, “Enhanced knowledge-leverage-based tsf fuzzy system modeling for inductive transfer learning,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 1, pp. 1–21, 2016.
- [68] C. Yang, Z. Deng, K.-S. Choi, and S. Wang, “Takagi–sugeno–kang transfer learning fuzzy logic system for the adaptive recognition of epileptic electroencephalogram signals,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 5, pp. 1079–1094, 2015.



- [69] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, “Fuzzy regression transfer learning in takagi–sugeno fuzzy models,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1795–1807, 2016.
- [70] X. Peng, Z. Huang, X. Sun, and K. Saenko, “Domain agnostic learning with disentangled representations,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 5102–5112.
- [71] A. Shorten and C. Moorley, “Selecting the sample,” *Evidence-based nursing*, vol. 17, no. 2, pp. 32–33, 2014.
- [72] A. E. Berndt, “Sampling methods,” *Journal of Human Lactation*, vol. 36, no. 2, pp. 224–226, 2020.
- [73] J. Eichhorn, *Survey research and sampling*. Sage, 2022.
- [74] S. L. Gill, “Qualitative sampling methods,” *Journal of Human Lactation*, vol. 36, no. 4, pp. 579–581, 2020.
- [75] A. Omair *et al.*, “Sample size estimation and sampling techniques for selecting a representative sample,” *Journal of Health specialties*, vol. 2, no. 4, p. 142, 2014.
- [76] S. Tyrer and B. Heyman, “Sampling in epidemiological research: Issues, hazards and pitfalls,” *BJPsych bulletin*, vol. 40, no. 2, pp. 57–60, 2016.
- [77] A. Chaudhuri and H. Stenger, *Survey sampling: theory and methods*. CRC Press, 2005.
- [78] M. Elfil and A. Negida, “Sampling methods in clinical research; an educational review,” *Emergency*, vol. 5, no. 1, 2017.
- [79] O. Yazdanbakhsh and S. Dick, “A systematic review of complex fuzzy sets and logic,” *Fuzzy Sets and Systems*, vol. 338, pp. 1–22, 2018.
- [80] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan, *Transfer learning*. Cambridge University Press, 2020.

- [81] U. M. L. Repository, “Medical datasets,” <https://archive.ics.uci.edu/ml/index.php>, 2022.
- [82] L. T. H. Lan, *Một số mở rộng của hệ suy diễn mờ phức cho bài toán hỗ trợ ra quyết định*. Luận án Tiến sĩ Khoa học máy tính, Học viện Khoa học và Công nghệ-Viện Hàn lâm Khoa học và Công nghệ Việt Nam, 2021.
- [83] L. A. Breslow, D. W. Aha, *et al.*, “Simplifying decision trees: A survey,” *Knowledge engineering review*, vol. 12, no. 1, pp. 1–40, 1997.
- [84] S. Fletcher and M. Z. Islam, “Decision tree classification with differential privacy: A survey,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–33, 2019.
- [85] V. G. Costa and C. E. Pedreira, “Recent advances in decision trees: An updated survey,” *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4765–4800, 2023.
- [86] A. Liaw, M. Wiener, *et al.*, “Classification and regression by randomforest,” *R news*, vol. 2, no. 3, pp. 18–22, 2002.
- [87] A. B. Shaik and S. Srinivasan, “A brief survey on random forest ensembles in classification model,” in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2*, Springer, 2019, pp. 253–260.
- [88] Z. He, D. Lin, T. Lau, and M. Wu, “Gradient boosting machine: A survey,” *arXiv preprint arXiv:1908.06951*, 2019.
- [89] Y.-l. Chen, T. Wang, B.-s. Wang, and Z.-j. Li, “A survey of fuzzy decision tree classifier,” *Fuzzy Information and Engineering*, vol. 1, pp. 149–159, 2009.
- [90] P. Nagaraj and P. Deepalakshmi, “An intelligent fuzzy inference rule-based expert recommendation system for predictive diabetes diagnosis,” *Interna-*

- tional Journal of Imaging Systems and Technology*, vol. 32, no. 4, pp. 1373–1396, 2022.
- [91] M. P. Palwankar, R. K. Kapania, and D. C. Hammerand, “Making finite element modeling choices using decision-tree-based fuzzy inference system,” *AIAA Journal*, vol. 61, no. 3, pp. 1349–1365, 2023.
- [92] J. Ma, A. Zhang, F. Gao, W. Bi, and C. Tang, “A novel rule generation and activation method for extended belief rule-based system based on improved decision tree,” *Applied Intelligence*, vol. 53, no. 7, pp. 7355–7368, 2023.
- [93] N. Khazali, M. Sharifi, and M. A. Ahmadi, “Application of fuzzy decision tree in eor screening assessment,” *Journal of Petroleum Science and Engineering*, vol. 177, pp. 167–180, 2019.
- [94] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. Santhosh Kumar, M. Selvi, and K. Arputharaj, “Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks,” *IET Communications*, vol. 14, no. 5, pp. 888–895, 2020.
- [95] H. Li, “Application of intelligent fuzzy decision tree algorithm in english machine translation,” in *International Conference on Innovative Computing*, Springer, 2022, pp. 647–652.
- [96] M. Krasnyuk, I. Hrashchenko, S. Goncharenko, and S. Krasniuk, “Hybrid application of decision trees, fuzzy logic and production rules for supporting investment decision making (on the example of an oil and gas producing company),” *ACCESS Journal: Access to Science, Business, Innovation in Digital Economy*, 2022.
- [97] Z. Liu, A. Liu, G. Zhang, and J. Lu, “An empirical study of fuzzy decision tree for gradient boosting ensemble,” in *Australasian Joint Conference on Artificial Intelligence*, Springer, 2022, pp. 716–727.

- [98] J. Li, Q. Zhou, H. Williams, P. Xu, H. Xu, and G. Lu, “Fuzzy-tree-constructed data-efficient modelling methodology for volumetric efficiency of dedicated hybrid engines,” *Applied Energy*, vol. 310, p. 118 534, 2022.
- [99] N. Agarwal, A. Sondhi, K. Chopra, and G. Singh, “Transfer learning: Survey and classification,” *Smart Innovations in Communication and Computational Sciences: Proceedings of ICSICCS 2020*, pp. 145–155, 2021.
- [100] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [101] W. Fang, C. Chen, B. Song, L. Wang, J. Zhou, and K. Q. Zhu, “Adapted tree boosting for transfer learning,” in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 741–750.
- [102] Y. Sun, T. Lu, C. Wang, *et al.*, “Transboost: A boosting-tree kernel transfer learning algorithm for improving financial inclusion,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 12 181–12 190.
- [103] E. Girlich, “K. thulasiraman, mns swamy: Graphs: Theory and algorithms,” *ZOR*, vol. 41, pp. 122–122, 1995.
- [104] P. W. Tennant, E. J. Murray, K. F. Arnold, *et al.*, “Use of directed acyclic graphs (dags) to identify confounders in applied health research: Review and recommendations,” *International journal of epidemiology*, vol. 50, no. 2, pp. 620–632, 2021.
- [105] S. Chowdhury, R. Wang, Q. Yu, *et al.*, “Dagbagm: Learning directed acyclic graphs of mixed variables with an application to identify protein biomarkers for treatment response in ovarian cancer,” *BMC bioinformatics*, vol. 23, no. 1, p. 321, 2022.

- [106] M. Piccininni, S. Konigorski, J. L. Rohmann, and T. Kurth, “Directed acyclic graphs and causal thinking in clinical risk prediction modeling,” *BMC medical research methodology*, vol. 20, no. 1, pp. 1–9, 2020.

## PHỤ LỤC A. MỘT SỐ VÍ DỤ CHO CÁC THUẬT TOÁN TRÊN CẤU TRÚC CFRG

### Ví dụ về thêm luật vào cấu trúc CFRG

Để minh họa rõ ràng hơn các bước thêm một nút CFN vào CFRG, cũng như thêm luật CFR vào CFRG, một ví dụ được mô tả như trong Hình 1 và Hình 2. Hình 1, 2 biểu diễn các bước thêm luật  $CFR_1$  và  $CFR_2$  vào CFRG. Với hai luật  $CFR_1$  và  $CFR_2$  như sau:

$CFR_1$  : *If  $A_{X_1}$  is S and  $P_{X_1}$  is S and  $A_{X_2}$  is S and  $P_{X_2}$  is M and  $A_{X_3}$  is VS and  $P_{X_3}$  is VS then 0*

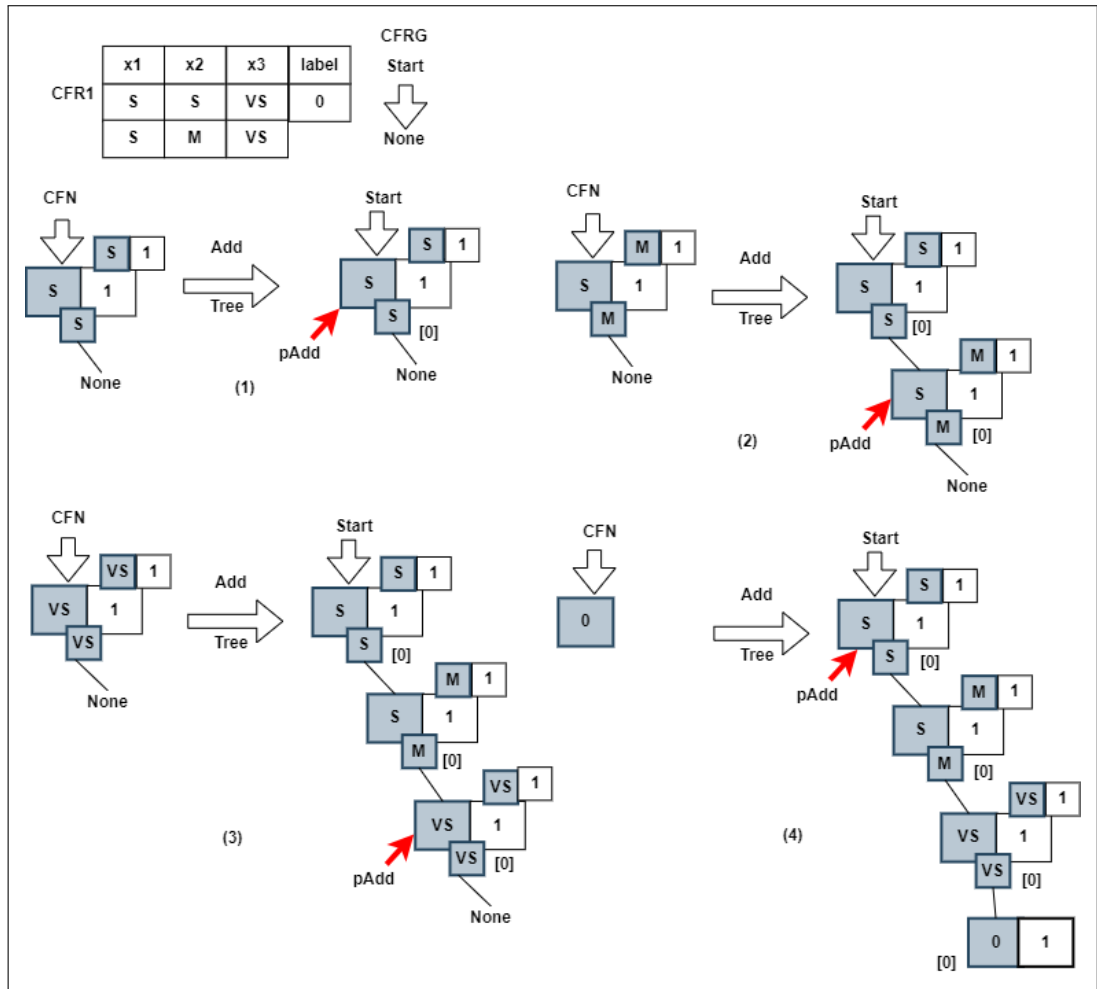
$CFR_2$  : *If  $A_{X_1}$  is S and  $P_{X_1}$  is VS and  $A_{X_2}$  is S and  $P_{X_2}$  is VS and  $A_{X_3}$  is L and  $P_{X_3}$  is L then 0* Ban đầu CFRG rỗng, Start trở đến None. Ứng với mỗi luật, mỗi cặp biến ngôn ngữ biên độ và pha sẽ tạo thành các nút, sau đó được thêm vào CFRG.

Với luật  $CFR_1$ , gồm 3 thuộc tính và một nhãn đầu ra sẽ tạo thành 4 nút mờ phức được thêm vào trên CFRG, trong đó nút lá sẽ có phần pha là None (như Hình 1(4)). Quá trình thêm nút, có sử dụng con trỏ  $pAdd$  trở đến nút vừa mới được thêm vào. Khi thêm đến nút lá thì con trỏ  $pAdd$  được gán lại về nút Start. Dễ dàng nhận thấy với mỗi nút phần pha được tạo ra để liên kết với nút con, tất cả các phần pha của các nút trong cùng một luật có chỉ số giống nhau.

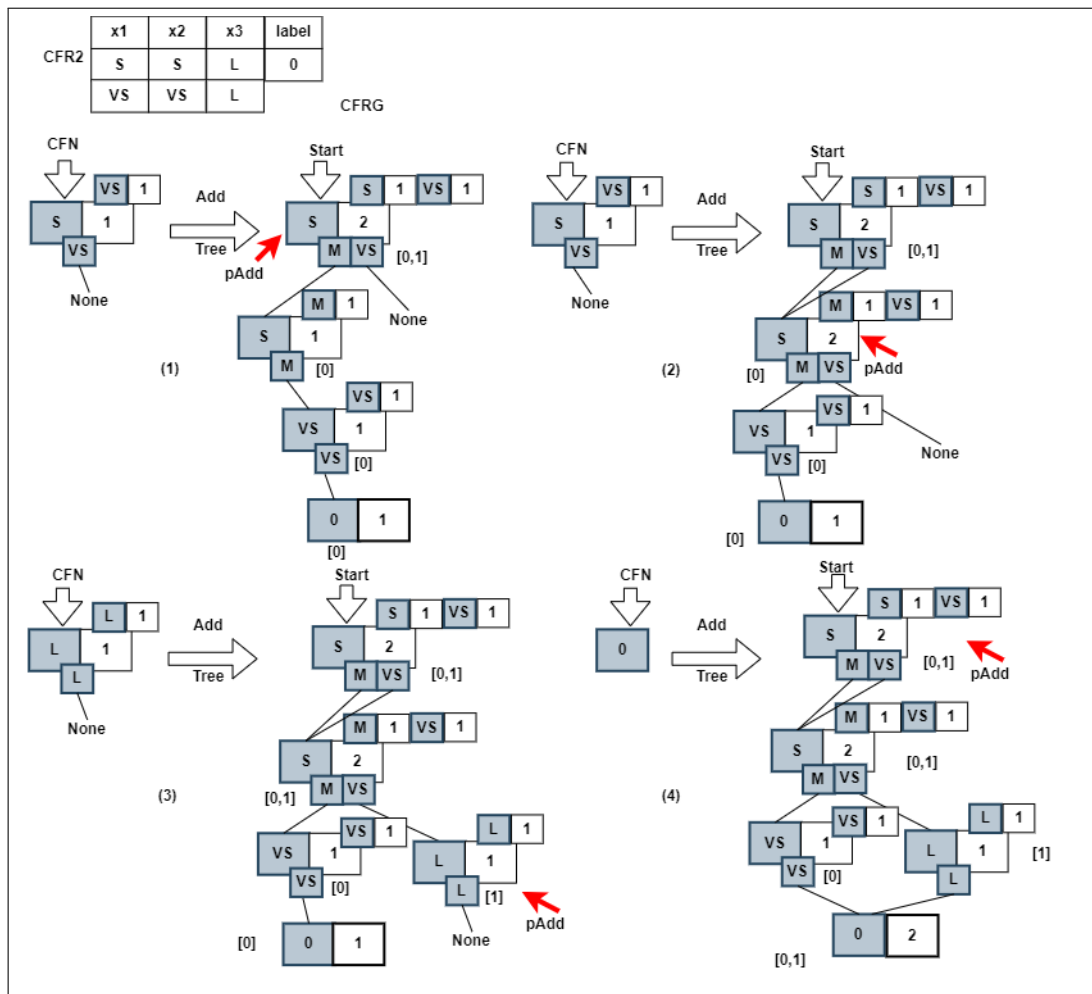
Trường hợp thêm luật  $CFR_1$  (Hình 1) thêm vào CFRG, đây là luật đầu tiên được thêm vào, CFRG ban đầu là None. Do đó, ở mỗi trường hợp thêm một nút mới, nút con của  $pAdd$  đều là None, nên các nút sẽ được thêm liên tiếp như Hình 1.

Thêm tiếp luật  $CFR_2$  vào CFRG (Hình 2), sẽ thấy sự khác biệt ở đây, mỗi nút mới được thêm vào, ở mỗi bước, nút con của  $pAdd$  đều là khác None, một quá trình kiểm tra giá trị  $aValue$  của con của  $pAdd$  có trùng với giá trị  $aValue$  của nút cần thêm không được thực hiện. Nếu con của  $pAdd$  có giá trị  $aValue$  trùng với giá trị  $aValue$

của nút cần thêm (như Hình 2 (2)), thì thêm nút không tạo ra nút mới trên CFRG mà chia sẻ nhau tại nút này bằng cách tăng tần suất của nút con và bổ sung giá trị phần pha.



**Hình 1:** Thêm luật CFR1 vào CFRG

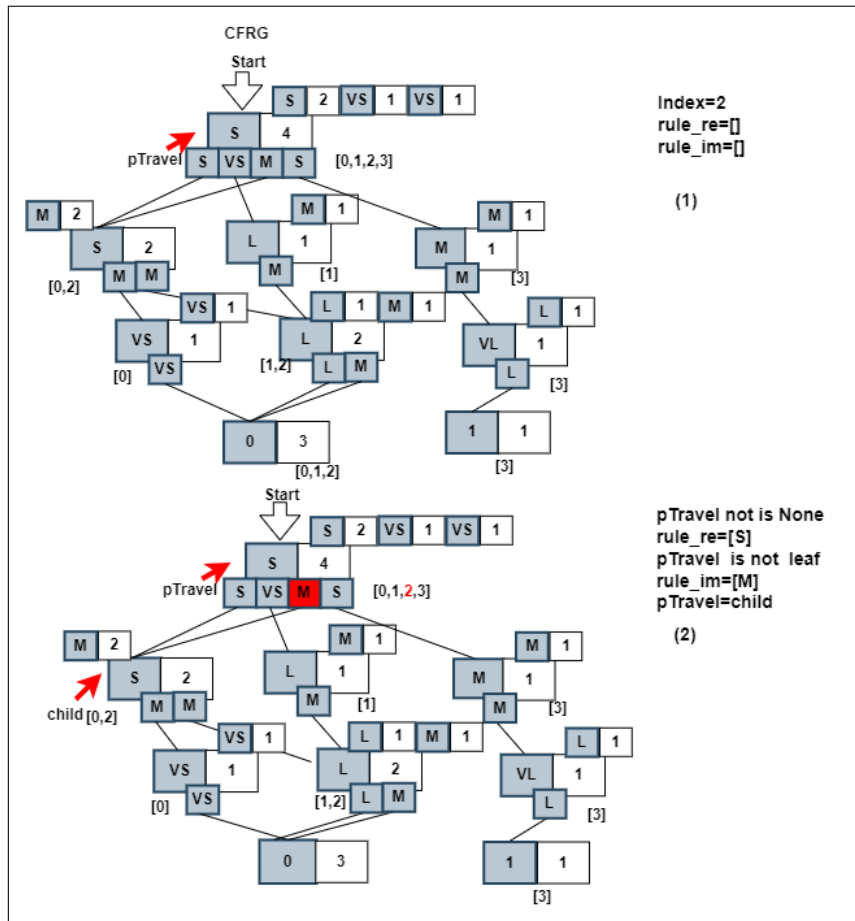


**Hình 2:** Thêm luật CFR2 vào CFRG

**Ví dụ về duyệt luật theo chiều sâu**

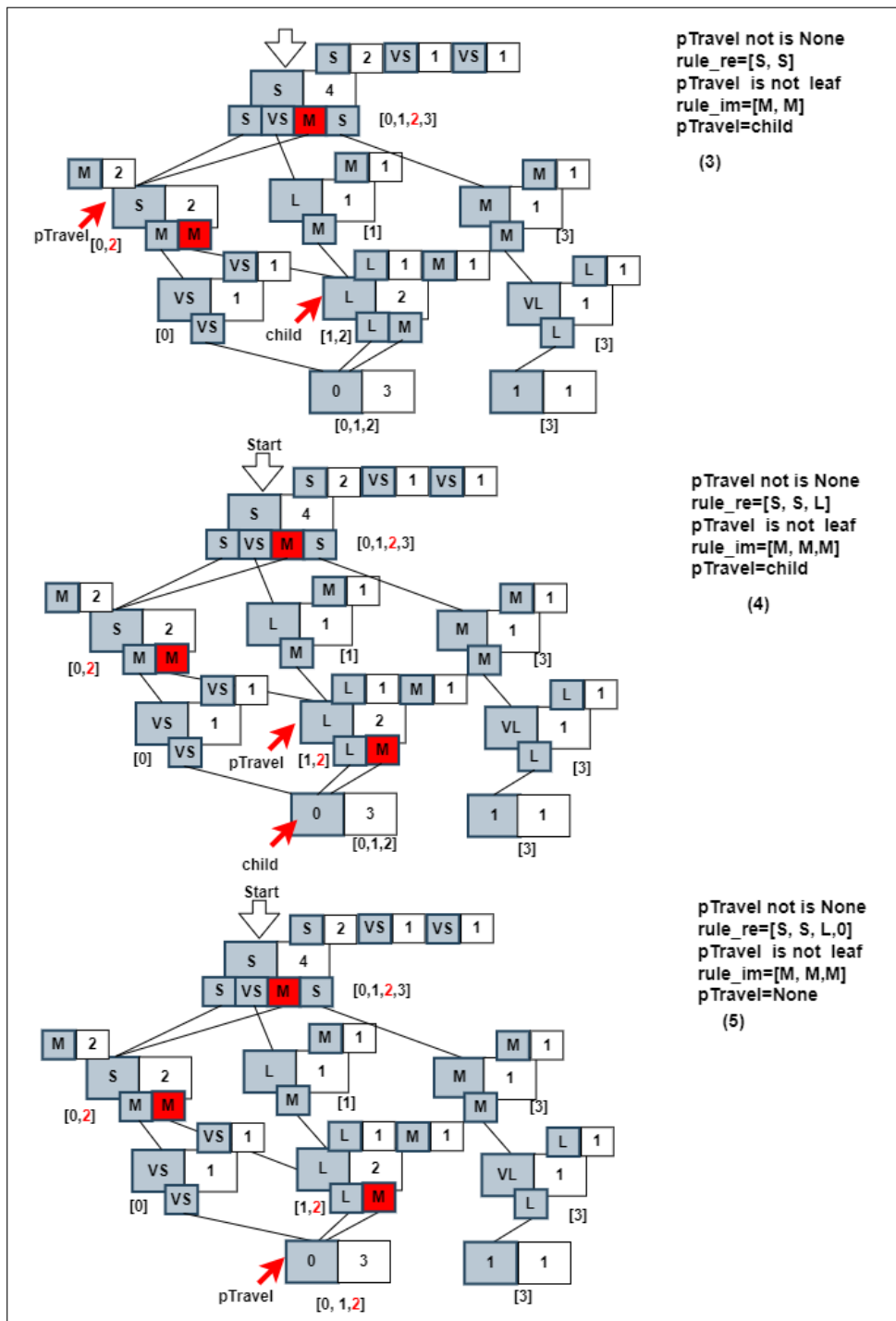
Hình 3 và 4 mô tả quá trình duyệt luật mờ phức theo chiều sâu, đây là quá trình duyệt luật mờ phức có chỉ số *index* là 2.





**Hình 3:** Duyệt luật mờ phức có chỉ mục 2 theo chiều sâu

Duyệt một luật, luôn xuất phát từ nút *Start* của CFGR và kết thúc khi duyệt đến nút lá. Khi duyệt sử dụng một con trỏ *pTravel* trở đến nút cần duyệt qua, ban đầu *pTravel* được trở đến nút gốc *Start* của CFGR. Tại mỗi nút giá trị biên độ và giá trị pha của nút được lấy để tạo thành luật. Như trong Hình 3, do giá trị chỉ số *index* là 2 nên phần pha được đọc là tương ứng sẽ là M. Ứng với giá trị phần pha và chỉ số *index* xác định được nút con được duyệt tiếp theo tương ứng như Hình 4.



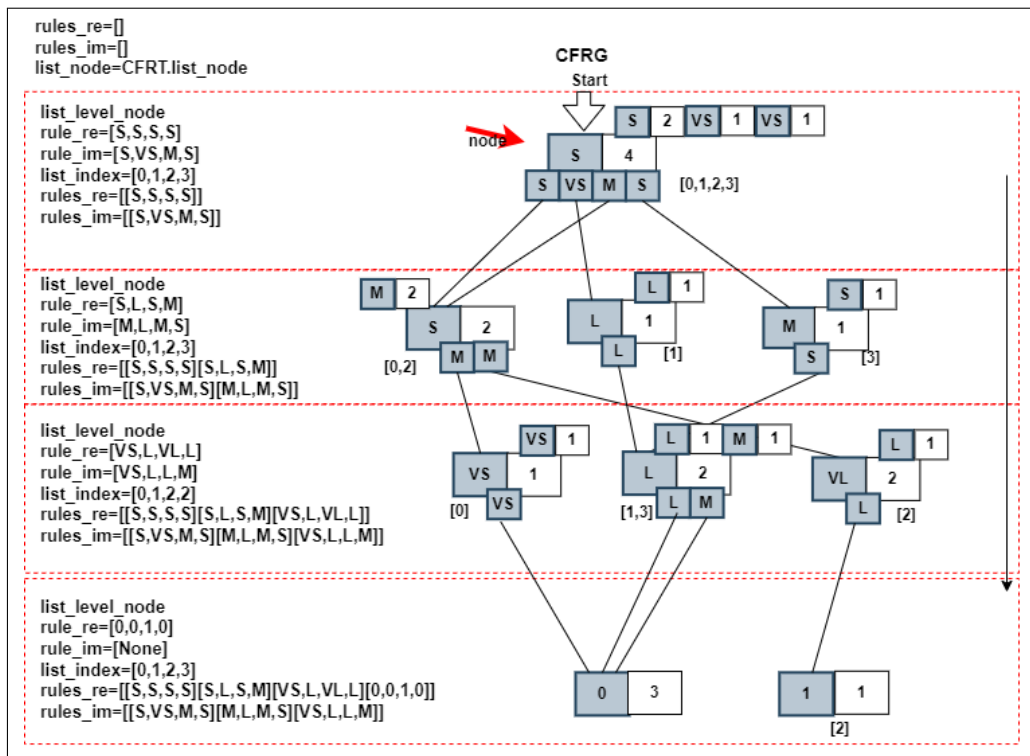
Hình 4: Duyệt luật mờ phức có chỉ mục là 2 theo chiều sâu (tiếp)

**Ví dụ về duyệt cây theo chiều rộng**

Một ví dụ về duyệt luật theo chiều rộng được trình bày trong Hình 5. Quá trình

duyet luật theo chiều rộng được mô tả như sau:

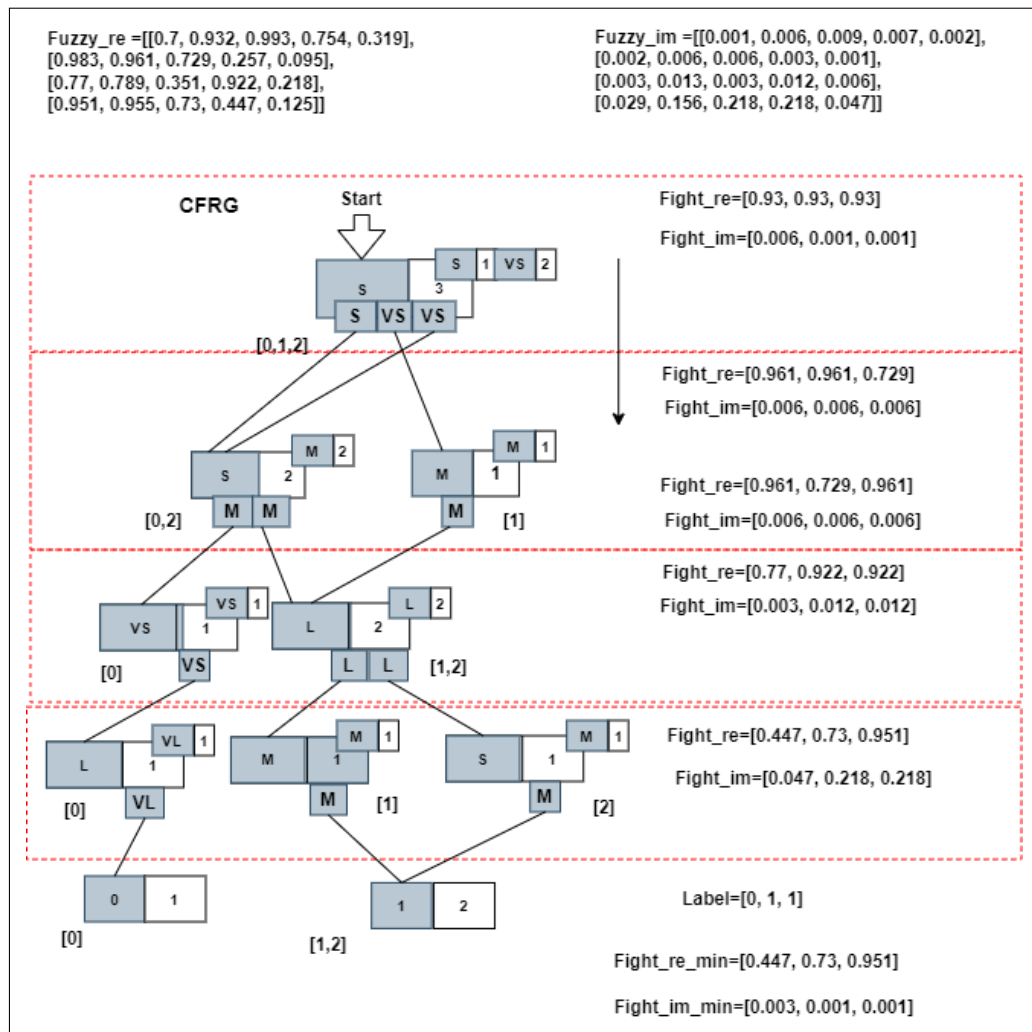
- Đầu tiên, các nút cấp 0 được duyệt, ở cấp này có duy nhất nút gốc. Luật phần biên độ sẽ là giá trị biên độ của nút được lấy ra đúng bằng tần suất của nút. Và ta sẽ có  $rule = [S, S, S, S]$  (như Hình 5). Luật phần pha được tạo thành theo chỉ số là thứ tự và giá trị là giá trị phần pha  $rule\_im = [S, VS, M, S]$ .
- Duyệt sang mức 1, danh sách các nút tại cấp này gồm 3 nút, mỗi nút sẽ được thăm tạo thành danh sách luật phần biên độ và danh sách luật phần pha theo cách, thăm phần pha của nút để lấy chỉ số thứ tự luật làm cơ sở để gán giá trị luật biên độ  $rule = [S, S, L, M]$  và giá trị luật phần pha  $rule = [M, L, M, S]$ .
- Duyệt tiếp sang mức 2, danh sách các nút sẽ được thăm và giá trị là  $rule = [VS, L, VL, L]$  và  $rule\_im = [VS, L, L, M]$ .
- Tương tự khi duyệt tiếp mức 3 danh sách các nút lá được đọc ra là  $rule = [0, 0, 1, 0]$
- Cuối cùng ta thu được tập luật mờ phức gồm phần biên độ  $rules = [[S, S, S, S]$   
 $[S, S, L, M]$   $[VS, L, VL, L]$   $[0, 0, 1, 0]$ ] và phần pha  $rules\_im = [[S, VS, M, S]$   $[M, L, M, S]$   
 $[VS, L, L, M]$



Hình 5: Duyệt cấu trúc CFRG theo chiều rộng

**Ví dụ về suy diễn mờ phức trên cấu trúc CFRG** Một ví dụ về suy diễn trên CFRG được biểu diễn trong hình dưới đây Hình 6. Với đầu vào đã mờ hóa cho phần thực (Fuzzy\_re) và phần ảo (Fuzzy\_im). Ở mỗi bước duyệt theo chiều rộng của CFRG, giá trị biên độ và pha của nút sẽ được ánh xạ lên Fuzzy\_re, Fuzzy\_im tương ứng để cho ra giá trị chấy phần thực và phân pha.

Như ở mức 0 sẽ có  $rule = [S, S, S]$  và  $rule\_im = [S, VS, VS]$  các giá trị này sẽ được ánh xạ tương ứng lên Fuzzy\_re, Fuzzy\_im cho kết quả là giá trị chấy  $Fight\_re = [0,932; 0,932; 0,932]$  và  $Fight\_im = [0,006; 0,001; 0,001]$ . Quá trình được thực hiện tương ở các mức tiếp theo. Sau đó các giá trị chấy phần thực và phân pha này sẽ được tổng hợp theo toán tử min để thu được giá trị chấy của mỗi luật mờ phức trên CFRG.

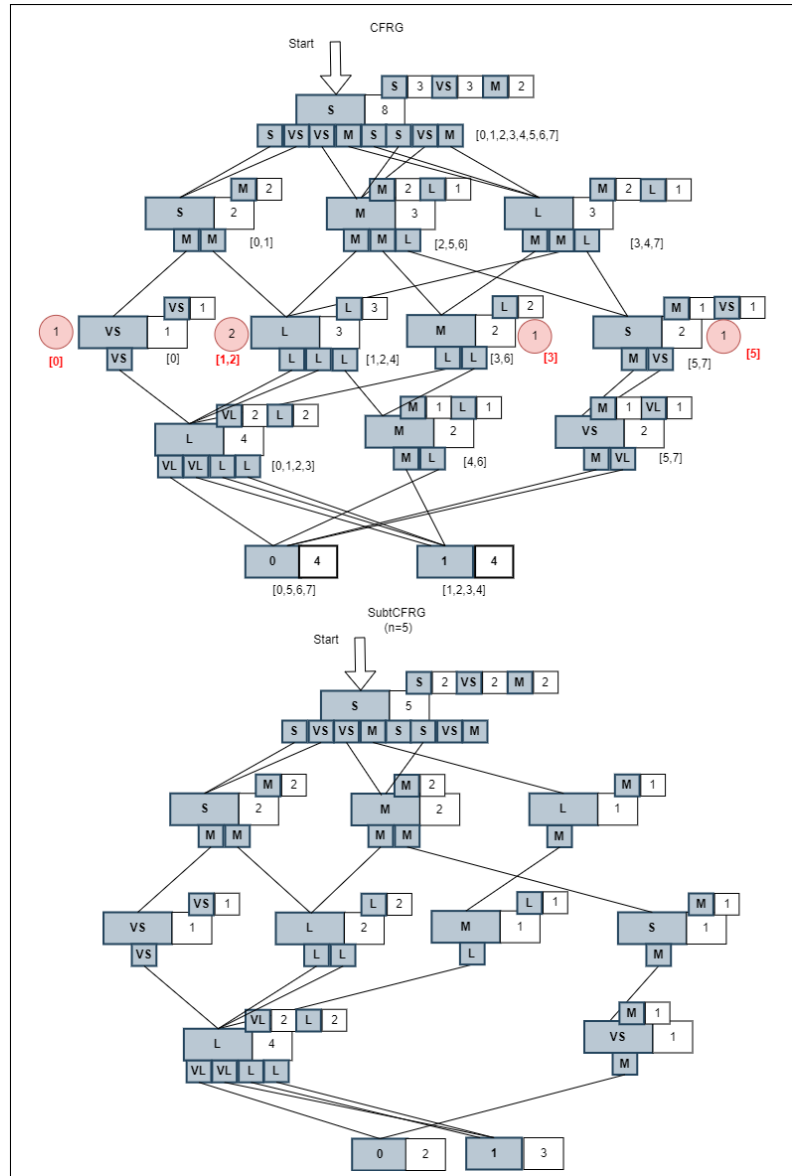


Hình 6: Suy diễn mờ phức trên cấu trúc CFGR

**Ví dụ về tách cấu trúc CFGR con từ cấu trúc CFGR cha**

Hình 7 mô tả kết quả tách CFGR con gồm 5 luật từ cấu trúc CFGR - gồm 8 luật ban đầu. Hình trên trong Hình 7 mô tả cấu trúc CFGR ban đầu gồm 8 luật, hình dưới mô tả cấu trúc CFGR\_child gồm 5 luật được tách ra từ cấu trúc CFGR. Quá trình tách được thực hiện theo thuật toán đề xuất phía trên. Việc xác định các luật được lựa chọn từ danh sách các nút nằm tại mức có nhiều nút nhất được biểu diễn như trong hình ban đầu. Ở mỗi nút sẽ xác định số lượng nút được tách ra theo thuật toán Như biểu diễn trong Hình 7 tại mức 3 của CFGR, xác định được các nút sẽ lấy ra trên mỗi nút của CFGR tương ứng là (1, 2, 1, 1). Sau đó sẽ xác định được các luật được lấy ra

tương ứng sẽ là chỉ số luật. Trong ví dụ tại hình 7 các chỉ số các luật được lấy ra là (0, 1, 2, 3, 5). Với danh sách chỉ số này sẽ thực hiện tiếp thuật toán xóa luật theo chỉ số trên CFRG cha và thêm luật vào CFRG con.



**Hình 7:** Tách một cấu trúc CFRG con từ cấu trúc CFRG cha

